

EXHIBIT E

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

In these Infringement Contentions, AGIS Software Development LLC (“AGIS”) contends that at least the following claims of U.S. Patent No. 9,467,838 (the “’838 Patent”) identified below are infringed by the Accused Products (e.g., phones and tablets) which are manufactured, sold, offered for sale, and/or used by Defendants ZTE (TX) Inc., ZTE (USA) Inc., and ZTE Corporation (collectively, “ZTE”).

The Accused Products comprise ZTE products running the Android mobile operating system and manufactured, used, or sold during and after 2011. For example, the Accused Products comprise the following Android-based phones and tablets: Adamant, Agent, Anthem 4G, Aspect, Avail / Merit, Avid 4, Avid 4G, Avid Plus, Avid Plus / Avid 828, Avid Trio / ZFive 2, Awe / Emblem, Axon, Axon 7, Axon 7 Max, Axon 7 mini, Axon Elite, Axon Lux, Axon M, Axon Max, Axon Mini, Axon Pro / Axon, Blade A1, Blade A2, Blade A2 Plus, Blade A2S, Blade A3, Blade A910, Blade C, Blade D Lux, Blade D2, Blade D6, Blade E, Blade Force, Blade G2, Blade L, Blade L3, Blade L5 Plus, Blade Max 3, Blade Max Blue, Blade Q, Blade Q Maxi, Blade Q Mini, Blade Qlux 4G, Blade S6, Blade S6 Lux, Blade S6 Plus, Blade S7, Blade Spark, Blade V580, Blade V6, Blade V7, Blade V7 Lite, Blade V7 Max, Blade V7 Plus, Blade V8, Blade V8 Lite, Blade V8 Mini, Blade V8 Pro, Blade Vantage, Blade Vec 3G, Blade Vec 4G, Blade X, Blade X Max, Blade X3, Blade X5, Blade X9, Blade Z Max, C78, C79, C88, CAPTR II / A210, Chorus, Citrine, Compel, Concord / Midnight, Concord II, CYMBAL LTE, Cymbal LTE (Verizon), Cymbal Z-320, Cymbal LTE, Cymbal-C LTE, Cymbal-G LTE, CYMBAL-T, Cymbal-T LTE, Engage, Engage LT / Engage MT, Essence C70, F160, Fanfare, Fanfare 2, Fanfare 3, Flash, Force, Fury / Director, Geek, Grand Memo II LTE, Grand S Flex, Grand S II, Grand S Pro, Grand S3, Grand X, Grand X 3, Grand X 4, Grand X 4, Grand X Max 2 / Imperial MAX, Grand X Max+, Grand X Quad Lite, Grand XMax, Groove, Hawkeye, Imperial, Imperial II, Jasper LTE, Kis 3 Max, Kis Flex, Majesty, Majesty Pro, Majesty Pro Plus LTE, Maven 2, Maven 2 / Sonata 3, Maven 3 / Overture 3, Max, MAX Blue LTE, MAX XL, Max+, Memo, Midnight PRO LTE, MSGM8 II, N919D, Nubia M2, Nubia M2 Lite, Nubia M2 Play, Nubia My Prague, Nubia N1, Nubia N1 lite, Nubia N1, Nubia N2, Nubia Prague S, Nubia X6, Nubia Z11, Nubia Z11 Max, Nubia Z11 mini, Nubia Z11 mini S, Nubia Z17, Nubia Z17 Lite, Nubia Z17 mini, Nubia Z17 miniS, Nubia Z17S, Nubia Z7, Nubia Z7 Max, Nubia Z7 mini, Nubia Z9 Classic, Nubia Z9 Elite, Nubia Z9 Exclusive, Nubia Z9 Max, Nubia Z9 mini, Obsidian, Open, Open C, Open II, Overture, Overture 2 / Maven, Prelude / Avail 2, Prelude+, Prestige, Prestige 2, Q519T, Quartz, R225, Reef, Render, Salute, Savvy, Score M / Score, Small Fresh 4, Small Fresh 5, Solar, Sonata / Radiant, Sonata 2 / Paragon, Sonata 3, Source, Speed, Star 1, Star 2, Tempo, Tempo X, TXXM8 3G, V3 Energy Edition, V3 Extreme Edition, V3 Youth Edition, V5, V870, Valet, Vital / Supreme, Warp, Warp 4G, Warp 7, Warp Elite, Warp Sequent, Warp Sync, Whirl, Z221, Z222 / Z223, Z331, Z431 / Altair, Z432 / Altair 2, Z667 / Zinger / Prelude 2 / Flame / Whirl 2, Z998 / Unico LTE, ZFive 2 LTE, ZFive L LTE, Zinger, ZMax, Zmax 2, ZMAX 2 (Unlocked), ZMAX Champ LTE, ZMAX Grand / Champ / Avid 916, ZMAX GRAND LTE,

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

ZMax Pro, and any variants thereof. AGIS reserves the right to amend this list of Accused Products as discovery progresses. For example, AGIS reviewed Android-based products from Android-based handset manufacturers, including three ZTE phones (serial numbers 329F75623FA5; 329F7562388B; and 329F75624913) which are available for inspection at ZTE's request. For example, the Accused Products comprise ZTE products, including but not limited to the phones and tablets as described herein, running the following versions (and all intervening updates and sub-versions) of the Android mobile operating system: Android 2.3, 4.0, 4.1, 4.2, 4.3, 4.4, 5.0, 5.1, 6.0, 7.0, 7.1, 8.0, and 8.1. For example, the Accused Products comprise ZTE products, including but not limited to the phones and tablets as described herein, running any versions of the following Android-based applications and/or software: Android Device Manager, Find My Phone, Find My Device, Google Latitude, Google Plus, Google Hangouts, Google Maps, Google Assistant, Google Search, Google Messages, Android Messenger, Google Allo, Google Duo, Gmail, and Google Chrome. For example, the Accused Products comprise ZTE products, including but not limited to the phones and tablets described herein, participating in any networks and/or services related to the execution and/or use of the Android mobile operating system versions and Android-based applications and/or software described herein.

AGIS does not concede that any claims of the '838 Patent that are not listed below are not infringed by the identified products. Moreover, the citations to certain documents and other information below are intended to be exemplary only and in no way foreclose AGIS from citing or relying on additional documents, information, source code, and/or testimony at a later time. These contentions are preliminary in nature, and an analysis of ZTE's products, internal documentation, source code, and/or testimony from relevant witnesses may more fully and accurately describe the infringing features of its accused products. Accordingly, AGIS reserves the right to supplement, correct, modify, and/or amend these contentions once such additional information is made available to AGIS. Furthermore, AGIS reserves the right to supplement, correct, modify, and/or amend these contentions as discovery in this case progresses; in view of the Court's claim construction order(s); in view of any positions taken by ZTE, including but not limited to positions on claim construction, invalidity, and/or non-infringement; and in connection with the preparation and exchange of expert reports.

| US9467838 | Exemplary Supporting Evidence Regarding Accused Products |
|-------------------------------------|--|
| 1[P]. A computer-implemented method | ZTE infringes directly and/or indirectly by performing, inducing others to perform, and/or contributing to the performance of: a computer-implemented method [of claim 1]. |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| | |
|---|--|
| <p>US9467838 comprising:</p> | <p>Exemplary Supporting Evidence Regarding Accused Products</p> <p>The Accused Products meet the claim limitations by providing device-location tracking features such as those features described below. For example, the Accused Products meet the claim limitations because they are pre-installed with Android mobile operating systems containing code for providing device-location tracking features as provided in the claims limitations herein. For example, the Accused Products run applications and/or software that run within the Android mobile operating system and that use components of the Android mobile operating system to provide device-location tracking features. Upon information and belief, in addition to the components and features of the Android mobile operating system itself, the following applications and/or software run within the Android mobile operating system and use components of the Android mobile operating system to provide device-location tracking features: Android Device Manager, Find My Phone, Find My Device, Google Latitude, Google Plus, Google Hangouts, Google Maps, Google Messages, Android Messenger, Google Allo, Google Duo, and Google Chrome.</p> <p>The Accused Products meet the claim limitations under at least two exemplary methods and/or systems, as shown below, which may be useable together or separately.</p> <p>Find My Device: Android Device Manager is the predecessor to Find My Device and has been available as a standard, pre-installed feature since 2013 and downloadable as a software application. The current iteration, Find My Device, often called the “new and improved Android Device Manager” or “rebranded Android Device Manager” is now part of the standard Google Play Protect suite which is “built in and enabled on all devices,” <i>i.e.</i>, the Accused Products running Android OS. Upon information and belief, the Find My Device method also uses and/or works in conjunction with functionalities associated with Google Maps, Google Messages, Android Messenger, Location Access, Google Chrome, and other features which come pre-installed on the Accused Products. For the purposes of avoiding needlessly presenting cumulative and duplicative evidence, AGIS sets forth the Find My Device feature of the Accused Products as representative of this first exemplary method. AGIS reserves the right to supplement these contentions to the extent that defendant requires additional information in accordance with P.R. 3-1 and for any other reason for which it may deem necessary.</p> <p><i>See, e.g.</i>, https://www.androidcentral.com/find-my-device; https://support.google.com/android/answer/6160491?hl=en; https://android.googleblog.com/2013/08/find-your-lost-phone-with-android.html;</p> |
|---|--|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| | |
|------------------|---|
| <p>US9467838</p> | <p>Exemplary Supporting Evidence Regarding Accused Products https://play.google.com/store/apps/details?id=com.google.android.apps.adm&hl=en; https://www.blog.google/products/android/google-play-protect/</p> <p>Google Maps Share Location: Share Location is currently included as a standard feature on the Accused Devices operating as a feature of Google Maps. Google Maps is a pre-installed software application in Android OS. The Accused Devices have included the Share Location functionalities since 2009 as part of Google Latitude which was an opt-in feature for Google Maps on Android OS-based mobile devices, such as the Accused Products. Share Location functionalities were briefly shifted from Latitude for Google Maps to Google Plus and Google Hangouts, until reappearing as a standard feature in Google Maps. Upon information and belief, the Share Location method also uses and/or works in conjunction with functionalities associated with Google Maps, Google Messages, Android Messenger, Location Access, Google Allo, Google Duo, Google Chrome, and other features which come pre-installed on the Accused Products. For the purposes of avoiding needlessly presenting cumulative and duplicative evidence, AGIS sets forth Google Maps' Share Location feature of the Accused Products as representative of this second exemplary method. AGIS reserves the right to supplement these contentions to the extent that defendant requires additional information in accordance with P.R. 3-1 and for any other reason for which it may deem necessary.</p> <p>See, e.g., https://techcrunch.com/2017/03/22/google-maps-now-lets-you-share-your-location-with-friends-and-family-for-a-specific-period-of-time/; https://googleblog.blogspot.com/2009/02/see-where-your-friends-are-with-google.html; https://googleblog.blogspot.jp/2013/07/a-new-google-maps-app-for-smartphones.html; http://googleplusproject.blogspot.com/2013/05/google-for-android-42.html; https://googleblog.blogspot.com/2013/10/google-hangouts-and-photos-save-some.html</p> |
|------------------|---|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE



| | |
|------------------|---|
| <p>US9467838</p> | <p>Exemplary Supporting Evidence Regarding Accused Products</p> |
| | <p>Control within reach, even when your device isn't</p> <p>One of the biggest security risks you're likely to face is simply losing your phone. To help in these times of need, we're launching Find My Device as part of Google Play Protect. With Find My Device you can locate, ring, lock and erase your Android devices—phones, tablets, and even watches. This feature is built in and enabled on all devices; visit android.com/find or check out the app.</p> <p>See, e.g., https://www.blog.google/products/android/google-play-protect/</p> <p>Android Device Manager</p> <p>If you lose your phone or tablet, the Android Device Manager can help you find its approximate location, make it ring, lock it, or erase its data.</p> <p>These options are turned on by default. To view or change them, open the  Google Settings app (found in  All Apps) and touch Android Device Manager.</p> <hr/> <p>ANDROID QUICK START GUIDE ESSENTIALS 28</p> <p>https://static.googleusercontent.com/media/www.google.com/en/us/help/hc/images/android/android_ug_50/Android-Lollipop-Quick-Start-Guide.pdf</p> <p>As discussed above and shown below, Google Latitude provided similar features to Google Maps' location sharing feature since 2009:</p> |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

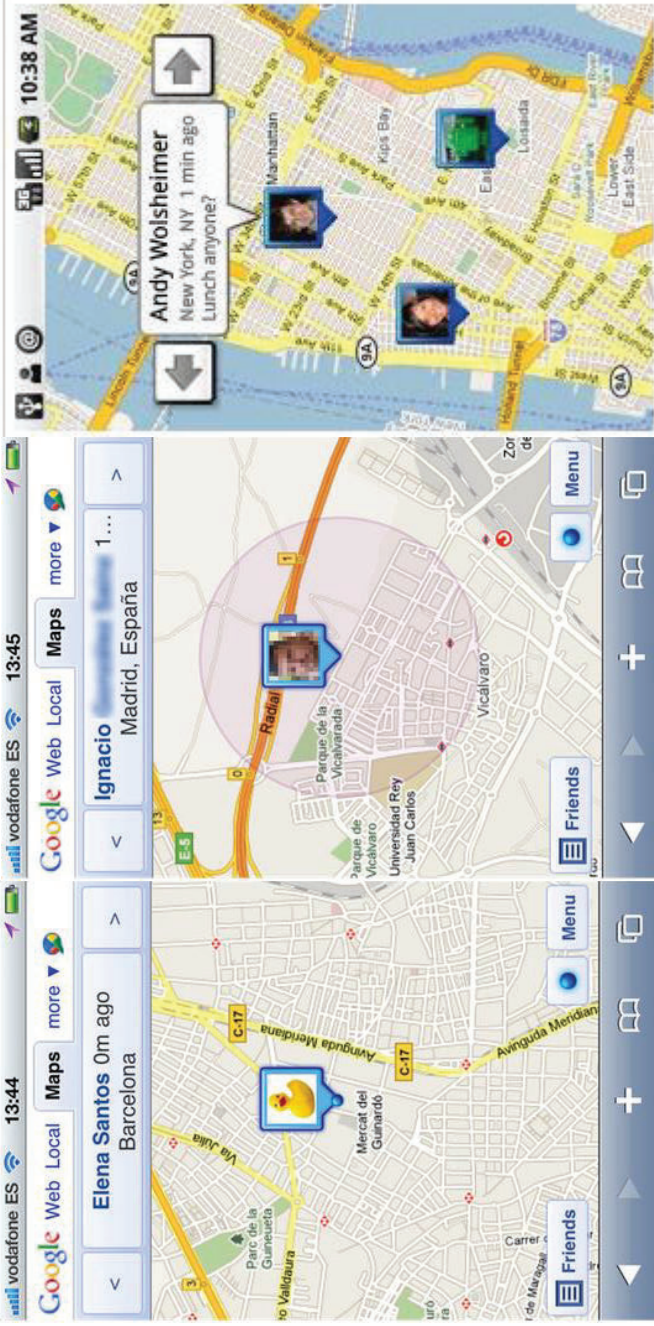
| | |
|---|---|
| <p>US9467838</p> | <p>Exemplary Supporting Evidence Regarding Accused Products</p>  <p>http://googlemobile.blogspot.com/2009/02/locate-your-friends-in-real-time-with.html</p> |
| <p>[1A] performing, by a first device:</p> | <p>ZTE infringes directly and/or indirectly by performing, inducing others to perform, and/or contributing to the performance, by a first device [the operations of claim 1].</p> <p>For example, the ZTE products, such as Android-based phones and tablets run device-location tracking features and/or software such as Android Device Manager, Find My Phone, Find My Device, Google Latitude, Google Plus, Google Hangouts, Google Maps, Google Messages, and Android Messenger, Google Allo, Google Duo, and Google Chrome.</p> |
| <p>[1B] joining a communication network</p> | <p>ZTE infringes directly and/or indirectly by performing, inducing others to perform, and/or contributing to the performance of: joining a communication network corresponding to a group, wherein joining the</p> |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| | |
|---|---|
| <p>US9467838</p> | <p>Exemplary Supporting Evidence Regarding Accused Products</p> |
| <p>corresponding to a group, wherein joining the communication network comprises transmitting a message including an identifier corresponding to the group;</p> | <p>communication network comprises transmitting a message including an identifier corresponding to the group.</p> <p>Regarding Find My Device and Android Device Manager, the Accused Products require a user to join the corresponding network by: signing-in to the device with an identifier (e.g., Google Account) or linking the device to the identifier by remote means. The sign-in process takes place within the Find My Device software on the Accused Product; by navigating to android.com/find within the Google Chrome browser on the Accused Product; by navigating to google.com and typing “find my phone” within the Google Chrome browser on the Accused Product; or by searching “find my phone” within the search bar of the Google home screen and/or Google application on the Accused Product. Alternatively, the sign-in process may partially or completely take place using credentials already provided when the user associates a Google Account with the Accused Product, e.g., during initial setup of the Accused Product. Subject to discovery, one or more additional or substitute identifiers may correspond to the group. The sign-in process involves a user entering its Google Account and additional authentication data on the interface of the Accused Product and sending a message containing the Google Account and additional authentication data over a network to the group. The group comprises the multiple devices linked to the identifier.</p> <p>Regarding Google Maps, Google Latitude, Google Plus, Google Hangouts, Google Messages, and Android Messenger, Google Duo, Google Allo, and Google Chrome, the Accused Products require a user to join the corresponding network by signing-in to the device with an identifier (e.g., Google Account). The sign-in process takes place within the Google Maps software on the Accused Product or by navigating to maps.google.com within the Google Chrome browser on the Accused Product. Alternatively, the sign-in process may partially or completely take place using credentials already provided when the user associates a Google Account with the Accused Product, e.g., during initial setup of the Accused Product. Subject to discovery, one or more additional or substitute identifiers may correspond to the group. The sign-in process involves a user entering its Google Account and additional authentication data on the interface of the Accused Product and sending a message containing the Google Account and additional authentication data over a network to the group. The group comprises the multiple identifiers, individuals, profiles, and/or devices associated with the group.</p> <p>Further regarding Google Maps, Google Latitude, Google Plus, Google Hangouts, Google Messages, and Android Messenger, Google Duo, Google Allo, and Google Chrome, the Accused Products alternatively</p> |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| | |
|------------------|--|
| US9467838 | <p data-bbox="186 176 228 1923"><u>Exemplary Supporting Evidence Regarding Accused Products</u></p> <p data-bbox="228 176 446 1923">require a user to send a request containing the identifier (e.g., Google Account). Subject to discovery, additional identifiers may be assigned or used to correspond to the group. The request may be an invitation or message that associates a Google Account with one or more Google Accounts for the purposes of sharing locations within the group. The group comprises the multiple identifiers, individuals, profiles, and/or devices associated with the group</p> <p data-bbox="446 176 519 1923"><u>Exemplary Support for Find My Device:</u></p> |
|------------------|--|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

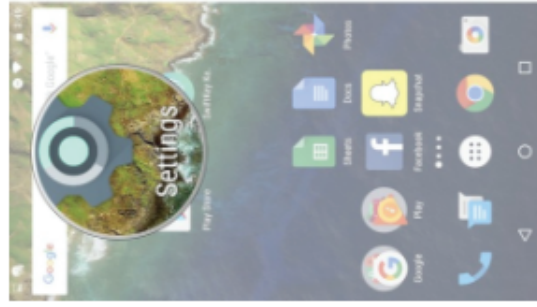
US9467838

Exemplary Supporting Evidence Regarding Accused Products

How to enable Find My Device on your phone

In newer Android phones, the Find My Device service is already located conveniently in your Settings app, but if you can't find it you can always download Find My Device from the Google Play Store. This locating service has essentially amalgamated with Google to make finding your phone easier. There are just a couple of things you'll need to activate.

1. Launch Settings.
2. Tap Security.
3. Tap Device Administration.



4. Tap Find My Device so that a checkmark appears in the checkbox.

<https://www.androidcentral.com/how-track-android-phone>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

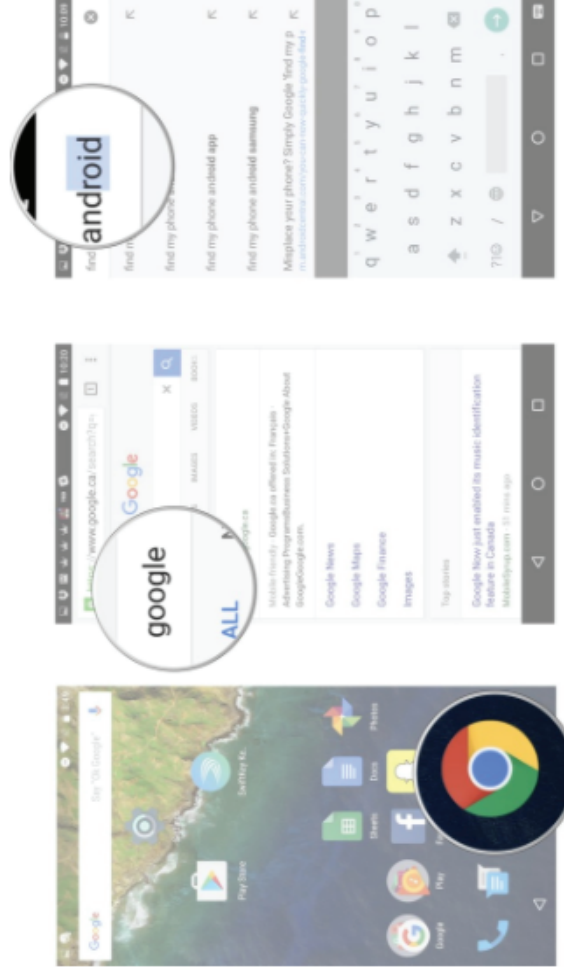
US9467838

Exemplary Supporting Evidence Regarding Accused Products

How to locate your phone with Google

Should you happen to lose your phone, you can locate its whereabouts by logging into your Google account from any computer or even from another phone.

1. Launch a web browser from a phone, tablet, or computer.
2. Navigate to Google if it is not your default search engine or home page.
3. Type find my phone android in the Google search bar.



4. Tap on Find My Device (usually the first option in the search).
5. Enter your email address and password just as though you were checking your email. If you have 2-step verification set up on your Google account (and you most certainly should), you'll need to complete that process as well.

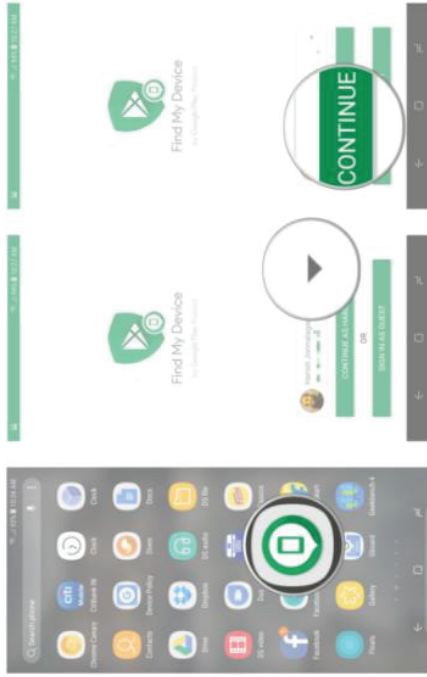
<https://www.androidcentral.com/how-track-android-phone>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products

1. Open Find Device from your home screen or app drawer.
2. Select the Google account you want to use the service with.
3. Hit the Continue as button.



4. Enter your Google account password.
5. Tap Sign in.
6. Give location access to the service.



<https://www.androidcentral.com/find-my-device>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

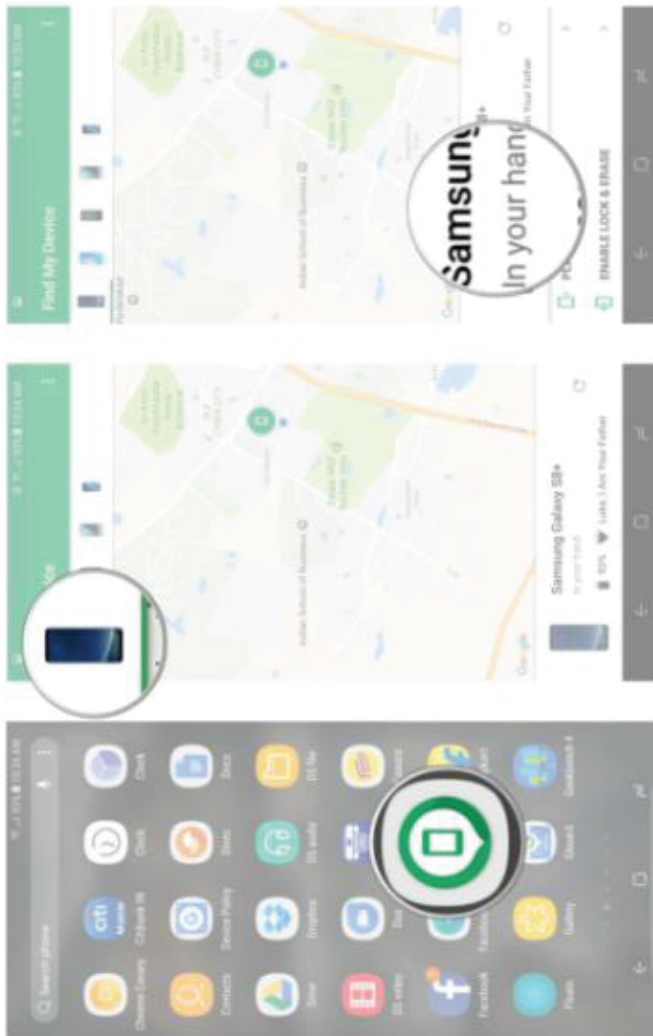
US9467838

Exemplary Supporting Evidence Regarding Accused Products

Once you're signed in to Find My Device, you'll see a map with your current location as well as the make and model of your phone, and two options — Play Sound, and Enable Lock & Erase. Hitting the latter option will allow you to start using the Lock and Erase functions.

If you've signed into more than one phone, you can select a particular device by browsing the list at the top of the screen.

1. Open Find My Device from your home screen or app drawer.
2. Select your phone from the list of devices at the top of the screen.
3. See if your phone is discoverable.



<https://www.androidcentral.com/find-my-device>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

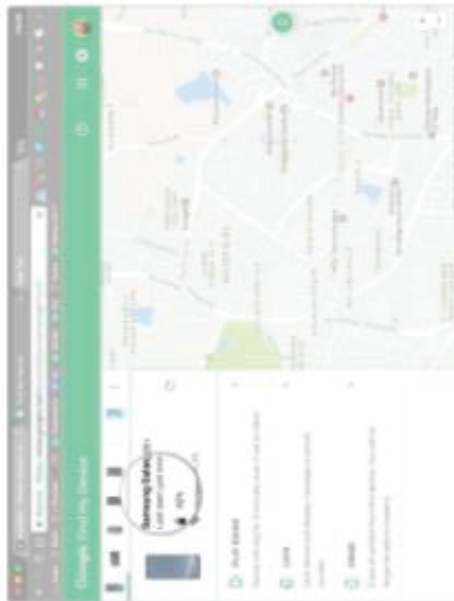
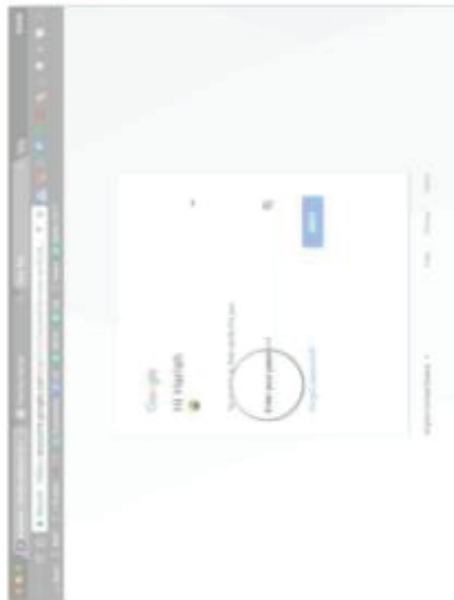
US9467838

Exemplary Supporting Evidence Regarding Accused Products

How to locate your phone over the internet

If you've lost your phone, you can remotely locate it through the Find My Device website. You'll need to sign in to the Google account that was used to set up Find My Device. It takes a few seconds, but the service should be able to track your phone. Alternatively, you can also do a Google search for "find my phone" to locate your handset.

1. Head to the **Find My Device website**.
2. Sign in to your **Google account**.
3. Check if your device is **visible**.



<https://www.androidcentral.com/find-my-device>

Exemplary Support for Google Maps:

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE












| | |
|-----------|---|
| US9467838 | <p data-bbox="191 709 228 1564">Exemplary Supporting Evidence Regarding Accused Products</p> <p data-bbox="240 1108 261 1528">COMPUTER ANDROID IPHONE & IPAD</p> <hr data-bbox="289 506 293 1564"/> <p data-bbox="342 1052 380 1541">If they have a Google Account</p> <ol data-bbox="402 657 695 1535" style="list-style-type: none"> <li data-bbox="402 856 423 1535">1. If you haven't already, add their Gmail address to your Google Contacts []. <li data-bbox="440 657 461 1535">2. On your Android phone or tablet, open the Google Maps app  and sign in. Learn how to sign in. <li data-bbox="477 1045 498 1535">3. Tap Menu  > Location sharing > Add People . <li data-bbox="514 1073 535 1535">4. Choose how long you want to share your location. <li data-bbox="552 1346 573 1535">5. Tap Select People. <ul data-bbox="589 940 610 1514" style="list-style-type: none"> <li data-bbox="589 940 610 1514">• If you're asked about your contacts, give Google Maps access. <li data-bbox="634 1192 656 1535">6. Choose who you want to share with. <li data-bbox="672 1423 693 1535">7. Tap Share. <p data-bbox="753 968 790 1541">If they don't have a Google Account</p> <ol data-bbox="813 520 938 1535" style="list-style-type: none"> <li data-bbox="813 657 834 1535">1. On your Android phone or tablet, open the Google Maps app  and sign in. Learn how to sign in. <li data-bbox="850 1045 872 1535">2. Tap Menu  > Location sharing > Add People . <li data-bbox="888 520 938 1535">3. Tap More  > Copy to clipboard. People with this link can see your location for as long as you choose, up to 72 hours. <p data-bbox="987 1213 1024 1541">Share using another app</p> <p data-bbox="1040 877 1062 1541">You can also share through messaging apps. Tap More  > select an app.</p> <p data-bbox="1122 1339 1159 1541">Stop sharing</p> <ol data-bbox="1182 877 1279 1535" style="list-style-type: none"> <li data-bbox="1182 1234 1203 1535">1. Open the Google Maps app . <li data-bbox="1219 1213 1240 1535">2. Tap Menu  > Location sharing. <li data-bbox="1256 877 1278 1535">3. Next to the person with whom you want to stop sharing, tap Remove . <p data-bbox="1295 373 1328 1564">https://support.google.com/maps/answer/7326816?co=GENIE.Platform%3DAndroid&oco=1</p> |
|-----------|---|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE









| | |
|-----------|---|
| US9467838 | <p data-bbox="185 296 228 1923">Exemplary Supporting Evidence Regarding Accused Products</p> <p data-bbox="243 296 292 1923">Share your E.T.A</p> <p data-bbox="308 296 349 1923">After you start your drive, you can share your destination, estimated arrival time, and where you are on the route.</p> <ol data-bbox="365 296 617 1923" style="list-style-type: none"> 1. Open the Google Maps app . 2. Set a driving destination. Learn how to navigate to a place. 3. After you start navigation, tap More  > Share trip progress. 4. Choose a person from the list. 5. Tap Share. 6. Location Sharing will stop when you reach your destination or stop navigating. <ul data-bbox="633 296 673 1923" style="list-style-type: none"> • To stop sharing before you arrive, tap More  > Stop sharing. <p data-bbox="730 296 779 1923">See where someone is</p> <p data-bbox="795 296 836 1923">If someone shares their location with you, you can see them on the map.</p> <ol data-bbox="852 296 974 1923" style="list-style-type: none"> 1. Open the Google Maps app . 2. Tap Menu  > Location sharing. 3. Choose someone. <ul data-bbox="990 296 1031 1923" style="list-style-type: none"> • To see an updated location, tap on a friend's icon > More  > Refresh. <p data-bbox="1071 296 1120 1923">Stop seeing someone's location</p> <ol data-bbox="1136 296 1299 1923" style="list-style-type: none"> 1. Open the Google Maps app . 2. On the map, tap their icon. 3. At the bottom, tap More . 4. To temporarily hide someone, tap Hide from map. You can stop hiding them at any time. <p data-bbox="1315 296 1356 1923">Note: You can stop someone's location from ever appearing on your map. Learn how to block another person's account.</p> <p data-bbox="1364 296 1403 1923">https://support.google.com/maps/answer/7326816?co=GENIE.Platform%3DAndroid&oco=1</p> |
|-----------|---|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE








| | |
|------------------|---|
| <p>US9467838</p> | <p>Exemplary Supporting Evidence Regarding Accused Products</p> <h2>Create a list of places</h2> <p>In Google Maps, you can create a list of places, like your favorite places or places you want to visit.</p> <p>COMPUTER ANDROID IPHONE & IPAD</p> <h3>Make a new list</h3> <ol style="list-style-type: none">1. On your Android phone or tablet, open the Google Maps app .2. Tap Menu  > Your places > Saved.3. In the bottom right, tap Add .4. Enter a name and description.5. Tap Save. <h3>Save a place to a list</h3> <ol style="list-style-type: none">1. Open the Google Maps app .2. Search for a place or tap it on the map.3. At the bottom, tap the place's name or address.4. Tap Save.5. Choose a list. To create a new list, tap New list . <h3>See your lists</h3> <ol style="list-style-type: none">1. Open the Google Maps app .2. Tap Menu  > Your places > Saved. <p>https://support.google.com/maps/answer/7280933?hl=en&ref_topic=7301134&co=GENIE.Platform%3DAndroid&oco=1</p> |
|------------------|---|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE





| | |
|-----------|---|
| US9467838 | <p data-bbox="191 1570 228 1919">Exemplary Supporting Evidence Regarding Accused Products</p> <h2 data-bbox="245 1192 282 1528">Hide or share lists</h2> <p data-bbox="310 1163 337 1528">Note: You can't share starred places.</p> <ol data-bbox="365 630 516 1528" style="list-style-type: none"> <li data-bbox="365 1184 393 1528">1. Open the Google Maps app . <li data-bbox="407 1108 435 1528">2. Tap Menu  > Your places > Saved. <li data-bbox="449 823 477 1528">3. Next to the list you want to share, tap More  > choose an option: <ul data-bbox="492 394 633 1495" style="list-style-type: none"> <li data-bbox="492 630 519 1495">• Hide/Show on your map: Display or hide your saved places when looking at the map. <li data-bbox="534 1016 561 1495">• Share list: Allow others to see your saved list. <li data-bbox="576 394 633 1495">• Sharing options: You can make your list public, private, or shared. To let anyone with the link see your list, tap Shared. To let anyone find and follow your list, tap Public. <h2 data-bbox="699 1310 737 1528">Follow a list</h2> <p data-bbox="764 348 824 1528">If you follow a list made by someone else, their saved places will show up in Your Places. The places will also appear as suggested locations in Google Maps.</p> <h2 data-bbox="878 1163 915 1528">Follow a list using a link</h2> <ol data-bbox="943 722 1045 1528" style="list-style-type: none"> <li data-bbox="943 1117 971 1528">1. Tap on the link you received to open it. <li data-bbox="985 806 1013 1528">2. Tap Follow. This list will now be added to the group of lists you follow. <li data-bbox="1027 722 1045 1528">3. optional: To unfollow a list someone shared with you, tap the list > Following. <h2 data-bbox="1099 1150 1136 1528">See lists made by others</h2> <p data-bbox="1164 743 1192 1528">If a user has any Google Maps lists that were made public, you can follow them.</p> <ol data-bbox="1219 197 1321 1528" style="list-style-type: none"> <li data-bbox="1219 940 1247 1528">1. Tap on the name of a user whose list you want to follow. <li data-bbox="1261 1402 1289 1528">2. Tap Lists. <li data-bbox="1304 940 1321 1528">3. Tap on the list you want to follow > More  > Follow. <p data-bbox="1336 197 1396 1562">https://support.google.com/maps/answer/7280933?hl=en&ref_topic=7301134&co=GENIE.Platform%3DAndroid&oco=1</p> |
|-----------|---|

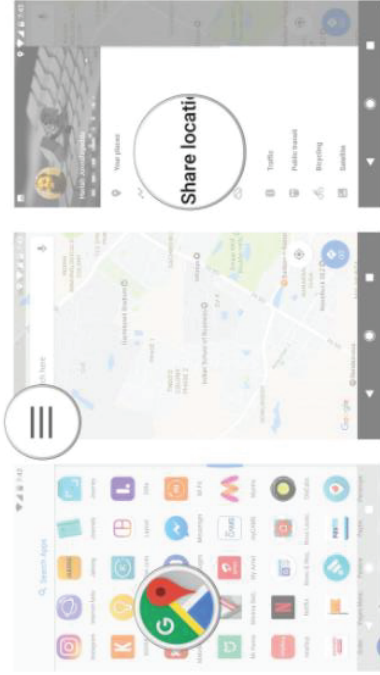
Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

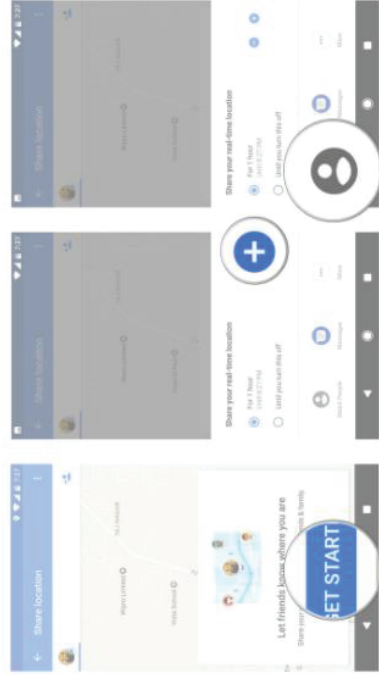
Exemplary Supporting Evidence Regarding Accused Products

How to share your location in Google Maps

1. Open Google Maps from the app drawer or the home screen.
2. Tap the hamburger menu (the three horizontal lines) on the top left corner of the screen.
3. Select Share location.



4. Tap Get Started.
5. Use the + icon to select a time period or select the **Until you turn this off** setting to share your location indefinitely.
6. Tap **Select People**.



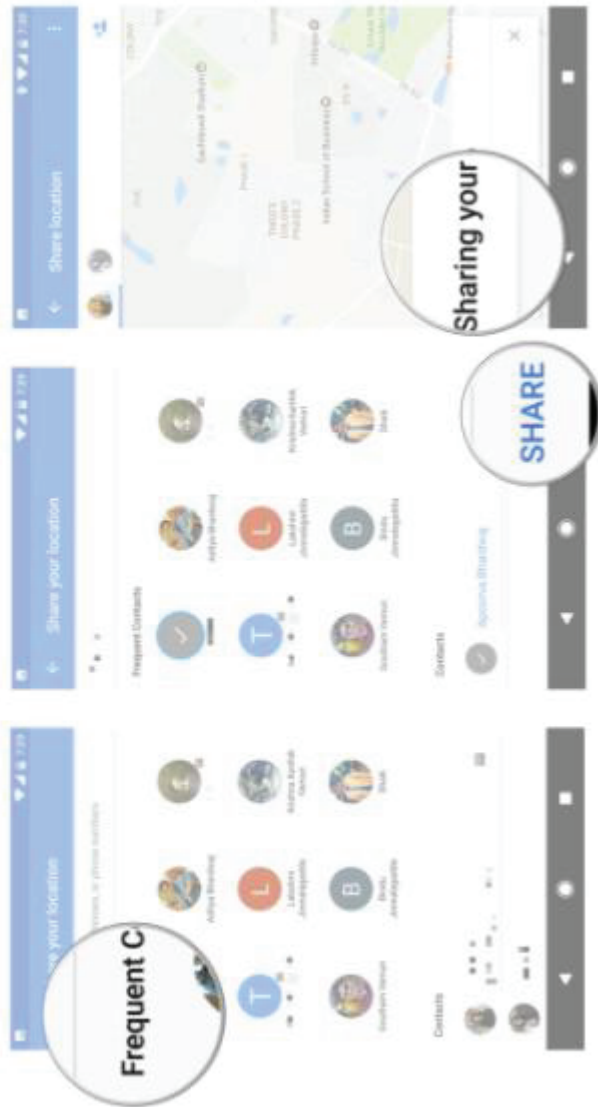
<https://www.androidcentral.com/how-share-location-google-maps>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products

- 7. You'll see a list of your frequent contacts at the top, along with a full list of contacts. Pick the contacts by tapping their name.
- 8. Once you've selected the contacts you want to share your location to, tap Share.
- 9. You'll see a message saying that the selected contact can view your location.



<https://www.androidcentral.com/how-share-location-google-maps>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

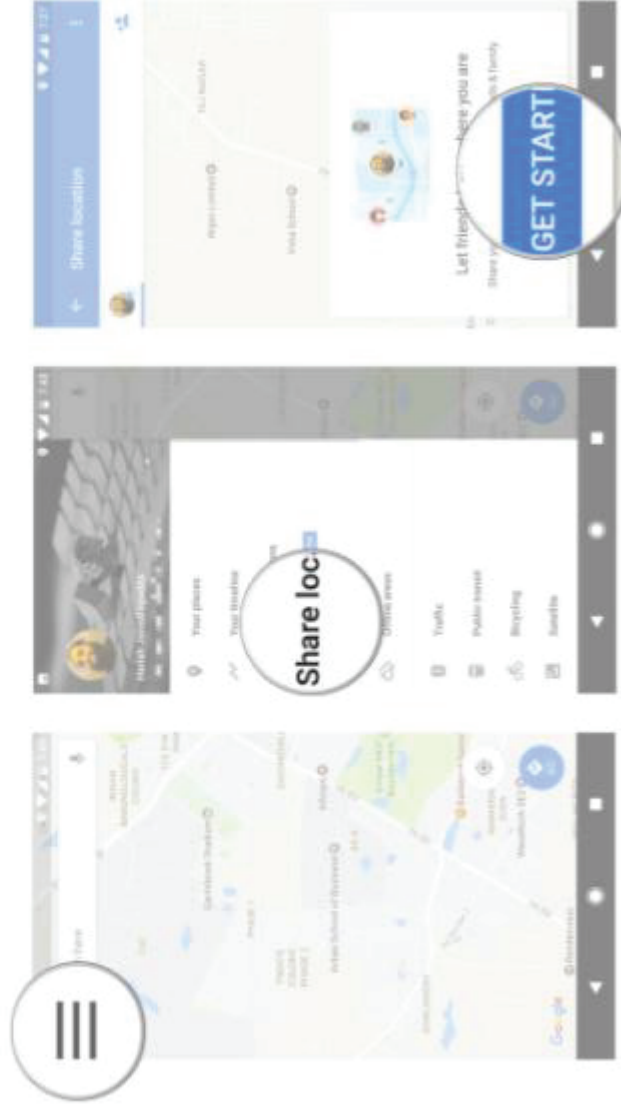
US9467838

Exemplary Supporting Evidence Regarding Accused Products

How to create a shareable link

You can also create a link and use it to share your location easily. Here's how to do it:

1. Tap the hamburger menu on the top left corner of the screen.
2. Select Share location.
3. Tap Get Started.



<https://www.androidcentral.com/how-share-location-google-maps>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| | |
|------------------|---|
| <p>US9467838</p> | <p>Exemplary Supporting Evidence Regarding Accused Products</p> |
| | <p>4. Select the amount of time you want to share your location.</p> <p>5. Tap More.</p> <p>6. Select your app of choice to create and send a unique URL that broadcasts your current location. You can email it, send the link via Messenger, or even tweet it to the intended recipient.</p>  <p>https://www.androidcentral.com/how-share-location-google-maps</p> |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

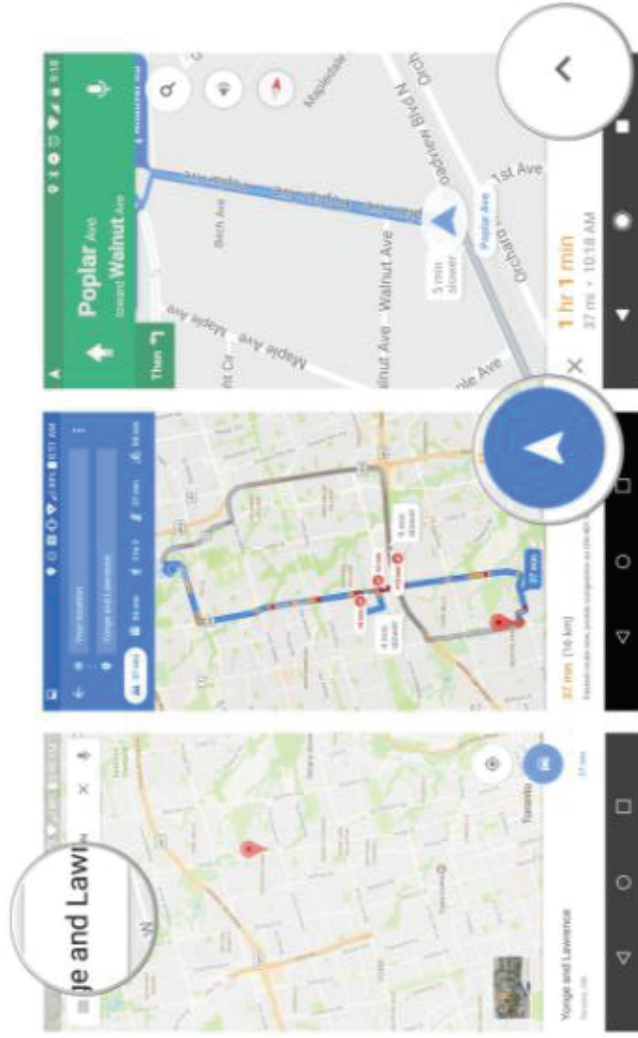
US9467838

Exemplary Supporting Evidence Regarding Accused Products

How to share your navigation directions while you walk, drive or transit

One of the best ways to share your location with someone is during a drive, walk or transit. If you're meeting a friend or family member somewhere, or navigating towards their home, Google Maps lets you share your location with them for the duration of the trip. It's magic!

1. In the search bar enter your destination.
2. Pick your navigation type (drive, transit, walk) and press the blue navigate button.
3. Tap the arrow next to the time-to-destination number at the bottom of the screen.



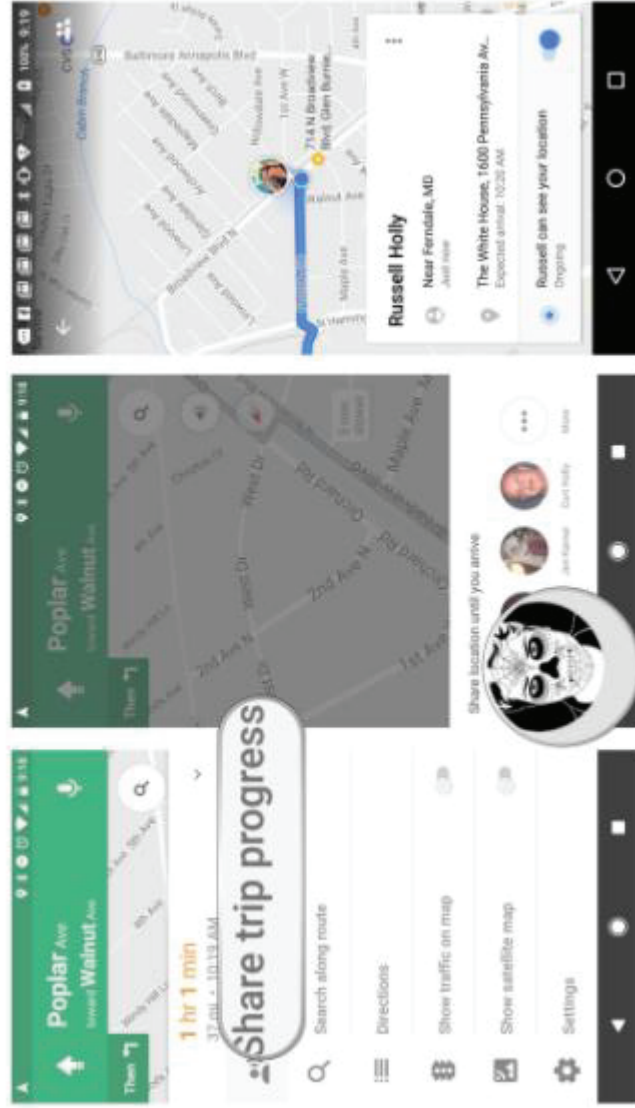
<https://www.androidcentral.com/how-share-location-google-maps>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products

- 4. Tap Share trip progress.
- 5. Choose one or more contacts to share trip progress.



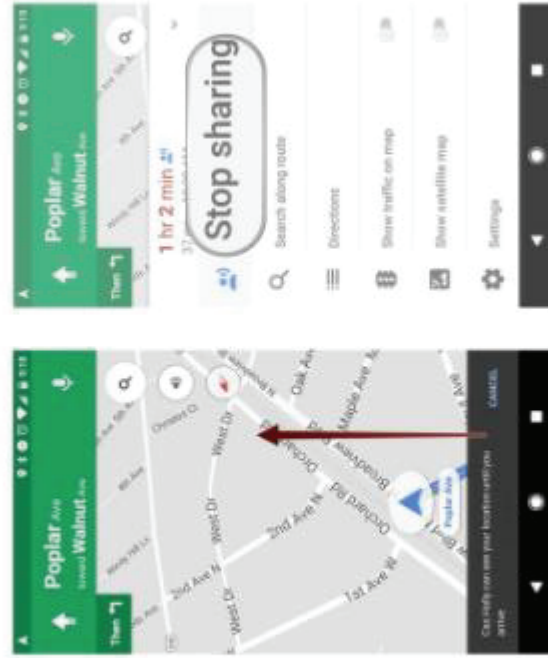
You can also stop sharing your location with someone before a trip ends.
<https://www.androidcentral.com/how-share-location-google-maps>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products

1. Tap the arrow next to the time-to-destination number at the bottom of the screen.
2. Tap Stop sharing.



That's it!

Are you excited that location sharing is back in Google Maps? How often do you use the feature?
<https://www.androidcentral.com/how-share-location-google-maps>

As shown below, a group may also be defined within Google Contacts.

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE











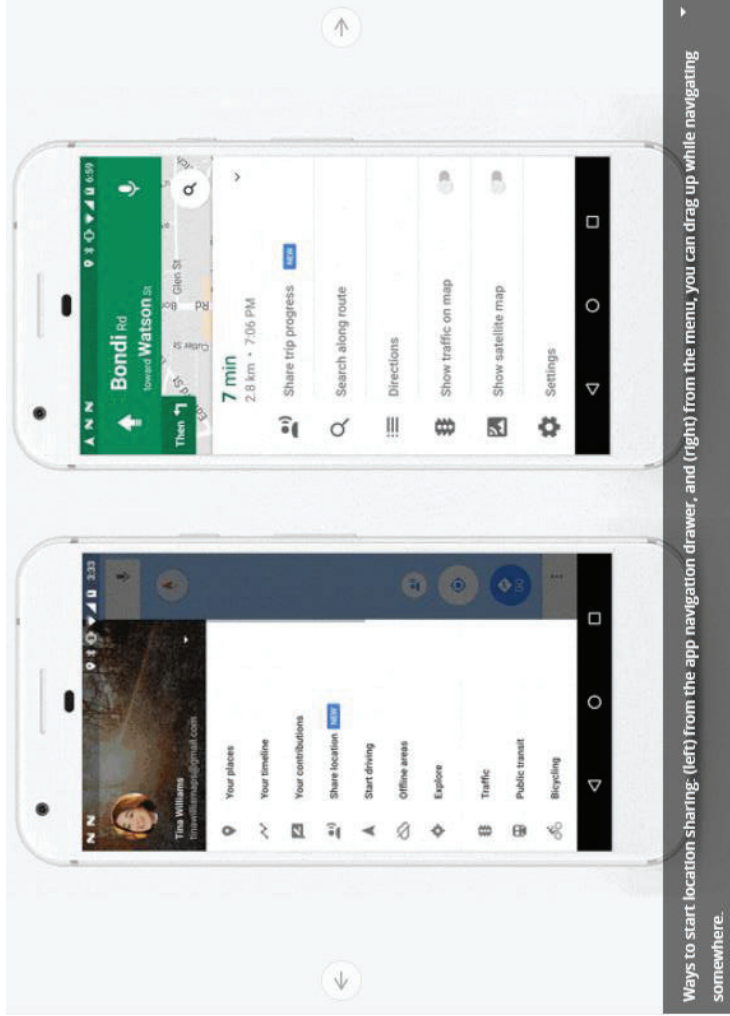
| | |
|-----------|---|
| US9467838 | <p data-bbox="191 247 228 1822">Exemplary Supporting Evidence Regarding Accused Products</p> <h2 data-bbox="235 304 284 1822">See your contacts</h2> <ol data-bbox="300 304 381 1822" style="list-style-type: none"> <li data-bbox="300 304 341 1822">1. Open your device's Contacts app . <li data-bbox="349 304 381 1822">2. Tap Menu . <ul data-bbox="406 304 527 1822" style="list-style-type: none"> <li data-bbox="406 304 446 1822">• See contacts by label: Choose a label from the list. <li data-bbox="454 304 495 1822">• See contacts for another account: Tap Down arrow  > pick an account. <li data-bbox="503 304 527 1822">• See the contacts for all your accounts: Choose All contacts. <p data-bbox="535 304 576 1822">Tip: If you have multiple contacts with the same information, the information will be grouped into one contact.</p> <ul data-bbox="584 304 625 1822" style="list-style-type: none"> <li data-bbox="584 304 625 1822">• See your Google Account contacts on the web: Go to Google Contacts . <p data-bbox="633 304 673 1822">https://support.google.com/android/answer/6118731?hl=en&ref_topic=6118711</p> <h2 data-bbox="714 304 763 1822">Label your contacts</h2> <p data-bbox="779 304 803 1822">You can group contacts together using labels.</p> <ol data-bbox="828 304 950 1822" style="list-style-type: none"> <li data-bbox="828 304 868 1822">1. Open your device's Contacts app . <li data-bbox="876 304 917 1822">2. Tap Menu  > Create label. <li data-bbox="925 304 950 1822">3. Enter a label name and tap Ok. <ul data-bbox="974 304 1047 1822" style="list-style-type: none"> <li data-bbox="974 304 1015 1822">• Add one contact to a label: Tap Add contact  > choose a contact. <li data-bbox="1023 304 1047 1822">• Add multiple contacts to a label: Tap Add contact  > touch and hold a contact > tap the other contacts > tap Add. <p data-bbox="1055 304 1096 1822">https://support.google.com/android/answer/6118731?hl=en&ref_topic=6118711</p> <h2 data-bbox="1136 304 1185 1822">Share your contacts</h2> <ol data-bbox="1201 304 1372 1822" style="list-style-type: none"> <li data-bbox="1201 304 1242 1822">1. Open your device's Contacts app . <li data-bbox="1250 304 1291 1822">2. Tap a contact in the list. <li data-bbox="1299 304 1339 1822">3. Tap More  > Share. <li data-bbox="1347 304 1372 1822">4. Choose how you want to share the contact. <p data-bbox="1380 304 1421 1822">https://support.google.com/android/answer/6118731?hl=en&ref_topic=6118711</p> |
|-----------|---|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products



<https://arstechnica.com/gadgets/2017/03/location-sharing-finally-returns-to-google-maps/>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products

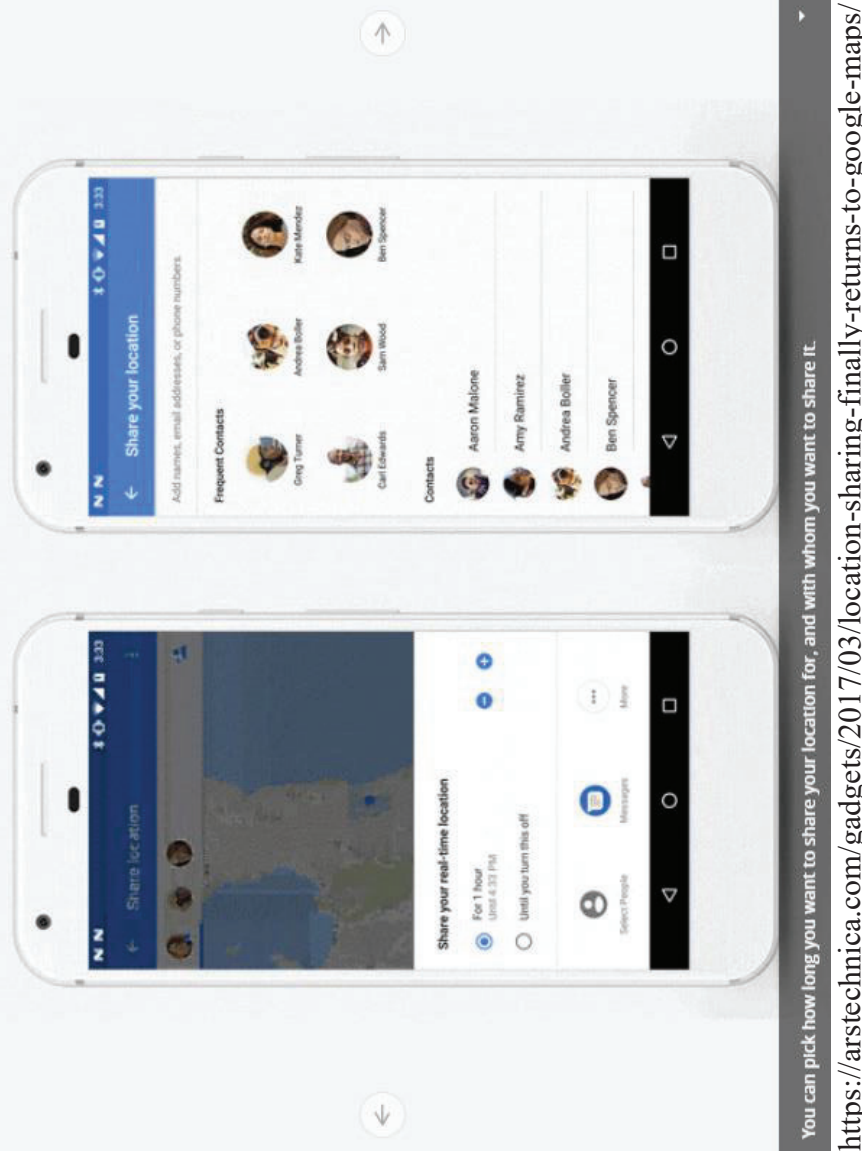
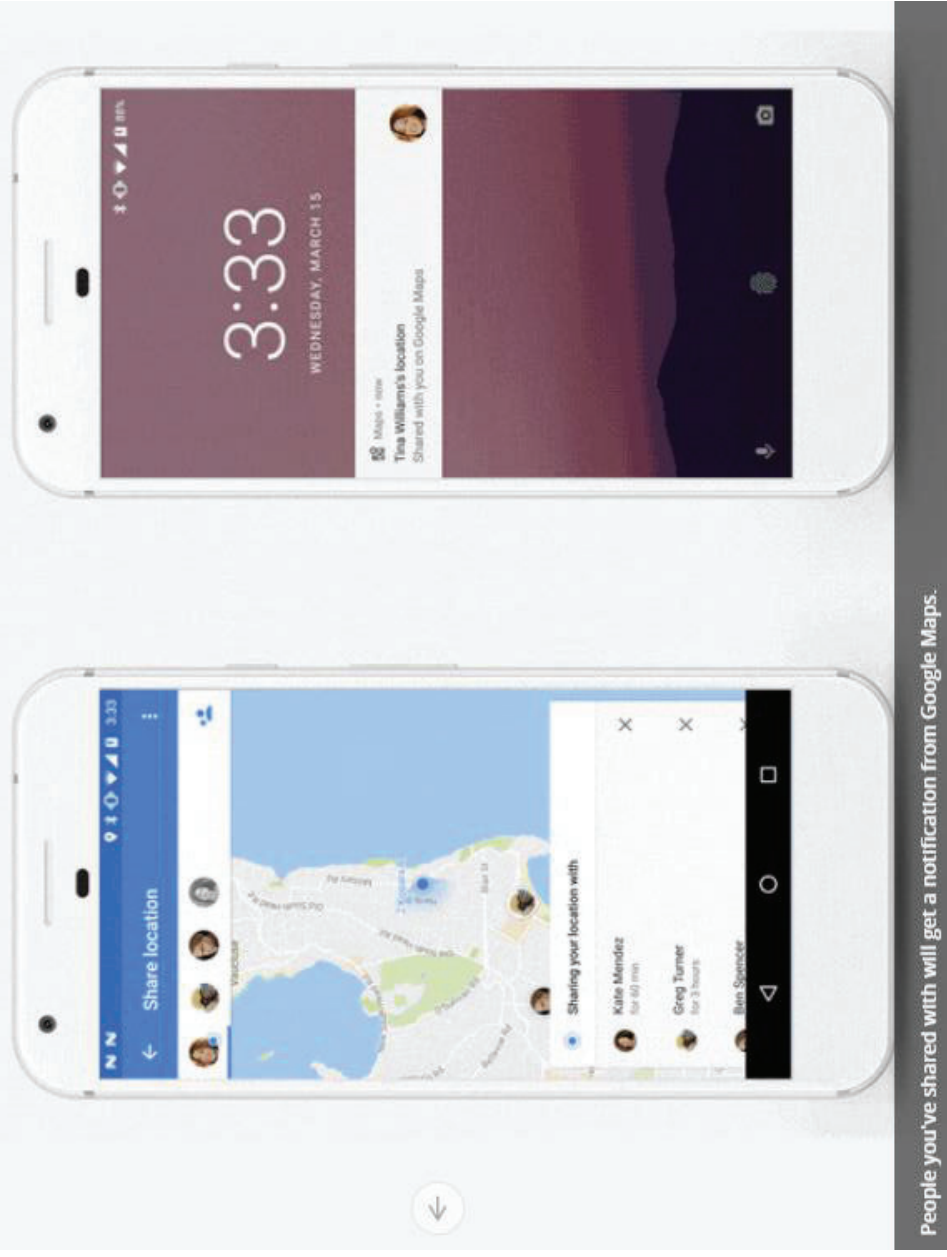


Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products



<https://arstechnica.com/gadgets/2017/03/location-sharing-finally-returns-to-google-maps/>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

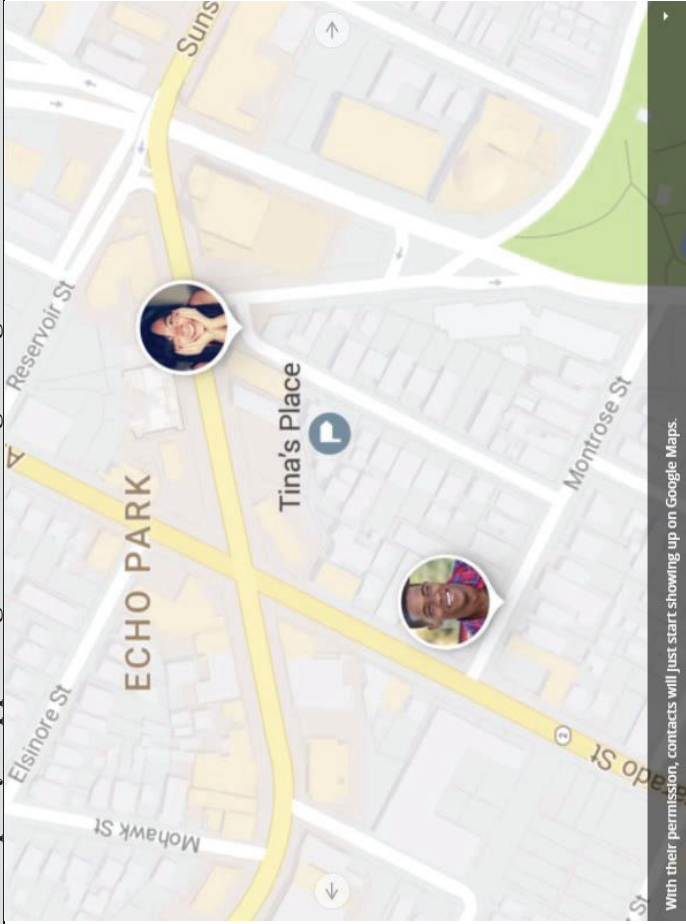
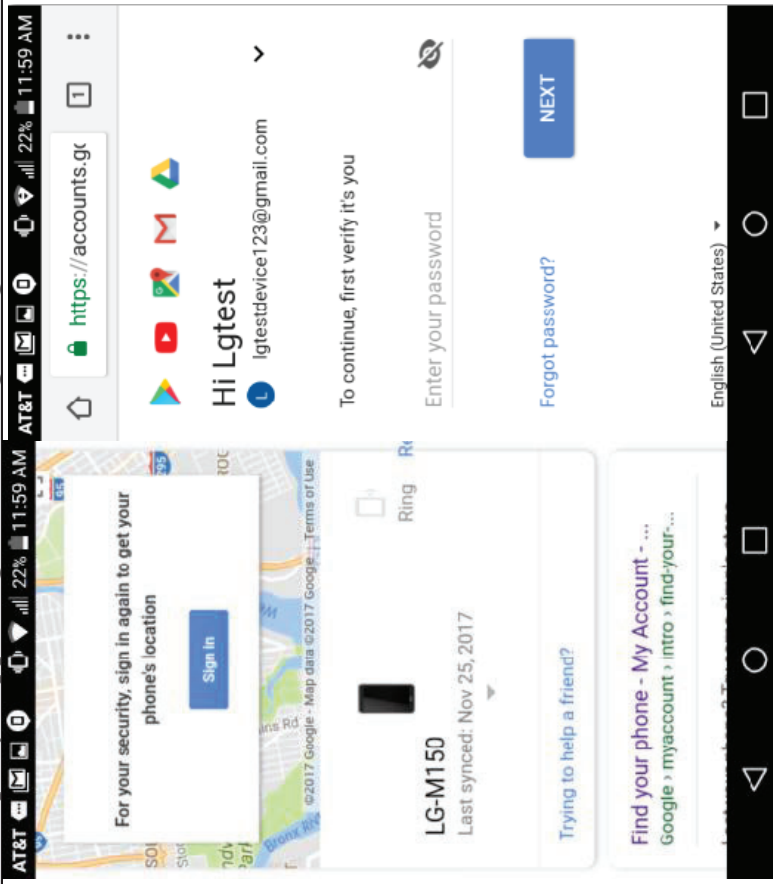
| | |
|-------------------------|--|
| <p>US9467838</p> | <p>Exemplary Supporting Evidence Regarding Accused Products</p>  <p>The screenshot shows a Google Maps view of the Echo Park neighborhood in Los Angeles. Two circular location pins are overlaid on the map, each containing a profile picture of a person. One pin is located near the intersection of Reservoir St and Echo Park Blvd, and the other is near Tina's Place. A parking 'P' icon is also visible near Tina's Place. The map includes street names such as Mohawk St, Elsinore St, Reservoir St, Montrose St, and Echo Park Blvd. A text overlay at the bottom of the map reads: "With their permission, contacts will just start showing up on Google Maps." Below the map, a URL is provided: https://arstechnica.com/gadgets/2017/03/location-sharing-finally-returns-to-google-maps/</p> <p><u>Exemplary Find My Device Screenshots:</u></p> |
|-------------------------|--|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products



Exemplary Google Maps Screenshots:

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

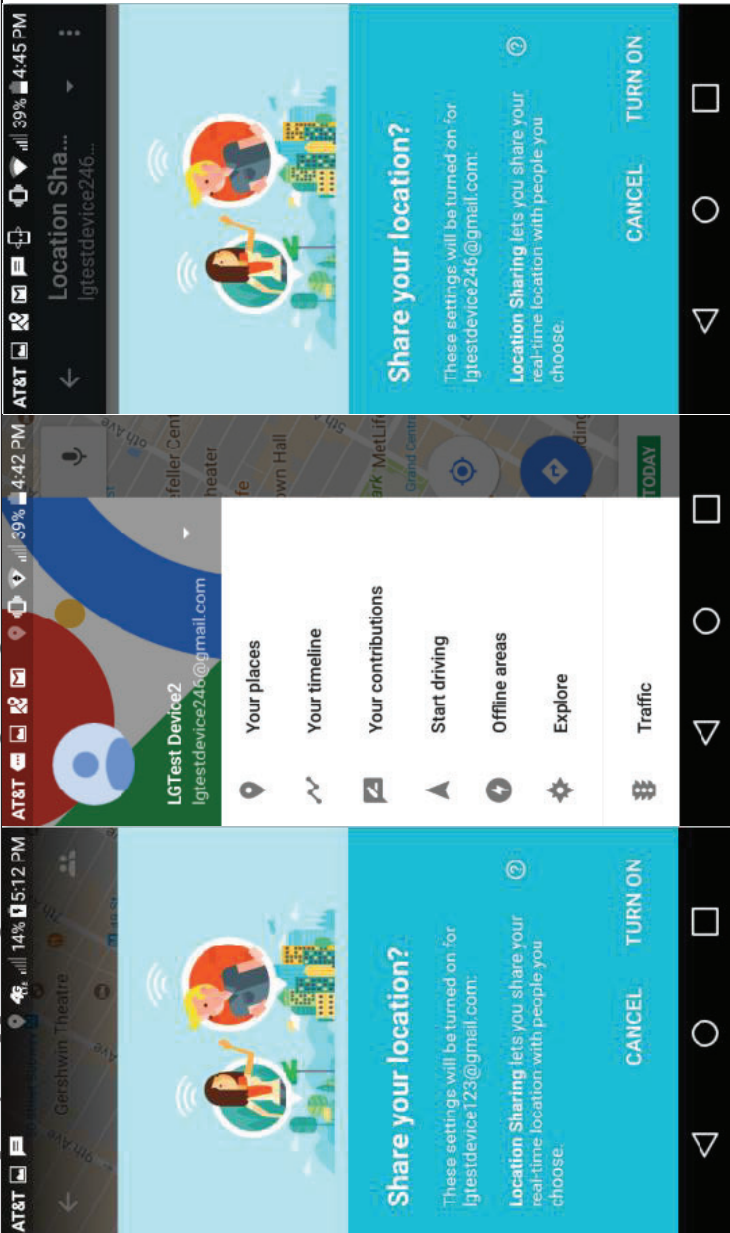
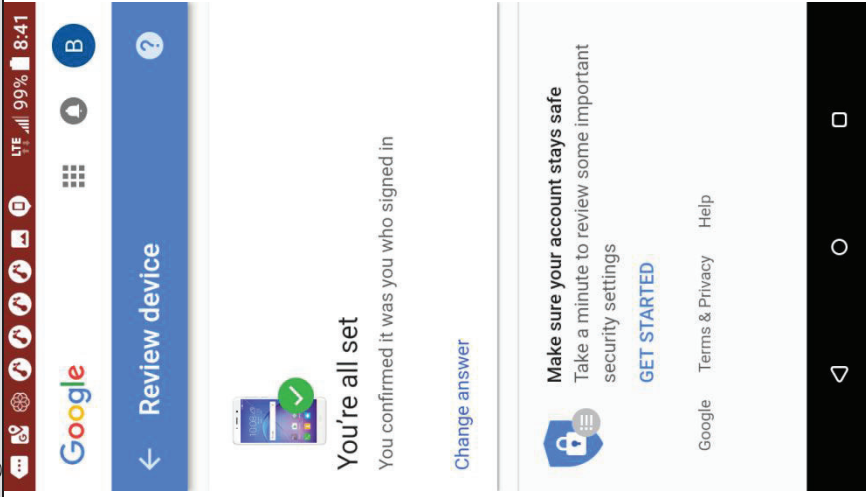
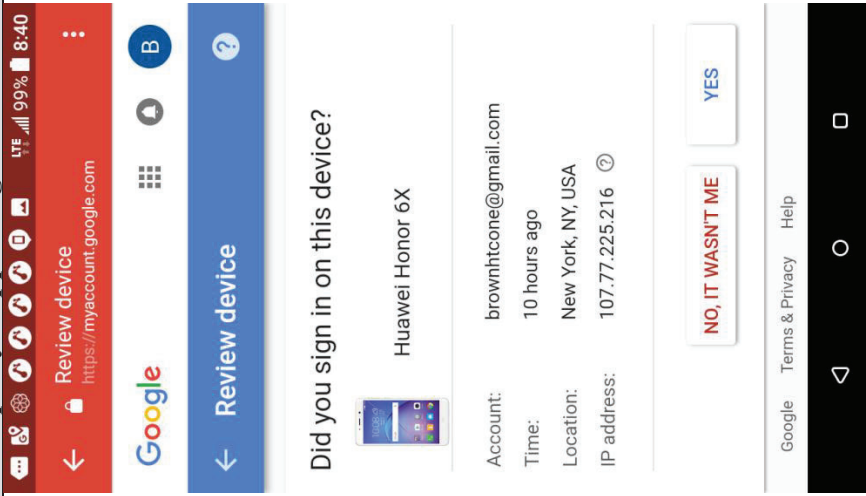
| | |
|------------------|--|
| <p>US9467838</p> | <p>Exemplary Supporting Evidence Regarding Accused Products</p>  <p>The Accused Products require a user to join the corresponding network by: signing-in to the device with an identifier (e.g., Google Account) or linking the device to the identifier by remote means. The message received by the first device relates to the second device joining into a group with the first device.</p> |
|------------------|--|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products



e.g., above, the device receives a message indicating that the second device has joined the group, i.e. the google account.

Exemplary Source Code:

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| | |
|---|--|
| <p>US9467838</p> | <p><u>Exemplary Supporting Evidence Regarding Accused Products</u></p> |
| <p>The above functionality is performed at least in part by the following publicly available source code and/or source code that invokes or is invoked by the following source code (or a substantially similar copy compiled and loaded onto the Accused Products by ZTE): AGIS reserves the right to amend these contentions to include additional source code as discovery progresses and as additional source code is made available.</p> | <p style="text-align: center;">Contacts Provider</p> <p>The Contacts Provider is a powerful and flexible Android component that manages the device's central repository of data about people. The Contacts Provider is the source of data you see in the device's contacts application, and you can also access its data in your own application and transfer data between the device and online services. The provider accommodates a wide range of data sources and tries to manage as much data as possible for each person, with the result that its organization is complex. Because of this, the provider's API includes an extensive set of contract classes and interfaces that facilitate both data retrieval and modification.</p> <p>This guide describes the following:</p> <ul style="list-style-type: none"> • The basic provider structure. • How to retrieve data from the provider. • How to modify data in the provider. • How to write a sync adapter for synchronizing data from your server to the Contacts Provider. <p>https://developer.android.com/guide/topics/providers/contacts-provider.html</p> |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products

Overview

ContactsContract defines an extensible database of contact-related information. Contact information is stored in a three-tier data model:

- A row in the `ContactsContract.Data` table can store any kind of personal data, such as a phone number or email addresses. The set of data kinds that can be stored in this table is open-ended. There is a predefined set of common kinds, but any application can add its own data kinds.
- A row in the `ContactsContract.RawContacts` table represents a set of data describing a person and associated with a single account (for example, one of the user's Gmail accounts).
- A row in the `ContactsContract.Contacts` table represents an aggregate of one or more `RawContacts` presumably describing the same person. When data in or associated with the `RawContacts` table is changed, the affected aggregate contacts are updated as necessary.

Other tables include:

- `ContactsContract.Groups`, which contains information about raw contact groups such as Gmail contact groups. The current API does not support the notion of groups spanning multiple accounts.
- `ContactsContract.StatusUpdates`, which contains social status updates including IMI availability.
- `ContactsContract.AggregationExceptions`, which is used for manual aggregation and disaggregation of raw contacts
- `ContactsContract.Settings`, which contains visibility and sync settings for accounts and groups.
- `ContactsContract.SyncState`, which contains free-form data maintained on behalf of sync adapters
- `ContactsContract.PhoneLookup`, which is used for quick caller-ID lookup

<https://developer.android.com/reference/android/provider/ContactsContract.html>

Data

As noted previously, the data for a raw contact is stored in a `ContactsContract.Data` row that is linked to the raw contact's `_id` value. This allows a single raw contact to have multiple instances of the same type of data such as email addresses or phone numbers. For example, if "Thomas Higginson" for `emilyd@gmail.com` (the raw contact row for Thomas Higginson associated with the Google account `emilyd@gmail.com`) has a home email address of `thigg@gmail.com` and a work email address of `thomas.higginson@gmail.com`, the `Contacts Provider` stores the two email address rows and links them both to the raw contact.

Notice that different types of data are stored in this single table. Display name, phone number, email, postal address, photo, and website detail rows are all found in the `ContactsContract.Data` table. To help manage this, the `ContactsContract.Data` table has some columns with descriptive names, and others with generic names. The contents of a descriptive-name column have the same meaning regardless of the type of data in the row, while the contents of a generic-name column have different meanings depending on the type of data.

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| <p>US9467838</p> | <p>Exemplary Supporting Evidence Regarding Accused Products https://developer.android.com/guide/topics/providers/contacts-provider.html</p> <table border="1" data-bbox="370 327 1044 1556"> <thead> <tr> <th data-bbox="370 327 415 478">Task</th> <th data-bbox="370 478 415 590">Action</th> <th data-bbox="370 590 415 779">Data</th> <th data-bbox="370 779 415 890">MIME type</th> <th data-bbox="370 890 415 1350">Notes</th> </tr> </thead> <tbody> <tr> <td data-bbox="415 327 548 478">Pick a contact from a list</td> <td data-bbox="415 478 548 590">ACTION_PICK</td> <td data-bbox="415 590 548 779"> One of: <ul style="list-style-type: none"> • <code>Contacts.CONTENT_URI</code>, which displays a list of contacts. • <code>Phone.CONTENT_URI</code>, which displays a list of phone numbers for a raw contact. • <code>StructuredPostal.CONTENT_URI</code>, which displays a list of postal addresses for a raw contact. • <code>Email.CONTENT_URI</code>, which displays a list of email addresses for a raw contact. </td> <td data-bbox="415 779 548 890">Not used</td> <td data-bbox="415 890 548 1350"> Displays a list of raw contacts or a list of data from a raw contact, depending on the content URI type you supply. Call <code>startActivityForResult()</code>, which returns the content URI of the selected row. The form of the URI is the table's content URI with the row's <code>LOOKUP_ID</code> appended to it. The device's contacts app delegates read and write permissions to this content URI for the life of your activity. See the Content Provider Basics guide for more details. </td> </tr> </tbody> </table> <p>https://developer.android.com/guide/topics/providers/contacts-provider.html</p> | Task | Action | Data | MIME type | Notes | Pick a contact from a list | ACTION_PICK | One of: <ul style="list-style-type: none"> • <code>Contacts.CONTENT_URI</code>, which displays a list of contacts. • <code>Phone.CONTENT_URI</code>, which displays a list of phone numbers for a raw contact. • <code>StructuredPostal.CONTENT_URI</code>, which displays a list of postal addresses for a raw contact. • <code>Email.CONTENT_URI</code>, which displays a list of email addresses for a raw contact. | Not used | Displays a list of raw contacts or a list of data from a raw contact, depending on the content URI type you supply. Call <code>startActivityForResult()</code> , which returns the content URI of the selected row. The form of the URI is the table's content URI with the row's <code>LOOKUP_ID</code> appended to it. The device's contacts app delegates read and write permissions to this content URI for the life of your activity. See the Content Provider Basics guide for more details. |
|----------------------------|---|--|-----------|---|-----------|-------|----------------------------|-------------|--|----------|---|
| Task | Action | Data | MIME type | Notes | | | | | | | |
| Pick a contact from a list | ACTION_PICK | One of: <ul style="list-style-type: none"> • <code>Contacts.CONTENT_URI</code>, which displays a list of contacts. • <code>Phone.CONTENT_URI</code>, which displays a list of phone numbers for a raw contact. • <code>StructuredPostal.CONTENT_URI</code>, which displays a list of postal addresses for a raw contact. • <code>Email.CONTENT_URI</code>, which displays a list of email addresses for a raw contact. | Not used | Displays a list of raw contacts or a list of data from a raw contact, depending on the content URI type you supply. Call <code>startActivityForResult()</code> , which returns the content URI of the selected row. The form of the URI is the table's content URI with the row's <code>LOOKUP_ID</code> appended to it. The device's contacts app delegates read and write permissions to this content URI for the life of your activity. See the Content Provider Basics guide for more details. | | | | | | | |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| US9467838 | Exemplary Supporting Evidence Regarding Accused Products |
|-----------|---|
| 25 | * Parsed form of the intent sent to the Contacts application. |
| 26 | */ |
| 27 | public class ContactsRequest { |
| 28 | |
| 29 | /** Default mode: browse contacts */ |
| 30 | public static final int ACTION_DEFAULT = 10; |
| 31 | |
| 32 | /** Show all contacts */ |
| 33 | public static final int ACTION_ALL_CONTACTS = 15; |
| 34 | |
| 35 | /** Show all contacts with phone numbers */ |
| 36 | public static final int ACTION_CONTACTS_WITH_PHONES = 17; |
| 37 | |
| 38 | /** Show contents of a specific group */ |
| 39 | public static final int ACTION_GROUP = 20; |
| 40 | |
| 41 | /** Show all starred contacts */ |
| 42 | public static final int ACTION_STARRED = 30; |
| 43 | |
| 44 | /** Show frequently contacted contacts */ |
| 45 | public static final int ACTION_FREQUENT = 40; |
| 46 | |
| 47 | /** Show starred and the frequent */ |
| 48 | public static final int ACTION_STREQUENT = 50; |
| 49 | |
| 50 | /** Show all contacts and pick them when clicking */ |
| 51 | public static final int ACTION_PICK_CONTACT = 60; |
| 52 | |
| 53 | /** Show all contacts as well as the option to create a new one */ |
| 54 | public static final int ACTION_PICK_OR_CREATE_CONTACT = 70; |
| 55 | |
| 56 | /** Show all contacts and pick them for edit when clicking, and allow creating a new contact */ |
| 57 | public static final int ACTION_INSERT_OR_EDIT_CONTACT = 80; |
| | https://android.googlesource.com/platform/packages/apps/Contacts/+nougat-mr1-release/src/com/android/contacts/list/ContactsRequest.java |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| US9467838 | Exemplary Supporting Evidence Regarding Accused Products |
|-----------|--|
| 59 | /** Show all phone numbers and pick them when clicking */ |
| 60 | public static final int ACTION_PICK_PHONE = 90; |
| 61 | /** Show all postal addresses and pick them when clicking */ |
| 62 | public static final int ACTION_PICK_POSTAL = 100; |
| 63 | /** Show all postal addresses and pick them when clicking */ |
| 64 | public static final int ACTION_PICK_EMAIL = 105; |
| 65 | /** Show all contacts and create a shortcut for the picked contact */ |
| 66 | public static final int ACTION_CREATE_SHORTCUT_CONTACT = 110; |
| 67 | /** Show all phone numbers and create a call shortcut for the picked number */ |
| 68 | public static final int ACTION_CREATE_SHORTCUT_CALL = 120; |
| 69 | /** Show all phone numbers and create an SMS shortcut for the picked number */ |
| 70 | public static final int ACTION_CREATE_SHORTCUT_SMS = 130; |
| 71 | /** Show all contacts and activate the specified one */ |
| 72 | public static final int ACTION_VIEW_CONTACT = 140; |
| 73 | /** Show contacts recommended for joining with a specified target contact */ |
| 74 | public static final int ACTION_PICK_JOIN = 150; |
| 75 | https://android.googlesource.com/platform/packages/apps/Contacts/+nougat-mr1- |
| 76 | release/src/com/android/contacts/list/ContactsRequest.java |
| 77 | |
| 78 | |
| 79 | |
| 80 | |
| 81 | |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| US9467838 | Exemplary Supporting Evidence Regarding Accused Products |
|-----------|---|
| 104 | * Displays a list to browse contacts. |
| 105 | */ |
| 106 | public class PeopleActivity extends ContactsActivity implements |
| 107 | View.OnCreateContextMenuListener, |
| 108 | View.OnClickListener, |
| 109 | ActionBarAdapter.Listener, |
| 110 | DialogManager.DialogShowingViewActivity, |
| 111 | ContactListFilterController.ContactListFilterListener, |
| 112 | ProviderStatusListener, |
| 113 | MultiContactDeleteListener, |
| 114 | JoinContactsListener { |
| | https://android.googlesource.com/platform/packages/apps/Contacts/+/nougat-mr1-release/src/com/android/contacts/activities/PeopleActivity.java |
| 145 | * Showing a list of Contacts. Also used for showing search results in search mode. |
| 146 | */ |
| 147 | private MultiSelectContactsListFragment mAllFragment; |
| 148 | private ContactTileListFragment mFavoritesFragment; |
| | https://android.googlesource.com/platform/packages/apps/Contacts/+/nougat-mr1-release/src/com/android/contacts/activities/PeopleActivity.java |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| US9467838 | Exemplary Supporting Evidence Regarding Accused Products |
|--|--|
| <pre> 458 private void configureFragments(boolean fromRequest) { 459 if (fromRequest) { 460 ContactListFilter filter = null; 461 int actionCode = mRequest.getActionCode(); 462 boolean searchMode = mRequest.isSearchMode(); 463 final int tabToOpen; 464 switch (actionCode) { 465 case ContactsRequest.ACTION_ALL_CONTACTS: 466 filter = ContactListFilter.createFilterWithType(467 ContactListFilter.FILTER_TYPE_ALL_ACCOUNTS); 468 tabToOpen = TabState.ALL; 469 break; 470 case ContactsRequest.ACTION_CONTACTS_WITH_PHONES: 471 filter = ContactListFilter.createFilterWithType(472 ContactListFilter.FILTER_TYPE_WITH_PHONE_NUMBERS_ONLY); 473 tabToOpen = TabState.ALL; 474 break; 475 case ContactsRequest.ACTION_FREQUENT: 476 case ContactsRequest.ACTION_STREQUENT: 477 case ContactsRequest.ACTION_STARRED: 478 tabToOpen = TabState.FAVORITES; 479 break; 480 case ContactsRequest.ACTION_VIEW_CONTACT: 481 tabToOpen = TabState.ALL; 482 break; 483 default: 484 tabToOpen = -1; 485 break; 486 } 487 } </pre> | <p>https://android.googlesource.com/platform/packages/apps/Contacts/+nougat-mr1-</p> |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| US9467838 | Exemplary Supporting Evidence Regarding Accused Products |
|-----------|---|
| | release/src/com/android/contacts/activities/PeopleActivity.java |
| | |
| 488 | if (tabToOpen != -1) { |
| 489 | mActionBarAdapter.setCurrentTab(tabToOpen); |
| 490 | } |
| 491 | |
| 492 | if (filter != null) { |
| 493 | mContactListFilterController.setContactListFilter(filter, false); |
| 494 | searchMode = false; |
| 495 | } |
| 496 | |
| 497 | if (mRequest.getContactUri() != null) { |
| 498 | searchMode = false; |
| 499 | } |
| 500 | |
| 501 | mActionBarAdapter.setSearchMode(searchMode); |
| 502 | configureContactListFragmentForRequest(); |
| 503 | } |
| 504 | |
| 505 | configureContactListFragment(); |
| 506 | |
| 507 | invalidateOptionsMenuIfNeeded(); |
| 508 | } |
| | |
| | https://android.googlesource.com/platform/packages/apps/Contacts/+nougat-mr1- |
| | release/src/com/android/contacts/activities/PeopleActivity.java |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| US9467838 | Exemplary Supporting Evidence Regarding Accused Products |
|-----------|--|
| | <pre> 35 public class ProfileAndContactsLoader extends CursorLoader { 36 37 private boolean mLoadProfile; 38 39 private String[] mProjection; 40 41 private Uri mExtraUri; 42 private String[] mExtraProjection; 43 private String mExtraSelection; 44 private String[] mExtraSelectionArgs; 45 private boolean mMergeExtraContactsAfterPrimary; 46 47 public ProfileAndContactsLoader(Context context) { 48 super(context); 49 } </pre> <p data-bbox="836 388 917 1564">https://android.googlesource.com/platform/packages/apps/ContactsCommon/+nougat-mr1-release/src/com/android/contacts/common/list/ProfileAndContactsLoader.java</p> |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products

```

35 public final class GroupMemberLoader extends CursorLoader {
36
37     public static class GroupEditorQuery {
38         private static final String[] PROJECTION = new String[] {
39             Data.CONTACT_ID, // 0
40             Data.RAW_CONTACT_ID, // 1
41             Data.DISPLAY_NAME_PRIMARY, // 2
42             Data.PHOTO_URI, // 3
43             Data.LOOKUP_KEY, // 4
44         };
45
46         public static final int CONTACT_ID = 0;
47         public static final int RAW_CONTACT_ID = 1;
48         public static final int CONTACT_DISPLAY_NAME_PRIMARY = 2;
49         public static final int CONTACT_PHOTO_URI = 3;
50         public static final int CONTACT_LOOKUP_KEY = 4;
51     }
52
53     public static class GroupDetailQuery {
54         private static final String[] PROJECTION = new String[] {
55             Data.CONTACT_ID, // 0
56             Data.PHOTO_URI, // 1
57             Data.LOOKUP_KEY, // 2
58             Data.DISPLAY_NAME_PRIMARY, // 3
59             Data.CONTACT_PRESENCE, // 4
60             Data.CONTACT_STATUS, // 5
61         };
62
63         public static final int CONTACT_ID = 0;
64         public static final int CONTACT_PHOTO_URI = 1;
65         public static final int CONTACT_LOOKUP_KEY = 2;
66         public static final int CONTACT_DISPLAY_NAME_PRIMARY = 3;
67         public static final int CONTACT_PRESENCE_STATUS = 4;
68         public static final int CONTACT_STATUS = 5;
69     }
70
71     private final long mGroupId;

```

<https://android.googlesource.com/platform/packages/apps/Contacts/+nougat-mr1-release/src/com/android/contacts/GroupMemberLoader.java>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products

24 * Group loader for the group list that includes details such as the number of contacts per group
 25 * and number of groups per account. This list is sorted by account type, account name, where the
 26 * group names are in alphabetical order. Note that the list excludes default, favorite, and deleted
 27 * groups.
 28 */
 29 public final class GroupListLoader extends CursorLoader {
 30
 31 private final static String[] COLUMNS = new String[] {
 32 Groups.ACCOUNT_NAME,
 33 Groups.ACCOUNT_TYPE,
 34 Groups.DATA_SET,
 35 Groups._ID,
 36 Groups.TITLE,
 37 Groups.SUMMARY_COUNT,
 38 };
 39
 40 public final static int ACCOUNT_NAME = 0;
 41 public final static int ACCOUNT_TYPE = 1;
 42 public final static int DATA_SET = 2;
 43 public final static int GROUP_ID = 3;
 44 public final static int TITLE = 4;
 45 public final static int MEMBER_COUNT = 5;
 46
 47 private static final Uri GROUP_LIST_URI = Groups.CONTENT_SUMMARY_URI;
 48
 49 public GroupListLoader(Context context) {
 50 super(context, GROUP_LIST_URI, COLUMNS, Groups.ACCOUNT_TYPE + " NOT NULL AND "
 51 + Groups.ACCOUNT_NAME + " NOT NULL AND " + Groups.AUTO_ADD + "=0 AND "
 52 Groups.FAVORITES + "=0 AND " + Groups.DELETED + "=0", null,
 53 Groups.ACCOUNT_TYPE + ", " + Groups.ACCOUNT_NAME + ", " + Groups.DATA_SET + ", " +
 54 Groups.TITLE + " COLLATE LOCALIZED ASC");
 55 }
 56 }

<https://android.googlesource.com/platform/packages/apps/Contacts/+nougat-mr1-release/src/com/android/contacts/GroupListLoader.java>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products

```

24 * Group meta-data loader. Loads all groups or just a single group from the
25 * database (if given a {@link Uri}).
26 */
27 public final class GroupMetadataLoader extends CursorLoader {
28
29     private final static String[] COLUMNS = new String[] {
30         Groups.ACCOUNT_NAME,
31         Groups.ACCOUNT_TYPE,
32         Groups.DATA_SET,
33         Groups._ID,
34         Groups.TITLE,
35         Groups.AUTO_ADD,
36         Groups.FAVORITES,
37         Groups.GROUP_IS_READ_ONLY,
38         Groups.DELETED,
39     };
40
41     public final static int ACCOUNT_NAME = 0;
42     public final static int ACCOUNT_TYPE = 1;
43     public final static int DATA_SET = 2;
44     public final static int GROUP_ID = 3;
45     public final static int TITLE = 4;
46     public final static int AUTO_ADD = 5;
47     public final static int FAVORITES = 6;
48     public final static int IS_READ_ONLY = 7;
49     public final static int DELETED = 8;
50
51     public GroupMetadataLoader(Context context, Uri groupUri) {
52         super(context, ensureIsGroupUri(groupUri), COLUMNS, Groups.ACCOUNT_TYPE + " NOT NULL AND "
53             + Groups.ACCOUNT_NAME + " NOT NULL", null, null);
54     }
55
56     /**
57      * Ensures that this is a valid group URI. If invalid, then an exception is
58      * thrown. Otherwise, the original URI is returned.
59      */
60     private static Uri ensureIsGroupUri(final Uri groupUri) {
61         // TODO: Fix ContactsProvider2.getType method to resolve the group Uri
62         if (groupUri == null) {
63             throw new IllegalArgumentException("Uri must not be null");
64         }
65         if (!groupUri.toString().startsWith(Groups.CONTENT_URI.toString())) {
66             throw new IllegalArgumentException("Invalid group Uri: " + groupUri);
67         }
68         return groupUri;
69     }
70 }

```

<https://android.googlesource.com/platform/packages/apps/Contacts/+/nougat-ml1-release/src/com/android/contacts/GroupMetadataLoader.java>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products

```

19  * Meta-data for a contact group. We load all groups associated with the contact's
20  * constituent accounts.
21  */
22  public final class GroupMetadata {
23  private String mAccountName;
24  private String mAccountType;
25  private String mDataSet;
26  private long mGroupId;
27  private String mTitle;
28  private boolean mDefaultGroup;
29  private boolean mFavorites;
30
31  public GroupMetadata(String accountName, String accountType, String dataSet, long groupId,
32  String title, boolean defaultGroup, boolean favorites) {
33  this.mAccountName = accountName;
34  this.mAccountType = accountType;
35  this.mDataSet = dataSet;
36  this.mGroupId = groupId;
37  this.mTitle = title;
38  this.mDefaultGroup = defaultGroup;
39  this.mFavorites = favorites;
40  }
41
42  public String getAccountName() {
43  return mAccountName;
44  }
45
46  public String getAccountType() {
47  return mAccountType;
48  }
49
50  public String getDataSet() {
51  return mDataSet;
52  }
53
54  public long getGroupId() {
55  return mGroupId;
56  }
57
58  public String getTitle() {
59  return mTitle;
60  }
61
62  public boolean isDefaultGroup() {
63  return mDefaultGroup;
64  }
65
66  public boolean isFavorites() {
67  return mFavorites;

```

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| | |
|-----------|--|
| US9467838 | <p data-bbox="196 716 228 1556">Exemplary Supporting Evidence Regarding Accused Products</p> <p data-bbox="233 394 305 1556">https://android.googlesource.com/platform/packages/apps/ContactsCommon/+/nougat-mr1-release/src/com/android/contacts/common/GroupMetaData.java</p> <pre data-bbox="553 478 1365 1535"> 44 * Class that sends chat message via SMS. 45 * 46 * The interface emulates a blocking sending similar to making an HTTP request. 47 * It calls the SmsManager to send a (potentially multipart) message and waits 48 * on the sent status on each part. The waiting has a timeout so it won't wait 49 * forever. Once the sent status of all parts received, the call returns. 50 * A successful sending requires success status for all parts. Otherwise, we 51 * pick the highest level of failure as the error for the whole message, which 52 * is used to determine if we need to retry the sending. 53 */ 54 public class SmsSender { 55 private static final String TAG = LogUtil.BUGLE_TAG; 56 57 public static final String EXTRA_PART_ID = "part_id"; 58 59 /* 60 * A map for pending sms messages. The key is the random request UUID. 61 */ 62 private static ConcurrentHashMap<Uri, SendResult> sPendingMessageMap = 63 new ConcurrentHashMap<Uri, SendResult>(); 64 65 private static final Random RANDOM = new Random(); 66 67 // Whether we should send multipart SMS as separate messages 68 private static Boolean sSendMultipartSmsAsSeparateMessages = null; 69 </pre> |
|-----------|--|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products

<https://android.googlesource.com/platform/packages/apps/Messaging/+nougat-mr1-release/src/com/android/messaging/sms/SmsSender.java>

```

253 // Actually sending the message using SmsManager
254 private static void sendInternal(final Context context, final int subId, String dest,
255     final ArrayList<String> messages, final String serviceCenter,
256     final boolean requireDeliveryReport, final Uri messageUri) throws SmsException {
257     Assert.notNull(context);
258     final SmsManager smsManager = PhoneUtils.get(subId).getSmsManager();
259     final int messageCount = messages.size();
260     final ArrayList<PendingIntent> deliveryIntents = new ArrayList<PendingIntent>(messageCount);
261     final ArrayList<PendingIntent> sentIntents = new ArrayList<PendingIntent>(messageCount);
262     for (int i = 0; i < messageCount; i++) {
263         // Make pending intents different for each message part
264         final int partId = (messageCount <= 1 ? 0 : i + 1);
265         if (requireDeliveryReport && (i == (messageCount - 1))) {
266             // TODO we only care about the delivery status of the last part
267             // Shall we have better tracking of delivery status of all parts?
268             deliveryIntents.add(PendingIntent.getBroadcast(
269                 context,
270                 partId,
271                 getSendStatusIntent(context, SendStatusReceiver.MESSAGE_DELIVERED_ACTION,
272                     messageUri, partId, subId),
273                 0/**flag*/));
274         } else {
275             deliveryIntents.add(null);
276         }
277         sentIntents.add(PendingIntent.getBroadcast(
278             context,
279             partId,
280             getSendStatusIntent(context, SendStatusReceiver.MESSAGE_SENT_ACTION,
281                 messageUri, partId, subId),
282             0/**flag*/));
283     }
284     if (sSendMultipartSmsAsSeparateMessages == null) {
285         sSendMultipartSmsAsSeparateMessages = MmsConfig.get(subId)
286             .getSendMultipartSmsAsSeparateMessages();
287     }

```

<https://android.googlesource.com/platform/packages/apps/Messaging/+nougat-mr1->

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| | |
|-----------|--|
| US9467838 | <p data-bbox="191 247 228 1564">Exemplary Supporting Evidence Regarding Accused Products release/src/com/android/messaging/sms/SmsSender.java</p> <pre data-bbox="305 394 868 1564"> 288 try { 289 if (sendMultipartSmsAsSeparateMessages) { 290 // If multipart sms is not supported, send them as separate messages 291 for (int i = 0; i < messageCount; i++) { 292 smsManager.sendMessage(dest, 293 serviceCenter, 294 messages.get(i), 295 sentIntents.get(i), 296 deliveryIntents.get(i)); 297 } 298 } else { 299 smsManager.sendMultipartTextMessage(300 dest, serviceCenter, messages, sentIntents, deliveryIntents); 301 } 302 } catch (final Exception e) { 303 throw new SmsSenderException("SmsSender: caught exception in sending " + e); 304 } 305 } </pre> <p data-bbox="917 478 990 1564">https://android.googlesource.com/platform/packages/apps/Messaging/+/nougat-mr1-release/src/com/android/messaging/sms/SmsSender.java</p> |
|-----------|--|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| US9467838 | Exemplary Supporting Evidence Regarding Accused Products |
|-----------|---|
| 56 | * Class that receives incoming SMS messages through android.provider.Telephony.SMS_RECEIVED |
| 57 | * |
| 58 | * This class serves two purposes: |
| 59 | * - Process phone verification SMS messages |
| 60 | * - Handle SMS messages when the user has enabled us to be the default SMS app (Pre-KLP) |
| 61 | */ |
| 62 | public final class SmsReceiver extends BroadcastReceiver { |
| 63 | private static final String TAG = LogUtil.BUGLE_TAG; |
| 64 | } |
| 65 | private static ArrayList<Pattern> ignoreSmsPatterns; |
| 66 | } |
| | https://android.googlesource.com/platform/packages/apps/Messaging/+nougat-mr1-release/src/com/android/messaging/receiver/SmsReceiver.java |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| US9467838 | Exemplary Supporting Evidence Regarding Accused Products |
|-----------|---|
| 200 | public static void deliverSmsMessages(final Context context, final int subId, |
| 201 | final int errorCode, final android.telephony.SmsMessage[] messages) { |
| 202 | final ContentValues messageValues = |
| 203 | MmsUtils.parseReceivedSmsMessage(context, messages, errorCode); |
| 204 | |
| 205 | LogUtil.v(TAG, "SmsReceiver.deliverSmsMessages"); |
| 206 | |
| 207 | final long nowInMillis = System.currentTimeMillis(); |
| 208 | final long receivedTimestamps = MmsUtils.getMessageDate(messages[0], nowInMillis); |
| 209 | |
| 210 | messageValues.put(Sms.Inbox.DATE, receivedTimestamps); |
| 211 | // Default to unread and unseen for us but ReceiveSmsMessageAction will override |
| 212 | // seen for the telephony db. |
| 213 | messageValues.put(Sms.Inbox.READ, 0); |
| 214 | messageValues.put(Sms.Inbox.SEEN, 0); |
| 215 | if (OsUtil.isAtLeastL_MRI()) { |
| 216 | messageValues.put(Sms.SUBSCRIPTION_ID, subId); |
| 217 | } |
| 218 | |
| 219 | if (messages[0].getMessageClass() == android.telephony.SmsMessage.MessageClass.CLASS_0 |
| 220 | DebugUtils.isDebugEnabled()) { |
| 221 | Factory.get().getUIIntents().launchClassZeroActivity(context, messageValues); |
| 222 | } else { |
| 223 | final ReceiveSmsMessageAction action = new ReceiveSmsMessageAction(messageValues); |
| 224 | action.start(); |
| 225 | } |
| 226 | } |
| | https://android.googlesource.com/platform/packages/apps/Messaging/+nougat-mr1-release/src/com/android/messaging/receiver/SmsReceiver.java |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| US9467838 | Exemplary Supporting Evidence Regarding Accused Products |
|-----------|---|
| | <pre> 228 @Override 229 public void onReceive(final Context context, final Intent intent) { 230 LogUtil.v(TAG, "SmsReceiver.onReceive " + intent); 231 // On KLP+ we only take delivery of SMS messages in SmsDeliverReceiver. 232 if (PhoneUtils.getDefault().isSmsEnabled()) { 233 final String action = intent.getAction(); 234 if (OsUtil.isSecondaryUser() && 235 (Telephony.Sms.Intents.SMS_RECEIVED_ACTION.equals(action) 236 // TODO: update this with the actual constant from Telephony 237 "android.provider.Telephony.MMS_DOWNLOADED".equals(action))) { 238 postNewMessageSecondaryUserNotification(); 239 } else if (!OsUtil.isAtLeastKLP()) { 240 deliverSmsIntent(context, intent); 241 } 242 } 243 } </pre> <p>https://android.googlesource.com/platform/packages/apps/Messaging/+nougat-mr1-release/src/com/android/messaging/receiver/SmsReceiver.java</p> |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| US9467838 | Exemplary Supporting Evidence Regarding Accused Products |
|-----------|---|
| 52 | * Class that sends chat message via MMS. |
| 53 | * |
| 54 | * The interface emulates a blocking send similar to making an HTTP request. |
| 55 | */ |
| 56 | public class MmsSender { |
| 57 | private static final String TAG = LogUtil.BUGLE_TAG; |
| 58 | |
| 59 | /** |
| 60 | * Send an MMS message. |
| 61 | * |
| 62 | * @param context Context |
| 63 | * @param messageUri The unique URI of the message for identifying it during sending |
| 64 | * @param sendReq The SendReq PDU of the message |
| 65 | * @throws MmsFailureException |
| 66 | */ |
| 67 | public static void sendMms(final Context context, final int subId, final Uri messageUri, |
| 68 | final SendReq sendReq, final Bundle sentIntentExtras) throws MmsFailureException { |
| 69 | sendMms(context, |
| 70 | subId, |
| 71 | messageUri, |
| 72 | null /* locationUrl */, |
| 73 | sendReq, |
| 74 | true /* responseImportant */, |
| 75 | sentIntentExtras); |
| 76 | } |
| | https://android.googlesource.com/platform/packages/apps/Messaging/+nougat-mr1-release/src/com/android/messaging/sms/MmsSender.java |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| US9467838 | Exemplary Supporting Evidence Regarding Accused Products |
|---|---|
| 240 | * Download an MMS message. |
| 241 | * |
| 242 | * @param context Context |
| 243 | * @param contentLocation The url of the MMS message |
| 244 | * @throws MmsFailureException |
| 245 | * @throws InvalidHeaderValueException |
| 246 | */ |
| 247 | public static void downloadMms(final Context context, final int subId, |
| 248 | final String contentLocation, Bundle extras) throws MmsFailureException, |
| 249 | InvalidHeaderValueException { |
| 250 | final Uri requestUri = Uri.parse(contentLocation); |
| 251 | final Uri contentUri = MmsFileProvider.buildRawMmsUri(); |
| 252 | |
| 253 | final Intent downloadedIntent = new Intent(SendStatusReceiver.MMS_DOWNLOADED_ACTION, |
| 254 | requestUri, |
| 255 | context, |
| 256 | SendStatusReceiver.class); |
| 257 | downloadedIntent.putExtra(SendMessageAction.EXTRA_CONTENT_URI, contentUri); |
| 258 | if (extras != null) { |
| 259 | downloadedIntent.putExtras(extras); |
| 260 | } |
| 261 | final PendingIntent downloadedPendingIntent = PendingIntent.getBroadcast(context, |
| 262 | 0 /*request code*/, |
| 263 | downloadedIntent, |
| 264 | PendingIntent.FLAG_UPDATE_CURRENT); |
| 265 | |
| 266 | |
| 267 | MmsManager.downloadMultimediaMessage(subId, context, contentLocation, contentUri, |
| 268 | downloadedPendingIntent); |
| 269 | } |
| https://android.googlesource.com/platform/packages/apps/Messaging/+nougat-mr1-release/src/com/android/messaging/sms/MmsSender.java | |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| US9467838 | Exemplary Supporting Evidence Regarding Accused Products |
|-----------|---|
| 97 | * Execute an MMS HTTP request, either a POST (sending) or a GET (downloading) |
| 98 | x |
| 99 | * @param urlString The request URL, for sending it is usually the MMSC, and for downloading |
| 100 | * it is the message URL |
| 101 | * @param pdu For POST (sending) only, the PDU to send |
| 102 | * @param method HTTP method, POST for sending and GET for downloading |
| 103 | * @param isProxySet Is there a proxy for the MMSC |
| 104 | * @param proxyHost The proxy host |
| 105 | * @param proxyPort The proxy port |
| 106 | * @param mmsConfig The MMS config to use |
| 107 | * @param userAgent The user agent header value |
| 108 | * @param uaProfUrl The UA Prof URL header value |
| 109 | * @return The HTTP response body |
| 110 | * @throws MmsHttpException For any failures |
| 111 | */ |
| 112 | public byte[] execute(String urlString, byte[] pdu, String method, boolean isProxySet, |
| 113 | String proxyHost, int proxyPort, Bundle mmsConfig, String userAgent, String uaProfUrl) |
| 114 | throws MmsHttpException { |
| 115 | Log.d(MmsService.TAG, "HTTP: " + method + " " + Utils.redactUrlForNonVerbose(urlString) |
| 116 | + (isProxySet ? (" proxy=" + proxyHost + " : " + proxyPort) : ""); |
| 117 | + " PDU size=" + (pdu != null ? pdu.length : 0)); |
| 118 | checkMethod(method); |
| 119 | URLConnection connection = null; |
| 120 | try { |
| 121 | Proxy proxy = Proxy.NO_PROXY; |
| 122 | if (isProxySet) { |
| 123 | proxy = new Proxy(Proxy.Type.HTTP, new InetSocketAddress(proxyHost, proxyPort)); |
| 124 | } |
| 125 | final URL url = new URL(urlString); |
| 126 | // Now get the connection |
| 127 | connection = (URLConnection) url.openConnection(proxy); |
| 128 | connection.setDoInput(true); |
| 129 | connection.setConnectTimeout(|
| 130 | mmsConfig.getInt(CarrierConfigValuesLoader.CONFIG_HTTP_SOCKET_TIMEOUT, |
| 131 | CarrierConfigValuesLoader.CONFIG_HTTP_SOCKET_TIMEOUT_DEFAULT)); |

https://android.googlesource.com/platform/packages/apps/Messaging/+nougat-mr1-release/src/android/support/v7/mms/MmsHttpClient.java

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products

```

132 // ----- COMMON HEADERS -----
133 // Header: Accept
134 connection.setRequestProperty(HEADER_ACCEPT, HEADER_VALUE_ACCEPT);
135 // Header: Accept-Language
136 connection.setRequestProperty(
137     HEADER_ACCEPT_LANGUAGE, getLocale().getLanguage(Locale.getDefault()));
138 // Header: User-Agent
139 Log.i("MmsService.TAG", "HTTP: User-Agent=" + userAgent);
140 connection.setRequestProperty(HEADER_USER_AGENT, userAgent);
141 // Header: x-wap-profile
142 final String uaProfileTagName = mmsConfig.getString(
143     CarrierConfigValuesLoader.CONFIG_UA_PROF_TAG_NAME,
144     CarrierConfigValuesLoader.CONFIG_UA_PROF_TAG_NAME_DEFAULT);
145 if (uaProfile != null) {
146     Log.i("MmsService.TAG", "HTTP: UaProfile=" + uaProfile);
147     connection.setRequestProperty(uaProfileTagName, uaProfile);
148 }
149 // Add extra headers specified by mms_config.xml's httpparams
150 addExtraHeaders(connection, mmsConfig);
151 // Different stuff for GET and POST
152 if (METHOD_POST.equals(method)) {
153     if (pdu == null || pdu.length < 1) {
154         Log.e("MmsService.TAG", "HTTP: empty pdu");
155         throw new MmsHttpException(0/*statusCode*/, "Sending empty PDU");
156     }
157     connection.setDoOutput(true);
158     connection.setRequestMethod(METHOD_POST);
159     if (mmsConfig.getBoolean(
160         CarrierConfigValuesLoader.CONFIG_SUPPORT_HTTP_CHARSET_HEADER,
161         CarrierConfigValuesLoader.CONFIG_SUPPORT_HTTP_CHARSET_HEADER_DEFAULT)) {
162         connection.setRequestProperty(HEADER_CONTENT_TYPE,
163             HEADER_VALUE_CONTENT_TYPE_WITH_CHARSET);
164     } else {
165         connection.setRequestProperty(HEADER_CONTENT_TYPE,
166             HEADER_VALUE_CONTENT_TYPE_WITHOUT_CHARSET);

```

<https://android.googlesource.com/platform/packages/apps/Messaging/+/nougat-mr1-release/src/android/support/v7/mms/MmsHttpClient.java>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products

```

167     }
168     if (Log.isLoggable(MmsService.TAG, Log.VERBOSE)) {
169         logHttpHeaders(connection.getRequestProperties());
170     }
171     connection.setFixedLengthStreamingMode(pdu.length);
172     // Sending request body
173     final OutputStream out =
174         new BufferedOutputStream(connection.getOutputStream());
175     out.write(pdu);
176     out.flush();
177     out.close();
178     } else if (METHOD_GET.equals(method)) {
179         if (Log.isLoggable(MmsService.TAG, Log.VERBOSE)) {
180             logHttpHeaders(connection.getRequestProperties());
181         }
182         connection.setRequestMethod(METHOD_GET);
183     }
184     // Get response
185     final int responseCode = connection.getResponseCode();
186     final String responseMessage = connection.getResponseMessage();
187     Log.d(MmsService.TAG, "HTTP: " + responseCode + " " + responseMessage);
188     if (Log.isLoggable(MmsService.TAG, Log.VERBOSE)) {
189         logHttpHeaders(connection.getHeaderFields());
190     }
191     if (responseCode / 100 != 2) {
192         throw new MmsHttpException(responseCode, responseMessage);
193     }
194     final InputStream in = new BufferedInputStream(connection.getInputStream());
195     final ByteArrayOutputStream bytesOut = new ByteArrayOutputStream();
196     final byte[] buf = new byte[4096];
197     int count = 0;
198     while ((count = in.read(buf)) > 0) {
199         bytesOut.write(buf, 0, count);
200     }
201     in.close();
202     final byte[] responseBody = bytesOut.toByteArray();
203     Log.d(MmsService.TAG, "HTTP: response size="
204         + (responseBody != null ? responseBody.length : 0));
205     return responseBody;

```

<https://android.googlesource.com/platform/packages/apps/Messaging/+nougat-mr1->

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| | |
|------------------|---|
| US9467838 | Exemplary Supporting Evidence Regarding Accused Products |
| | release/src/android/support/v7/mms/MmsHttpClient.java |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products

```

38 * Request to send an MMS
39 */
40 class SendRequest extends MmsRequest {
41     // Max send response PDU size in bytes (exceeding this may cause problem with
42     // system intent delivery).
43     private static final int MAX_SEND_RESPONSE_SIZE = 1000 * 1024;
44
45     private byte[] mPduData;
46
47     SendRequest(final String locationUrl, final Uri pduUri, final PendingIntent sentIntent) {
48         super(locationUrl, pduUri, sentIntent);
49     }
50
51     @Override
52     protected boolean loadRequest(final Context context, final Bundle mmsConfig) {
53         mPduData = readPduFromContentUri(
54             context,
55             pduUri,
56             mmsConfig.getInt(
57                 CarrierConfigValuesLoader.CONFIG_MAX_MESSAGE_SIZE,
58                 CarrierConfigValuesLoader.CONFIG_MAX_MESSAGE_SIZE_DEFAULT));
59         return (mPduData != null);
60     }
61
62     @Override
63     protected boolean transferResponse(final Context context, final Intent fillIn,
64         final byte[] response) {
65         // SendConf pdus are always small and can be included in the intent
66         if (response != null && fillIn != null) {
67             if (response.length > MAX_SEND_RESPONSE_SIZE) {
68                 // If the response PDU is too large, it won't be able to fit in
69                 // the PendingIntent to be transferred via system IPC.
70                 return false;
71             }
72             fillIn.putExtra(SmsManager.EXTRA_MMS_DATA, response);
73         }
74         return true;
75     }

```

<https://android.googlesource.com/platform/packages/apps/Messaging/+/nougat-mr1-release/src/android/support/v7/mms/SendRequest.java>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| <p>US9467838</p> | <p>Exemplary Supporting Evidence Regarding Accused Products</p> |
|--|---|
| <p>[IC] participating in the group, wherein participating in the group includes sending first location information to a first server and receiving second location information from the first server, the first location information comprising a location of the first device, the second location information comprising one or more locations of one or more respective second devices included in the group;</p> <p>server and receiving second location information from the first server, the first location information comprising a location of the first device, the second location information comprising one or more locations of one or more respective second devices included in the group;</p> <p>one or more respective second devices included in the group;</p> | <p>ZTE infringes directly and/or indirectly by performing, inducing others to perform, and/or contributing to the performance of: participating in the group, wherein participating in the group includes sending first location information to a first server and receiving second location information from the first server, the first location information comprising a location of the first device, the second location information comprising one or more locations of one or more respective second devices included in the group.</p> <p>Regarding Find My Device and Android Device Manager, the Accused Products require a user to join the corresponding network by: signing-in to the device with an identifier (e.g., Google Account) or linking the device to the identifier by remote means. The Accused Products require that a signed-in user of a first device to share its location by enabling Location Access or Location Services. When a user of a first device associated with an identity (e.g. Google Account) enables the Find My Device feature and Location Services on the first device, the user shares its location and the first device's location is sent to a server. When the user makes a request to view a map of device locations associated with the identity, the device receives one or more locations corresponding to one or more second devices associated with the identity. Alternatively, the locations corresponding to one or more second devices associated with the identity are sent to the first device on a rolling basis.</p> <p>Regarding Google Maps, Google Latitude, Google Plus, Google Hangouts, Google Messages, Google Allo, Google Duo, Google Chrome, and Android Messenger, the Accused Products require a user to join the corresponding network by signing-in to the device with an identifier (e.g., Google Account). When the signed-in user enables Google Maps Share Location and Location Services on the first device, the user shares its location and the first device's location is sent to a server. When the user enables sharing to one or more contacts (of respective devices) and the one or more contacts enable sharing their location to the user of the first device, the user of the first device receives the locations of the one or more contacts.</p> <p>Further regarding Google Maps, Google Latitude, Google Plus, Google Hangouts, Google Messages, Google Allo, Google Duo, Google Chrome, and Android Messenger, the Accused Products alternatively require a user to send a request containing the identifier (e.g., Google Account). When the signed-in user enables Google Maps Share Location and Location Services on the first device, the user shares its location and the first device's location is sent to a server. When the user sends a message to another contact through Google Maps, Google Messages, and/or another means from within the Google Maps application, the message including location information are sent to a server before transmission to the intended contact.</p> |

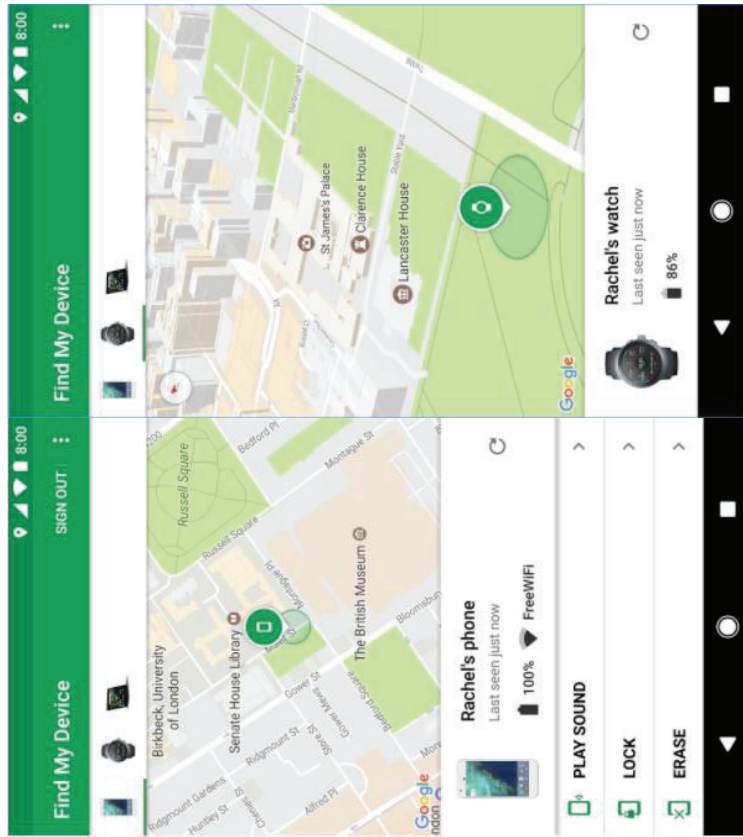
Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products

When one or more contacts enable sharing their location to the user of the first device, or alternatively send a message containing location information, or alternatively accept a request to share their location with the first user, the user of the first device receives the locations of the one or more contacts.

Exemplary Support for Find My Device:



<https://play.google.com/store/apps/details?id=com.google.android.apps.adm&hl=en>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

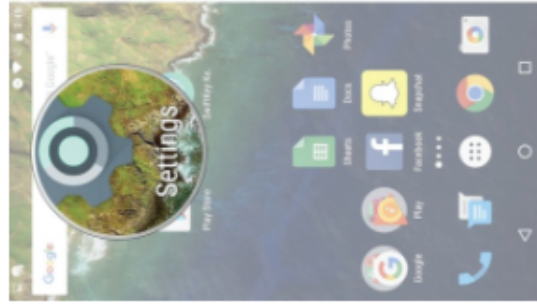
US9467838

Exemplary Supporting Evidence Regarding Accused Products

How to enable Find My Device on your phone

In newer Android phones, the Find My Device service is already located conveniently in your Settings app, but if you can't find it you can always download Find My Device from the Google Play Store. This locating service has essentially amalgamated with Google to make finding your phone easier. There are just a couple of things you'll need to activate.

1. Launch Settings.
2. Tap Security.
3. Tap Device Administration.



4. Tap Find My Device so that a checkmark appears in the checkbox.

<https://www.androidcentral.com/how-track-android-phone>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

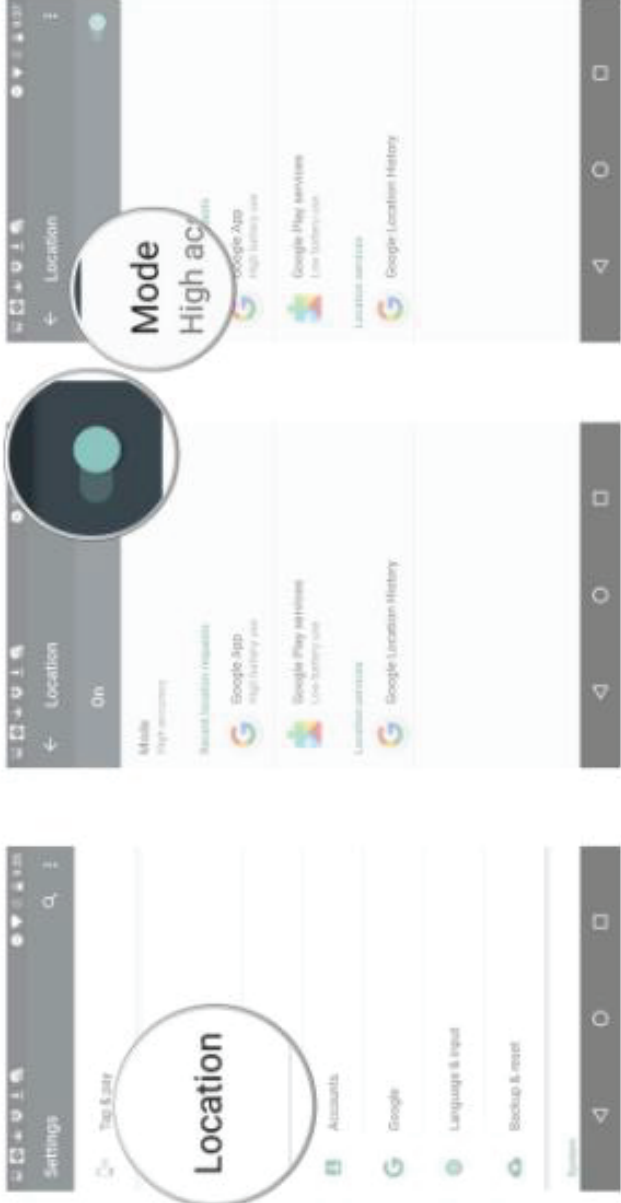
| | |
|------------------|--|
| <p>US9467838</p> | <p>Exemplary Supporting Evidence Regarding Accused Products</p> <p>7. Tap Location in the main Settings menu.</p> <p>8. Tap the switch beside Location at the top of the screen so that it turns on.</p> <p>9. Tap Mode.</p>  <p>https://www.androidcentral.com/how-track-android-phone</p> |
|------------------|--|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

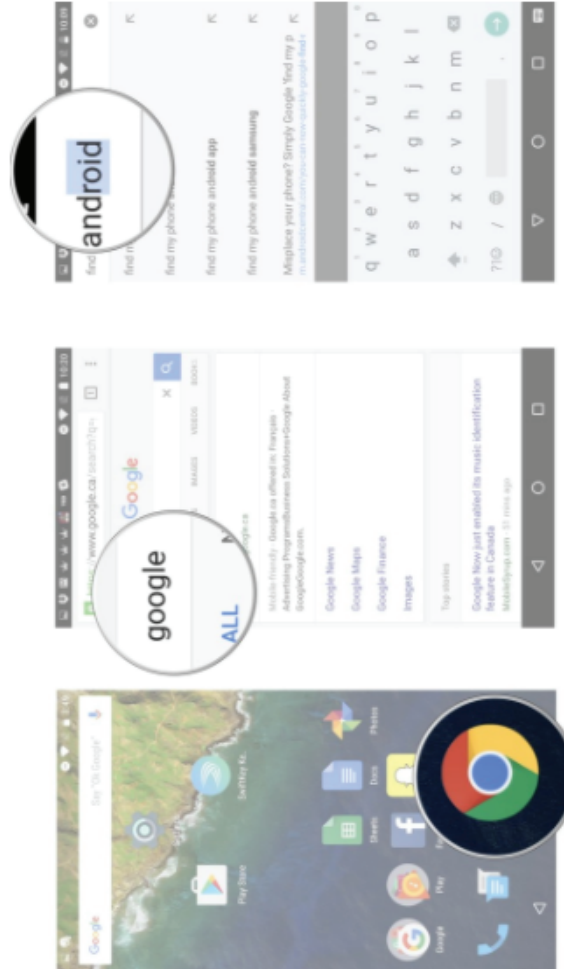
US9467838

Exemplary Supporting Evidence Regarding Accused Products

How to locate your phone with Google

Should you happen to lose your phone, you can locate its whereabouts by logging into your Google account from any computer or even from another phone.

1. Launch a web browser from a phone, tablet, or computer.
2. Navigate to Google if it is not your default search engine or home page.
3. Type find my phone android in the Google search bar.



4. Tap on Find My Device (usually the first option in the search).
5. Enter your email address and password just as though you were checking your email. If you have 2-step verification set up on your Google account (and you most certainly should), you'll need to complete that process as well.

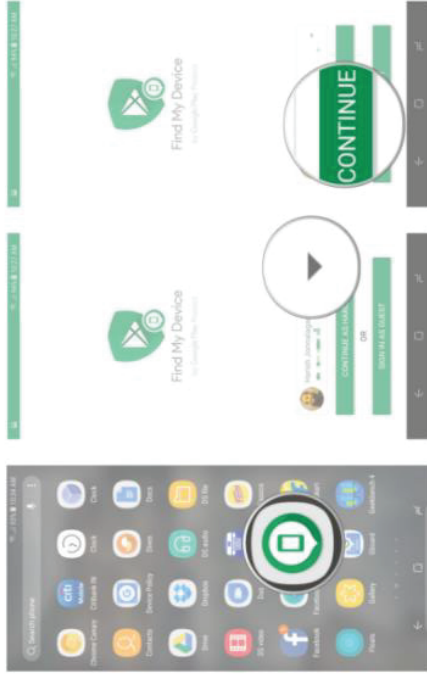
<https://www.androidcentral.com/how-track-android-phone>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

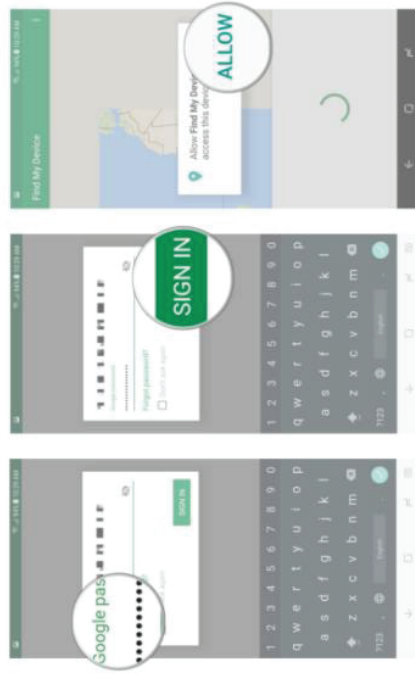
US9467838

Exemplary Supporting Evidence Regarding Accused Products

1. Open Find Device from your home screen or app drawer.
2. Select the Google account you want to use the service with.
3. Hit the Continue as button.



4. Enter your Google account password.
5. Tap Sign in.
6. Give location access to the service.



<https://www.androidcentral.com/find-my-device>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

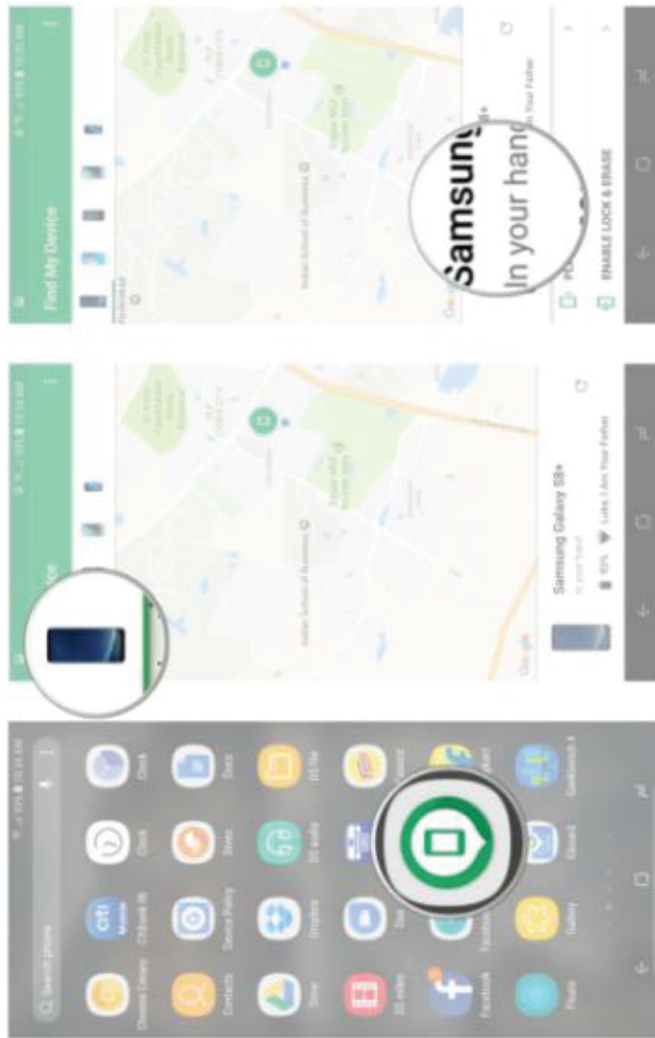
US9467838

Exemplary Supporting Evidence Regarding Accused Products

Once you're signed in to Find My Device, you'll see a map with your current location as well as the make and model of your phone, and two options — Play Sound, and Enable Lock & Erase. Hitting the latter option will allow you to start using the Lock and Erase functions.

If you've signed into more than one phone, you can select a particular device by browsing the list at the top of the screen.

1. Open **Find My Device** from your home screen or app drawer.
2. Select your phone from the **list of devices at the top of the screen**.
3. See if your phone is **discoverable**.



<https://www.androidcentral.com/find-my-device>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

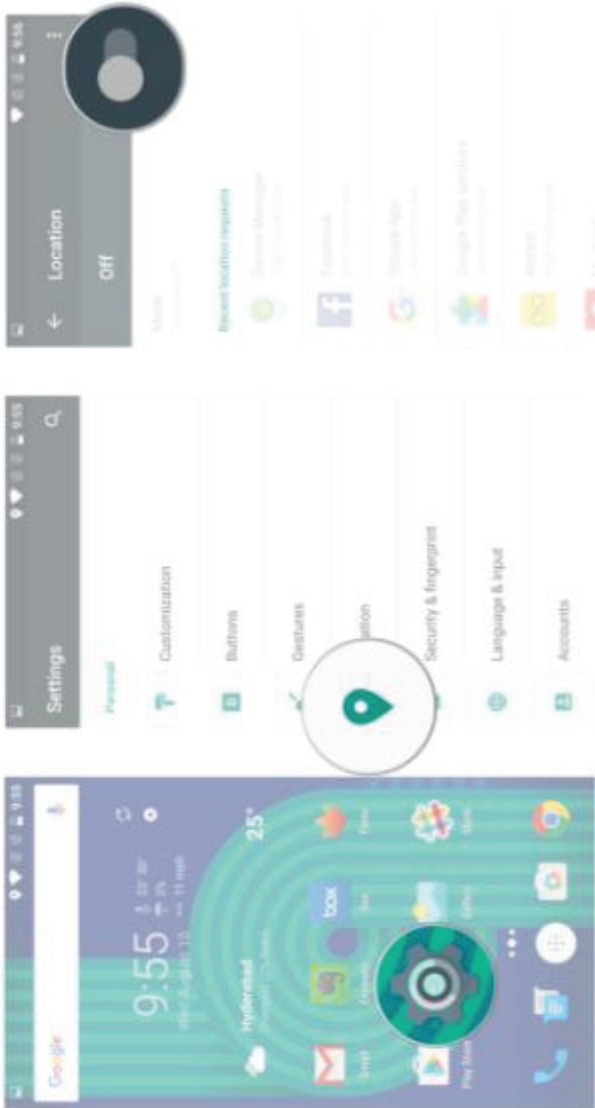
| | |
|-------------------------|---|
| <p>US9467838</p> | <p>Exemplary Supporting Evidence Regarding Accused Products</p> <p>if you're not able to find your phone or if it says that the device is unavailable, it is likely that the location services are disabled. Find My Device relies on GPS to track your phone, so now would be a good time to enable location services.</p> <ol style="list-style-type: none"> 1. Open Settings from your home screen or app drawer. 2. Tap Location. 3. Toggle Enable location services.  <p>https://www.androidcentral.com/find-my-device</p> <p>Exemplary Support for Google Maps:</p> |
|-------------------------|---|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE












| | |
|-----------|---|
| US9467838 | <p data-bbox="191 709 228 1564">Exemplary Supporting Evidence Regarding Accused Products</p> <p data-bbox="240 1108 261 1528">COMPUTER ANDROID IPHONE & IPAD</p> <hr data-bbox="289 506 293 1564"/> <p data-bbox="342 1052 380 1541">If they have a Google Account</p> <ol data-bbox="402 657 695 1535" style="list-style-type: none"> <li data-bbox="402 856 423 1535">1. If you haven't already, add their Gmail address to your Google Contacts []. <li data-bbox="440 657 461 1535">2. On your Android phone or tablet, open the Google Maps app  and sign in. Learn how to sign in. <li data-bbox="477 1045 498 1535">3. Tap Menu  > Location sharing > Add People . <li data-bbox="514 1073 535 1535">4. Choose how long you want to share your location. <li data-bbox="552 1346 573 1535">5. Tap Select People. <ul data-bbox="589 940 610 1514" style="list-style-type: none"> <li data-bbox="589 940 610 1514">• If you're asked about your contacts, give Google Maps access. <li data-bbox="634 1192 656 1535">6. Choose who you want to share with. <li data-bbox="672 1423 693 1535">7. Tap Share. <p data-bbox="753 968 790 1541">If they don't have a Google Account</p> <ol data-bbox="813 520 938 1535" style="list-style-type: none"> <li data-bbox="813 657 834 1535">1. On your Android phone or tablet, open the Google Maps app  and sign in. Learn how to sign in. <li data-bbox="850 1045 872 1535">2. Tap Menu  > Location sharing > Add People . <li data-bbox="888 520 938 1535">3. Tap More  > Copy to clipboard. People with this link can see your location for as long as you choose, up to 72 hours. <p data-bbox="987 1213 1024 1541">Share using another app</p> <p data-bbox="1040 877 1062 1541">You can also share through messaging apps. Tap More  > select an app.</p> <p data-bbox="1122 1339 1159 1541">Stop sharing</p> <ol data-bbox="1182 877 1279 1535" style="list-style-type: none"> <li data-bbox="1182 1234 1203 1535">1. Open the Google Maps app . <li data-bbox="1219 1213 1240 1535">2. Tap Menu  > Location sharing. <li data-bbox="1256 877 1278 1535">3. Next to the person with whom you want to stop sharing, tap Remove . <p data-bbox="1295 380 1328 1564">https://support.google.com/maps/answer/7326816?co=GENIE.Platform%3DAndroid&oco=1</p> |
|-----------|---|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE









| | |
|-----------|--|
| US9467838 | <p data-bbox="191 716 228 1566">Exemplary Supporting Evidence Regarding Accused Products</p> <h2 data-bbox="253 1236 297 1545">Share your E.T.A</h2> <p data-bbox="321 415 345 1545">After you start your drive, you can share your destination, estimated arrival time, and where you are on the route.</p> <ol data-bbox="378 877 618 1545" style="list-style-type: none"> <li data-bbox="378 1192 402 1545">1. Open the Google Maps app . <li data-bbox="418 919 443 1545">2. Set a driving destination. Learn how to navigate to a place. <li data-bbox="459 877 483 1545">3. After you start navigation, tap More  > Share trip progress. <li data-bbox="500 1203 524 1545">4. Choose a person from the list. <li data-bbox="540 1402 565 1545">5. Tap Share. <li data-bbox="581 720 605 1545">6. Location Sharing will stop when you reach your destination or stop navigating. <ul data-bbox="646 877 670 1545" style="list-style-type: none"> <li data-bbox="646 877 670 1545">• To stop sharing before you arrive, tap More  > Stop sharing. <h2 data-bbox="740 1119 784 1545">See where someone is</h2> <p data-bbox="808 814 833 1545">If someone shares their location with you, you can see them on the map.</p> <ol data-bbox="865 1161 976 1545" style="list-style-type: none"> <li data-bbox="865 1192 889 1545">1. Open the Google Maps app . <li data-bbox="906 1161 930 1545">2. Tap Menu  > Location sharing. <li data-bbox="946 1329 971 1545">3. Choose someone. <ul data-bbox="1003 772 1027 1545" style="list-style-type: none"> <li data-bbox="1003 772 1027 1545">• To see an updated location, tap on a friend's icon > More  > Refresh. <h2 data-bbox="1084 1045 1128 1545">Stop seeing someone's location</h2> <ol data-bbox="1144 1192 1295 1545" style="list-style-type: none"> <li data-bbox="1144 1192 1169 1545">1. Open the Google Maps app . <li data-bbox="1185 1245 1209 1545">2. On the map, tap their icon. <li data-bbox="1226 1224 1250 1545">3. At the bottom, tap More . <li data-bbox="1266 615 1291 1545">4. To temporarily hide someone, tap Hide from map. You can stop hiding them at any time. <p data-bbox="1328 342 1352 1545">Note: You can stop someone's location from ever appearing on your map. Learn how to block another person's account.</p> <p data-bbox="1377 373 1401 1566">https://support.google.com/maps/answer/7326816?co=GENIE.Platform%3DAndroid&oco=1</p> |
|-----------|--|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE








| | |
|------------------|---|
| <p>US9467838</p> | <p>Exemplary Supporting Evidence Regarding Accused Products</p> <h2>Create a list of places</h2> <p>In Google Maps, you can create a list of places, like your favorite places or places you want to visit.</p> <p>COMPUTER ANDROID IPHONE & IPAD</p> <h3>Make a new list</h3> <ol style="list-style-type: none">1. On your Android phone or tablet, open the Google Maps app .2. Tap Menu  > Your places > Saved.3. In the bottom right, tap Add .4. Enter a name and description.5. Tap Save. <h3>Save a place to a list</h3> <ol style="list-style-type: none">1. Open the Google Maps app .2. Search for a place or tap it on the map.3. At the bottom, tap the place's name or address.4. Tap Save.5. Choose a list. To create a new list, tap New list . <h3>See your lists</h3> <ol style="list-style-type: none">1. Open the Google Maps app .2. Tap Menu  > Your places > Saved. <p>https://support.google.com/maps/answer/7280933?hl=en&ref_topic=7301134&co=GENIE.Platform%3DAndroid&oco=1</p> |
|------------------|---|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE





| | |
|-----------|---|
| US9467838 | <p data-bbox="191 1570 228 1923">Exemplary Supporting Evidence Regarding Accused Products</p> <h2 data-bbox="245 1192 282 1528">Hide or share lists</h2> <p data-bbox="310 1163 337 1528">Note: You can't share starred places.</p> <ol data-bbox="365 630 516 1528" style="list-style-type: none"> <li data-bbox="365 1184 393 1528">1. Open the Google Maps app . <li data-bbox="407 1108 435 1528">2. Tap Menu  > Your places > Saved. <li data-bbox="449 823 477 1528">3. Next to the list you want to share, tap More  > choose an option: <ul data-bbox="492 394 633 1495" style="list-style-type: none"> <li data-bbox="492 630 519 1495">• Hide/Show on your map: Display or hide your saved places when looking at the map. <li data-bbox="534 1016 561 1495">• Share list: Allow others to see your saved list. <li data-bbox="576 394 633 1495">• Sharing options: You can make your list public, private, or shared. To let anyone with the link see your list, tap Shared. To let anyone find and follow your list, tap Public. <h2 data-bbox="699 1310 737 1528">Follow a list</h2> <p data-bbox="764 348 824 1528">If you follow a list made by someone else, their saved places will show up in Your Places. The places will also appear as suggested locations in Google Maps.</p> <h2 data-bbox="878 1163 915 1528">Follow a list using a link</h2> <ol data-bbox="943 722 1045 1528" style="list-style-type: none"> <li data-bbox="943 1117 971 1528">1. Tap on the link you received to open it. <li data-bbox="985 806 1013 1528">2. Tap Follow. This list will now be added to the group of lists you follow. <li data-bbox="1027 722 1045 1528">3. optional: To unfollow a list someone shared with you, tap the list > Following. <h2 data-bbox="1099 1150 1136 1528">See lists made by others</h2> <p data-bbox="1164 743 1192 1528">If a user has any Google Maps lists that were made public, you can follow them.</p> <ol data-bbox="1219 197 1321 1528" style="list-style-type: none"> <li data-bbox="1219 940 1247 1528">1. Tap on the name of a user whose list you want to follow. <li data-bbox="1261 1402 1289 1528">2. Tap Lists. <li data-bbox="1304 940 1321 1528">3. Tap on the list you want to follow > More  > Follow. <p data-bbox="1336 197 1399 1562">https://support.google.com/maps/answer/7280933?hl=en&ref_topic=7301134&co=GENIE.Platform%3DAndroid&oco=1</p> |
|-----------|---|

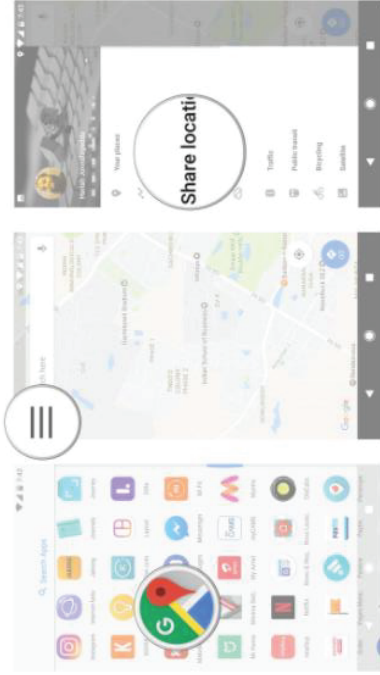
Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

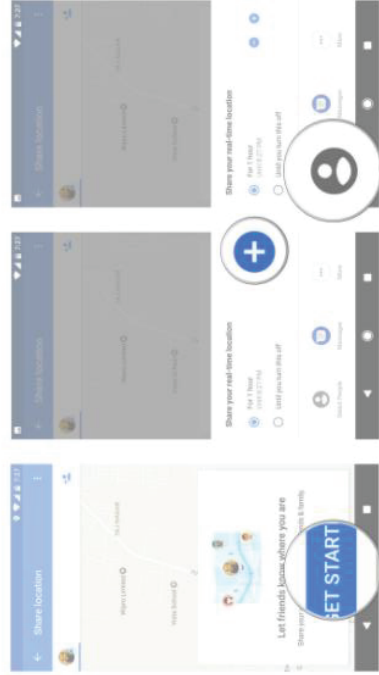
Exemplary Supporting Evidence Regarding Accused Products

How to share your location in Google Maps

1. Open Google Maps from the app drawer or the home screen.
2. Tap the hamburger menu (the three horizontal lines) on the top left corner of the screen.
3. Select Share location.



4. Tap Get Started.
5. Use the + icon to select a time period or select the **Until you turn this off** setting to share your location indefinitely.
6. Tap Select People.



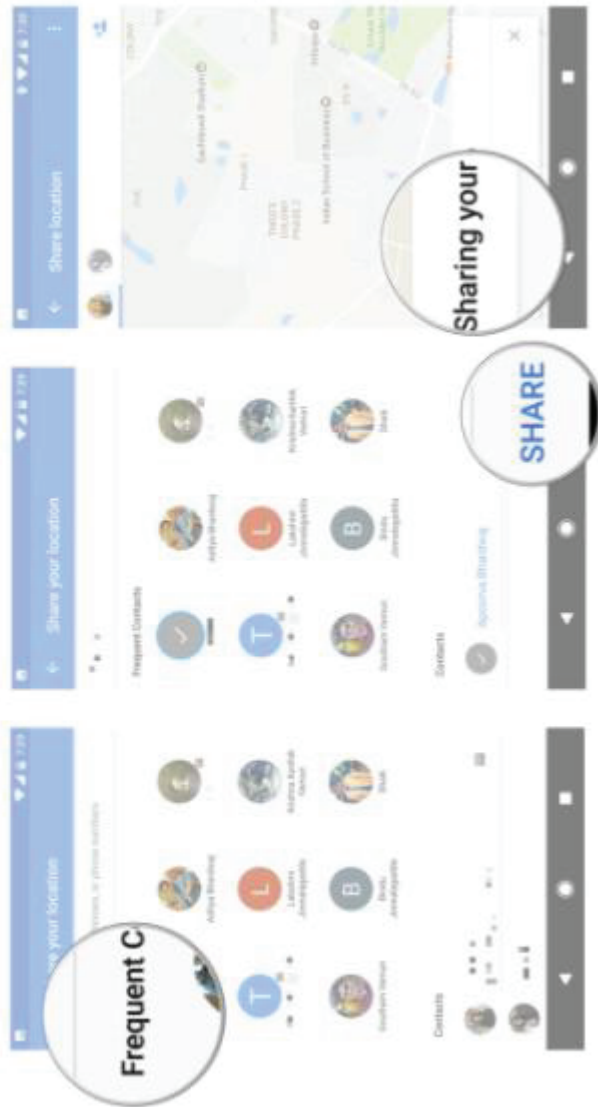
<https://www.androidcentral.com/how-share-location-google-maps>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products

- 7. You'll see a list of your frequent contacts at the top, along with a full list of contacts. Pick the contacts by tapping their name.
- 8. Once you've selected the contacts you want to share your location to, tap Share.
- 9. You'll see a message saying that the selected contact can view your location.



<https://www.androidcentral.com/how-share-location-google-maps>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

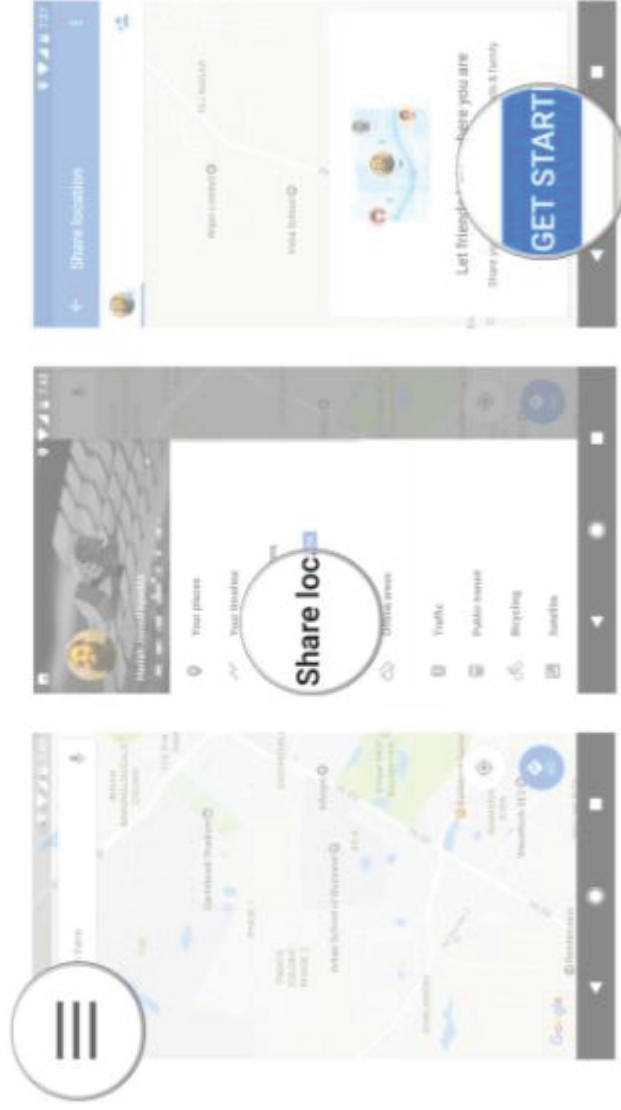
US9467838

Exemplary Supporting Evidence Regarding Accused Products

How to create a shareable link

You can also create a link and use it to share your location easily. Here's how to do it:

1. Tap the hamburger menu on the top left corner of the screen.
2. Select Share location.
3. Tap Get Started.



<https://www.androidcentral.com/how-share-location-google-maps>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

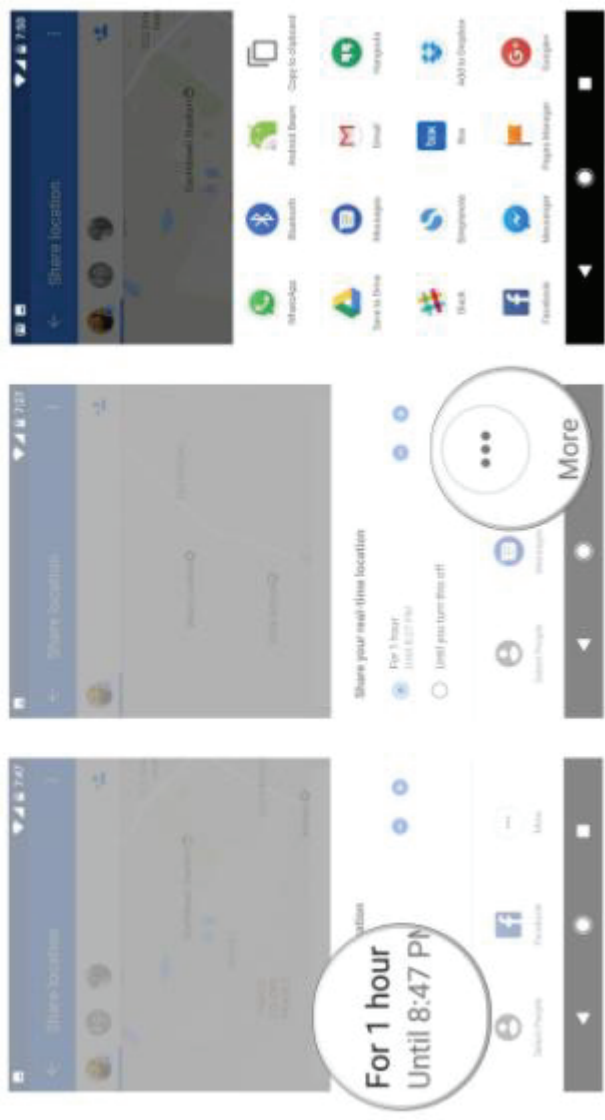
| | |
|------------------|---|
| <p>US9467838</p> | <p>Exemplary Supporting Evidence Regarding Accused Products</p> <ol style="list-style-type: none"> 4. Select the amount of time you want to share your location. 5. Tap More. 6. Select your app of choice to create and send a unique URL that broadcasts your current location. You can email it, send the link via Messenger, or even tweet it to the intended recipient.  <p>https://www.androidcentral.com/how-share-location-google-maps</p> |
|------------------|---|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

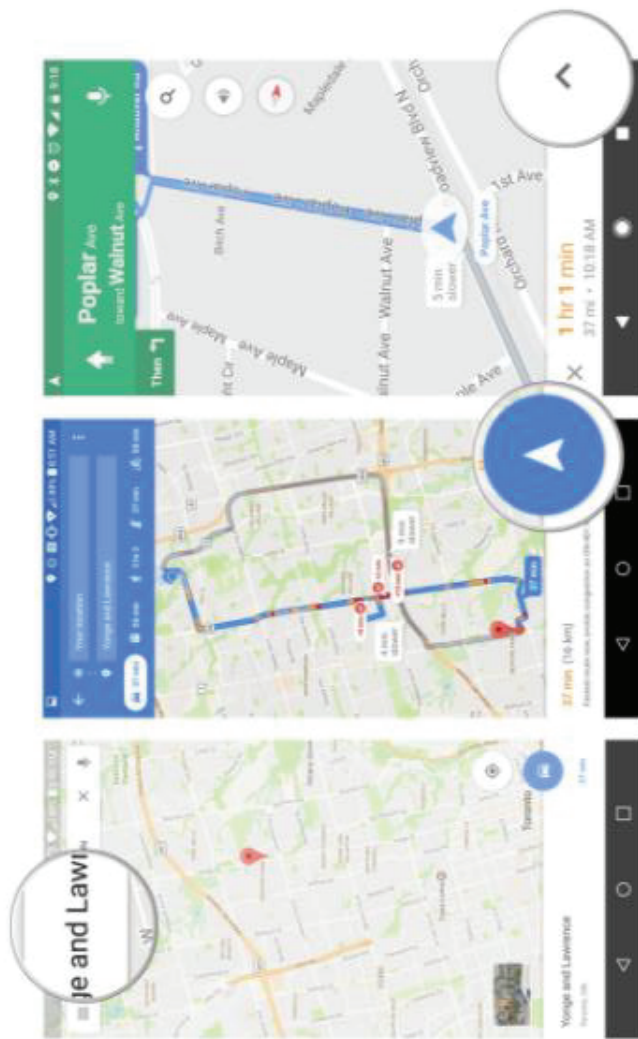
US9467838

Exemplary Supporting Evidence Regarding Accused Products

How to share your navigation directions while you walk, drive or transit

One of the best ways to share your location with someone is during a drive, walk or transit. If you're meeting a friend or family member somewhere, or navigating towards their home, Google Maps lets you share your location with them for the duration of the trip. It's magic!

1. In the search bar enter your destination.
2. Pick your navigation type (drive, transit, walk) and press the blue navigate button.
3. Tap the arrow next to the time-to-destination number at the bottom of the screen.



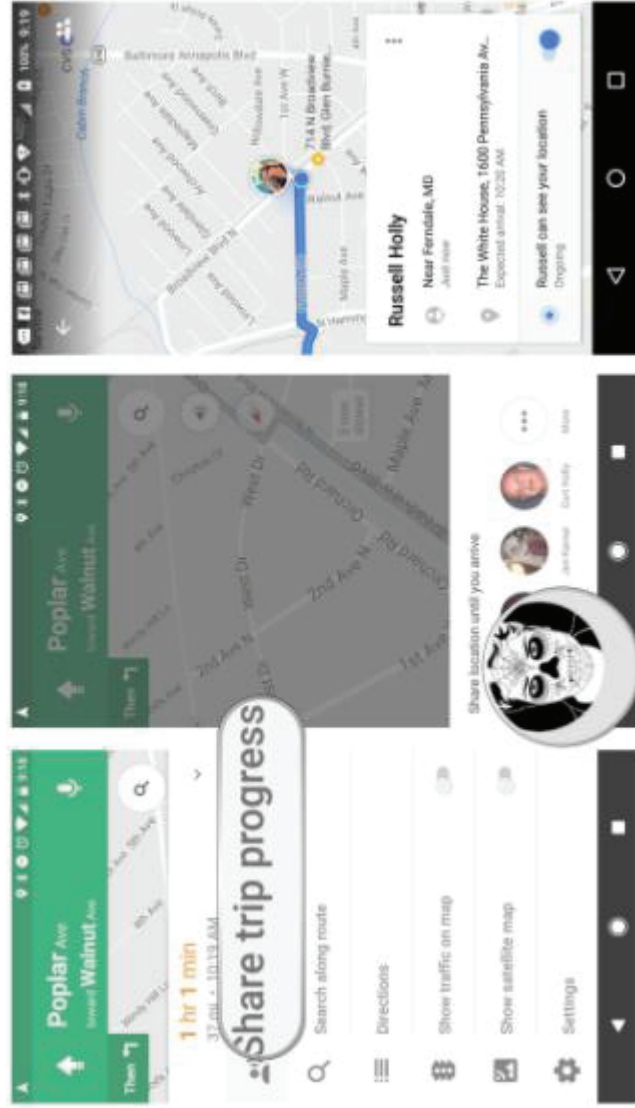
<https://www.androidcentral.com/how-share-location-google-maps>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products

- 4. Tap Share trip progress.
- 5. Choose one or more contacts to share trip progress.



You can also stop sharing your location with someone before a trip ends.
<https://www.androidcentral.com/how-share-location-google-maps>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| | |
|------------------|--|
| <p>US9467838</p> | <p>Exemplary Supporting Evidence Regarding Accused Products</p> |
| | <p>1. Tap the arrow next to the time-to-destination number at the bottom of the screen.</p> <p>2. Tap Stop sharing.</p>  <p>That's it!</p> <p>Are you excited that location sharing is back in Google Maps? How often do you use the feature? https://www.androidcentral.com/how-share-location-google-maps</p> <p>As shown below, a group may also be defined within Google Contacts.</p> |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE











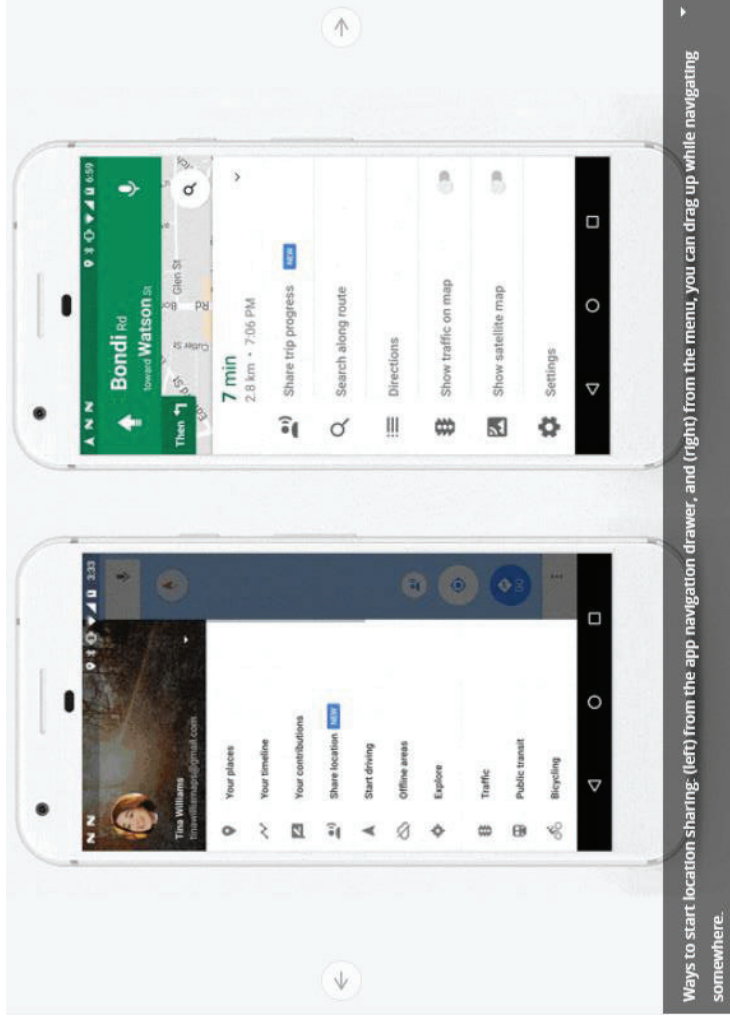
| | |
|-----------|---|
| US9467838 | <p data-bbox="196 716 228 1556">Exemplary Supporting Evidence Regarding Accused Products</p> <h2 data-bbox="237 1178 277 1535">See your contacts</h2> <ol data-bbox="305 338 602 1528" style="list-style-type: none"> <li data-bbox="305 1108 337 1528">1. Open your device's Contacts app . <li data-bbox="350 1346 383 1528">2. Tap Menu . <li data-bbox="407 961 440 1528">• See contacts by label: Choose a label from the list. <li data-bbox="453 709 485 1528">• See contacts for another account: Tap Down arrow  > pick an account. <li data-bbox="498 863 531 1528">• See the contacts for all your accounts: Choose All contacts. <p data-bbox="537 338 570 1507">Tip: If you have multiple contacts with the same information, the information will be grouped into one contact.</p> <ul data-bbox="576 716 609 1528" style="list-style-type: none"> <li data-bbox="576 716 609 1528">• See your Google Account contacts on the web: Go to Google Contacts . <p data-bbox="634 541 667 1556">https://support.google.com/android/answer/6118731?hl=en&ref_topic=6118711</p> <h2 data-bbox="711 1178 751 1535">Label your contacts</h2> <p data-bbox="776 1087 808 1535">You can group contacts together using labels.</p> <ol data-bbox="829 359 1024 1528" style="list-style-type: none"> <li data-bbox="829 1150 862 1528">1. Open your device's Contacts app . <li data-bbox="875 1213 907 1528">2. Tap Menu  > Create label. <li data-bbox="920 1213 953 1528">3. Enter a label name and tap Ok. <li data-bbox="966 842 998 1528">• Add one contact to a label: Tap Add contact  > choose a contact. <li data-bbox="1011 359 1044 1528">• Add multiple contacts to a label: Tap Add contact  > touch and hold a contact > tap the other contacts > tap Add. <p data-bbox="1044 541 1076 1556">https://support.google.com/android/answer/6118731?hl=en&ref_topic=6118711</p> <h2 data-bbox="1133 1136 1174 1535">Share your contacts</h2> <ol data-bbox="1198 1031 1360 1528" style="list-style-type: none"> <li data-bbox="1198 1108 1230 1528">1. Open your device's Contacts app . <li data-bbox="1243 1234 1276 1528">2. Tap a contact in the list. <li data-bbox="1289 1241 1321 1528">3. Tap More  > Share. <li data-bbox="1334 1031 1367 1528">4. Choose how you want to share the contact. <p data-bbox="1377 541 1409 1556">https://support.google.com/android/answer/6118731?hl=en&ref_topic=6118711</p> |
|-----------|---|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products



<https://arstechnica.com/gadgets/2017/03/location-sharing-finally-returns-to-google-maps/>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products

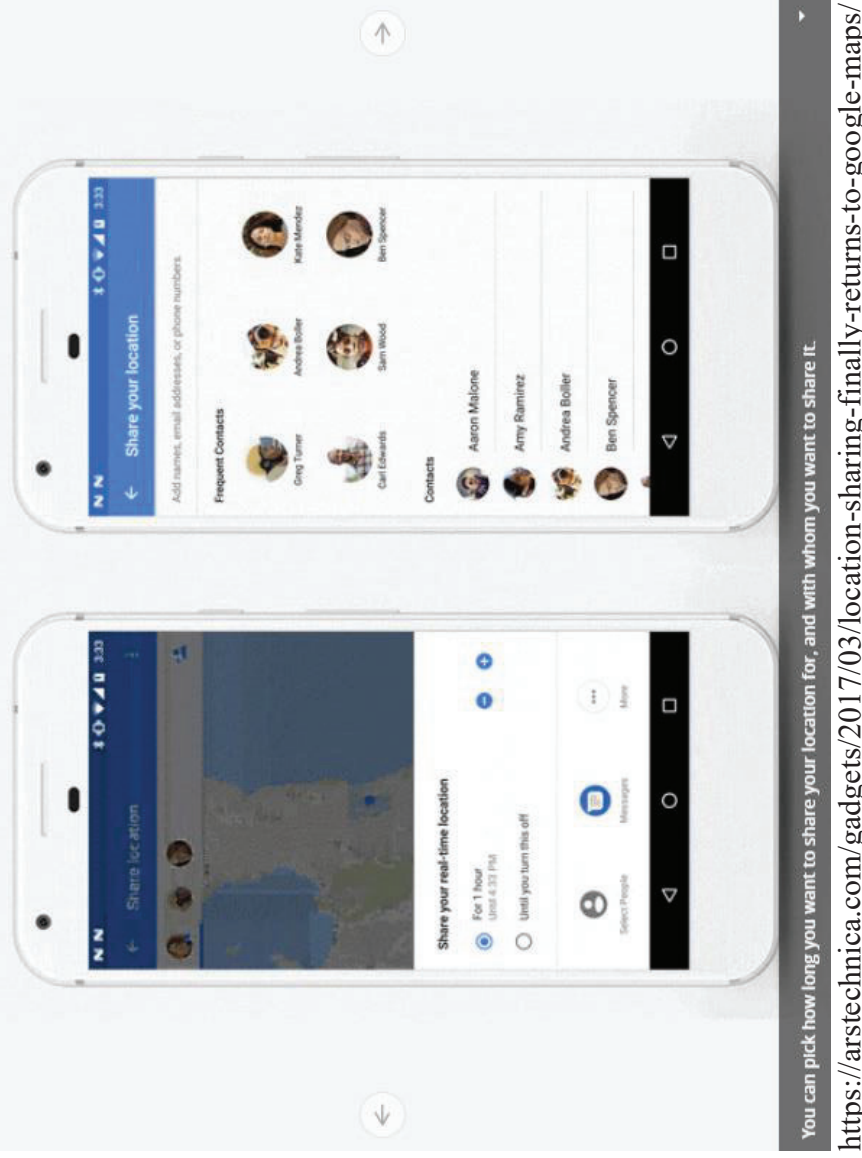
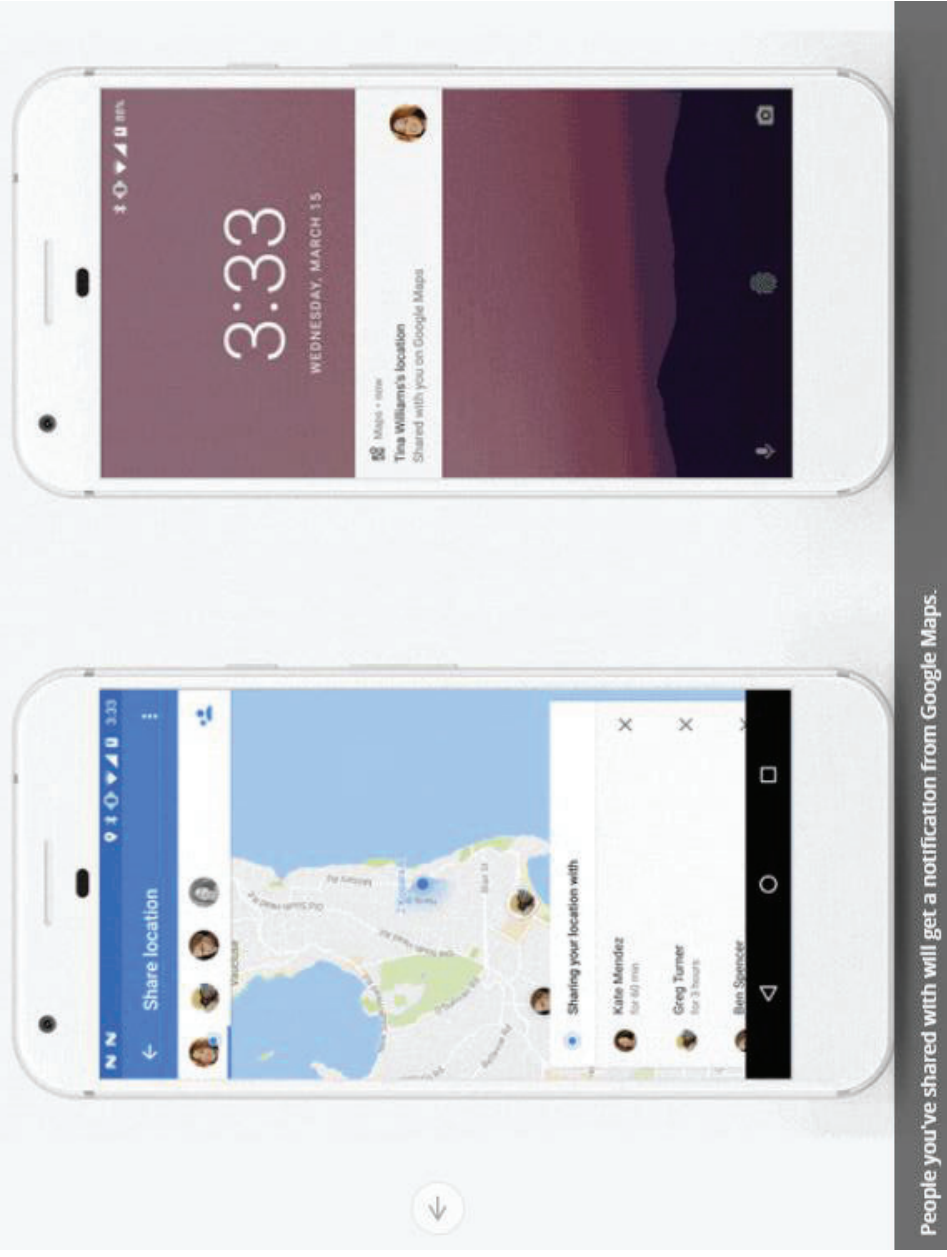


Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products



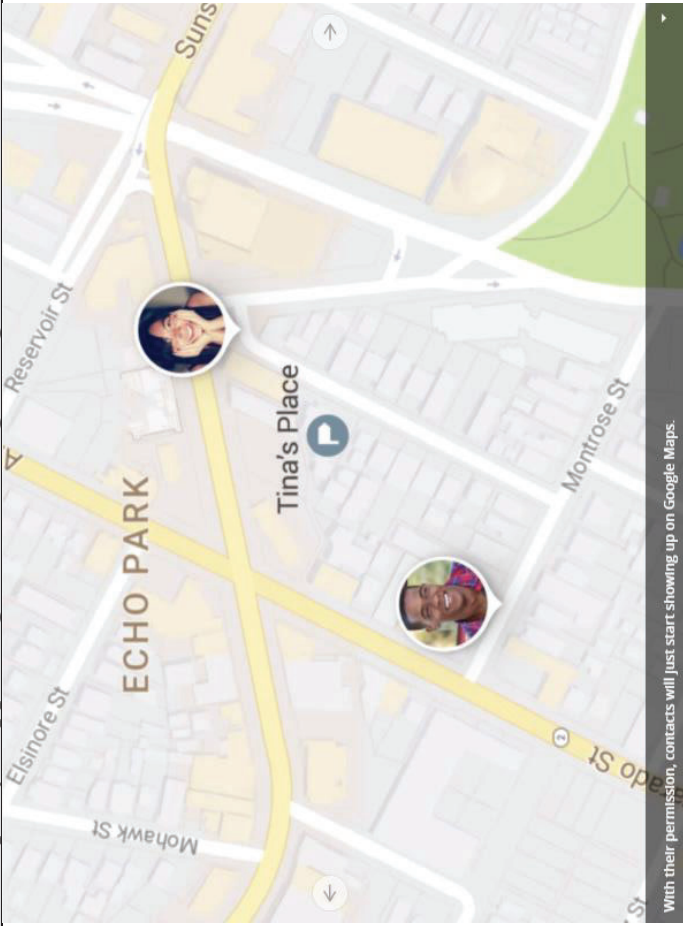
People you've shared with will get a notification from Google Maps.

<https://arstechnica.com/gadgets/2017/03/location-sharing-finally-returns-to-google-maps/>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products



<https://arstechnica.com/gadgets/2017/03/location-sharing-finally-returns-to-google-maps/>

Exemplary Find My Device Screenshots:

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

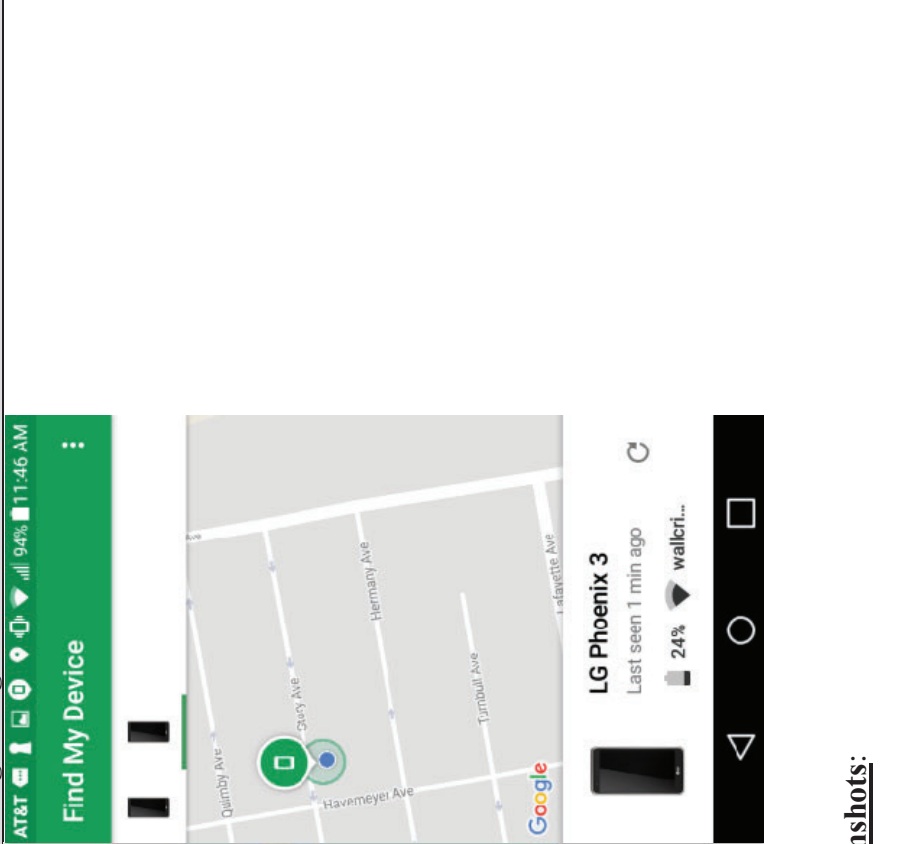
| | |
|------------------|---|
| <p>US9467838</p> | <p>Exemplary Supporting Evidence Regarding Accused Products</p>  <p>Exemplary Google Maps Screenshots:</p> |
|------------------|---|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

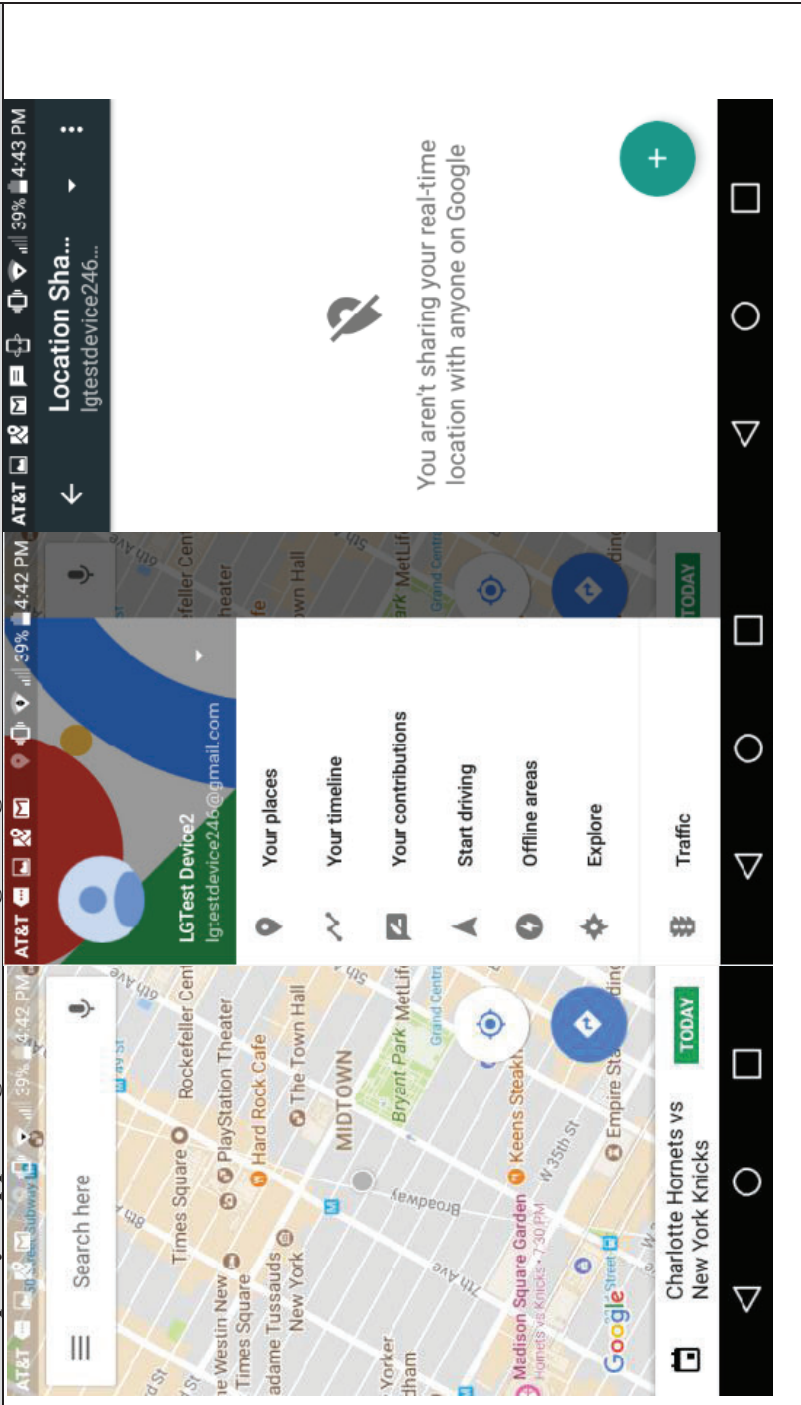
| | |
|-----------|--|
| US9467838 | <p data-bbox="188 709 228 1570">Exemplary Supporting Evidence Regarding Accused Products</p>  |
|-----------|--|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

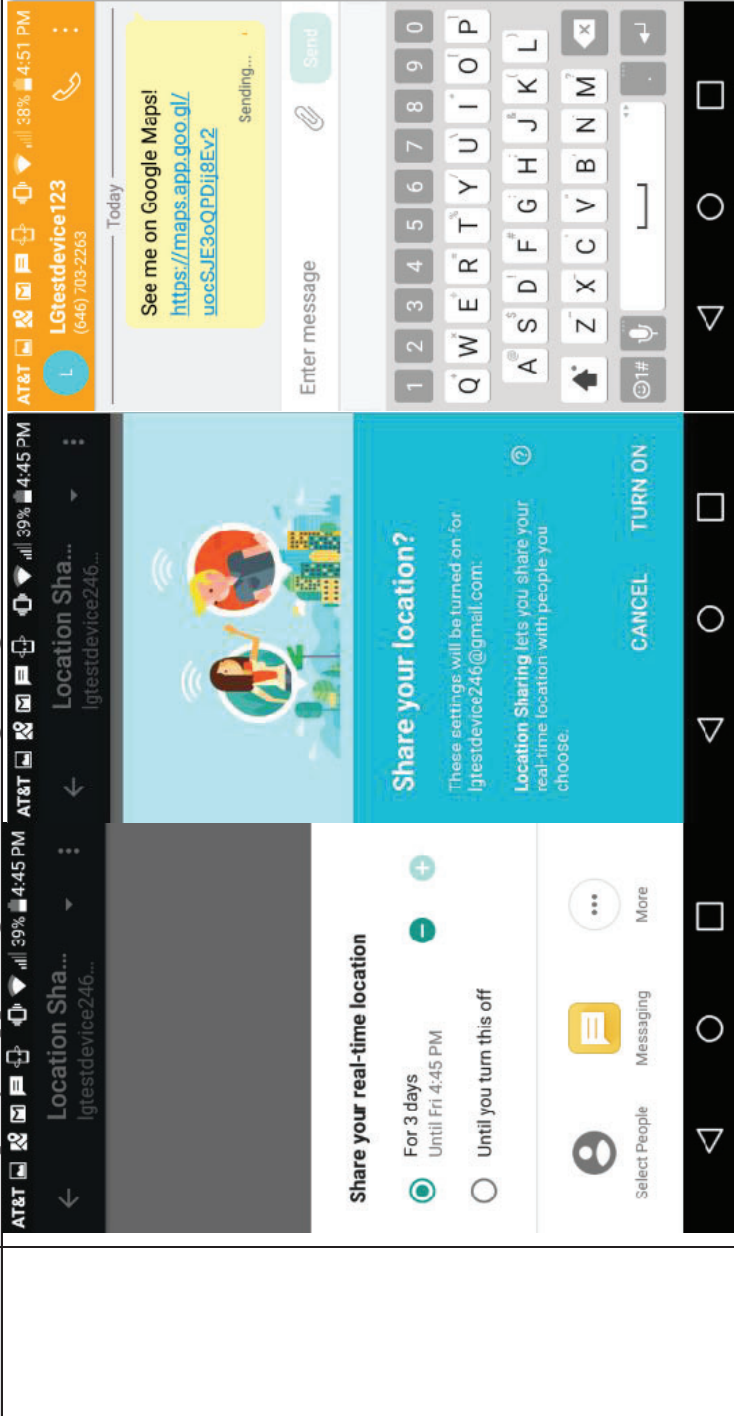
| | |
|-----------|---|
| US9467838 | <p>Exemplary Supporting Evidence Regarding Accused Products</p>  <p>The screenshot displays an Android messaging app interface. At the top, the status bar shows 'AT&T', signal strength, 39% battery, and the time 4:45 PM. The messaging app header shows 'AT&T' and 'LGtestdevice123 (646) 703-2263'. A text message from 'LGtestdevice123' is shown with the content: 'See me on Google Maps! https://maps.app.goo.gl/uocSJE3oQPdJj8Ev2 Sending...'. Below the message is a keyboard. A notification from 'Location Sha... lgtestdevice246...' is visible, with the text: 'Share your real-time location', 'For 3 days Until Fri 4:45 PM', and 'Until you turn this off'. The notification also includes icons for 'Select People' and 'Messaging', and a 'More' option. The notification background features an illustration of two people and the text: 'Share your location? These settings will be turned on for lgtestdevice246@gmail.com. Location Sharing lets you share your real-time location with people you choose. CANCEL TURN ON'.</p> |
|-----------|---|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

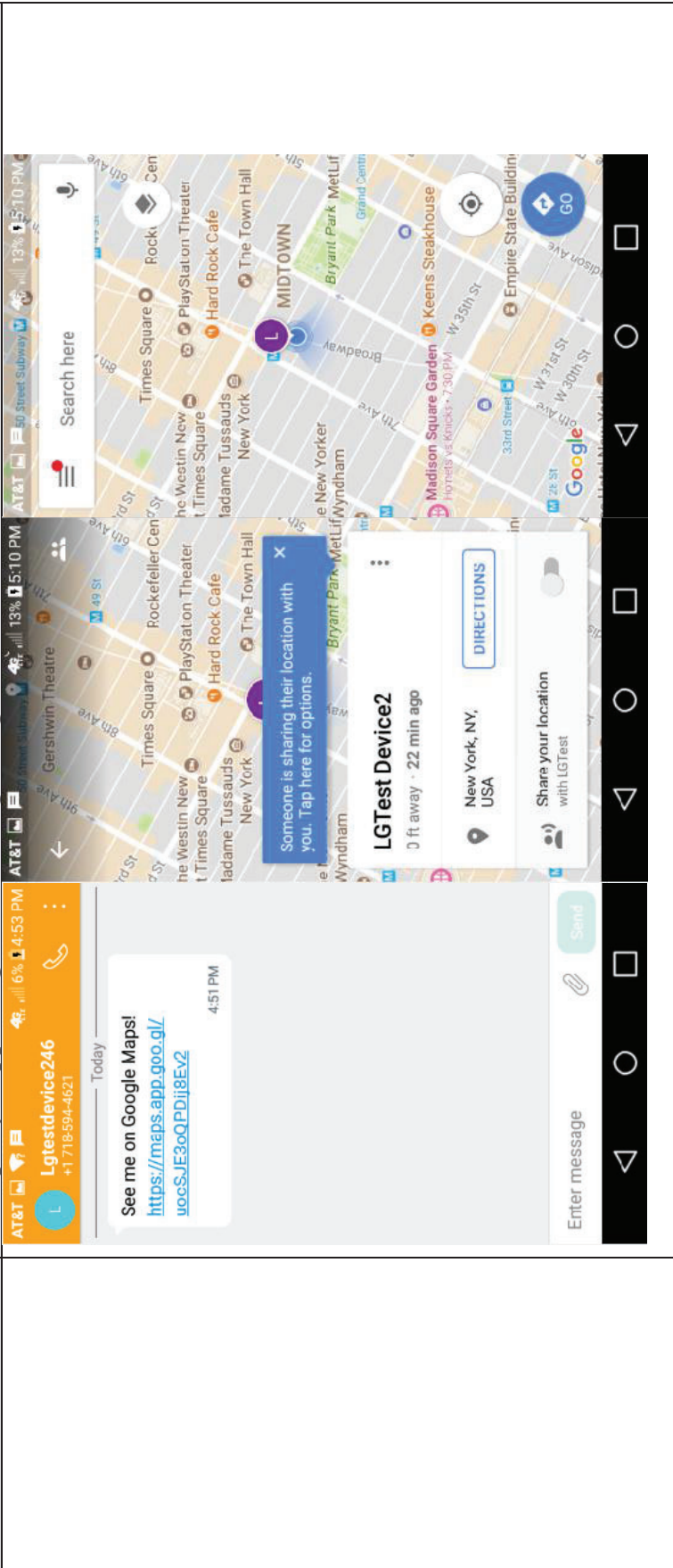
| | |
|------------------|--|
| <p>US9467838</p> | <p>Exemplary Supporting Evidence Regarding Accused Products</p>  |
|------------------|--|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

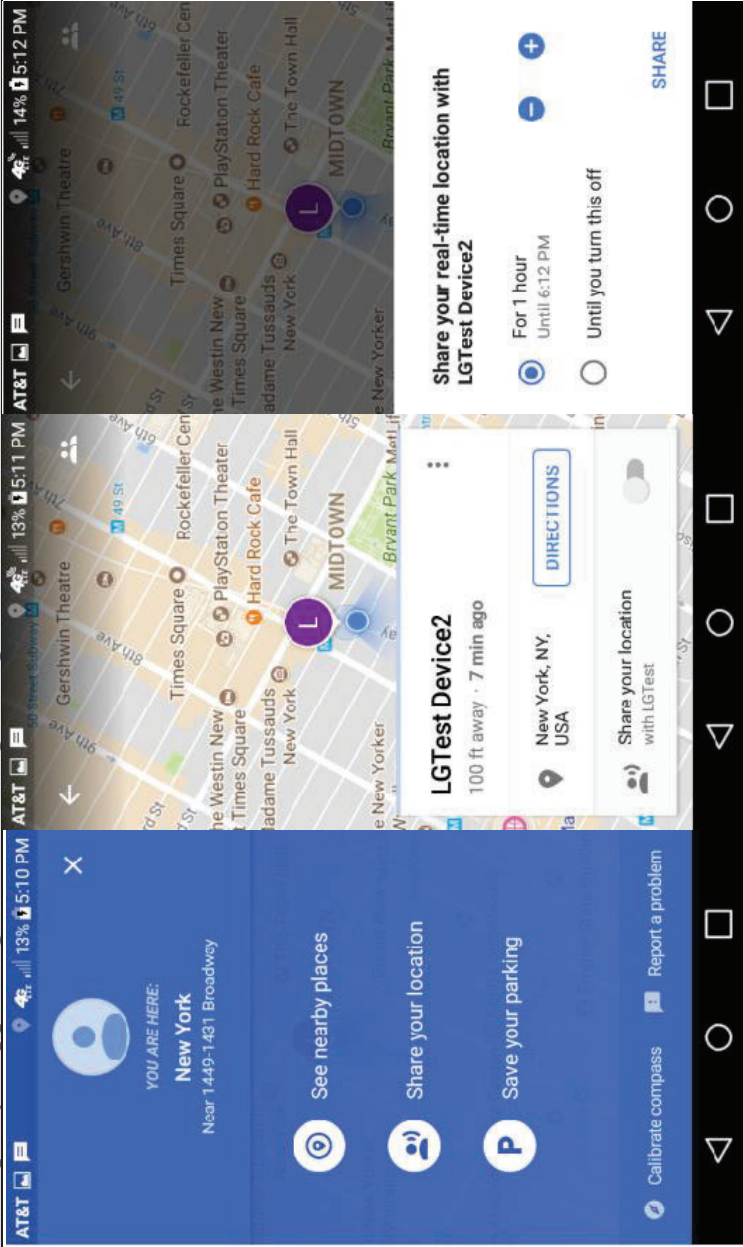
| | |
|------------------|--|
| <p>US9467838</p> | <p>Exemplary Supporting Evidence Regarding Accused Products</p>  |
|------------------|--|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

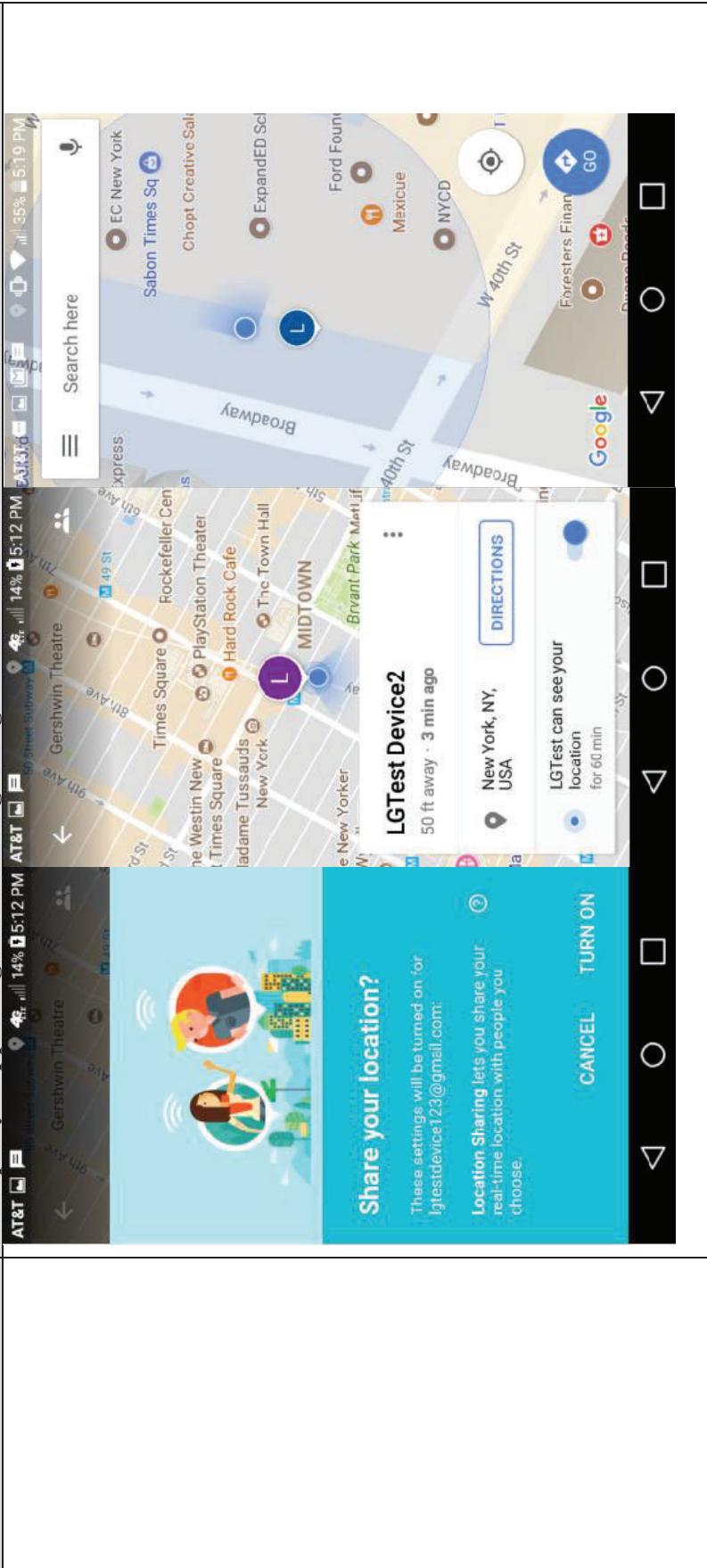
| | |
|------------------|---|
| <p>US9467838</p> | <p>Exemplary Supporting Evidence Regarding Accused Products</p>  |
|------------------|---|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products

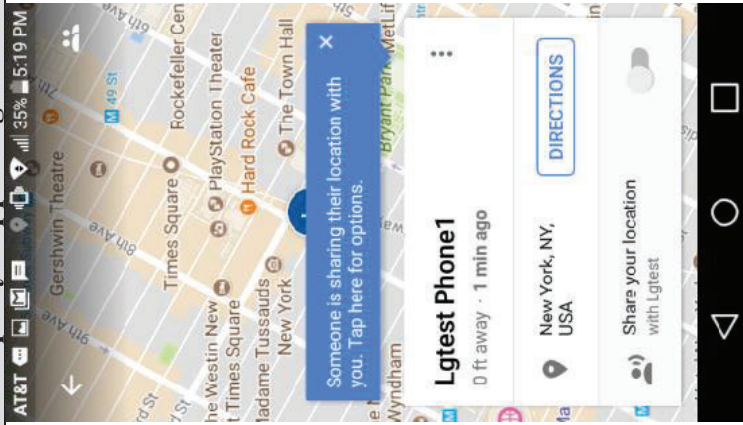


Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

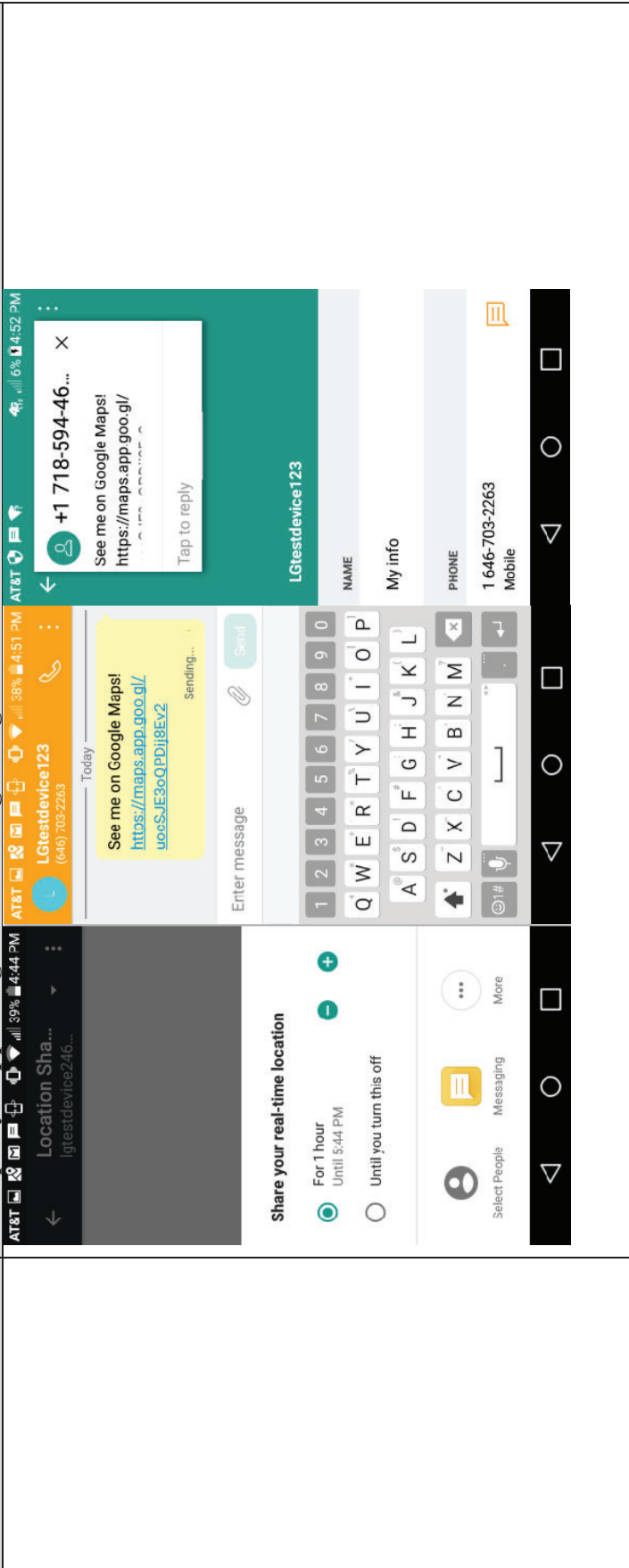
| | |
|------------------|--|
| <p>US9467838</p> |  <p>The screenshot displays a text message conversation on an Android phone. The contact is 'LGtestdevice123'. The messages are as follows:</p> <ul style="list-style-type: none">Message 1 (Received): "Location Sha... [Location Share Icon] LGtestdevice246..."Message 2 (Received): "See me on Google Maps! https://maps.app.goo.gl/uocSJE3oQDii8EV2 Sending..."Message 3 (Received): "See me on Google Maps! https://maps.app.goo.gl/ Tap to reply" <p>The contact information card at the bottom shows:</p> <ul style="list-style-type: none">NAME: My infoPHONE: 1 646-703-2263 Mobile <p>The phone's status bar at the top shows AT&T service, 39% battery, and the time 4:44 PM.</p> |
|------------------|--|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

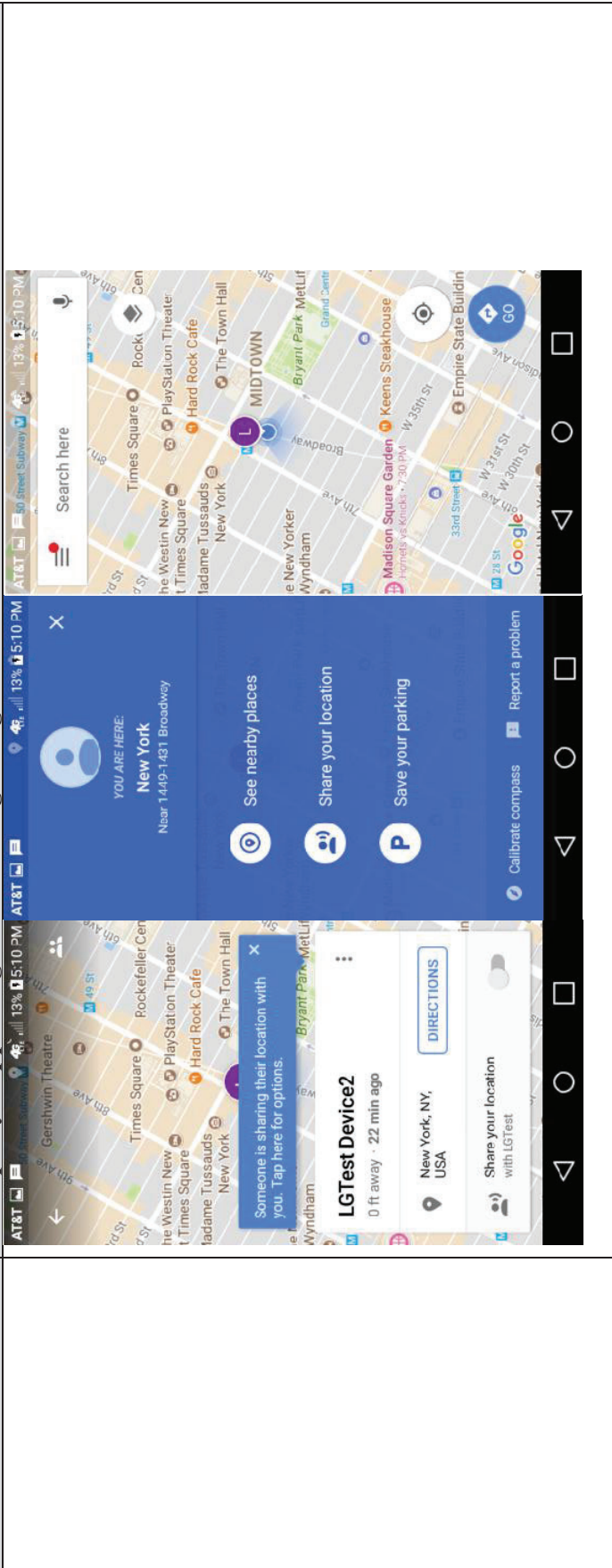
| | |
|------------------|---|
| <p>US9467838</p> | <p>Exemplary Supporting Evidence Regarding Accused Products</p>  |
|------------------|---|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

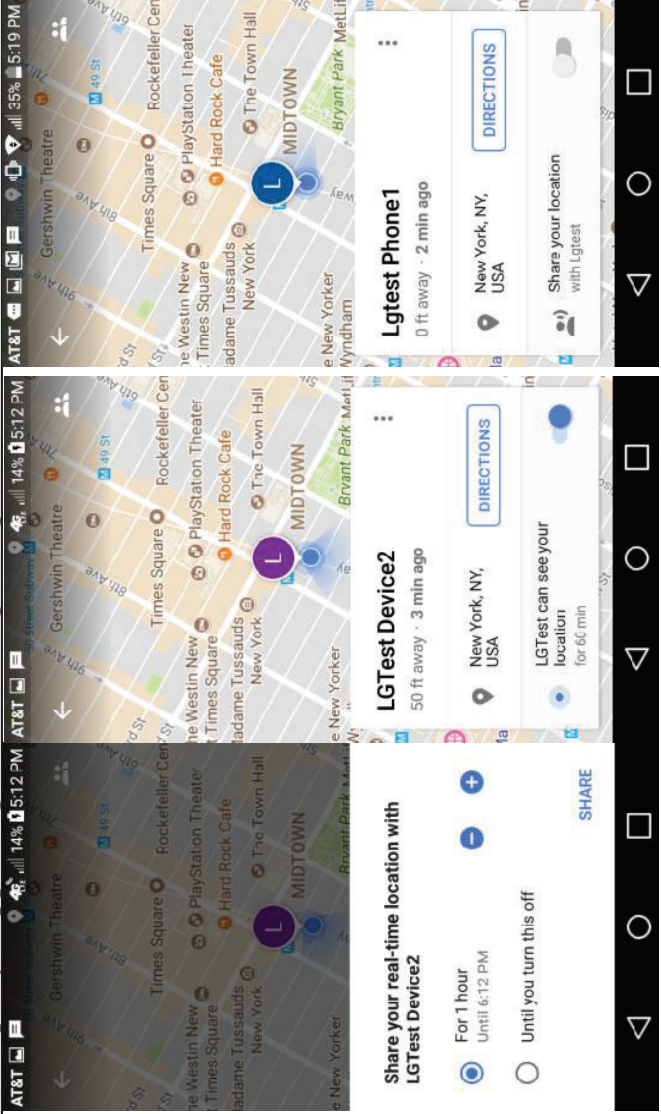
| | |
|------------------|--|
| <p>US9467838</p> | <p>Exemplary Supporting Evidence Regarding Accused Products</p>  <p>Exemplary Source Code:</p> <p>The above functionality is performed at least in part by the following publicly available source code and/or source code that invokes or is invoked by the following source code (or a substantially similar copy compiled and loaded onto the Accused Products by ZTE). AGIS reserves the right to amend these contentions to include additional source code as discovery progresses and as additional source code is made available.</p> |
|------------------|--|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| | |
|------------------|---|
| US9467838 | <p data-bbox="186 709 228 1570">Exemplary Supporting Evidence Regarding Accused Products</p> <h2 data-bbox="235 1056 293 1549">Contacts Provider</h2> <p data-bbox="326 604 591 1549">The Contacts Provider is a powerful and flexible Android component that manages the device's central repository of data about people. The Contacts Provider is the source of data you see in the device's contacts application, and you can also access its data in your own application and transfer data between the device and online services. The provider accommodates a wide range of data sources and tries to manage as much data as possible for each person, with the result that its organization is complex. Because of this, the provider's API includes an extensive set of contract classes and interfaces that facilitate both data retrieval and modification.</p> <p data-bbox="623 1234 647 1549">This guide describes the following:</p> <ul data-bbox="672 590 850 1549" style="list-style-type: none">• The basic provider structure.• How to retrieve data from the provider.• How to modify data in the provider.• How to write a sync adapter for synchronizing data from your server to the Contacts Provider. <p data-bbox="859 590 889 1570">https://developer.android.com/guide/topics/providers/provider.html</p> |
|------------------|---|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| | |
|-----------|---|
| US9467838 | <p data-bbox="191 247 228 1564">Exemplary Supporting Evidence Regarding Accused Products</p> <p data-bbox="240 247 267 1564">Overview</p> <p data-bbox="289 247 332 1564">ContactsContract defines an extensible database of contact-related information. Contact information is stored in a three-tier data model:</p> <ul data-bbox="341 247 544 1564" style="list-style-type: none"> <li data-bbox="341 247 397 1564">• A row in the ContactsContract.Data table can store any kind of personal data, such as a phone number or email addresses. The set of data kinds that can be stored in this table is open-ended. There is a predefined set of common kinds, but any application can add its own data kinds. <li data-bbox="414 247 470 1564">• A row in the ContactsContract.RawContacts table represents a set of data describing a person and associated with a single account (for example, one of the user's Gmail accounts). <li data-bbox="487 247 544 1564">• A row in the ContactsContract.Contacts table represents an aggregate of one or more RawContacts presumably describing the same person. When data in or associated with the RawContacts table is changed, the affected aggregate contacts are updated as necessary. <p data-bbox="568 247 592 1564">Other tables include:</p> <ul data-bbox="617 247 901 1564" style="list-style-type: none"> <li data-bbox="617 247 673 1564">• ContactsContract.Groups, which contains information about raw contact groups such as Gmail contact groups. The current API does not support the notion of groups spanning multiple accounts. <li data-bbox="690 247 714 1564">• ContactsContract.StatusUpdates, which contains social status updates including IM availability. <li data-bbox="730 247 755 1564">• ContactsContract.AggregationExceptions, which is used for manual aggregation and disaggregation of raw contacts <li data-bbox="771 247 795 1564">• ContactsContract.Settings, which contains visibility and sync settings for accounts and groups. <li data-bbox="812 247 836 1564">• ContactsContract.SyncState, which contains free-form data maintained on behalf of sync adapters <li data-bbox="852 247 876 1564">• ContactsContract.PhoneLookup, which is used for quick caller-ID lookup <p data-bbox="917 247 950 1564">https://developer.android.com/reference/android/provider/ContactsContract.html</p> <p data-bbox="958 247 990 1564">Data</p> <p data-bbox="1023 247 1177 1564">As noted previously, the data for a raw contact is stored in a ContactsContract.Data row that is linked to the raw contact's _id value. This allows a single raw contact to have multiple instances of the same type of data such as email addresses or phone numbers. For example, if "Thomas Higginson" for emilyd@gmail.com (the raw contact row for Thomas Higginson associated with the Google account emilyd@gmail.com) has a home email address of thigg@gmail.com and a work email address of thomas.higginson@gmail.com, the Contacts Provider stores the two email address rows and links them both to the raw contact.</p> <p data-bbox="1193 247 1307 1564">Notice that different types of data are stored in this single table. Display name, phone number, email, postal address, photo, and website detail rows are all found in the ContactsContract.Data table. To help manage this, the ContactsContract.Data table has some columns with descriptive names, and others with generic names. The contents of a descriptive-name column have the same meaning regardless of the type of data in the row, while the contents of a generic-name column have different meanings depending on the type of data.</p> <p data-bbox="1323 247 1356 1564">https://developer.android.com/guide/topics/providers/contacts-provider.html</p> |
|-----------|---|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| Exemplary Supporting Evidence Regarding Accused Products | | | | |
|---|--------------------------|--|-----------|--|
| Task | Action | Data | MIME type | Notes |
| Pick a contact from a list | <code>ACTION_PICK</code> | One of: <ul style="list-style-type: none"> <code>Contacts.CONTENT_URI</code>, which displays a list of contacts. <code>Phone.CONTENT_URI</code>, which displays a list of phone numbers for a raw contact. <code>StructuredPostal.CONTENT_URI</code>, which displays a list of postal addresses for a raw contact. <code>Email.CONTENT_URI</code>, which displays a list of email addresses for a raw contact. | Not used | Displays a list of raw contacts or a list of data from a raw contact, depending on the content URI type you supply. Call <code>startActivityForResult()</code> , which returns the content URI of the selected row. The form of the URI is the table's content URI with the row's <code>LOOKUP_ID</code> appended to it. The device's contacts app delegates read and write permissions to this content URI for the life of your activity. See the Content Provider Basics guide for more details. |
| https://developer.android.com/guide/topics/providers/contacts-provider.html | | | | |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| US9467838 | Exemplary Supporting Evidence Regarding Accused Products |
|-----------|---|
| 25 | * Parsed form of the intent sent to the Contacts application. |
| 26 | */ |
| 27 | public class ContactsRequest { |
| 28 | |
| 29 | /** Default mode: browse contacts */ |
| 30 | public static final int ACTION_DEFAULT = 10; |
| 31 | |
| 32 | /** Show all contacts */ |
| 33 | public static final int ACTION_ALL_CONTACTS = 15; |
| 34 | |
| 35 | /** Show all contacts with phone numbers */ |
| 36 | public static final int ACTION_CONTACTS_WITH_PHONES = 17; |
| 37 | |
| 38 | /** Show contents of a specific group */ |
| 39 | public static final int ACTION_GROUP = 20; |
| 40 | |
| 41 | /** Show all starred contacts */ |
| 42 | public static final int ACTION_STARRED = 30; |
| 43 | |
| 44 | /** Show frequently contacted contacts */ |
| 45 | public static final int ACTION_FREQUENT = 40; |
| 46 | |
| 47 | /** Show starred and the frequent */ |
| 48 | public static final int ACTION_STREQUENT = 50; |
| 49 | |
| 50 | /** Show all contacts and pick them when clicking */ |
| 51 | public static final int ACTION_PICK_CONTACT = 60; |
| 52 | |
| 53 | /** Show all contacts as well as the option to create a new one */ |
| 54 | public static final int ACTION_PICK_OR_CREATE_CONTACT = 70; |
| 55 | |
| 56 | /** Show all contacts and pick them for edit when clicking, and allow creating a new contact */ |
| 57 | public static final int ACTION_INSERT_OR_EDIT_CONTACT = 80; |
| | https://android.googlesource.com/platform/packages/apps/Contacts/+nougat-mr1- release/src/com/android/contacts/list/ContactsRequest.java |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| US9467838 | Exemplary Supporting Evidence Regarding Accused Products |
|-----------|---|
| 59 | /** Show all phone numbers and pick them when clicking */ |
| 60 | public static final int ACTION_PICK_PHONE = 90; |
| 61 | |
| 62 | /** Show all postal addresses and pick them when clicking */ |
| 63 | public static final int ACTION_PICK_POSTAL = 100; |
| 64 | |
| 65 | /** Show all postal addresses and pick them when clicking */ |
| 66 | public static final int ACTION_PICK_EMAIL = 105; |
| 67 | |
| 68 | /** Show all contacts and create a shortcut for the picked contact */ |
| 69 | public static final int ACTION_CREATE_SHORTCUT_CONTACT = 110; |
| 70 | |
| 71 | /** Show all phone numbers and create a call shortcut for the picked number */ |
| 72 | public static final int ACTION_CREATE_SHORTCUT_CALL = 120; |
| 73 | |
| 74 | /** Show all phone numbers and create an SMS shortcut for the picked number */ |
| 75 | public static final int ACTION_CREATE_SHORTCUT_SMS = 130; |
| 76 | |
| 77 | /** Show all contacts and activate the specified one */ |
| 78 | public static final int ACTION_VIEW_CONTACT = 140; |
| 79 | |
| 80 | /** Show contacts recommended for joining with a specified target contact */ |
| 81 | public static final int ACTION_PICK_JOIN = 150; |
| | https://android.googlesource.com/platform/packages/apps/Contacts/+nougat-mr1- release/src/com/android/contacts/list/ContactsRequest.java |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| US9467838 | Exemplary Supporting Evidence Regarding Accused Products |
|-----------|---|
| 104 | * Displays a list to browse contacts. |
| 105 | */ |
| 106 | public class PeopleActivity extends ContactsActivity implements |
| 107 | View.OnCreateContextMenuListener, |
| 108 | View.OnClickListener, |
| 109 | ActionBarAdapter.Listener, |
| 110 | DialogManager.DialogShowingViewActivity, |
| 111 | ContactListFilterController.ContactListFilterListener, |
| 112 | ProviderStatusListener, |
| 113 | MultiContactDeleteListener, |
| 114 | JoinContactsListener { |
| | https://android.googlesource.com/platform/packages/apps/Contacts/+nougat-mr1-release/src/com/android/contacts/activities/PeopleActivity.java |
| 145 | * Showing a list of Contacts. Also used for showing search results in search mode. |
| 146 | */ |
| 147 | private MultiSelectContactsListFragment mAllFragment; |
| 148 | private ContactTileListFragment mFavoritesFragment; |
| | https://android.googlesource.com/platform/packages/apps/Contacts/+nougat-mr1-release/src/com/android/contacts/activities/PeopleActivity.java |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| US9467838 | Exemplary Supporting Evidence Regarding Accused Products |
|---|--|
| <pre> 458 private void configureFragments(boolean fromRequest) { 459 if (fromRequest) { 460 ContactListFilter filter = null; 461 int actionCode = mRequest.getActionCode(); 462 boolean searchMode = mRequest.isSearchMode(); 463 final int tabToOpen; 464 switch (actionCode) { 465 case ContactsRequest.ACTION_ALL_CONTACTS: 466 filter = ContactListFilter.createFilterWithType(467 ContactListFilter.FILTER_TYPE_ALL_ACCOUNTS); 468 tabToOpen = TabState.ALL; 469 break; 470 case ContactsRequest.ACTION_CONTACTS_WITH_PHONES: 471 filter = ContactListFilter.createFilterWithType(472 ContactListFilter.FILTER_TYPE_WITH_PHONE_NUMBERS_ONLY); 473 tabToOpen = TabState.ALL; 474 break; 475 case ContactsRequest.ACTION_FREQUENT: 476 case ContactsRequest.ACTION_STREQUENT: 477 case ContactsRequest.ACTION_STARRED: 478 tabToOpen = TabState.FAVORITES; 479 break; 480 case ContactsRequest.ACTION_VIEW_CONTACT: 481 tabToOpen = TabState.ALL; 482 break; 483 default: 484 tabToOpen = -1; 485 break; 486 } 487 } </pre> <p> https://android.googlesource.com/platform/packages/apps/Contacts/+nougat-mr1-release/src/com/android/contacts/activities/PeopleActivity.java </p> | |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| US9467838 | Exemplary Supporting Evidence Regarding Accused Products |
|-----------|--|
| 488 | if (tabToOpen != -1) { |
| 489 | mActionBarAdapter.setCurrentTab(tabToOpen); |
| 490 | } |
| 491 | if (filter != null) { |
| 492 | mContactListFilterController.setContactListFilter(filter, false); |
| 493 | searchMode = false; |
| 494 | } |
| 495 | if (mRequest.getContactUri() != null) { |
| 496 | searchMode = false; |
| 497 | } |
| 498 | mActionBarAdapter.setSearchMode(searchMode); |
| 499 | configureContactListFragmentForRequest(); |
| 500 | } |
| 501 | configureContactListFragment(); |
| 502 | invalidateOptionsMenuIfNeeded(); |
| 503 | } |
| 504 | } |
| 505 | } |
| 506 | } |
| 507 | } |
| 508 | } |
| | https://android.googlesource.com/platform/packages/apps/Contacts/+/nougat-mr1- |
| | release/src/com/android/contacts/activities/PeopleActivity.java |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| US9467838 | Exemplary Supporting Evidence Regarding Accused Products |
|-----------|---|
| | <pre> 35 public class ProfileAndContactsLoader extends CursorLoader { 36 37 private boolean mLoadProfile; 38 39 private String[] mProjection; 40 41 private Uri mExtraUri; 42 private String[] mExtraProjection; 43 private String mExtraSelection; 44 private String[] mExtraSelectionArgs; 45 private boolean mMergeExtraContactsAfterPrimary; 46 47 public ProfileAndContactsLoader(Context context) { 48 super(context); 49 } </pre> <p> https://android.googlesource.com/platform/packages/apps/ContactsCommon/+nougat-mr1-release/src/com/android/contacts/common/list/ProfileAndContactsLoader.java </p> |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| US9467838 | Exemplary Supporting Evidence Regarding Accused Products |
|-----------|--|
| 35 | public final class GroupMemberLoader extends CursorLoader { |
| 36 | |
| 37 | public static class GroupEditorQuery { |
| 38 | private static final String[] PROJECTION = new String[] { |
| 39 | Data.CONTACT_ID, // 0 |
| 40 | Data.RAW_CONTACT_ID, // 1 |
| 41 | Data.DISPLAY_NAME_PRIMARY, // 2 |
| 42 | Data.PHOTO_URI, // 3 |
| 43 | Data.LOOKUP_KEY, // 4 |
| 44 | }; |
| 45 | |
| 46 | public static final int CONTACT_ID = 0; |
| 47 | public static final int RAW_CONTACT_ID = 1; |
| 48 | public static final int CONTACT_DISPLAY_NAME_PRIMARY = 2; |
| 49 | public static final int CONTACT_PHOTO_URI = 3; |
| 50 | public static final int CONTACT_LOOKUP_KEY = 4; |
| 51 | } |
| 52 | |
| 53 | public static class GroupDetailQuery { |
| 54 | private static final String[] PROJECTION = new String[] { |
| 55 | Data.CONTACT_ID, // 0 |
| 56 | Data.PHOTO_URI, // 1 |
| 57 | Data.LOOKUP_KEY, // 2 |
| 58 | Data.DISPLAY_NAME_PRIMARY, // 3 |
| 59 | Data.CONTACT_PRESENCE, // 4 |
| 60 | Data.CONTACT_STATUS, // 5 |
| 61 | }; |
| 62 | |
| 63 | public static final int CONTACT_ID = 0; |
| 64 | public static final int CONTACT_PHOTO_URI = 1; |
| 65 | public static final int CONTACT_LOOKUP_KEY = 2; |
| 66 | public static final int CONTACT_DISPLAY_NAME_PRIMARY = 3; |
| 67 | public static final int CONTACT_PRESENCE_STATUS = 4; |
| 68 | public static final int CONTACT_STATUS = 5; |
| 69 | } |
| 70 | |
| 71 | private final long mGroupId; |
| | https://android.googlesource.com/platform/packages/apps/Contacts/+nougat-mr1- |
| | release/src/com/android/contacts/GroupMemberLoader.java |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products

24 * Group loader for the group list that includes details such as the number of contacts per group
 25 * and number of groups per account. This list is sorted by account type, account name, where the
 26 * group names are in alphabetical order. Note that the list excludes default, favorite, and deleted
 27 * groups.
 28 */

```

29 public final class GroupListLoader extends CursorLoader {
30
31     private final static String[] COLUMNS = new String[] {
32         Groups.ACCOUNT_NAME,
33         Groups.ACCOUNT_TYPE,
34         Groups.DATA_SET,
35         Groups._ID,
36         Groups.TITLE,
37         Groups.SUMMARY_COUNT,
38     };
39
40     public final static int ACCOUNT_NAME = 0;
41     public final static int ACCOUNT_TYPE = 1;
42     public final static int DATA_SET = 2;
43     public final static int GROUP_ID = 3;
44     public final static int TITLE = 4;
45     public final static int MEMBER_COUNT = 5;
46
47     private static final Uri GROUP_LIST_URI = Groups.CONTENT_SUMMARY_URI;
48
49     public GroupListLoader(Context context) {
50         super(context, GROUP_LIST_URI, COLUMNS, Groups.ACCOUNT_TYPE + " NOT NULL AND "
51             + Groups.ACCOUNT_NAME + " NOT NULL AND " + Groups.AUTO_ADD + "=0 AND "
52             + Groups.FAVORITES + "=0 AND " + Groups.DELETED + "=0", null,
53             Groups.ACCOUNT_TYPE + ", " + Groups.ACCOUNT_NAME + ", " + Groups.DATA_SET + ", " +
54             Groups.TITLE + " COLLATE LOCALIZED ASC");
55     }
56 }

```

<https://android.googlesource.com/platform/packages/apps/Contacts/+/nougat-mr1-release/src/com/android/contacts/GroupListLoader.java>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products

```

24 * Group meta-data loader. Loads all groups or just a single group from the
25 * database (if given a {@link Uri}).
26 */
27 public final class GroupMetadataLoader extends CursorLoader {
28
29     private final static String[] COLUMNS = new String[] {
30         Groups.ACCOUNT_NAME,
31         Groups.ACCOUNT_TYPE,
32         Groups.DATA_SET,
33         Groups._ID,
34         Groups.TITLE,
35         Groups.AUTO_ADD,
36         Groups.FAVORITES,
37         Groups.GROUP_IS_READ_ONLY,
38         Groups.DELETED,
39     };
40
41     public final static int ACCOUNT_NAME = 0;
42     public final static int ACCOUNT_TYPE = 1;
43     public final static int DATA_SET = 2;
44     public final static int GROUP_ID = 3;
45     public final static int TITLE = 4;
46     public final static int AUTO_ADD = 5;
47     public final static int FAVORITES = 6;
48     public final static int IS_READ_ONLY = 7;
49     public final static int DELETED = 8;
50
51     public GroupMetadataLoader(Context context, Uri groupUri) {
52         super(context, ensureIsGroupUri(groupUri), COLUMNS, Groups.ACCOUNT_TYPE + " NOT NULL AND "
53             + Groups.ACCOUNT_NAME + " NOT NULL", null, null);
54     }
55
56     /**
57      * Ensures that this is a valid group URI. If invalid, then an exception is
58      * thrown. Otherwise, the original URI is returned.
59      */
60     private static Uri ensureIsGroupUri(final Uri groupUri) {
61         // TODO: Fix ContactsProvider2.getType method to resolve the group Uri
62         if (groupUri == null) {
63             throw new IllegalArgumentException("Uri must not be null");
64         }
65         if (!groupUri.toString().startsWith(Groups.CONTENT_URI.toString())) {
66             throw new IllegalArgumentException("Invalid group Uri: " + groupUri);
67         }
68         return groupUri;
69     }
70 }

```

<https://android.googlesource.com/platform/packages/apps/Contacts/+nougat-mr1-release/src/com/android/contacts/GroupMetadataLoader.java>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products

```

19 * Meta-data for a contact group. We load all groups associated with the contact's
20 * constituent accounts.
21 */
22 public final class GroupMetadata {
23     private String mAccountName;
24     private String mAccountType;
25     private String mDataSet;
26     private long mGroupId;
27     private String mTitle;
28     private boolean mDefaultGroup;
29     private boolean mFavorites;
30
31     public GroupMetadata(String accountName, String accountType, String dataSet, long groupId,
32         String title, boolean defaultGroup, boolean favorites) {
33         this.mAccountName = accountName;
34         this.mAccountType = accountType;
35         this.mDataSet = dataSet;
36         this.mGroupId = groupId;
37         this.mTitle = title;
38         this.mDefaultGroup = defaultGroup;
39         this.mFavorites = favorites;
40     }
41
42     public String getAccountName() {
43         return mAccountName;
44     }
45
46     public String getAccountType() {
47         return mAccountType;
48     }
49
50     public String getDataSet() {
51         return mDataSet;
52     }
53
54     public long getGroupId() {
55         return mGroupId;
56     }
57
58     public String getTitle() {
59         return mTitle;
60     }
61
62     public boolean isDefaultGroup() {
63         return mDefaultGroup;
64     }
65
66     public boolean isFavorites() {
67         return mFavorites;

```

<https://android.googlesource.com/platform/packages/apps/ContactsCommon/+/nougat-mr1-release/src/com/android/contacts/common/GroupMetadata.java>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products

```

44 * Class that sends chat message via SMS.
45 *
46 * The interface emulates a blocking sending similar to making an HTTP request.
47 * It calls the SmsManager to send a (potentially multipart) message and waits
48 * on the sent status on each part. The waiting has a timeout so it won't wait
49 * forever. Once the sent status of all parts received, the call returns.
50 * A successful sending requires success status for all parts. Otherwise, we
51 * pick the highest level of failure as the error for the whole message, which
52 * is used to determine if we need to retry the sending.
53 */
54 public class SmsSender {
55     private static final String TAG = LogUtil.BUGLE_TAG;
56
57     public static final String EXTRA_PART_ID = "part_id";
58
59     /*
60     * A map for pending sms messages. The key is the random request UUID.
61     */
62     private static ConcurrentHashMap<Uri, SendResult> sPendingMessageMap =
63         new ConcurrentHashMap<Uri, SendResult>();
64
65     private static final Random RANDOM = new Random();
66
67     // Whether we should send multipart SMS as separate messages
68     private static Boolean sSendMultipartSmsAsSeparateMessages = null;
69

```

<https://android.googlesource.com/platform/packages/apps/Messaging/+nougat-mr1-release/src/com/android/messaging/sms/SmsSender.java>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products

```

253 // Actually sending the message using SmsManager
254 private static void sendInternal(final Context context, final int subId, String dest,
255     final ArrayList<String> messages, final String serviceCenter,
256     final boolean requireDeliveryReport, final Uri messageUri) throws SmsException {
257     Assert.notNull(context);
258     final SmsManager smsManager = PhoneUtils.get(subId).getSmsManager();
259     final int messageCount = messages.size();
260     final ArrayList<PendingIntent> deliveryIntents = new ArrayList<PendingIntent>(messageCount);
261     final ArrayList<PendingIntent> sentIntents = new ArrayList<PendingIntent>(messageCount);
262     for (int i = 0; i < messageCount; i++) {
263         // Make pending intents different for each message part
264         final int partId = (messageCount <= 1 ? 0 : i + 1);
265         if (requireDeliveryReport && (i == (messageCount - 1))) {
266             // TODO we only care about the delivery status of the last part
267             // Shall we have better tracking of delivery status of all parts?
268             deliveryIntents.add((PendingIntent)getBroadcast(
269                 context,
270                 partId,
271                 getSendStatusIntent(context, SendStatusReceiver.MESSAGE_DELIVERED_ACTION,
272                     messageUri, partId, subId),
273                 0/*flag*/));
274         } else {
275             deliveryIntents.add(null);
276         }
277         sentIntents.add((PendingIntent)getBroadcast(
278             context,
279             partId,
280             getSendStatusIntent(context, SendStatusReceiver.MESSAGE_SENT_ACTION,
281                 messageUri, partId, subId),
282             0/*flag*/));
283     }
284     if (sSendMultiPartSmsAsSeparateMessages == null) {
285         sSendMultiPartSmsAsSeparateMessages = MmsConfig.get(subId)
286             .getSendMultiPartSmsAsSeparateMessages();
287     }

```

<https://android.googlesource.com/platform/packages/apps/Messaging/+/nougat-mr1-release/src/com/android/messaging/sms/SmsSender.java>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products

```

288     try {
289         if (sSendMultipartSmsAsSeparateMessages) {
290             // If multipart sms is not supported, send them as separate messages
291             for (int i = 0; i < messageCount; i++) {
292                 smsManager.sendTextMessage(dest,
293                     serviceCenter,
294                     messages.get(i),
295                     sentIntents.get(i),
296                     deliveryIntents.get(i));
297             }
298         } else {
299             smsManager.sendMultipartTextMessage(
300                 dest, serviceCenter, messages, sentIntents, deliveryIntents);
301         }
302     } catch (final Exception e) {
303         throw new SmsException("SmsSender: caught exception in sending " + e);
304     }
305 }

```

<https://android.googlesource.com/platform/packages/apps/Messaging/+/nougat-mr1-release/src/com/android/messaging/sms/SmsSender.java>

```

56 * Class that receives incoming SMS messages through android.provider.Telephony.SMS_RECEIVED
57 *
58 * This class serves two purposes:
59 * - Process phone verification SMS messages
60 * - Handle SMS messages when the user has enabled us to be the default SMS app (Pre-KLP)
61 */
62 public final class SmsReceiver extends BroadcastReceiver {
63     private static final String TAG = LogUtil.BUGLE_TAG;
64
65     private static ArrayList<Pattern> sIgnoreSmsPatterns;
66

```

<https://android.googlesource.com/platform/packages/apps/Messaging/+/nougat-mr1-release/src/com/android/messaging/receiver/SmsReceiver.java>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| US9467838 | Exemplary Supporting Evidence Regarding Accused Products |
|-----------|---|
| 200 | public static void deliverSmsMessages(final Context context, final int subId, |
| 201 | final int errorCode, final android.telephony.SmsMessage[] messages) { |
| 202 | final ContentValues messageValues = |
| 203 | MmsUtils.parseReceivedSmsMessage(context, messages, errorCode); |
| 204 | |
| 205 | LogUtil.v(TAG, "SmsReceiver.deliverSmsMessages"); |
| 206 | |
| 207 | final long nowInMillis = System.currentTimeMillis(); |
| 208 | final long receivedTimestamps = MmsUtils.getMessageDate(messages[0], nowInMillis); |
| 209 | |
| 210 | messageValues.put(Sms.Inbox.DATE, receivedTimestamps); |
| 211 | // Default to unread and unseen for us but ReceiveSmsMessageAction will override |
| 212 | // seen for the telephony db. |
| 213 | messageValues.put(Sms.Inbox.READ, 0); |
| 214 | messageValues.put(Sms.Inbox.SEEN, 0); |
| 215 | if (OsUtil.isAtLeastL_MRI()) { |
| 216 | messageValues.put(Sms.SUBSCRIPTION_ID, subId); |
| 217 | } |
| 218 | |
| 219 | if (messages[0].getMessageClass() == android.telephony.SmsMessage.MessageClass.CLASS_0 |
| 220 | DebugUtils.isDebugEnabled()) { |
| 221 | Factory.get().getUIIntents().launchClassZeroActivity(context, messageValues); |
| 222 | } else { |
| 223 | final ReceiveSmsMessageAction action = new ReceiveSmsMessageAction(messageValues); |
| 224 | action.start(); |
| 225 | } |
| 226 | } |
| | https://android.googlesource.com/platform/packages/apps/Messaging/+nougat-mr1-release/src/com/android/messaging/receiver/SmsReceiver.java |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| US9467838 | Exemplary Supporting Evidence Regarding Accused Products |
|-----------|---|
| | <pre> 228 @Override 229 public void onReceive(final Context context, final Intent intent) { 230 LogUtil.v(TAG, "SmsReceiver.onReceive " + intent); 231 // On KLP+ we only take delivery of SMS messages in SmsDeliverReceiver. 232 if (PhoneUtils.getDefault().isSmsEnabled()) { 233 final String action = intent.getAction(); 234 if (OsUtil.isSecondaryUser() && 235 (Telephony.Sms.Intents.SMS_RECEIVED_ACTION.equals(action) 236 // TODO: update this with the actual constant from Telephony 237 "android.provider.Telephony.MMS_DOWNLOADED".equals(action))) { 238 postNewMessageSecondaryUserNotification(); 239 } else if (!OsUtil.isAtLeastKLP()) { 240 deliverSmsIntent(context, intent); 241 } 242 } 243 } </pre> <p>https://android.googlesource.com/platform/packages/apps/Messaging/+nougat-mr1-release/src/com/android/messaging/receiver/SmsReceiver.java</p> |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| US9467838 | Exemplary Supporting Evidence Regarding Accused Products |
|-----------|---|
| 52 | * Class that sends chat message via MMS. |
| 53 | * The interface emulates a blocking send similar to making an HTTP request. |
| 54 | * The interface emulates a blocking send similar to making an HTTP request. |
| 55 | */ |
| 56 | public class MmsSender { |
| 57 | private static final String TAG = LogUtil.BUGLE_TAG; |
| 58 | /** |
| 59 | * Send an MMS message. |
| 60 | * @param context Context |
| 61 | * @param messageUri The unique URI of the message for identifying it during sending |
| 62 | * @param sendReq The SendReq PDU of the message |
| 63 | * @throws MmsFailureException |
| 64 | */ |
| 65 | public static void sendMms(final Context context, final int subId, final Uri messageUri, |
| 66 | final SendReq sendReq, final Bundle sentIntentExtras) throws MmsFailureException { |
| 67 | sendMms(context, |
| 68 | subId, |
| 69 | messageUri, |
| 70 | null /* locationUrl */, |
| 71 | sendReq, |
| 72 | true /* responseImportant */, |
| 73 | sentIntentExtras); |
| 74 | } |
| 75 | https://android.googlesource.com/platform/packages/apps/Messaging/+/nougat-mr1- |
| 76 | release/src/com/android/messaging/sms/MmsSender.java |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| US9467838 | Exemplary Supporting Evidence Regarding Accused Products |
|-----------|---|
| 240 | * Download an MMS message. |
| 241 | * |
| 242 | * @param context Context |
| 243 | * @param contentLocation The url of the MMS message |
| 244 | * @throws MmsFailureException |
| 245 | * @throws InvalidHeaderValueException |
| 246 | */ |
| 247 | public static void downloadMms(final Context context, final int subId, |
| 248 | final String contentLocation, Bundle extras) throws MmsFailureException, |
| 249 | InvalidHeaderValueException { |
| 250 | final Uri requestUri = Uri.parse(contentLocation); |
| 251 | final Uri contentUri = MmsFileProvider.buildRawMmsUri(); |
| 252 | |
| 253 | final Intent downloadedIntent = new Intent(SendStatusReceiver.MMS_DOWNLOADED_ACTION, |
| 254 | requestUri, |
| 255 | context, |
| 256 | SendStatusReceiver.class); |
| 257 | downloadedIntent.putExtra(SendMessageAction.EXTRA_CONTENT_URI, contentUri); |
| 258 | if (extras != null) { |
| 259 | downloadedIntent.putExtras(extras); |
| 260 | } |
| 261 | final PendingIntent downloadedPendingIntent = PendingIntent.getBroadcast(context, |
| 262 | 0 /*request code*/, |
| 263 | downloadedIntent, |
| 264 | PendingIntent.FLAG_UPDATE_CURRENT); |
| 265 | |
| 266 | |
| 267 | MmsManager.downloadMultimediaMessage(subId, context, contentLocation, contentUri, |
| 268 | downloadedPendingIntent); |
| 269 | } |
| | https://android.googlesource.com/platform/packages/apps/Messaging/+nougat-mr1-release/src/com/android/messaging/sms/MmsSender.java |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| US9467838 | Exemplary Supporting Evidence Regarding Accused Products |
|-----------|---|
| 97 | * Execute an MMS HTTP request, either a POST (sending) or a GET (downloading) |
| 98 | x |
| 99 | * @param urlString The request URL, for sending it is usually the MMSC, and for downloading |
| 100 | * it is the message URL |
| 101 | * @param pdu For POST (sending) only, the PDU to send |
| 102 | * @param method HTTP method, POST for sending and GET for downloading |
| 103 | * @param isProxySet Is there a proxy for the MMSC |
| 104 | * @param proxyHost The proxy host |
| 105 | * @param proxyPort The proxy port |
| 106 | * @param mmsConfig The MMS config to use |
| 107 | * @param userAgent The user agent header value |
| 108 | * @param uaProfUrl The UA Prof URL header value |
| 109 | * @return The HTTP response body |
| 110 | * @throws MmsHttpException For any failures |
| 111 | */ |
| 112 | public byte[] execute(String urlString, byte[] pdu, String method, boolean isProxySet, |
| 113 | String proxyHost, int proxyPort, Bundle mmsConfig, String userAgent, String uaProfUrl) |
| 114 | throws MmsHttpException { |
| 115 | Log.d(MmsService.TAG, "HTTP: " + method + " " + Utils.redactUrlForNonVerbose(urlString) |
| 116 | + (isProxySet ? (" proxy=" + proxyHost + " : " + proxyPort) : "") |
| 117 | + ", PDU size=" + (pdu != null ? pdu.length : 0)); |
| 118 | checkMethod(method); |
| 119 | URLConnection connection = null; |
| 120 | try { |
| 121 | Proxy proxy = Proxy.NO_PROXY; |
| 122 | if (isProxySet) { |
| 123 | proxy = new Proxy(Proxy.Type.HTTP, new InetSocketAddress(proxyHost, proxyPort)); |
| 124 | } |
| 125 | final URL url = new URL(urlString); |
| 126 | // Now get the connection |
| 127 | connection = (URLConnection) url.openConnection(proxy); |
| 128 | connection.setDoInput(true); |
| 129 | connection.setConnectTimeout(|
| 130 | mmsConfig.getInt(CarrierConfigValuesLoader.CONFIG_HTTP_SOCKET_TIMEOUT, |
| 131 | CarrierConfigValuesLoader.CONFIG_HTTP_SOCKET_TIMEOUT_DEFAULT)); |

<https://android.googlesource.com/platform/packages/apps/Messaging+/nougat-mr1-release/src/android/support/v7/mms/MmsHttpClient.java>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products

```

132 // ----- COMMON HEADERS -----
133 // Header: Accept
134 connection.setRequestProperty(HEADER_ACCEPT, HEADER_VALUE_ACCEPT);
135 // Header: Accept-Language
136 connection.setRequestProperty(
137     HEADER_ACCEPT_LANGUAGE, getLocale().getLanguage(Locale.getDefault()));
138 // Header: User-Agent
139 Log.i("MmsService.TAG", "HTTP: User-Agent=" + userAgent);
140 connection.setRequestProperty(HEADER_USER_AGENT, userAgent);
141 // Header: x-wap-profile
142 final String uaProfUrlTagName = mmsConfig.getString(
143     CarrierConfigValuesLoader.CONFIG_UA_PROF_TAG_NAME,
144     CarrierConfigValuesLoader.CONFIG_UA_PROF_TAG_NAME_DEFAULT);
145 if (uaProfUrl != null) {
146     Log.i("MmsService.TAG", "HTTP: UaProfUrl=" + uaProfUrl);
147     connection.setRequestProperty(uaProfUrlTagName, uaProfUrl);
148 }
149 // Add extra headers specified by mms_config.xml's httpparams
150 addExtraHeaders(connection, mmsConfig);
151 // Different stuff for GET and POST
152 if (METHOD_POST.equals(method)) {
153     if (pdu == null || pdu.length < 1) {
154         Log.e("MmsService.TAG", "HTTP: empty pdu");
155         throw new MmsHttpException(0/*statusCode*/, "Sending empty PDU");
156     }
157     connection.setDoOutput(true);
158     connection.setRequestMethod(METHOD_POST);
159     if (mmsConfig.getBoolean(
160         CarrierConfigValuesLoader.CONFIG_SUPPORT_HTTP_CHARSET_HEADER,
161         CarrierConfigValuesLoader.CONFIG_SUPPORT_HTTP_CHARSET_HEADER_DEFAULT)) {
162         connection.setRequestProperty(HEADER_CONTENT_TYPE,
163             HEADER_VALUE_CONTENT_TYPE_WITH_CHARSET);
164     } else {
165         connection.setRequestProperty(HEADER_CONTENT_TYPE,
166             HEADER_VALUE_CONTENT_TYPE_WITHOUT_CHARSET);

```

<https://android.googlesource.com/platform/packages/apps/Messaging/+nougat-mr1-release/src/android/support/v7/mms/MmsHttpClient.java>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products

```

167     }
168     if (Log.isLoggable(MmsService.TAG, Log.VERBOSE)) {
169         logHttpHeaders(connection.getRequestProperties());
170     }
171     connection.setFixedLengthStreamingMode(pdu.length);
172     // Sending request body
173     final OutputStream out =
174         new BufferedOutputStream(connection.getOutputStream());
175     out.write(pdu);
176     out.flush();
177     out.close();
178     } else if (METHOD_GET.equals(method)) {
179         if (Log.isLoggable(MmsService.TAG, Log.VERBOSE)) {
180             logHttpHeaders(connection.getRequestProperties());
181         }
182         connection.setRequestMethod(METHOD_GET);
183     }
184     // Get response
185     final int responseCode = connection.getResponseCode();
186     final String responseMessage = connection.getResponseMessage();
187     Log.d(MmsService.TAG, "HTTP: " + responseCode + " " + responseMessage);
188     if (Log.isLoggable(MmsService.TAG, Log.VERBOSE)) {
189         logHttpHeaders(connection.getHeaderFields());
190     }
191     if (responseCode / 100 != 2) {
192         throw new MmsHttpException(responseCode, responseMessage);
193     }
194     final InputStream in = new BufferedInputStream(connection.getInputStream());
195     final ByteArrayOutputStream byteOut = new ByteArrayOutputStream();
196     final byte[] buf = new byte[4096];
197     int count = 0;
198     while ((count = in.read(buf)) > 0) {
199         byteOut.write(buf, 0, count);
200     }
201     in.close();
202     final byte[] responseBody = byteOut.toByteArray();
203     Log.d(MmsService.TAG, "HTTP: response size="
204         + (responseBody != null ? responseBody.length : 0));
205     return responseBody;

```

<https://android.googlesource.com/platform/packages/apps/Messaging/+nougat-mr1-release/src/android/support/v7/mms/MmsHttpClient.java>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products

```

38 * Request to send an MMS
39 */
40 class SendRequest extends MmsRequest {
41     // Max send response PDU size in bytes (exceeding this may cause problem with
42     // system intent delivery).
43     private static final int MAX_SEND_RESPONSE_SIZE = 1000 * 1024;
44
45     private byte[] mPduData;
46
47     SendRequest(final String locationUrl, final Uri pduUri, final PendingIntent sentIntent) {
48         super(locationUrl, pduUri, sentIntent);
49     }
50
51     @Override
52     protected boolean loadRequest(final Context context, final Bundle mmsConfig) {
53         mPduData = readPduFromContentUri(
54             context,
55             pduUri,
56             mmsConfig.getInt(
57                 CarrierConfigValuesLoader.CONFIG_MAX_MESSAGE_SIZE,
58                 CarrierConfigValuesLoader.CONFIG_MAX_MESSAGE_SIZE_DEFAULT));
59         return (mPduData != null);
60     }
61
62     @Override
63     protected boolean transferResponse(final Context context, final Intent fillIn,
64         final byte[] response) {
65         // SendConf pdus are always small and can be included in the intent
66         if (response != null && fillIn != null) {
67             if (response.length > MAX_SEND_RESPONSE_SIZE) {
68                 // If the response PDU is too large, it won't be able to fit in
69                 // the PendingIntent to be transferred via system IPC.
70                 return false;
71             }
72             fillIn.putExtra(SmsManager.EXTRA_MMS_DATA, response);
73         }
74         return true;
75     }

```

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| | |
|------------------|--|
| <p>US9467838</p> | <p>Exemplary Supporting Evidence Regarding Accused Products</p> <p>https://android.googlesource.com/platform/packages/apps/Messaging/+nougat-mr1-release/src/android/support/v7/mms/SendRequest.java</p> <pre>public static LocationRequest create ()</pre> <p>Create a location request with default parameters.</p> <p>Default parameters are for a block accuracy, slowly updated location. It can then be adjusted as required by the applications before passing to the <code>FusedLocationProviderApi</code>.</p> <p>Returns</p> <ul style="list-style-type: none"> • a new location request <p>https://developers.google.com/android/reference/com/google/android/gms/location/LocationRequest</p> |
|------------------|--|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| | |
|---|---|
| <p>US9467838</p> | <p>Exemplary Supporting Evidence Regarding Accused Products</p> |
| <p>public static final int PRIORITY_BALANCED_POWER_ACCURACY</p> <p>Used with <code>setPriority(int)</code> to request "block" level accuracy.</p> <p>Block level accuracy is considered to be about 100 meter accuracy. Using a coarse accuracy such as this often consumes less power.</p> <p>Constant Value: 102</p> <p>public static final int PRIORITY_HIGH_ACCURACY</p> <p>Used with <code>setPriority(int)</code> to request the most accurate locations available.</p> <p>This will return the finest location available.</p> <p>Constant Value: 100</p> <p>public static final int PRIORITY_LOW_POWER</p> <p>Used with <code>setPriority(int)</code> to request "city" level accuracy.</p> <p>City level accuracy is considered to be about 10km accuracy. Using a coarse accuracy such as this often consumes less power.</p> <p>Constant Value: 104</p> <p>https://developers.google.com/android/reference/com/google/android/gms/location/LocationRequest</p> | |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| | |
|-----------|--|
| US9467838 | <p data-bbox="191 709 228 1564">Exemplary Supporting Evidence Regarding Accused Products</p> <pre data-bbox="248 1060 280 1539">public Task<Location> getLastLocation ()</pre> <p data-bbox="313 972 341 1551">Returns the best most recent location currently available.</p> <p data-bbox="370 380 431 1551">If a location is not available, which should happen very rarely, null will be returned. The best accuracy available while respecting the location permissions will be returned.</p> <p data-bbox="461 338 522 1551">This method provides a simplified way to get location. It is particularly well suited for applications that do not require an accurate location and that do not want to maintain extra logic for location updates.</p> <pre data-bbox="578 863 610 1539">public Task<LocationAvailability> getLocationAvailability ()</pre> <p data-bbox="643 386 704 1551">Returns the availability of location data. When <code>isLocationAvailable()</code> returns true, then the location returned by <code>getLastLocation()</code> will be reasonably up to date within the hints specified by the active <code>LocationRequest</code>s.</p> <p data-bbox="734 604 761 1551">If the client isn't connected to Google Play services and the request times out, null is returned.</p> <p data-bbox="790 403 852 1551">Note it's always possible for <code>getLastLocation()</code> to return null even when this method returns true (e.g. location settings were disabled between calls).</p> <p data-bbox="862 197 925 1564">https://developers.google.com/android/reference/com/google/android/gms/location/FusedLocationProviderClient</p> |
|-----------|--|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| | | | | | | | |
|-----------------------|--|----------------------|---------------------------------------|-----------------------|--|---------------------|--|
| <p>US9467838</p> | <p>Exemplary Supporting Evidence Regarding Accused Products</p> <p><code>public Task<Void> requestLocationUpdates (LocationRequest request, LocationCallback callback, Looper looper)</code></p> <p>Requests location updates with a callback on the specified Looper thread.</p> <p>This method is suited for the foreground use cases. For background use cases, the <code>PendingIntent</code> version of the method is recommended, see <code>requestLocationUpdates(LocationRequest, PendingIntent)</code>.</p> <p>Any previous <code>LocationRequests</code> registered on this <code>LocationListener</code> will be replaced.</p> <p>This call will keep the Google Play services connection active, so make sure to call <code>removeLocationUpdates(LocationCallback)</code> when you no longer need it, otherwise you lose the benefits of the automatic connection management.</p> <p>Callbacks for <code>LocationCallback</code> will be made on the specified thread, which must already be a prepared looper thread.</p> <p>Parameters</p> <table border="1"> <tr> <td><code>request</code></td> <td>The location request for the updates.</td> </tr> <tr> <td><code>callback</code></td> <td>The callback for the location updates.</td> </tr> <tr> <td><code>looper</code></td> <td>The Looper object whose message queue will be used to implement the callback mechanism, or null to make callbacks on the calling thread.</td> </tr> </table> <p>https://developers.google.com/android/reference/com/google/android/gms/location/FusedLocationProviderClient</p> | <code>request</code> | The location request for the updates. | <code>callback</code> | The callback for the location updates. | <code>looper</code> | The Looper object whose message queue will be used to implement the callback mechanism, or null to make callbacks on the calling thread. |
| <code>request</code> | The location request for the updates. | | | | | | |
| <code>callback</code> | The callback for the location updates. | | | | | | |
| <code>looper</code> | The Looper object whose message queue will be used to implement the callback mechanism, or null to make callbacks on the calling thread. | | | | | | |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| | | | | | |
|-----------------------------|--|----------------------|---------------------------------------|-----------------------------|---|
| <p>US9467838</p> | <p>Exemplary Supporting Evidence Regarding Accused Products</p> <pre>public Task<Void> requestLocationUpdates (LocationRequest request, PendingIntent callbackIntent)</pre> <p>Requests location updates with a callback on the specified PendingIntent.</p> <p>This method is suited for the background use cases, more specifically for receiving location updates, even when the app has been killed by the system. In order to do so, use a <code>PendingIntent</code> for a started service. For foreground use cases, the <code>LocationCallback</code> version of the method is recommended, see <code>requestLocationUpdates(LocationRequest, LocationCallback, Looper)</code>.</p> <p>Any previously registered requests that have the same <code>PendingIntent</code> (as defined by <code>equals(Object)</code>) will be replaced by this request.</p> <p>Both <code>LocationResult</code> and <code>LocationAvailability</code> are sent to the given <code>PendingIntent</code>. You can extract data from an <code>Intent</code> using <code>hasResult(Intent)</code>, <code>extractResult(Intent)</code>, <code>hasLocationAvailability(Intent)</code>, and <code>extractLocationAvailability(Intent)</code>.</p> <p>Parameters</p> <table border="1"> <tr> <td><code>request</code></td> <td>The location request for the updates.</td> </tr> <tr> <td><code>callbackIntent</code></td> <td>A pending intent to be sent for each location update.</td> </tr> </table> <p>Returns</p> <ul style="list-style-type: none"> a Task for the call, check <code>isSuccessful()</code> to determine if it was successful. <p>https://developers.google.com/android/reference/com/google/android/gms/location/FusedLocationProviderClient</p> | <code>request</code> | The location request for the updates. | <code>callbackIntent</code> | A pending intent to be sent for each location update. |
| <code>request</code> | The location request for the updates. | | | | |
| <code>callbackIntent</code> | A pending intent to be sent for each location update. | | | | |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| | |
|------------------|--|
| <p>US9467838</p> | <p>Exemplary Supporting Evidence Regarding Accused Products</p> <p>public void onLocationAvailability (LocationAvailability locationAvailability)</p> <p>Called when there is a change in the availability of location data.</p> <p>When <code>isLocationAvailable()</code> returns <code>false</code> you can assume that location will not be returned in <code>onLocationResult(LocationResult)</code> until something changes in the device's settings or environment. Even when <code>isLocationAvailable()</code> returns <code>true</code> the <code>onLocationResult(LocationResult)</code> may not always be called regularly, however the device location is known and both the most recently delivered location and <code>getLastLocation(GoogleApiClient)</code> will be reasonably up to date given the hints specified by the active <code>LocationRequest</code> s.</p> <p>Parameters</p> <p>locationAvailability The current status of location availability.</p> <p>public void onLocationResult (LocationResult result)</p> <p>Called when device location information is available.</p> <p>The most recent location returned by <code>getLastLocation()</code> is not guaranteed to be immediately fresh, but will be reasonably up to date given the hints specified by the active <code>LocationRequest</code> s.</p> <p>Parameters</p> <p>result The latest location result available.</p> <p>https://developers.google.com/android/reference/com/google/android/gms/location/LocationCallback</p> <p>public abstract void onLocationChanged (Location location)</p> <p>Called when the location has changed.</p> <p>Parameters</p> <p>location The updated location.</p> |
|------------------|--|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| | |
|------------------|---|
| <p>US9467838</p> | <p>Exemplary Supporting Evidence Regarding Accused Products https://developers.google.com/android/reference/com/google/android/gms/location/LocationListener</p> <p>Public Constructors</p> <p>public MapView (Context context)</p> <p>public MapView (Context context, AttributeSet attrs)</p> <p>public MapView (Context context, AttributeSet attrs, int defStyleAttr)</p> <p>public MapView (Context context, GoogleMapOptions options)</p> <p>https://developers.google.com/android/reference/com/google/android/gms/maps/MapView</p> |
|------------------|---|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| | |
|--|--|
| <p>US9467838</p> | <p>Exemplary Supporting Evidence Regarding Accused Products</p> <pre>public void getMapAsync (OnMapReadyCallback callback)</pre> <p>Returns a non-null instance of the <code>GoogleMap</code>, ready to be used.</p> <p>Note that:</p> <ul style="list-style-type: none"> • This method must be called from the main thread. • The callback will be executed in the main thread. • In the case where Google Play services is not installed on the user's device, the callback will not be triggered until the user installs it. • The <code>GoogleMap</code> object provided by the callback is non-null. <p>Parameters</p> <p>callback The callback object that will be triggered when the map is ready to be used.</p> <pre>public final void onCreate (Bundle savedInstanceState)</pre> <p>You must call this method from the parent Activity/Fragment's corresponding method. https://developers.google.com/android/reference/com/google/android/gms/maps/MapView</p> |
| <p>[1D] presenting, via an interactive display of the first device, a first interactive, georeferenced map and a first set of one or more user-selectable symbols corresponding to a first set of one or more second devices, wherein the first set of symbols are positioned on the first georeferenced map to spatial coordinates.</p> <p>Regarding Find My Device and Android Device Manager, the Accused Products display, to the user on the display of the first device, a map with one or more symbols corresponding to one or more second devices. The map is interactive because the user may control the display of the map, e.g., pan, zoom, and/or effect change to the map in an otherwise interactive manner. The map is georeferenced for at least the reason that one or more symbols are associated with spatial locations, i.e., coordinates. The symbols</p> | |

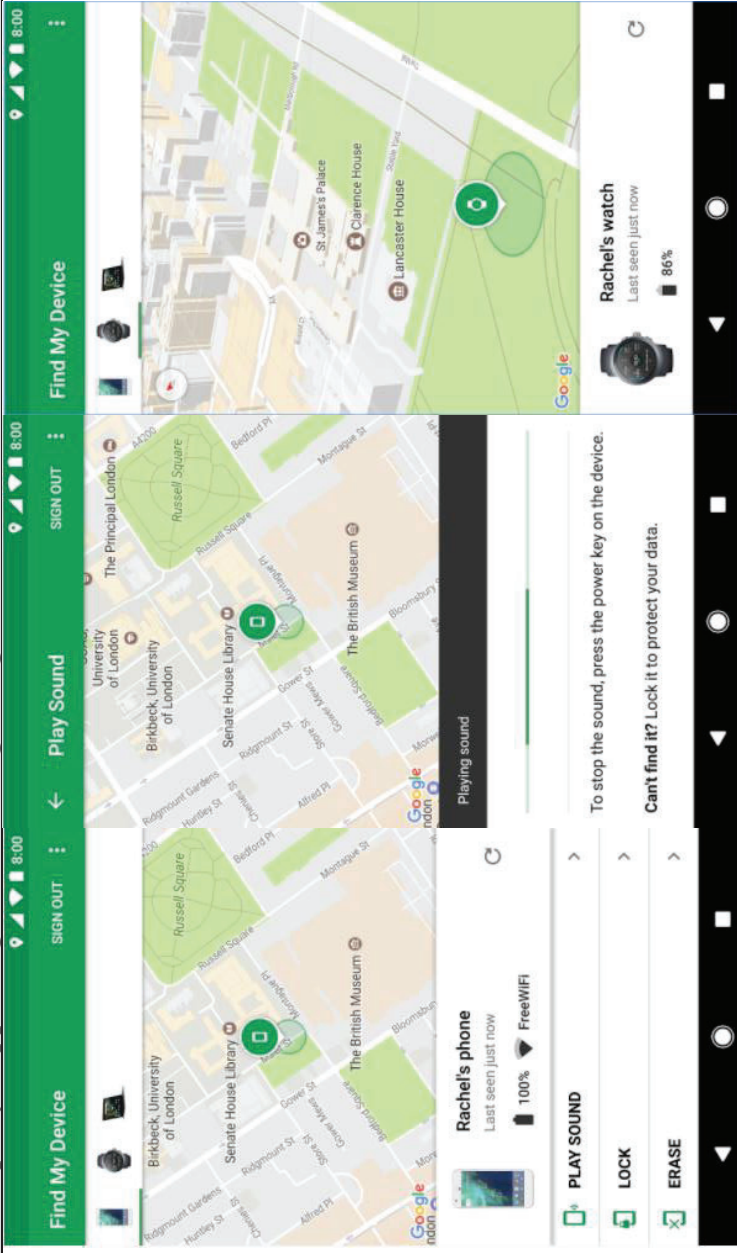
Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| | |
|--|--|
| <p>US9467838 first georeferenced map at respective positions corresponding to the locations of the first set of second devices, and wherein first georeferenced map data relate positions on the first georeferenced map to spatial coordinates;</p> | <p>Exemplary Supporting Evidence Regarding Accused Products are user-selectable because a user may touch the display to select the device associated with the symbol. For example, the user may input a touch selection directly on or near the portion of the display corresponding to the symbol's coordinates on the map to effect a selection of the device.</p> <p>Regarding Google Maps, Google Latitude, Google Plus, Google Hangouts, Google Messages, Google Allo, Google Duo, Google Chrome, and Android Messenger, the Accused Products display, to the user on the display of the first device, a map with one or more symbols corresponding to one or more second users (or second devices corresponding to the second users). The map is interactive because the user may control the display of the map, e.g., pan, zoom, and/or effect change to the map in an otherwise interactive manner. The map is georeferenced for at least the reason that one or more symbols are associated with spatial locations, i.e., coordinates. The symbols are user-selectable because a user may touch the display to select the user or device associated with the symbol. For example, the user may input a touch selection directly on or near the portion of the display corresponding to the symbol's coordinates on the map to effect a selection of the user or device.</p> <p>Exemplary Support for Find My Device:</p> |
|--|--|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products



<https://play.google.com/store/apps/details?id=com.google.android.apps.adm&hl=en>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

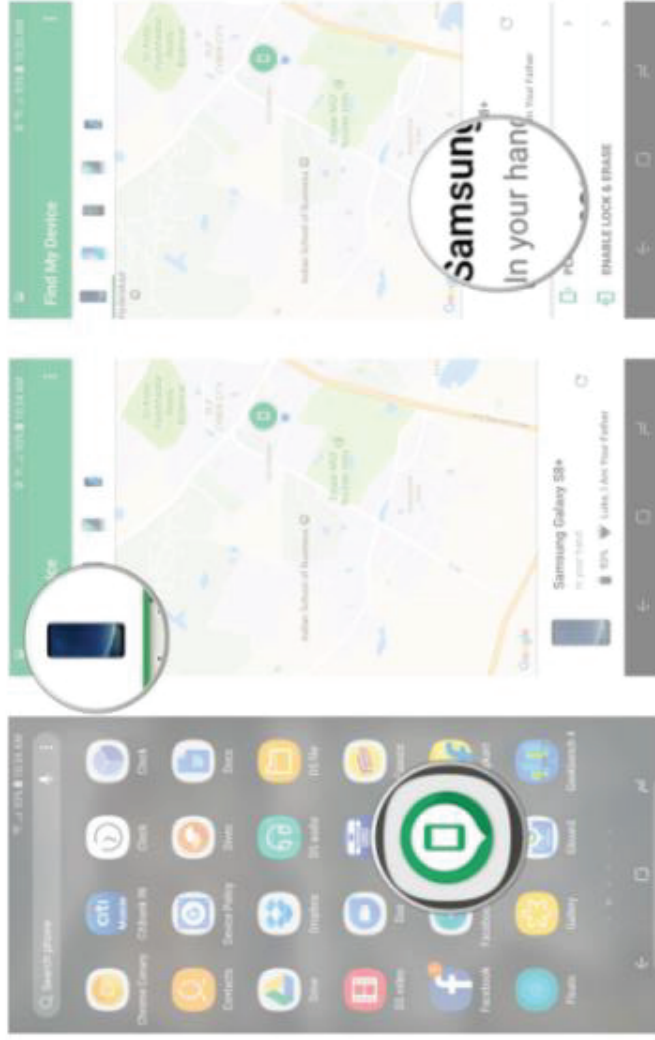
US9467838

Exemplary Supporting Evidence Regarding Accused Products

Once you're signed in to Find My Device, you'll see a map with your current location as well as the make and model of your phone, and two options — Play Sound, and Enable Lock & Erase. Hitting the latter option will allow you to start using the Lock and Erase functions.

If you've signed into more than one phone, you can select a particular device by browsing the list at the top of the screen.

1. Open **Find My Device** from your home screen or app drawer.
2. Select your phone from the **list of devices at the top of the screen**.
3. See if your phone is **discoverable**.



<https://www.androidcentral.com/find-my-device>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

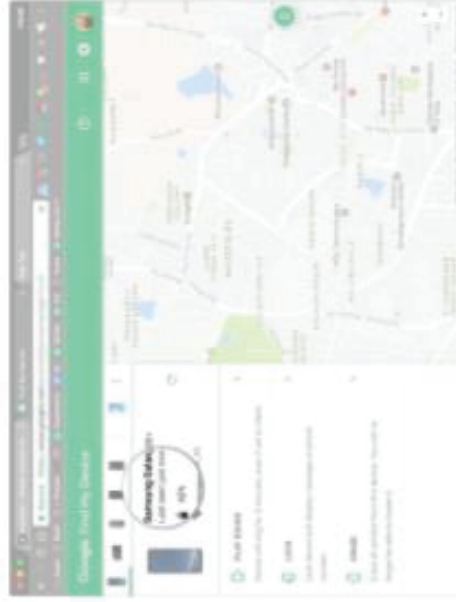
US9467838

Exemplary Supporting Evidence Regarding Accused Products

How to locate your phone over the internet

If you've lost your phone, you can remotely locate it through the Find My Device website. You'll need to sign in to the Google account that was used to set up Find My Device. It takes a few seconds, but the service should be able to track your phone. Alternatively, you can also do a Google search for "find my phone" to locate your handset.

1. Head to the [Find My Device website](#).
2. Sign in to your Google account.
3. Check if your device is visible.



<https://www.androidcentral.com/find-my-device>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| | |
|------------------|--|
| US9467838 | <u>Exemplary Supporting Evidence Regarding Accused Products</u> |
| | <u>Exemplary Support for Google Maps:</u> |




Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products

Share your E.T.A

After you start your drive, you can share your destination, estimated arrival time, and where you are on the route.

1. Open the Google Maps app .
 2. Set a driving destination. Learn how to navigate to a place.
 3. After you start navigation, tap More  > **Share trip progress**.
 4. Choose a person from the list.
 5. Tap **Share**.
 6. Location Sharing will stop when you reach your destination or stop navigating.
- To stop sharing before you arrive, tap More  > **Stop sharing**.

See where someone is

If someone shares their location with you, you can see them on the map.

1. Open the Google Maps app .
 2. Tap Menu  > **Location sharing**.
 3. Choose someone.
- To see an updated location, tap on a friend's icon > More  > **Refresh**.

Stop seeing someone's location

1. Open the Google Maps app .
2. On the map, tap their icon.
3. At the bottom, tap More .
4. To temporarily hide someone, tap **Hide from map**. You can stop hiding them at any time.

Note: You can stop someone's location from ever appearing on your map. Learn how to block another person's account.

<https://support.google.com/maps/answer/7326816?co=GENIE.Platform%3DAndroid&oco=1>

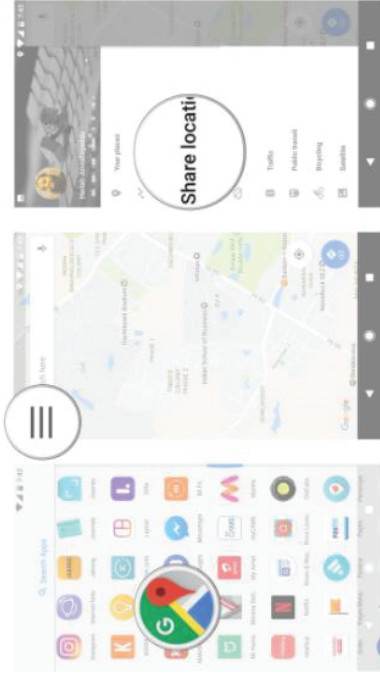
Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

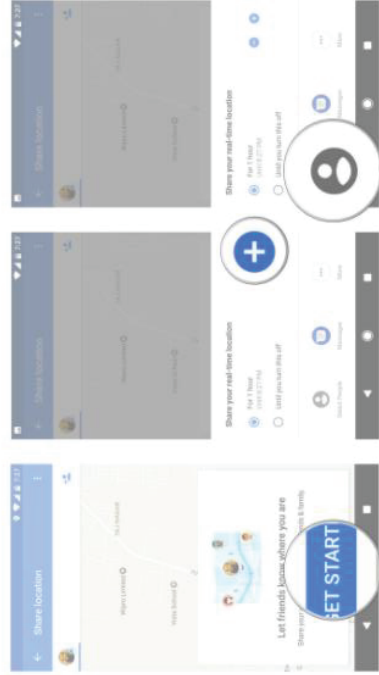
Exemplary Supporting Evidence Regarding Accused Products

How to share your location in Google Maps

1. Open Google Maps from the app drawer or the home screen.
2. Tap the hamburger menu (the three horizontal lines) on the top left corner of the screen.
3. Select Share location.



4. Tap Get Started.
5. Use the + icon to select a time period or select the **Until you turn this off** setting to share your location indefinitely.
6. Tap Select People.



<https://www.androidcentral.com/how-share-location-google-maps>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

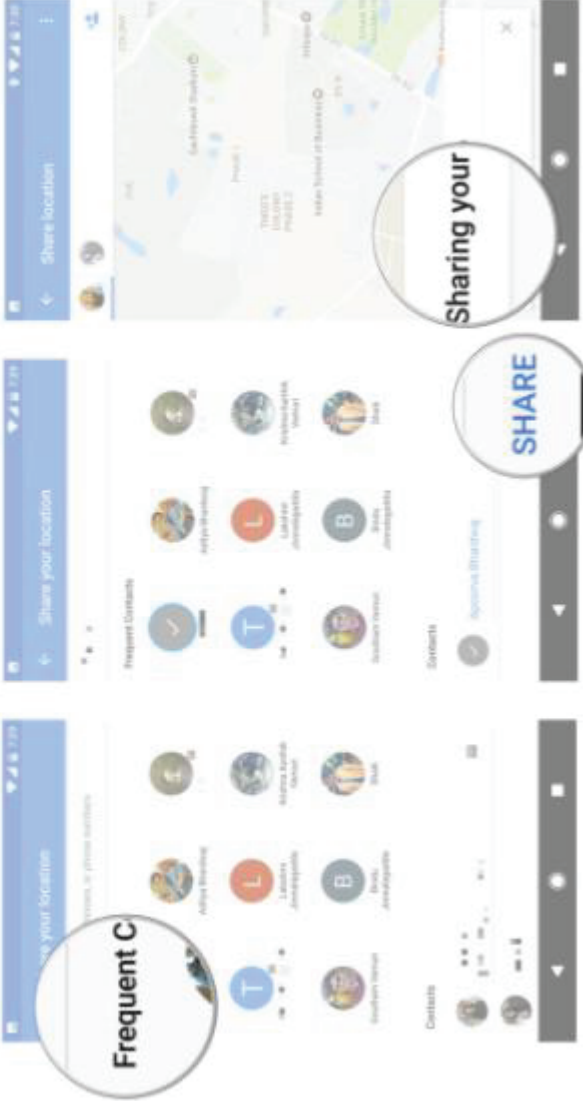
| | |
|------------------|---|
| <p>US9467838</p> | <p>Exemplary Supporting Evidence Regarding Accused Products</p> <p>7. You'll see a list of your frequent contacts at the top, along with a full list of contacts. Pick the contacts by tapping their name.</p> <p>8. Once you've selected the contacts you want to share your location to, tap Share.</p> <p>9. You'll see a message saying that the selected contact can view your location.</p>  <p>https://www.androidcentral.com/how-share-location-google-maps</p> |
|------------------|---|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

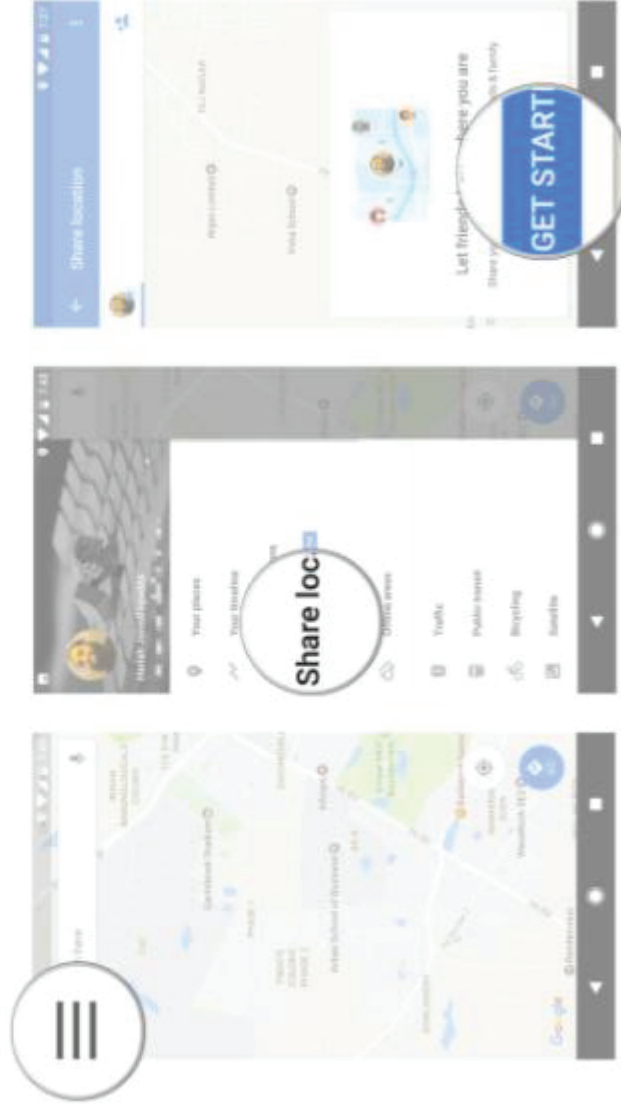
US9467838

Exemplary Supporting Evidence Regarding Accused Products

How to create a shareable link

You can also create a link and use it to share your location easily. Here's how to do it:

1. Tap the hamburger menu on the top left corner of the screen.
2. Select Share location.
3. Tap Get Started.



<https://www.androidcentral.com/how-share-location-google-maps>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

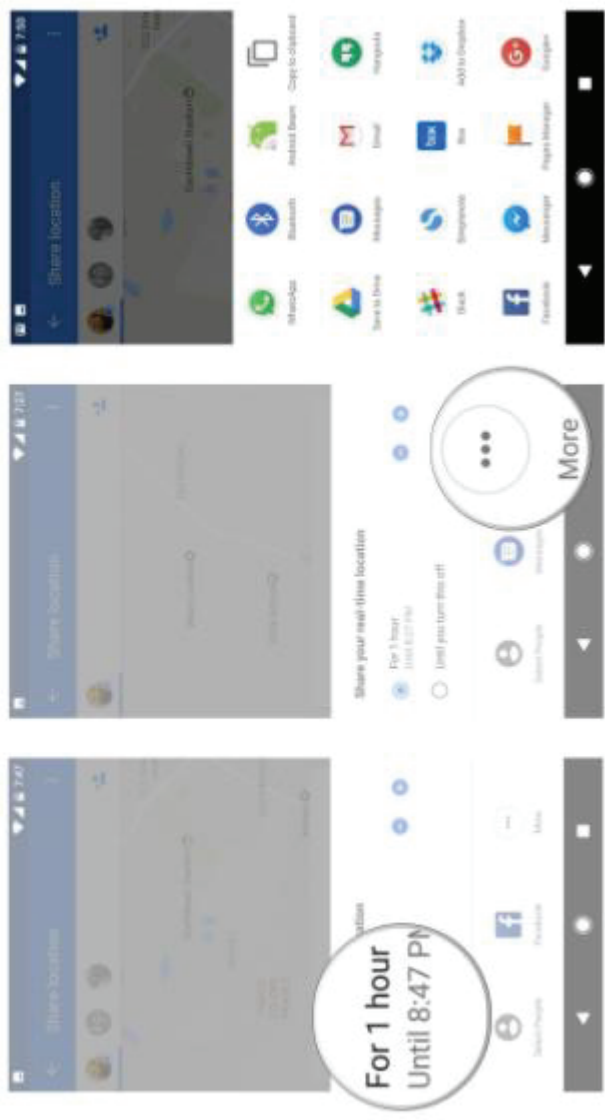
| | |
|------------------|---|
| <p>US9467838</p> | <p>Exemplary Supporting Evidence Regarding Accused Products</p> <ol style="list-style-type: none">4. Select the amount of time you want to share your location.5. Tap More.6. Select your app of choice to create and send a unique URL that broadcasts your current location. You can email it, send the link via Messenger, or even tweet it to the intended recipient.  <p>https://www.androidcentral.com/how-share-location-google-maps</p> |
|------------------|---|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

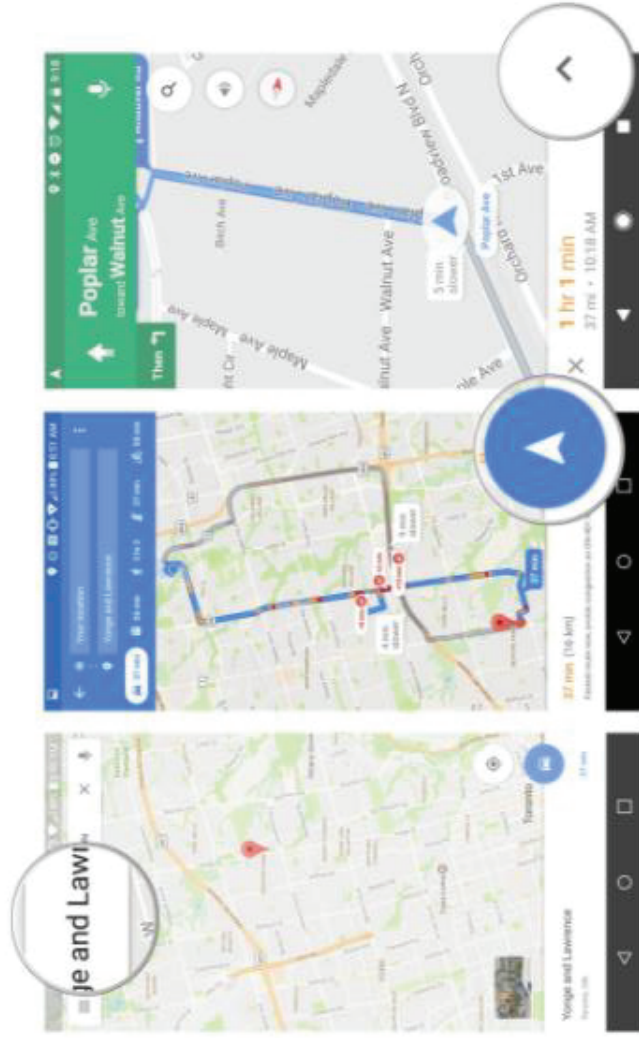
US9467838

Exemplary Supporting Evidence Regarding Accused Products

How to share your navigation directions while you walk, drive or transit

One of the best ways to share your location with someone is during a drive, walk or transit. If you're meeting a friend or family member somewhere, or navigating towards their home, Google Maps lets you share your location with them for the duration of the trip. It's magic!

1. In the **search bar** enter your destination.
2. Pick your navigation type (drive, transit, walk) and press the **blue navigate button**.
3. Tap the arrow next to the **time-to-destination** number at the bottom of the screen.



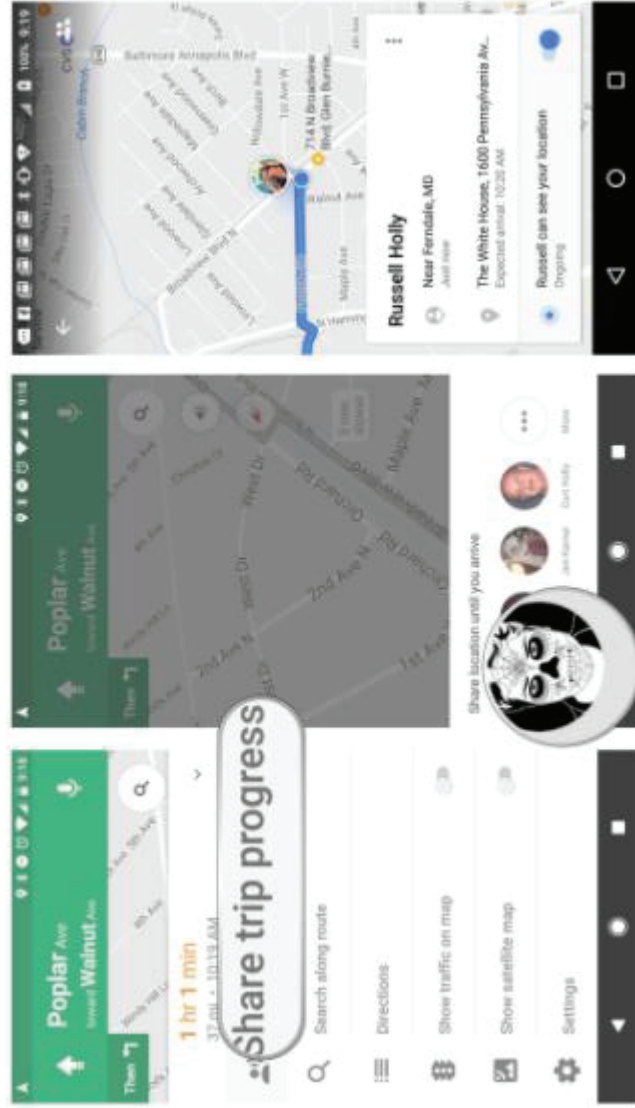
<https://www.androidcentral.com/how-share-location-google-maps>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products

- 4. Tap Share trip progress.
- 5. Choose one or more contacts to share trip progress.



You can also stop sharing your location with someone before a trip ends.
<https://www.androidcentral.com/how-share-location-google-maps>

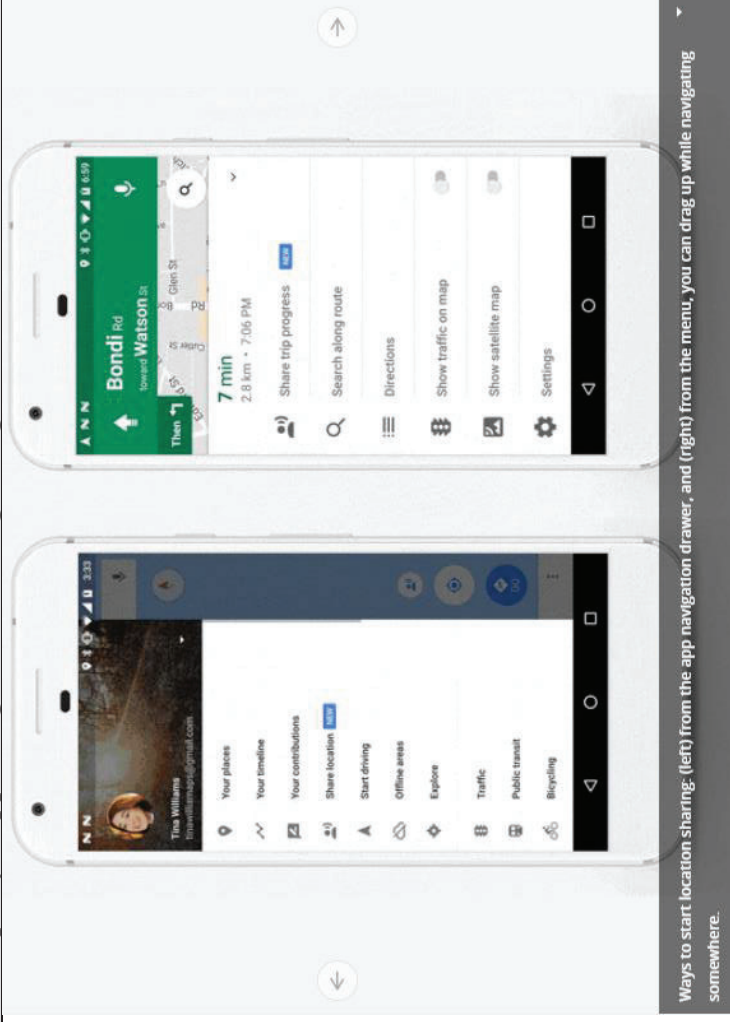
Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| | | |
|------------------|---|---|
| <p>US9467838</p> | <p>Exemplary Supporting Evidence Regarding Accused Products</p> | <p>1. Tap the arrow next to the time-to-destination number at the bottom of the screen.</p> <p>2. Tap Stop sharing.</p>  <p>That's it!</p> <p>Are you excited that location sharing is back in Google Maps? How often do you use the feature? https://www.androidcentral.com/how-share-location-google-maps</p> |
|------------------|---|---|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products



<https://arstechnica.com/gadgets/2017/03/location-sharing-finally-returns-to-google-maps/>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products

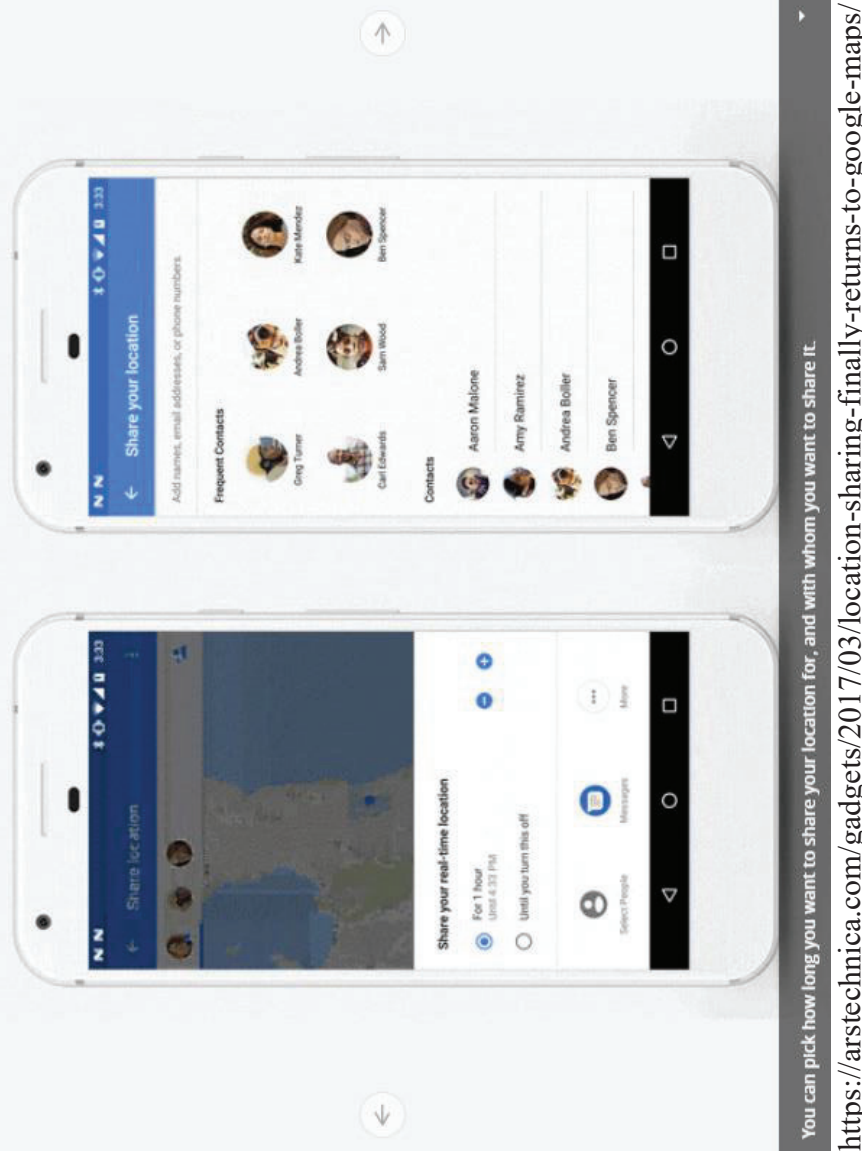
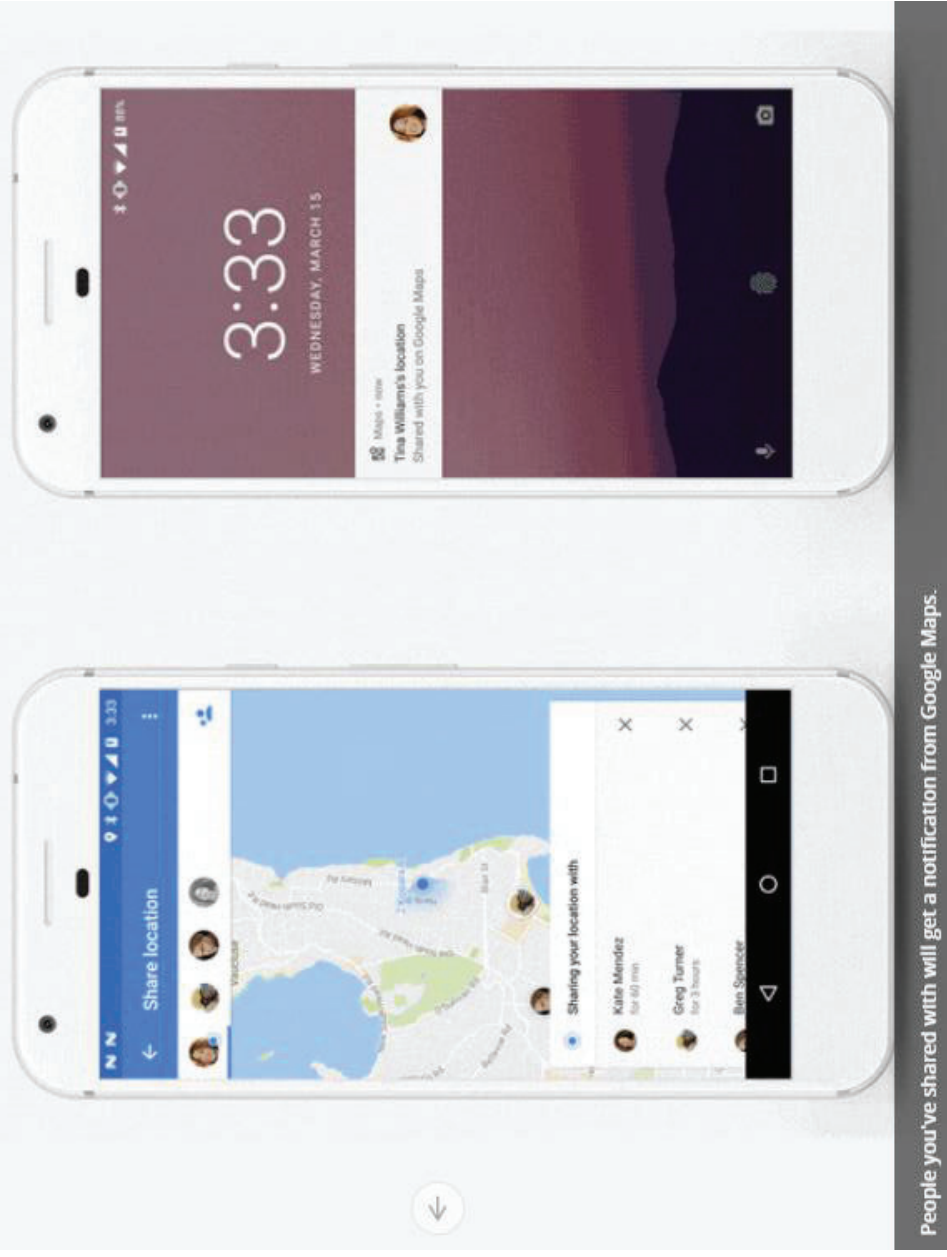


Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

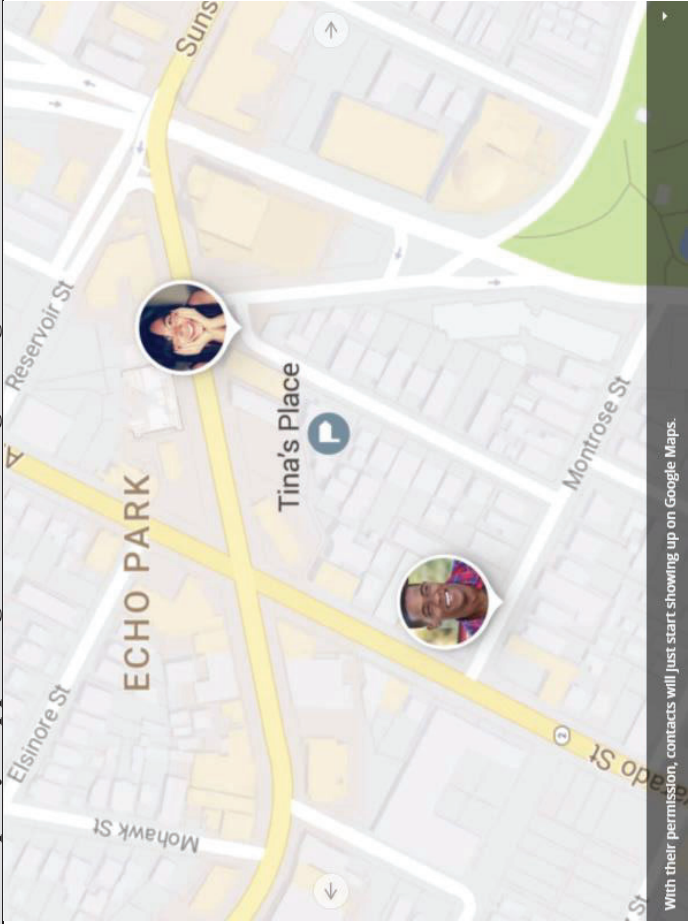
US9467838

Exemplary Supporting Evidence Regarding Accused Products



<https://arstechnica.com/gadgets/2017/03/location-sharing-finally-returns-to-google-maps/>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| | |
|-------------------------|--|
| <p>US9467838</p> | <p>Exemplary Supporting Evidence Regarding Accused Products</p>  <p>The screenshot shows a Google Maps view of Echo Park, Los Angeles. Two circular location-sharing pins are overlaid on the map. One pin is located on Reservoir St, and the other is on Montrose St. A parking icon labeled 'Tina's Place' is also visible. The map includes street names such as Mohawk St, Elsinore St, Reservoir St, Montrose St, and Echo St. A small text box at the bottom of the map reads: 'With their permission, contacts will just start showing up on Google Maps.'</p> <p>https://arstechnica.com/gadgets/2017/03/location-sharing-finally-returns-to-google-maps/</p> |
|-------------------------|--|

Exemplary Find My Device Screenshots:

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

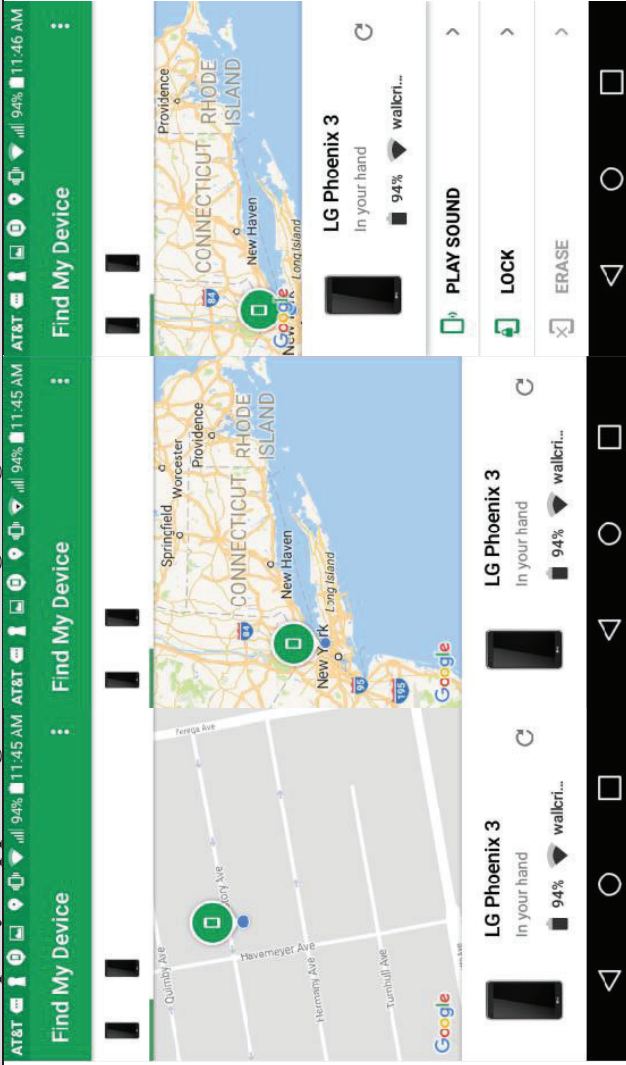
| | |
|-----------|--|
| US9467838 | <p data-bbox="191 709 228 1564">Exemplary Supporting Evidence Regarding Accused Products</p>  |
|-----------|--|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

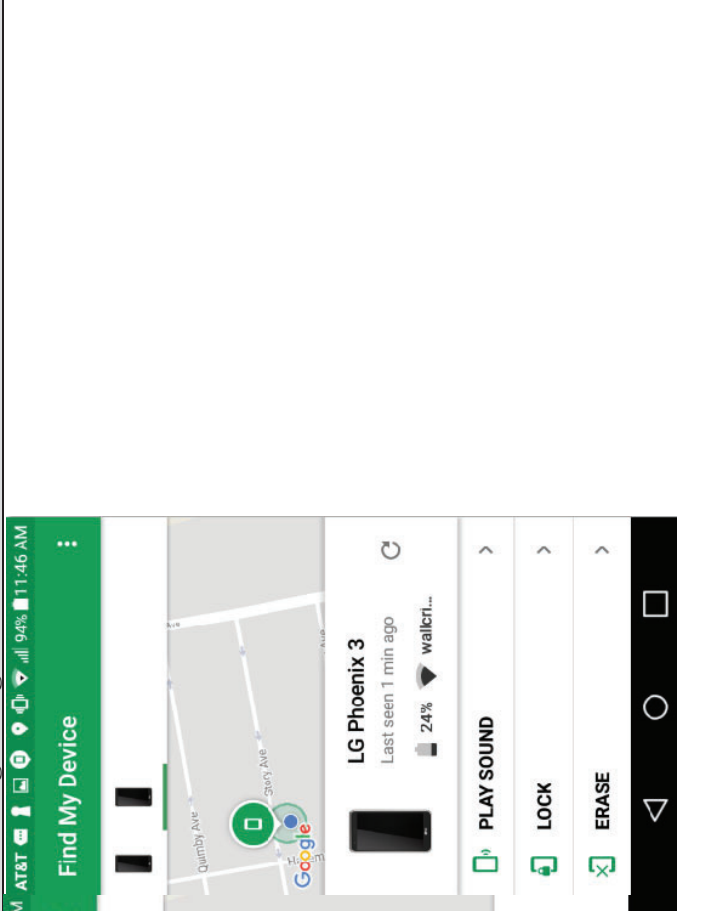
| | |
|------------------|--|
| <p>US9467838</p> | <p>Exemplary Supporting Evidence Regarding Accused Products</p>  |
|------------------|--|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

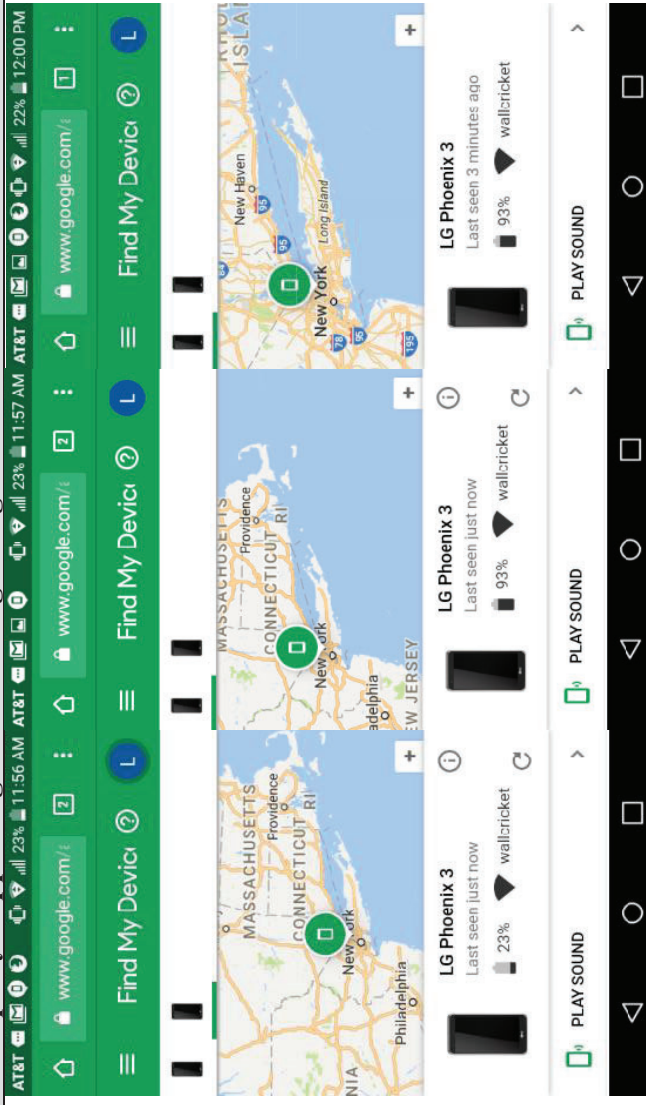
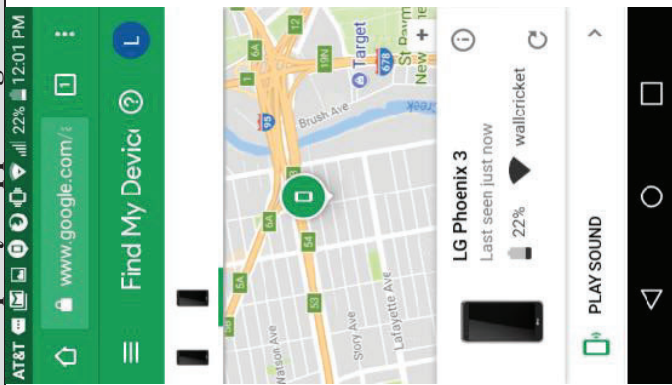
| | |
|-----------|---|
| US9467838 | <p>Exemplary Supporting Evidence Regarding Accused Products</p>  |
|-----------|---|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products



Exemplary Google Maps Screenshots:

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products

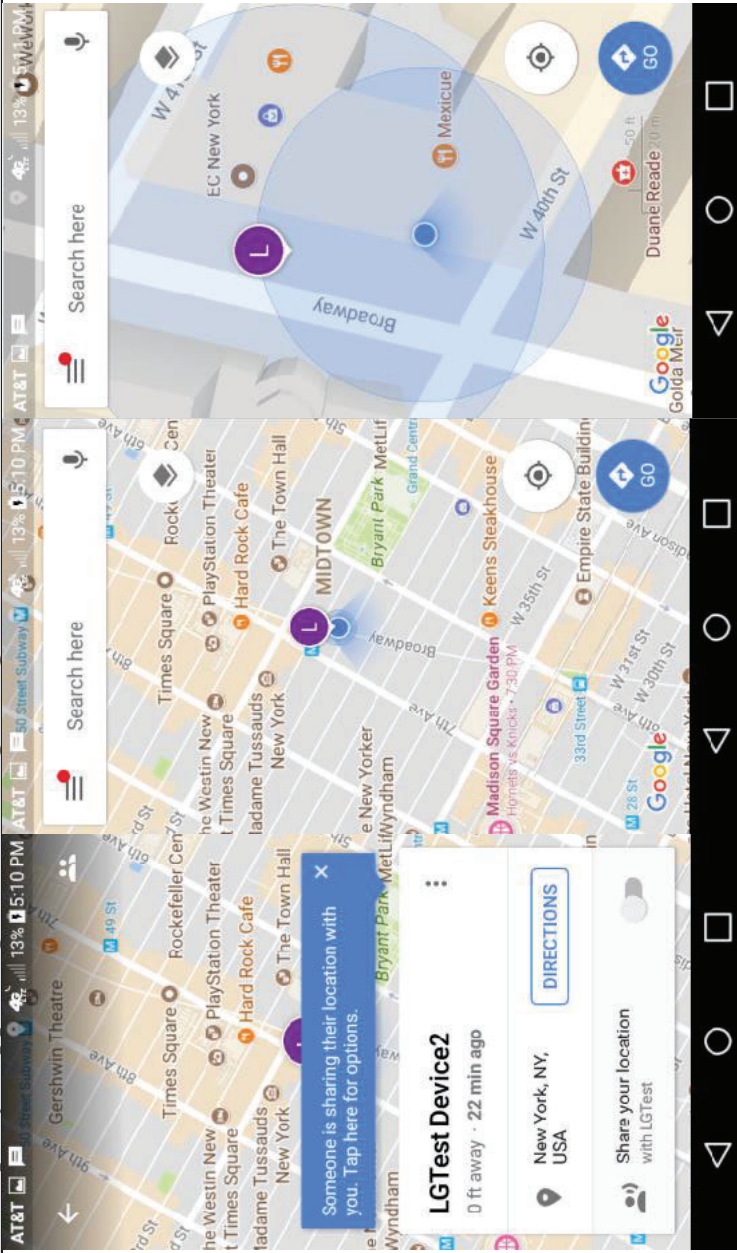


Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

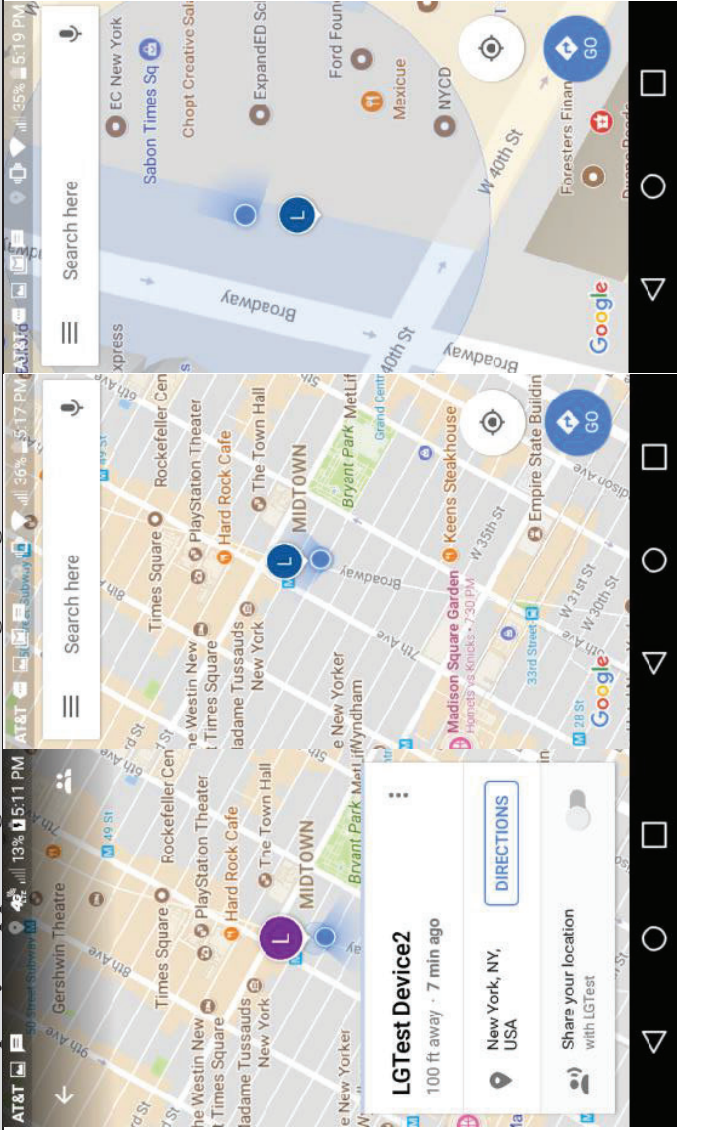
| | |
|------------------|--|
| <p>US9467838</p> | <p>Exemplary Supporting Evidence Regarding Accused Products</p>  |
|------------------|--|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

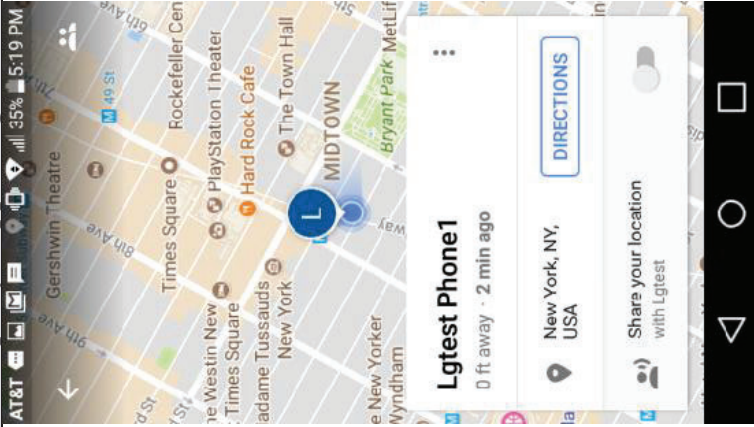
| | |
|------------------|--|
| <p>US9467838</p> | <p>Exemplary Supporting Evidence Regarding Accused Products</p>  <p>Exemplary Source Code:</p> <p>The above functionality is performed at least in part by the following publicly available source code and/or source code that invokes or is invoked by the following source code (or a substantially similar copy compiled and loaded onto the Accused Products by ZTE). AGIS reserves the right to supplement these contentions with additional source code as discovery progresses and as additional source code is made available.</p> |
|------------------|--|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| | |
|-----------|---|
| US9467838 | <p data-bbox="196 716 228 1556">Exemplary Supporting Evidence Regarding Accused Products</p> <p data-bbox="250 1100 282 1535">public static <code>LocationRequest create ()</code></p> <p data-bbox="315 1052 347 1535">Create a location request with default parameters.</p> <p data-bbox="371 443 436 1535">Default parameters are for a block accuracy, slowly updated location. It can then be adjusted as required by the applications before passing to the <code>FusedLocationProviderApi</code>.</p> <p data-bbox="456 1446 480 1535">Returns</p> <ul data-bbox="501 1268 526 1514" style="list-style-type: none"> • a new location request <p data-bbox="542 285 574 1556">https://developers.google.com/android/reference/com/google/android/gms/location/LocationRequest</p> <p data-bbox="594 806 626 1535">public static final int PRIORITY_BALANCED_POWER_ACCURACY</p> <p data-bbox="659 905 683 1535">Used with <code>setPriority(int)</code> to request "block" level accuracy.</p> <p data-bbox="716 443 781 1535">Block level accuracy is considered to be about 100 meter accuracy. Using a coarse accuracy such as this often consumes less power.</p> <p data-bbox="797 1352 821 1535">Constant Value: 102</p> <p data-bbox="883 974 915 1535">public static final int PRIORITY_HIGH_ACCURACY</p> <p data-bbox="948 743 972 1535">Used with <code>setPriority(int)</code> to request the most accurate locations available.</p> <p data-bbox="1005 1115 1029 1535">This will return the finest location available.</p> <p data-bbox="1062 1352 1086 1535">Constant Value: 100</p> <p data-bbox="1143 1026 1175 1535">public static final int PRIORITY_LOW_POWER</p> <p data-bbox="1208 926 1232 1535">Used with <code>setPriority(int)</code> to request "city" level accuracy.</p> <p data-bbox="1265 344 1330 1535">City level accuracy is considered to be about 10km accuracy. Using a coarse accuracy such as this often consumes less power.</p> <p data-bbox="1346 1352 1370 1535">Constant Value: 104</p> <p data-bbox="1386 285 1419 1556">https://developers.google.com/android/reference/com/google/android/gms/location/LocationRequest</p> |
|-----------|---|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| | |
|-----------|---|
| US9467838 | <p data-bbox="185 296 228 1921">Exemplary Supporting Evidence Regarding Accused Products</p> <pre data-bbox="276 325 324 1543">public Task<Location> getLastLocation ()</pre> <p data-bbox="341 325 381 1543">Returns the best most recent location currently available.</p> <p data-bbox="397 325 470 1543">If a location is not available, which should happen very rarely, null will be returned. The best accuracy available while respecting the location permissions will be returned.</p> <p data-bbox="487 325 560 1543">This method provides a simplified way to get location. It is particularly well suited for applications that do not require an accurate location and that do not want to maintain extra logic for location updates.</p> <pre data-bbox="600 325 649 1543">public Task<LocationAvailability> getLocationAvailability ()</pre> <p data-bbox="665 325 738 1543">Returns the availability of location data. When <code>isLocationAvailable()</code> returns true, then the location returned by <code>getLastLocation()</code> will be reasonably up to date within the hints specified by the active <code>LocationRequest</code>s.</p> <p data-bbox="755 325 795 1543">If the client isn't connected to Google Play services and the request times out, null is returned.</p> <p data-bbox="812 325 885 1543">Note it's always possible for <code>getLastLocation()</code> to return null even when this method returns true (e.g. location settings were disabled between calls).</p> <p data-bbox="893 325 963 1543">https://developers.google.com/android/reference/com/google/android/gms/location/FusedLocationProviderClient</p> |
|-----------|---|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| | | | | | | | |
|-----------------------|---|----------------------|---------------------------------------|-----------------------|--|---------------------|--|
| <p>US9467838</p> | <p>Exemplary Supporting Evidence Regarding Accused Products</p> <pre>public Task<Void> requestLocationUpdates (LocationRequest request, LocationCallback callback, Looper looper)</pre> <p>Requests location updates with a callback on the specified Looper thread.</p> <p>This method is suited for the foreground use cases. For background use cases, the <code>PendingIntent</code> version of the method is recommended, see <code>requestLocationUpdates(LocationRequest, PendingIntent)</code>.</p> <p>Any previous <code>LocationRequests</code> registered on this <code>LocationListener</code> will be replaced.</p> <p>This call will keep the Google Play services connection active, so make sure to call <code>removeLocationUpdates(LocationCallback)</code> when you no longer need it, otherwise you lose the benefits of the automatic connection management.</p> <p>Callbacks for <code>LocationCallback</code> will be made on the specified thread, which must already be a prepared looper thread.</p> <p>Parameters</p> <table border="1"> <tr> <td><code>request</code></td> <td>The location request for the updates.</td> </tr> <tr> <td><code>callback</code></td> <td>The callback for the location updates.</td> </tr> <tr> <td><code>looper</code></td> <td>The Looper object whose message queue will be used to implement the callback mechanism, or null to make callbacks on the calling thread.</td> </tr> </table> <p>https://developers.google.com/android/reference/com/google/android/gms/location/FusedLocationProviderClient</p> | <code>request</code> | The location request for the updates. | <code>callback</code> | The callback for the location updates. | <code>looper</code> | The Looper object whose message queue will be used to implement the callback mechanism, or null to make callbacks on the calling thread. |
| <code>request</code> | The location request for the updates. | | | | | | |
| <code>callback</code> | The callback for the location updates. | | | | | | |
| <code>looper</code> | The Looper object whose message queue will be used to implement the callback mechanism, or null to make callbacks on the calling thread. | | | | | | |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| | | | | | |
|-----------------------------|--|----------------------|---------------------------------------|-----------------------------|---|
| <p>US9467838</p> | <p>Exemplary Supporting Evidence Regarding Accused Products</p> <pre>public Task<Void> requestLocationUpdates (LocationRequest request, PendingIntent callbackIntent)</pre> <p>Requests location updates with a callback on the specified PendingIntent.</p> <p>This method is suited for the background use cases, more specifically for receiving location updates, even when the app has been killed by the system. In order to do so, use a <code>PendingIntent</code> for a started service. For foreground use cases, the <code>LocationCallback</code> version of the method is recommended, see <code>requestLocationUpdates(LocationRequest, LocationCallback, Looper)</code>.</p> <p>Any previously registered requests that have the same <code>PendingIntent</code> (as defined by <code>equals(Object)</code>) will be replaced by this request.</p> <p>Both <code>LocationResult</code> and <code>LocationAvailability</code> are sent to the given <code>PendingIntent</code>. You can extract data from an <code>Intent</code> using <code>hasResult(Intent)</code>, <code>extractResult(Intent)</code>, <code>hasLocationAvailability(Intent)</code>, and <code>extractLocationAvailability(Intent)</code>.</p> <p>Parameters</p> <table border="1"> <tr> <td><code>request</code></td> <td>The location request for the updates.</td> </tr> <tr> <td><code>callbackIntent</code></td> <td>A pending intent to be sent for each location update.</td> </tr> </table> <p>Returns</p> <ul style="list-style-type: none"> a Task for the call, check <code>isSuccessful()</code> to determine if it was successful. <p>https://developers.google.com/android/reference/com/google/android/gms/location/FusedLocationProviderClient</p> | <code>request</code> | The location request for the updates. | <code>callbackIntent</code> | A pending intent to be sent for each location update. |
| <code>request</code> | The location request for the updates. | | | | |
| <code>callbackIntent</code> | A pending intent to be sent for each location update. | | | | |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| | |
|-----------|--|
| US9467838 | <p data-bbox="191 247 228 1564">Exemplary Supporting Evidence Regarding Accused Products</p> <p data-bbox="240 247 277 1564">public void onLocationAvailability (LocationAvailability locationAvailability)</p> <p data-bbox="305 247 337 1564">Called when there is a change in the availability of location data.</p> <p data-bbox="365 247 560 1564">When <code>isLocationAvailable()</code> returns <code>false</code> you can assume that location will not be returned in <code>onLocationResult(LocationResult)</code> until something changes in the device's settings or environment. Even when <code>isLocationAvailable()</code> returns <code>true</code> the <code>onLocationResult(LocationResult)</code> may not always be called regularly, however the device location is known and both the most recently delivered location and <code>getLastLocation(GoogleApiClient)</code> will be reasonably up to date given the hints specified by the active <code>LocationRequest</code> s.</p> <p data-bbox="581 247 613 1564">Parameters</p> <p data-bbox="641 247 706 1564">locationAvailability The current status of location availability.</p> <p data-bbox="755 247 787 1564">public void onLocationResult (LocationResult result)</p> <p data-bbox="820 247 852 1564">Called when device location information is available.</p> <p data-bbox="876 247 941 1564">The most recent location returned by <code>getLastLocation()</code> is not guaranteed to be immediately fresh, but will be reasonably up to date given the hints specified by the active <code>LocationRequest</code> s.</p> <p data-bbox="966 247 998 1564">Parameters</p> <p data-bbox="1023 247 1088 1564">result The latest location result available.</p> <p data-bbox="1096 247 1128 1564">https://developers.google.com/android/reference/com/google/android/gms/location/LocationCallback</p> <p data-bbox="1144 247 1177 1564">public abstract void onLocationChanged (Location location)</p> <p data-bbox="1209 247 1242 1564">Called when the location has changed.</p> <p data-bbox="1258 247 1291 1564">Parameters</p> <p data-bbox="1315 247 1380 1564">location The updated location.</p> |
|-----------|--|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| | |
|------------------|---|
| <p>US9467838</p> | <p>Exemplary Supporting Evidence Regarding Accused Products https://developers.google.com/android/reference/com/google/android/gms/location/LocationListener</p> <p>Public Constructors</p> <p><code>public MapView (Context context)</code></p> <p><code>public MapView (Context context, AttributeSet attrs)</code></p> <p><code>public MapView (Context context, AttributeSet attrs, int defStyleAttr)</code></p> <p><code>public MapView (Context context, GoogleMapOptions options)</code></p> <p>https://developers.google.com/android/reference/com/google/android/gms/maps/MapView</p> |
|------------------|---|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| | |
|---|--|
| <p>US9467838</p> | <p>Exemplary Supporting Evidence Regarding Accused Products</p> <pre>public void getMapAsync (OnMapReadyCallback callback)</pre> <p>Returns a non-null instance of the <code>GoogleMap</code>, ready to be used.</p> <p>Note that:</p> <ul style="list-style-type: none"> • This method must be called from the main thread. • The callback will be executed in the main thread. • In the case where Google Play services is not installed on the user's device, the callback will not be triggered until the user installs it. • The <code>GoogleMap</code> object provided by the callback is non-null. <p>Parameters</p> <p>callback The callback object that will be triggered when the map is ready to be used.</p> <pre>public final void onCreate (Bundle savedInstanceState)</pre> <p>You must call this method from the parent Activity/Fragment's corresponding method. https://developers.google.com/android/reference/com/google/android/gms/maps/MapView</p> |
| <p>[1E] sending, to a second server, a request for second georeferenced map data different from the first georeferenced map data;</p> | <p>ZTE infringes directly and/or indirectly by performing, inducing others to perform, and/or contributing to the performance of: sending, to a second server, a request for second georeferenced map data different from the first georeferenced map data.</p> <p>Regarding Find My Device and Android Device Manager, the Accused Products display, to the user on the display of the first device, a map with one or more symbols corresponding to one or more second devices. The user, via the first device, or the device itself requests different map data from a second server. The request occurs responsive to user input (e.g., zoom, drag, pan, change focus, change map type, symbol or device selection, another device or user selection, refresh or reload request, change in position of first device, change in position of a second device). Alternatively, the request occurs responsive to an</p> |

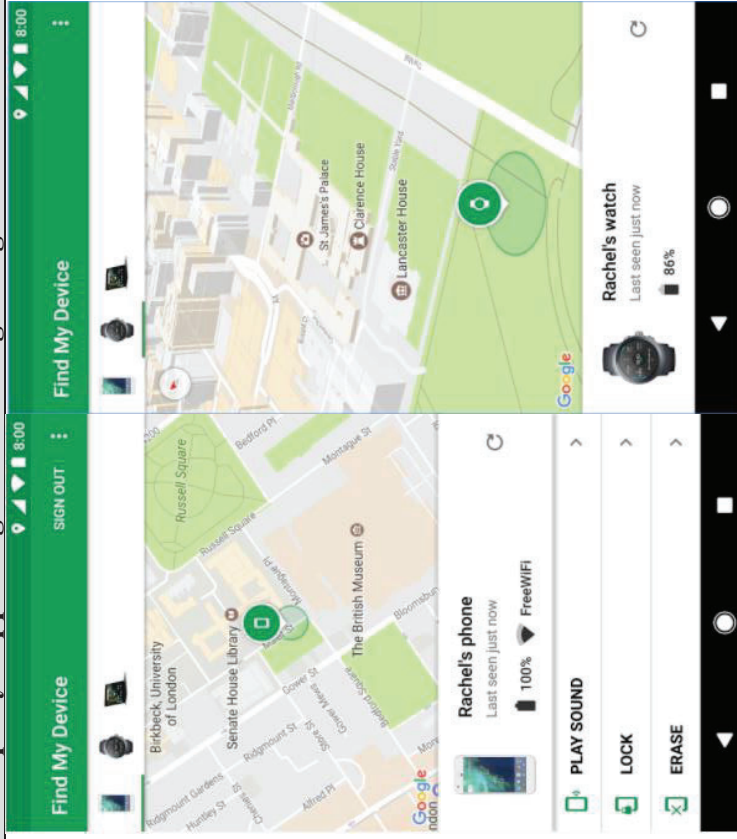
Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| | |
|-----------|--|
| US9467838 | <p data-bbox="233 205 370 1562">Exemplary Supporting Evidence Regarding Accused Products automatic and/or pre-determined control caused by an instruction from within the first device or from the one or more second devices, e.g. a refresh. The different map data includes an update to the first data or a replacement of the first data. Upon information and belief, the new map data may come from one of many sources and servers.</p> <p data-bbox="415 205 812 1562">Regarding Google Maps, Google Latitude, Google Plus, Google Hangouts, Google Messages, Google Allo, Google Duo, Google Chrome, and Android Messenger, the Accused Products display, to the user on the display of the first device, a map with one or more symbols corresponding to one or more second users (or second devices corresponding to the second users). The user, via the first device, or the device itself requests different map data from a second server. The request occurs responsive to user input (e.g., zoom, drag, pan, change focus, change map type, refresh or reload request, symbol or device selection, another device or user selection, change in position of first device, change in position of a second device). Alternatively, the request occurs responsive to an automatic and/or pre-determined control caused by an instruction from within the first device or from the one or more second devices, e.g. a refresh. The different map data includes an update to the first data or a replacement of the first data. Upon information and belief, the new map data may come from one of many sources and servers.</p> <p data-bbox="857 1012 886 1562"><u>Exemplary Support for Find My Device:</u></p> |
|-----------|--|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products



<https://play.google.com/store/apps/details?id=com.google.android.apps.adm&hl=en>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

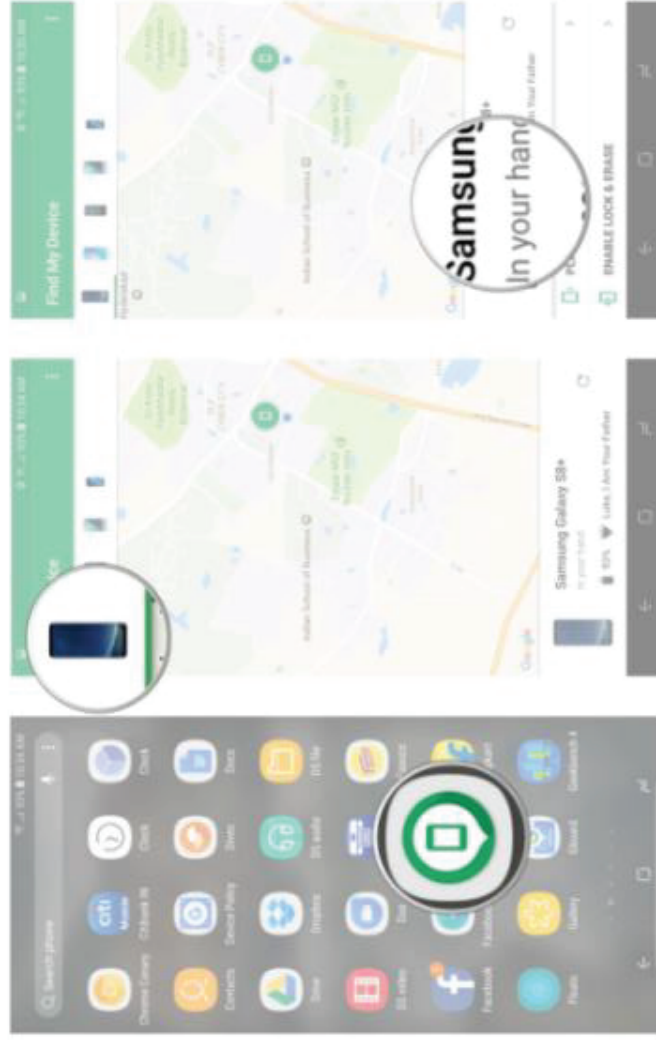
US9467838

Exemplary Supporting Evidence Regarding Accused Products

Once you're signed in to Find My Device, you'll see a map with your current location as well as the make and model of your phone, and two options — Play Sound, and Enable Lock & Erase. Hitting the latter option will allow you to start using the Lock and Erase functions.

If you've signed into more than one phone, you can select a particular device by browsing the list at the top of the screen.

1. Open **Find My Device** from your home screen or app drawer.
2. Select your phone from the **list of devices at the top of the screen**.
3. See if your phone is **discoverable**.



<https://www.androidcentral.com/find-my-device>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

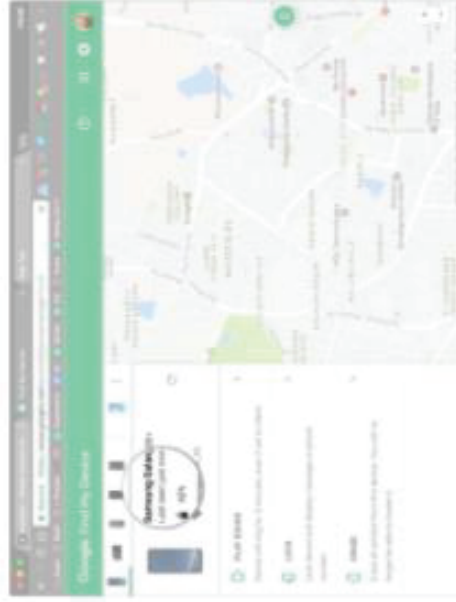
US9467838

Exemplary Supporting Evidence Regarding Accused Products

How to locate your phone over the internet

If you've lost your phone, you can remotely locate it through the Find My Device website. You'll need to sign in to the Google account that was used to set up Find My Device. It takes a few seconds, but the service should be able to track your phone. Alternatively, you can also do a Google search for "find my phone" to locate your handset.

1. Head to the [Find My Device website](#).
2. Sign in to your Google account.
3. Check if your device is visible.



<https://www.androidcentral.com/find-my-device>

Exemplary Support for Google Maps:

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

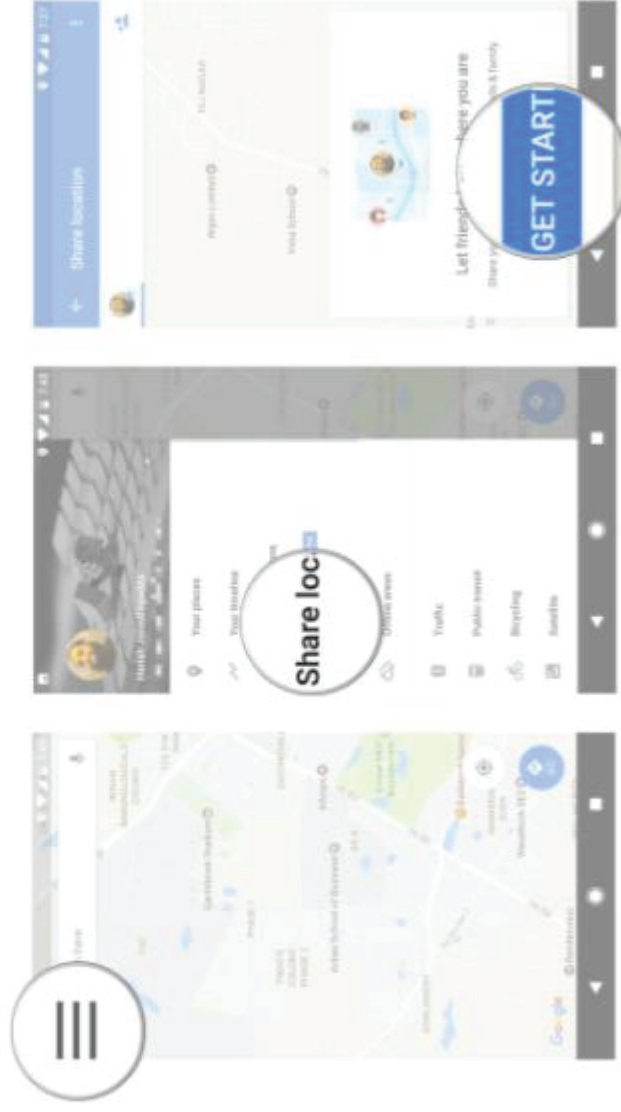
US9467838

Exemplary Supporting Evidence Regarding Accused Products

How to create a shareable link

You can also create a link and use it to share your location easily. Here's how to do it:

1. Tap the hamburger menu on the top left corner of the screen.
2. Select Share location.
3. Tap Get Started.



<https://www.androidcentral.com/how-share-location-google-maps>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

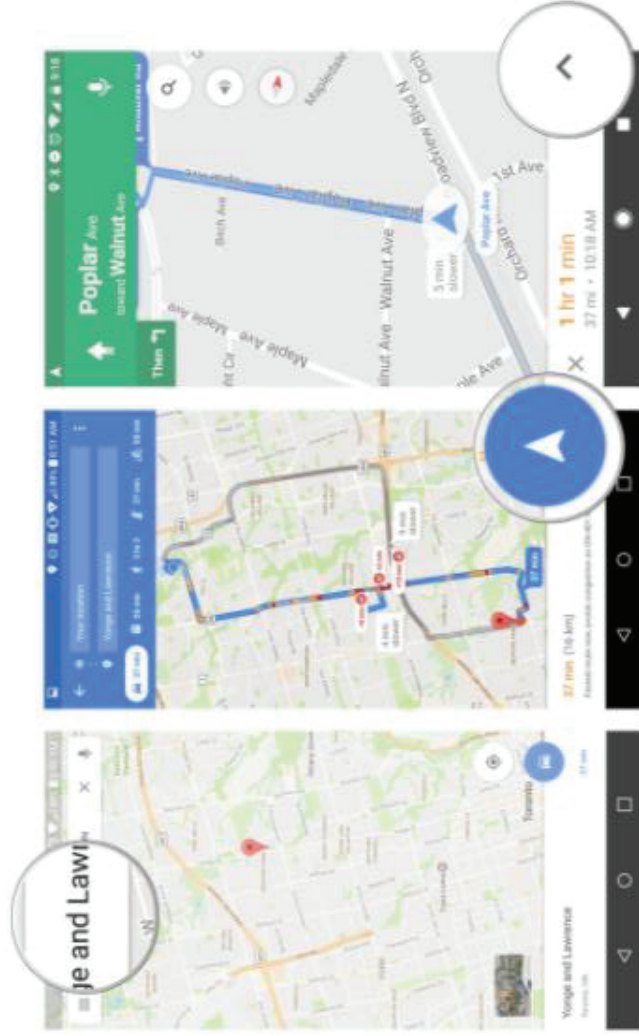
US9467838

Exemplary Supporting Evidence Regarding Accused Products

How to share your navigation directions while you walk, drive or transit

One of the best ways to share your location with someone is during a drive, walk or transit. If you're meeting a friend or family member somewhere, or navigating towards their home, Google Maps lets you share your location with them for the duration of the trip. It's magic!

1. In the **search bar** enter your destination.
2. Pick your navigation type (drive, transit, walk) and press the **blue navigate button**.
3. Tap the **arrow next to the time-to-destination number** at the bottom of the screen.



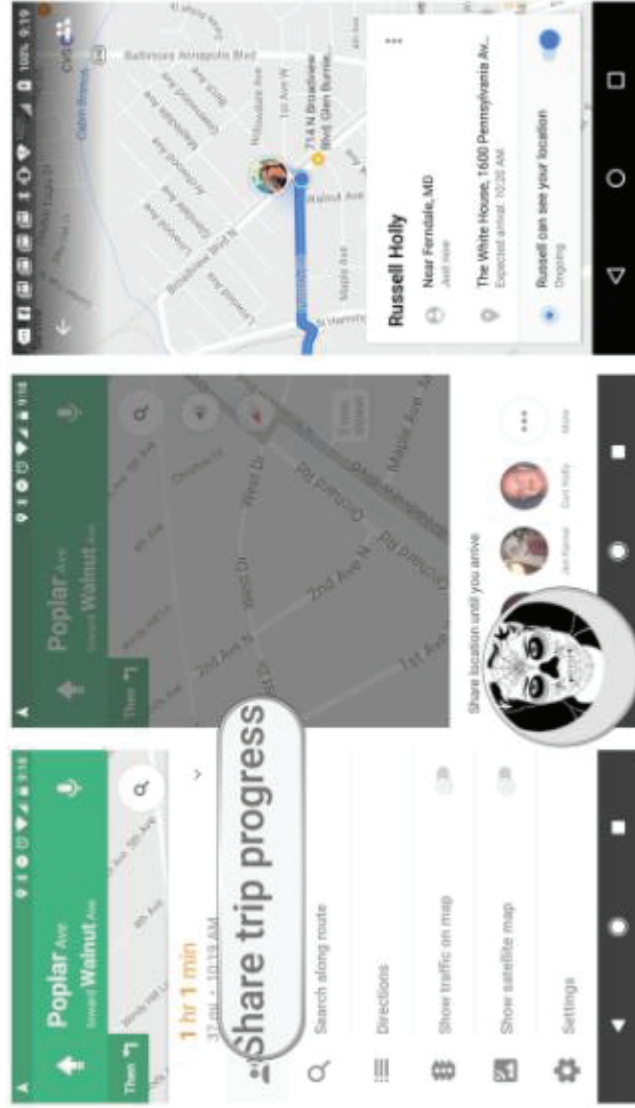
<https://www.androidcentral.com/how-share-location-google-maps>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products

- 4. Tap Share trip progress.
- 5. Choose one or more contacts to share trip progress.

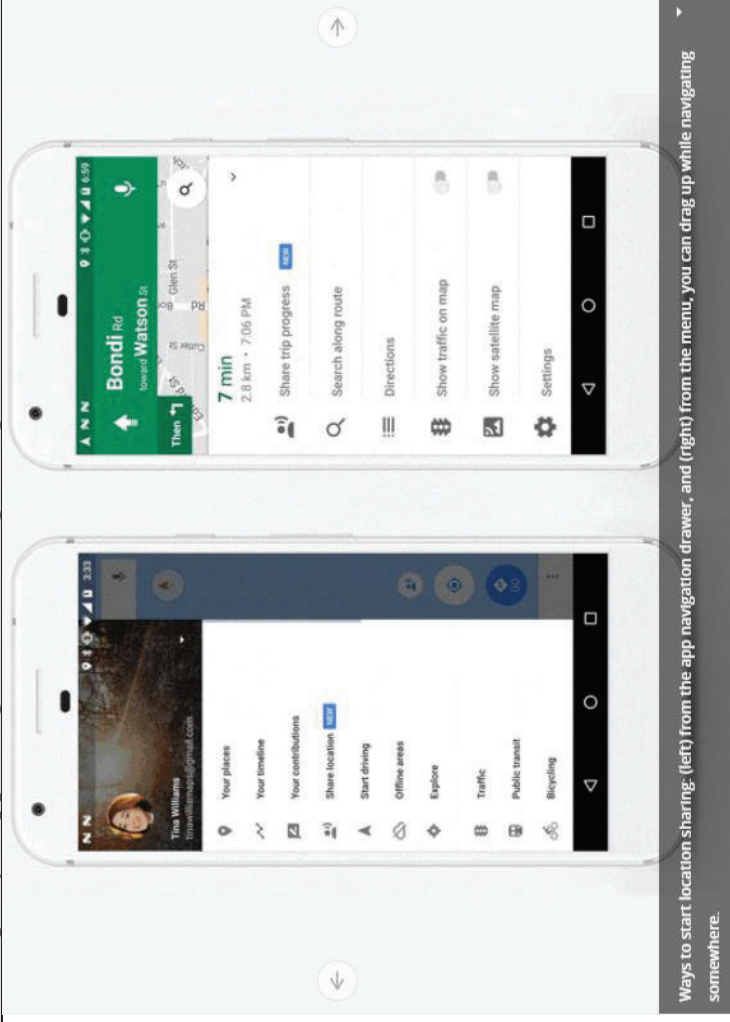


You can also stop sharing your location with someone before a trip ends.
<https://www.androidcentral.com/how-share-location-google-maps>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products



<https://arstechnica.com/gadgets/2017/03/location-sharing-finally-returns-to-google-maps/>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products

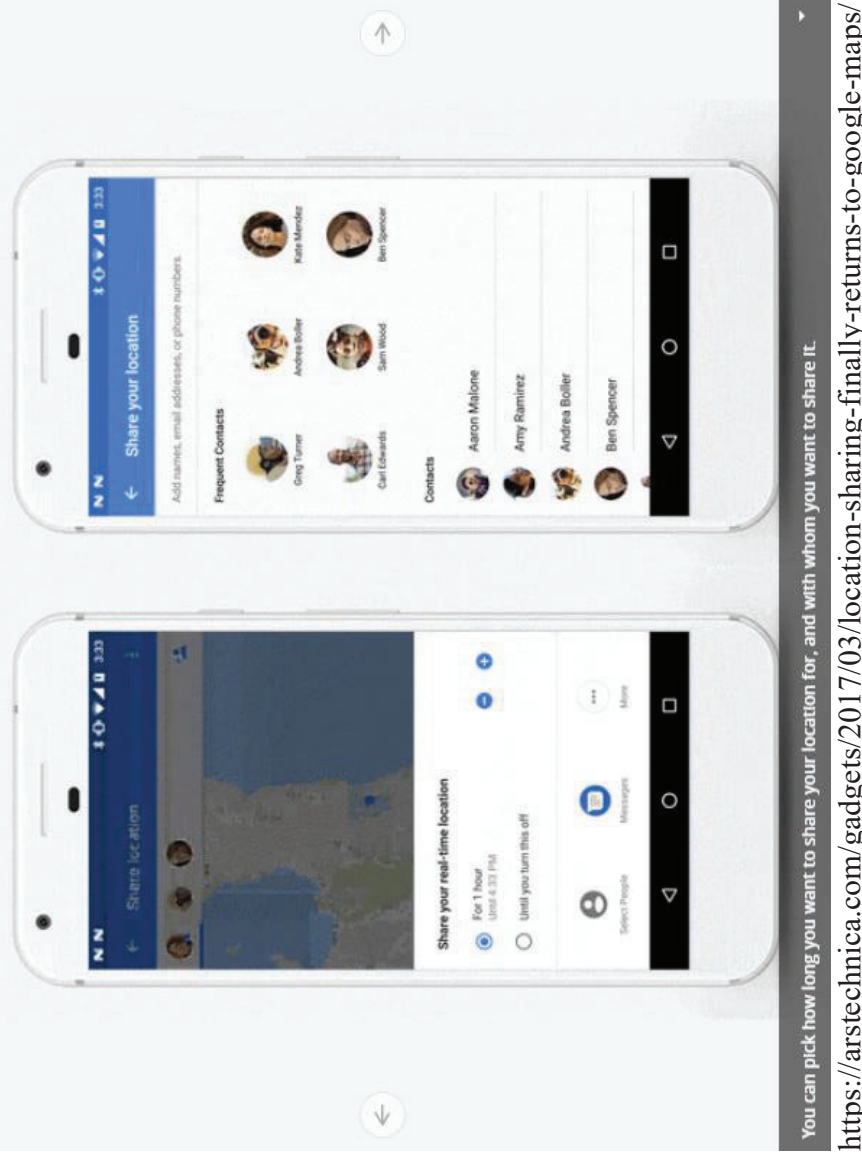


Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products

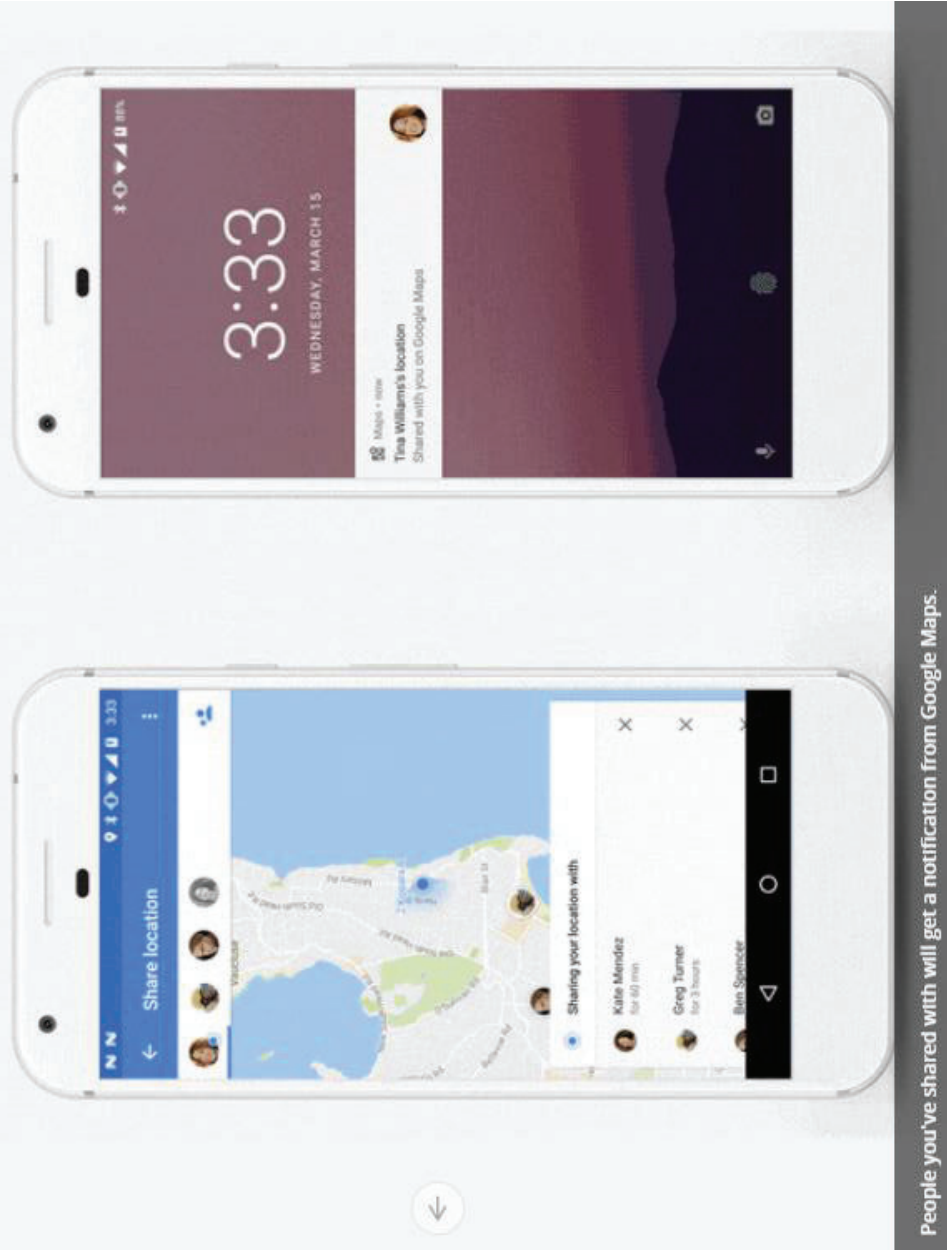
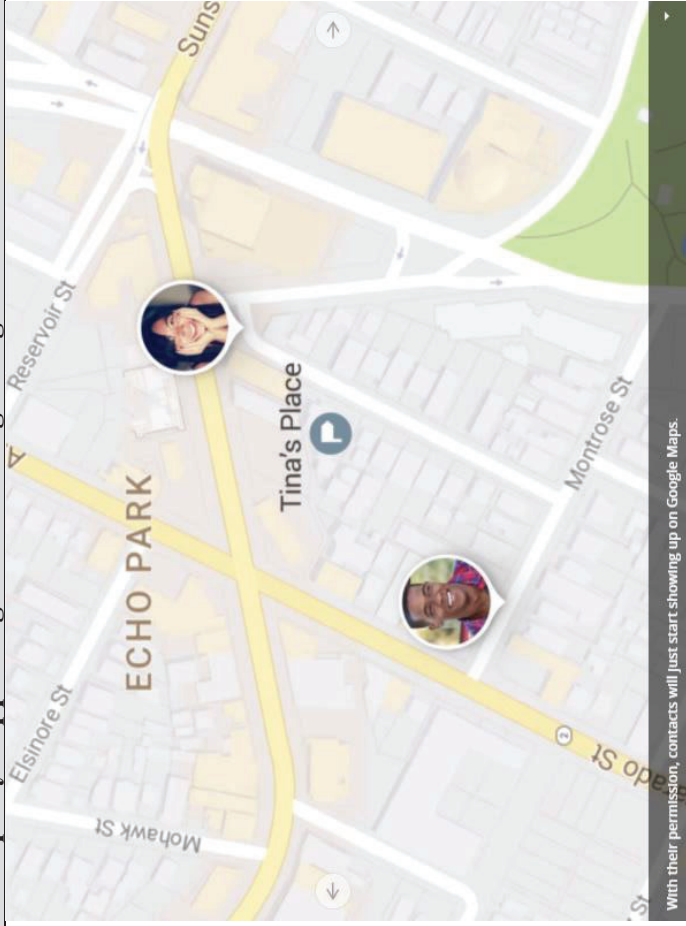


Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products



<https://arstechnica.com/gadgets/2017/03/location-sharing-finally-returns-to-google-maps/>

For example, Google requests and receives map data from many sources and servers, both internally and externally. In response to one of a number of user actions (e.g. zoom, drag or change focus, change map type, select another device or user), new map data is retrieved to complete the user's action and the displayed map is replaced or updated accordingly. The new map data may come from one of many sources and servers. The Accused Products request and receive the map data described above. Alternatively, any computer signed-in to the Google network services may request and receive the map data described above.

Another example of this limitation in practice includes toggling between map types. The first-displayed map can be the standard interface map with geographical points of interest. In response to user action, a second map can be

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| | |
|-------------------------|--|
| <p>US9467838</p> | <p>Exemplary Supporting Evidence Regarding Accused Products</p> <p>called with satellite imagery, which would be retrieved from a different server source having satellite information. Another known example is when a user zooms into a standard map and a transmit map is retrieved and overlaid or otherwise included to form a second map with transit data pulled from a transit data server.</p> <p>3.36 United States. Google maps of the United States include data provided from the following sources:</p> <ul style="list-style-type: none"> a. U.S. Fish and Wildlife Service - http://www.fws.gov/ b. U.S. Census Bureau - http://www.census.gov/ c. USDA Forest Service - http://www.fs.fed.us/ d. U.S. Geological Survey, Gap Analysis Program (GAP) - http://gapanalysis.usgs.gov/padus/ e. U.S. Geological Survey, U.S. Geographic Names Information System (GNIS) - http://geonames.usgs.gov/ f. U.S. Geological Survey, National Hydrography Dataset (NHD) - http://nhd.usgs.gov/ g. U.S. Geological Survey, Topographic Maps - http://topomaps.usgs.gov/ h. U.S. Geological Survey - http://www.usgs.gov/ i. U.S. Coast Guard - http://www.uscg.mil/ j. University of New Hampshire - http://ccom.unh.edu/ k. U.S. National Parks Service - http://www.nps.gov l. U.S. Department of Transportation, Research and Innovative Technology Administration - http://www.rita.dot.gov/ <p>https://www.google.com/intl/en_us/help/legalnotices_maps.html</p> |
|-------------------------|--|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products

Data center locations

We own and operate data centers around the world to keep our products running 24 hours a day, 7 days a week. Find out more about our data center locations, community involvement, and job opportunities in our locations around the world.

Americas

- Berkeley County, South Carolina
- Council Bluffs, Iowa
- Douglas County, Georgia
- Jackson County, Alabama
- Lenoir, North Carolina
- Mayes County, Oklahoma
- Montgomery County, Tennessee
- Quilicura, Chile
- The Dalles, Oregon

Asia

- Changhua County, Taiwan
- Singapore

Europe

- Dublin, Ireland
- Eemshaven, Netherlands
- Hamina, Finland
- St Ghislain, Belgium



<https://www.google.com/about/datacenters/inside/locations/index.html>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products

Request Location Updates

Before requesting location updates, your app must connect to location services and make a location request. The lesson on Changing Location Settings shows you how to do this. Once a location request is in place you can start the regular updates by calling `requestLocationUpdates()`. Do this in the `onConnected()` callback, provided by Google API Client, which is called when the client is ready.

Depending on the form of the request, the fused location provider either invokes the `LocationListener.onLocationChanged()` callback method and passes it a `Location` object, or issues a `PendingIntent` that contains the location in its extended data. The accuracy and frequency of the updates are affected by the location permissions you've requested and the options you set in the location request object.

This lesson shows you how to get the update using the `LocationListener` callback approach. Call `requestLocationUpdates()`, passing it your instance of the `GoogleApiClient`, the `LocationRequest` object, and a `LocationListener`. Define a `startLocationUpdates()` method, called from the `onConnected()` callback, as shown in the following code sample:

```
@Override
public void onConnected(Bundle connectionHint) {
    ...
    if (mRequestingLocationUpdates) {
        startLocationUpdates();
    }
}

protected void startLocationUpdates() {
    LocationServices.FusedLocationApi.requestLocationUpdates(
        mGoogleApiClient, mLocationRequest, this);
}
```

Notice that the above code snippet refers to a boolean flag, `mRequestingLocationUpdates`, used to track whether the user has turned location updates on or off. For more about retaining the value of this flag across instances of the activity, see [Save the State of the Activity](#).

<https://developer.android.com/training/location/receive-location-updates.html>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| | |
|------------------|--|
| <p>US9467838</p> | <p>Exemplary Supporting Evidence Regarding Accused Products</p> <p>Configure a firewall to allow access to the Google Maps APIs Services</p> <p><i>Why it's important:</i> Google Maps APIs services use a variety of domains, some of which do not belong to the *google.com domain. If you are behind a restrictive firewall, it is important to allow access to the domains used by each Maps API service. If your firewall doesn't allow access to these domains, API requests will fail, which can break your applications. You can find a complete list of domains used by the Maps APIs in the Support Portal:</p> <ol style="list-style-type: none"> 1. Log in to the Google Cloud Support Portal. 2. The Support Portal is available only to customers with the Google Maps APIs Premium Plan or a previous Google Maps APIs for Work or Google Maps for Business license. 3. Navigate to the Resources tab. 4. Select the list of domains used by the Google Maps APIs family. (Here's the direct link.) 5. Allow your applications to access the listed domains. <p>We do not recommend managing firewall restrictions by IP address, as the IPs associated with these domains are not static.</p> <p>Note: Google Maps APIs services use port 80 (http) and 443 (https) for inbound and outbound traffic. These services also require GET, POST, PUT, DELETE, and HEAD requests. Configure your firewall to allow traffic over these ports and to allow requests, depending on API and use case.</p> <p>https://developers.google.com/maps/premium/prelaunch-checklist#firewall</p> <p><u>Exemplary Screenshots:</u></p> <p>See, e.g., 1D and 1E above.</p> <p><u>Exemplary Source Code:</u></p> <p>The above functionality is performed at least in part by the following publicly available source code and/or source code that invokes or is invoked by the following source code (or a substantially similar copy compiled and loaded onto the Accused Products by ZTE). AGIS reserves the right to supplement these contentions with additional source code as discovery progresses and as additional source code is made available.</p> |
|------------------|--|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| US9467838 | Exemplary Supporting Evidence Regarding Accused Products |
|--|--|
| <pre> 44 * Class that sends chat message via SMS. 45 * 46 * The interface emulates a blocking sending similar to making an HTTP request. 47 * It calls the SmsManager to send a (potentially multipart) message and waits 48 * on the sent status on each part. The waiting has a timeout so it won't wait 49 * forever. Once the sent status of all parts received, the call returns. 50 * A successful sending requires success status for all parts. Otherwise, we 51 * pick the highest level of failure as the error for the whole message, which 52 * is used to determine if we need to retry the sending. 53 */ 54 public class SmsSender { 55 private static final String TAG = LogUtil.BUGLE_TAG; 56 57 public static final String EXTRA_PART_ID = "part_id"; 58 59 /* 60 * A map for pending sms messages. The key is the random request UUID. 61 */ 62 private static ConcurrentHashMap<Uri, SendResult> sPendingMessageMap = 63 new ConcurrentHashMap<Uri, SendResult>(); 64 65 private static final Random RANDOM = new Random(); 66 67 // Whether we should send multipart SMS as separate messages 68 private static Boolean sSendMultipartSmsAsSeparateMessages = null; 69 </pre> | <p>https://android.googlesource.com/platform/packages/apps/Messaging/+/nougat-mr1-release/src/com/android/messaging/sms/SmsSender.java</p> |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products

```

253 // Actually sending the message using SmsManager
254 private static void sendInternal(final Context context, final int subId, String dest,
255     final ArrayList<String> messages, final String serviceCenter,
256     final boolean requireDeliveryReport, final Uri messageUri) throws SmsException {
257     Assert.notNull(context);
258     final SmsManager smsManager = PhoneUtils.get(subId).getSmsManager();
259     final int messageCount = messages.size();
260     final ArrayList<PendingIntent> deliveryIntents = new ArrayList<PendingIntent>(messageCount);
261     final ArrayList<PendingIntent> sentIntents = new ArrayList<PendingIntent>(messageCount);
262     for (int i = 0; i < messageCount; i++) {
263         // Make pending intents different for each message part
264         final int partId = (messageCount <= 1 ? 0 : i + 1);
265         if (requireDeliveryReport && (i == (messageCount - 1))) {
266             // TODO we only care about the delivery status of the last part
267             // Shall we have better tracking of delivery status of all parts?
268             deliveryIntents.add((PendingIntent)getBroadcast(
269                 context,
270                 partId,
271                 getSendStatusIntent(context, SendStatusReceiver.MESSAGE_DELIVERED_ACTION,
272                     messageUri, partId, subId),
273                 0/*flag*/));
274         } else {
275             deliveryIntents.add(null);
276         }
277         sentIntents.add((PendingIntent)getBroadcast(
278             context,
279             partId,
280             getSendStatusIntent(context, SendStatusReceiver.MESSAGE_SENT_ACTION,
281                 messageUri, partId, subId),
282             0/*flag*/));
283     }
284     if (sSendMultiPartSmsAsSeparateMessages == null) {
285         sSendMultiPartSmsAsSeparateMessages = MmsConfig.get(subId)
286             .getSendMultiPartSmsAsSeparateMessages();
287     }

```

<https://android.googlesource.com/platform/packages/apps/Messaging/+/nougat-mr1-release/src/com/android/messaging/sms/SmsSender.java>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products

```

288     try {
289         if (sSendMultipartSmsAsSeparateMessages) {
290             // If multipart sms is not supported, send them as separate messages
291             for (int i = 0; i < messageCount; i++) {
292                 smsManager.sendTextMessage(dest,
293                     serviceCenter,
294                     messages.get(i),
295                     sentIntents.get(i),
296                     deliveryIntents.get(i));
297             }
298         } else {
299             smsManager.sendMultipartTextMessage(
300                 dest, serviceCenter, messages, sentIntents, deliveryIntents);
301         }
302     } catch (final Exception e) {
303         throw new SmsException("SmsSender: caught exception in sending " + e);
304     }
305 }

```

<https://android.googlesource.com/platform/packages/apps/Messaging/+/nougat-mr1-release/src/com/android/messaging/sms/SmsSender.java>

```

56 * Class that receives incoming SMS messages through android.provider.Telephony.SMS_RECEIVED
57 *
58 * This class serves two purposes:
59 * - Process phone verification SMS messages
60 * - Handle SMS messages when the user has enabled us to be the default SMS app (Pre-KLP)
61 */
62 public final class SmsReceiver extends BroadcastReceiver {
63     private static final String TAG = LogUtil.BUGLE_TAG;
64
65     private static ArrayList<Pattern> sIgnoreSmsPatterns;
66

```

<https://android.googlesource.com/platform/packages/apps/Messaging/+/nougat-mr1-release/src/com/android/messaging/receiver/SmsReceiver.java>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products

```

200 public static void deliverSmsMessages(final Context context, final int subId,
201     final int errorCode, final android.telephony.SmsMessage[] messages) {
202     final ContentValues messageValues =
203         MmsUtils.parseReceivedSmsMessage(context, messages, errorCode);
204
205     LogUtil.v(TAG, "SmsReceiver.deliverSmsMessages");
206
207     final long nowInMillis = System.currentTimeMillis();
208     final long receivedTimestamps = MmsUtils.getMessageDate(messages[0], nowInMillis);
209
210     messageValues.put(Sms.Inbox.DATE, receivedTimestamps);
211     // Default to unread and unseen for us but ReceiveSmsMessageAction will override
212     // seen for the telephony db.
213     messageValues.put(Sms.Inbox.READ, 0);
214     messageValues.put(Sms.Inbox.SEEN, 0);
215     if (OsUtil.isAtLeastL_MRI()) {
216         messageValues.put(Sms.SUBSCRIPTION_ID, subId);
217     }
218
219     if (messages[0].getMessageClass() == android.telephony.SmsMessage.MessageClass.CLASS_0 ||
220         DebugUtils.isDebugEnabled()) {
221         Factory.get().getUIIntents().launchClassZeroActivity(context, messageValues);
222     } else {
223         final ReceiveSmsMessageAction action = new ReceiveSmsMessageAction(messageValues);
224         action.start();
225     }
226 }

```

<https://android.googlesource.com/platform/packages/apps/Messaging/+nougat-mr1-release/src/com/android/messaging/receiver/SmsReceiver.java>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| US9467838 | Exemplary Supporting Evidence Regarding Accused Products |
|--|--|
| <pre> 228 @Override 229 public void onReceive(final Context context, final Intent intent) { 230 LogUtil.v(TAG, "SmsReceiver.onReceive " + intent); 231 // On KLP+ we only take delivery of SMS messages in SmsDeliverReceiver. 232 if (PhoneUtils.getDefault().isSmsEnabled()) { 233 final String action = intent.getAction(); 234 if (OsUtil.isSecondaryUser() && 235 (Telephony.Sms.Intents.SMS_RECEIVED_ACTION.equals(action) 236 // TODO: update this with the actual constant from Telephony 237 "android.provider.Telephony.MMS_DOWNLOADED".equals(action))) { 238 postNewMessageSecondaryUserNotification(); 239 } else if (!OsUtil.isAtLeastKLP()) { 240 deliverSmsIntent(context, intent); 241 } 242 } 243 } </pre> | <p>https://android.googlesource.com/platform/packages/apps/Messaging/+nougat-mr1-release/src/com/android/messaging/receiver/SmsReceiver.java</p> |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| US9467838 | Exemplary Supporting Evidence Regarding Accused Products |
|-----------|---|
| 52 | * Class that sends chat message via MMS. |
| 53 | * |
| 54 | * The interface emulates a blocking send similar to making an HTTP request. |
| 55 | */ |
| 56 | public class MmsSender { |
| 57 | private static final String TAG = LogUtil.BUGLE_TAG; |
| 58 | |
| 59 | /** |
| 60 | * Send an MMS message. |
| 61 | * |
| 62 | * @param context Context |
| 63 | * @param messageUri The unique URI of the message for identifying it during sending |
| 64 | * @param sendReq The SendReq PDU of the message |
| 65 | * @throws MmsFailureException |
| 66 | */ |
| 67 | public static void sendMms(final Context context, final int subId, final Uri messageUri, |
| 68 | final SendReq sendReq, final Bundle sentIntentExtras) throws MmsFailureException { |
| 69 | sendMms(context, |
| 70 | subId, |
| 71 | messageUri, |
| 72 | null /* locationUrl */, |
| 73 | sendReq, |
| 74 | true /* responseImportant */, |
| 75 | sentIntentExtras); |
| 76 | } |
| | https://android.googlesource.com/platform/packages/apps/Messaging/+/nougat-mr1-release/src/com/android/messaging/sms/MmsSender.java |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| US9467838 | Exemplary Supporting Evidence Regarding Accused Products |
|-----------|---|
| 240 | * Download an MMS message. |
| 241 | * |
| 242 | * @param context Context |
| 243 | * @param contentLocation The url of the MMS message |
| 244 | * @throws MmsFailureException |
| 245 | * @throws InvalidHeaderValueException |
| 246 | */ |
| 247 | public static void downloadMms(final Context context, final int subId, |
| 248 | final String contentLocation, Bundle extras) throws MmsFailureException, |
| 249 | InvalidHeaderValueException { |
| 250 | final Uri requestUri = Uri.parse(contentLocation); |
| 251 | final Uri contentUri = MmsFileProvider.buildRawMmsUri(); |
| 252 | |
| 253 | final Intent downloadedIntent = new Intent(SendStatusReceiver.MMS_DOWNLOADED_ACTION, |
| 254 | requestUri, |
| 255 | context, |
| 256 | SendStatusReceiver.class); |
| 257 | downloadedIntent.putExtra(SendMessageAction.EXTRA_CONTENT_URI, contentUri); |
| 258 | if (extras != null) { |
| 259 | downloadedIntent.putExtras(extras); |
| 260 | } |
| 261 | final PendingIntent downloadedPendingIntent = PendingIntent.getBroadcast(context, |
| 262 | 0 /*request code*/, |
| 263 | downloadedIntent, |
| 264 | PendingIntent.FLAG_UPDATE_CURRENT); |
| 265 | |
| 266 | |
| 267 | MmsManager.downloadMultimediaMessage(subId, context, contentLocation, contentUri, |
| 268 | downloadedPendingIntent); |
| 269 | } |
| | https://android.googlesource.com/platform/packages/apps/Messaging/+nougat-mr1-release/src/com/android/messaging/sms/MmsSender.java |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| US9467838 | Exemplary Supporting Evidence Regarding Accused Products |
|-----------|---|
| 97 | * Execute an MMS HTTP request, either a POST (sending) or a GET (downloading) |
| 98 | x |
| 99 | * @param urlString The request URL, for sending it is usually the MMSC, and for downloading |
| 100 | * it is the message URL |
| 101 | * @param pdu For POST (sending) only, the PDU to send |
| 102 | * @param method HTTP method, POST for sending and GET for downloading |
| 103 | * @param isProxySet Is there a proxy for the MMSC |
| 104 | * @param proxyHost The proxy host |
| 105 | * @param proxyPort The proxy port |
| 106 | * @param mmsConfig The MMS config to use |
| 107 | * @param userAgent The user agent header value |
| 108 | * @param uaProfUrl The UA Prof URL header value |
| 109 | * @return The HTTP response body |
| 110 | * @throws MmsHttpException For any failures |
| 111 | */ |
| 112 | public byte[] execute(String urlString, byte[] pdu, String method, boolean isProxySet, |
| 113 | String proxyHost, int proxyPort, Bundle mmsConfig, String userAgent, String uaProfUrl) |
| 114 | throws MmsHttpException { |
| 115 | Log.d(MmsService.TAG, "HTTP: " + method + " " + Utils.redactUrlForNonVerbose(urlString) |
| 116 | + (isProxySet ? (" proxy=" + proxyHost + " : " + proxyPort) : ""); |
| 117 | + ", PDU size=" + (pdu != null ? pdu.length : 0)); |
| 118 | checkMethod(method); |
| 119 | URLConnection connection = null; |
| 120 | try { |
| 121 | Proxy proxy = Proxy.NO_PROXY; |
| 122 | if (isProxySet) { |
| 123 | proxy = new Proxy(Proxy.Type.HTTP, new InetSocketAddress(proxyHost, proxyPort)); |
| 124 | } |
| 125 | final URL url = new URL(urlString); |
| 126 | // Now get the connection |
| 127 | connection = (URLConnection) url.openConnection(proxy); |
| 128 | connection.setDoInput(true); |
| 129 | connection.setConnectTimeout(|
| 130 | mmsConfig.getInt(CarrierConfigValuesLoader.CONFIG_HTTP_SOCKET_TIMEOUT, |
| 131 | CarrierConfigValuesLoader.CONFIG_HTTP_SOCKET_TIMEOUT_DEFAULT)); |

<https://android.googlesource.com/platform/packages/apps/Messaging+/nougat-mr1-release/src/android/support/v7/mms/MmsHttpClient.java>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products

```

132 // ----- COMMON HEADERS -----
133 // Header: Accept
134 connection.setRequestProperty(HEADER_ACCEPT, HEADER_VALUE_ACCEPT);
135 // Header: Accept-Language
136 connection.setRequestProperty(
137     HEADER_ACCEPT_LANGUAGE, getLocale().getLanguage(Locale.getDefault()));
138 // Header: User-Agent
139 Log.i("MmsService.TAG", "HTTP: User-Agent=" + userAgent);
140 connection.setRequestProperty(HEADER_USER_AGENT, userAgent);
141 // Header: x-wap-profile
142 final String uaProfileTagName = mmsConfig.getString(
143     CarrierConfigValuesLoader.CONFIG_UA_PROF_TAG_NAME,
144     CarrierConfigValuesLoader.CONFIG_UA_PROF_TAG_NAME_DEFAULT);
145 if (uaProfile != null) {
146     Log.i("MmsService.TAG", "HTTP: UaProfile=" + uaProfile);
147     connection.setRequestProperty(uaProfileTagName, uaProfile);
148 }
149 // Add extra headers specified by mms_config.xml's httpparams
150 addExtraHeaders(connection, mmsConfig);
151 // Different stuff for GET and POST
152 if (METHOD_POST.equals(method)) {
153     if (pdu == null || pdu.length < 1) {
154         Log.e("MmsService.TAG", "HTTP: empty pdu");
155         throw new MmsHttpException(0/*statusCode*/, "Sending empty PDU");
156     }
157     connection.setOutput(true);
158     connection.setRequestMethod(METHOD_POST);
159     if (mmsConfig.getBoolean(
160         CarrierConfigValuesLoader.CONFIG_SUPPORT_HTTP_CHARSET_HEADER,
161         CarrierConfigValuesLoader.CONFIG_SUPPORT_HTTP_CHARSET_HEADER_DEFAULT)) {
162         connection.setRequestProperty(HEADER_CONTENT_TYPE,
163             HEADER_VALUE_CONTENT_TYPE_WITH_CHARSET);
164     } else {
165         connection.setRequestProperty(HEADER_CONTENT_TYPE,
166             HEADER_VALUE_CONTENT_TYPE_WITHOUT_CHARSET);

```

<https://android.googlesource.com/platform/packages/apps/Messaging/+nougat-mr1-release/src/android/support/v7/mms/MmsHttpClient.java>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products

```

167     }
168     if (Log.isLoggable(MmsService.TAG, Log.VERBOSE)) {
169         logHttpHeaders(connection.getRequestProperties());
170     }
171     connection.setFixedLengthStreamingMode(pdu.length);
172     // Sending request body
173     final OutputStream out =
174         new BufferedOutputStream(connection.getOutputStream());
175     out.write(pdu);
176     out.flush();
177     out.close();
178 } else if (METHOD_GET.equals(method)) {
179     if (Log.isLoggable(MmsService.TAG, Log.VERBOSE)) {
180         logHttpHeaders(connection.getRequestProperties());
181     }
182     connection.setRequestMethod(METHOD_GET);
183 }
184 // Get response
185 final int responseCode = connection.getResponseCode();
186 final String responseMessage = connection.getResponseMessage();
187 Log.d(MmsService.TAG, "HTTP: " + responseCode + " " + responseMessage);
188 if (Log.isLoggable(MmsService.TAG, Log.VERBOSE)) {
189     logHttpHeaders(connection.getHeaderFields());
190 }
191 if (responseCode / 100 != 2) {
192     throw new MmsHttpException(responseCode, responseMessage);
193 }
194 final InputStream in = new BufferedInputStream(connection.getInputStream());
195 final ByteArrayOutputStream byteOut = new ByteArrayOutputStream();
196 final byte[] buf = new byte[4096];
197 int count = 0;
198 while ((count = in.read(buf)) > 0) {
199     byteOut.write(buf, 0, count);
200 }
201 in.close();
202 final byte[] responseBody = byteOut.toByteArray();
203 Log.d(MmsService.TAG, "HTTP: response size="
204     + (responseBody != null ? responseBody.length : 0));
205 return responseBody;

```

<https://android.googlesource.com/platform/packages/apps/Messaging/+nougat-mr1-release/src/android/support/v7/mms/MmsHttpClient.java>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products

```

38 * Request to send an MMS
39 */
40 class SendRequest extends MmsRequest {
41     // Max send response PDU size in bytes (exceeding this may cause problem with
42     // system intent delivery).
43     private static final int MAX_SEND_RESPONSE_SIZE = 1000 * 1024;
44
45     private byte[] mPduData;
46
47     SendRequest(final String locationUrl, final Uri pduUri, final PendingIntent sentIntent) {
48         super(locationUrl, pduUri, sentIntent);
49     }
50
51     @Override
52     protected boolean loadRequest(final Context context, final Bundle mmsConfig) {
53         mPduData = readPduFromContentUri(
54             context,
55             pduUri,
56             mmsConfig.getInt(
57                 CarrierConfigValuesLoader.CONFIG_MAX_MESSAGE_SIZE,
58                 CarrierConfigValuesLoader.CONFIG_MAX_MESSAGE_SIZE_DEFAULT));
59         return (mPduData != null);
60     }
61
62     @Override
63     protected boolean transferResponse(final Context context, final Intent fillIn,
64         final byte[] response) {
65         // SendConf pdus are always small and can be included in the intent
66         if (response != null && fillIn != null) {
67             if (response.length > MAX_SEND_RESPONSE_SIZE) {
68                 // If the response PDU is too large, it won't be able to fit in
69                 // the PendingIntent to be transferred via system IPC.
70                 return false;
71             }
72             fillIn.putExtra(SmsManager.EXTRA_MMS_DATA, response);
73         }
74         return true;
75     }

```

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| | |
|------------------|--|
| <p>US9467838</p> | <p>Exemplary Supporting Evidence Regarding Accused Products https://android.googlesource.com/platform/packages/apps/Messaging/+nougat-mr1-release/src/android/support/v7/mms/SendRequest.java</p> <pre>public static LocationRequest create ()</pre> <p>Create a location request with default parameters.</p> <p>Default parameters are for a block accuracy, slowly updated location. It can then be adjusted as required by the applications before passing to the <code>FusedLocationProviderApi</code>.</p> <p>Returns</p> <ul style="list-style-type: none"> • a new location request <p>https://developers.google.com/android/reference/com/google/android/gms/location/LocationRequest</p> |
|------------------|--|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| US9467838 | Exemplary Supporting Evidence Regarding Accused Products |
|-----------|---|
| | <p>public static final int PRIORITY_BALANCED_POWER_ACCURACY</p> <p>Used with <code>setPriority(int)</code> to request "block" level accuracy.</p> <p>Block level accuracy is considered to be about 100 meter accuracy. Using a coarse accuracy such as this often consumes less power.</p> <p>Constant Value: 102</p> |
| | <p>public static final int PRIORITY_HIGH_ACCURACY</p> <p>Used with <code>setPriority(int)</code> to request the most accurate locations available.</p> <p>This will return the finest location available.</p> <p>Constant Value: 100</p> |
| | <p>public static final int PRIORITY_LOW_POWER</p> <p>Used with <code>setPriority(int)</code> to request "city" level accuracy.</p> <p>City level accuracy is considered to be about 10km accuracy. Using a coarse accuracy such as this often consumes less power.</p> <p>Constant Value: 104</p> <p>https://developers.google.com/android/reference/com/google/android/gms/location/LocationRequest</p> |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| | |
|-----------|--|
| US9467838 | <p data-bbox="185 709 228 1562">Exemplary Supporting Evidence Regarding Accused Products</p> <pre data-bbox="245 1060 280 1539">public Task<Location> getLastLocation ()</pre> <p data-bbox="313 972 341 1551">Returns the best most recent location currently available.</p> <p data-bbox="370 380 431 1551">If a location is not available, which should happen very rarely, null will be returned. The best accuracy available while respecting the location permissions will be returned.</p> <p data-bbox="461 338 522 1551">This method provides a simplified way to get location. It is particularly well suited for applications that do not require an accurate location and that do not want to maintain extra logic for location updates.</p> <pre data-bbox="578 863 613 1539">public Task<LocationAvailability> getLocationAvailability ()</pre> <p data-bbox="646 386 708 1551">Returns the availability of location data. When <code>isLocationAvailable()</code> returns true, then the location returned by <code>getLastLocation()</code> will be reasonably up to date within the hints specified by the active <code>LocationRequest</code>s.</p> <p data-bbox="737 604 764 1551">If the client isn't connected to Google Play services and the request times out, null is returned.</p> <p data-bbox="794 401 855 1551">Note it's always possible for <code>getLastLocation()</code> to return null even when this method returns true (e.g. location settings were disabled between calls).</p> <p data-bbox="862 197 925 1562">https://developers.google.com/android/reference/com/google/android/gms/location/FusedLocationProviderClient</p> |
|-----------|--|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| | | | | | | | |
|------------------|--|----------------|---------------------------------------|-----------------|--|---------------|--|
| <p>US9467838</p> | <p>Exemplary Supporting Evidence Regarding Accused Products</p> <p>public Task<Void> requestLocationUpdates (LocationRequest request, LocationCallback callback, Looper looper)</p> <p>Requests location updates with a callback on the specified Looper thread.</p> <p>This method is suited for the foreground use cases. For background use cases, the <code>PendingIntent</code> version of the method is recommended, see <code>requestLocationUpdates(LocationRequest, PendingIntent)</code>.</p> <p>Any previous LocationRequests registered on this LocationListener will be replaced.</p> <p>This call will keep the Google Play services connection active, so make sure to call <code>removeLocationUpdates(LocationCallback)</code> when you no longer need it, otherwise you lose the benefits of the automatic connection management.</p> <p>Callbacks for <code>LocationCallback</code> will be made on the specified thread, which must already be a prepared looper thread.</p> <p>Parameters</p> <table border="1"> <tr> <td>request</td> <td>The location request for the updates.</td> </tr> <tr> <td>callback</td> <td>The callback for the location updates.</td> </tr> <tr> <td>looper</td> <td>The Looper object whose message queue will be used to implement the callback mechanism, or null to make callbacks on the calling thread.</td> </tr> </table> <p>https://developers.google.com/android/reference/com/google/android/gms/location/FusedLocationProviderClient</p> | request | The location request for the updates. | callback | The callback for the location updates. | looper | The Looper object whose message queue will be used to implement the callback mechanism, or null to make callbacks on the calling thread. |
| request | The location request for the updates. | | | | | | |
| callback | The callback for the location updates. | | | | | | |
| looper | The Looper object whose message queue will be used to implement the callback mechanism, or null to make callbacks on the calling thread. | | | | | | |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| | | | | | |
|-----------------------------|--|----------------------|---------------------------------------|-----------------------------|---|
| <p>US9467838</p> | <p>Exemplary Supporting Evidence Regarding Accused Products</p> <pre>public Task<Void> requestLocationUpdates (LocationRequest request, PendingIntent callbackIntent)</pre> <p>Requests location updates with a callback on the specified PendingIntent.</p> <p>This method is suited for the background use cases, more specifically for receiving location updates, even when the app has been killed by the system. In order to do so, use a <code>PendingIntent</code> for a started service. For foreground use cases, the <code>LocationCallback</code> version of the method is recommended, see <code>requestLocationUpdates(LocationRequest, LocationCallback, Looper)</code>.</p> <p>Any previously registered requests that have the same <code>PendingIntent</code> (as defined by <code>equals(Object)</code>) will be replaced by this request.</p> <p>Both <code>LocationResult</code> and <code>LocationAvailability</code> are sent to the given <code>PendingIntent</code>. You can extract data from an <code>Intent</code> using <code>hasResult(Intent)</code>, <code>extractResult(Intent)</code>, <code>hasLocationAvailability(Intent)</code>, and <code>extractLocationAvailability(Intent)</code>.</p> <p>Parameters</p> <table border="1"> <tr> <td><code>request</code></td> <td>The location request for the updates.</td> </tr> <tr> <td><code>callbackIntent</code></td> <td>A pending intent to be sent for each location update.</td> </tr> </table> <p>Returns</p> <ul style="list-style-type: none"> a Task for the call, check <code>isSuccessful()</code> to determine if it was successful. <p>https://developers.google.com/android/reference/com/google/android/gms/location/FusedLocationProviderClient</p> | <code>request</code> | The location request for the updates. | <code>callbackIntent</code> | A pending intent to be sent for each location update. |
| <code>request</code> | The location request for the updates. | | | | |
| <code>callbackIntent</code> | A pending intent to be sent for each location update. | | | | |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| | |
|------------------|--|
| <p>US9467838</p> | <p>Exemplary Supporting Evidence Regarding Accused Products</p> <p>public void onLocationAvailability (LocationAvailability locationAvailability)</p> <p>Called when there is a change in the availability of location data.</p> <p>When <code>isLocationAvailable()</code> returns <code>false</code> you can assume that location will not be returned in <code>onLocationResult(LocationResult)</code> until something changes in the device's settings or environment. Even when <code>isLocationAvailable()</code> returns <code>true</code> the <code>onLocationResult(LocationResult)</code> may not always be called regularly, however the device location is known and both the most recently delivered location and <code>getLastLocation(GoogleApiClient)</code> will be reasonably up to date given the hints specified by the active <code>LocationRequest</code> s.</p> <p>Parameters</p> <p>locationAvailability The current status of location availability.</p> <p>public void onLocationResult (LocationResult result)</p> <p>Called when device location information is available.</p> <p>The most recent location returned by <code>getLastLocation()</code> is not guaranteed to be immediately fresh, but will be reasonably up to date given the hints specified by the active <code>LocationRequest</code> s.</p> <p>Parameters</p> <p>result The latest location result available.</p> <p>https://developers.google.com/android/reference/com/google/android/gms/location/LocationCallback</p> <p>public abstract void onLocationChanged (Location location)</p> <p>Called when the location has changed.</p> <p>Parameters</p> <p>location The updated location.</p> |
|------------------|--|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| | |
|------------------|---|
| <p>US9467838</p> | <p>Exemplary Supporting Evidence Regarding Accused Products https://developers.google.com/android/reference/com/google/android/gms/location/LocationListener</p> <p>Public Constructors</p> <p><code>public MapView (Context context)</code></p> <p><code>public MapView (Context context, AttributeSet attrs)</code></p> <p><code>public MapView (Context context, AttributeSet attrs, int defStyleAttr)</code></p> <p><code>public MapView (Context context, GoogleMapOptions options)</code></p> <p>https://developers.google.com/android/reference/com/google/android/gms/maps/MapView</p> |
|------------------|---|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| | |
|---|--|
| <p>US9467838</p> | <p>Exemplary Supporting Evidence Regarding Accused Products</p> <pre>public void getMapAsync (OnMapReadyCallback callback)</pre> <p>Returns a non-null instance of the <code>GoogleMap</code>, ready to be used.</p> <p>Note that:</p> <ul style="list-style-type: none"> • This method must be called from the main thread. • The callback will be executed in the main thread. • In the case where Google Play services is not installed on the user's device, the callback will not be triggered until the user installs it. • The <code>GoogleMap</code> object provided by the callback is non-null. <p>Parameters</p> <p>callback The callback object that will be triggered when the map is ready to be used.</p> <pre>public final void onCreate (Bundle savedInstanceState)</pre> <p>You must call this method from the parent Activity/Fragment's corresponding method. https://developers.google.com/android/reference/com/google/android/gms/maps/MapView</p> |
| <p>[IF] receiving, from the second server, the second georeferenced map data;</p> | <p>ZTE infringes directly and/or indirectly by performing, inducing others to perform, and/or contributing to the performance of: receiving, from the second server, the second georeferenced map data.</p> <p>Regarding Find My Device and Android Device Manager, the user, via the first device, or the device itself receives second map data from a second server. The received second map data occurs responsive to user input (e.g., zoom, drag, pan, change focus, change map type, refresh or reload request device or symbol selection, another device or user selection, change in position of first device, change in position of a second device). Alternatively, the second map data may be received responsive to an automatic and/or pre-determined control caused by an instruction from within the first device or from the one or more second devices, e.g. a refresh. The received second map data includes an update to the first data or a</p> |

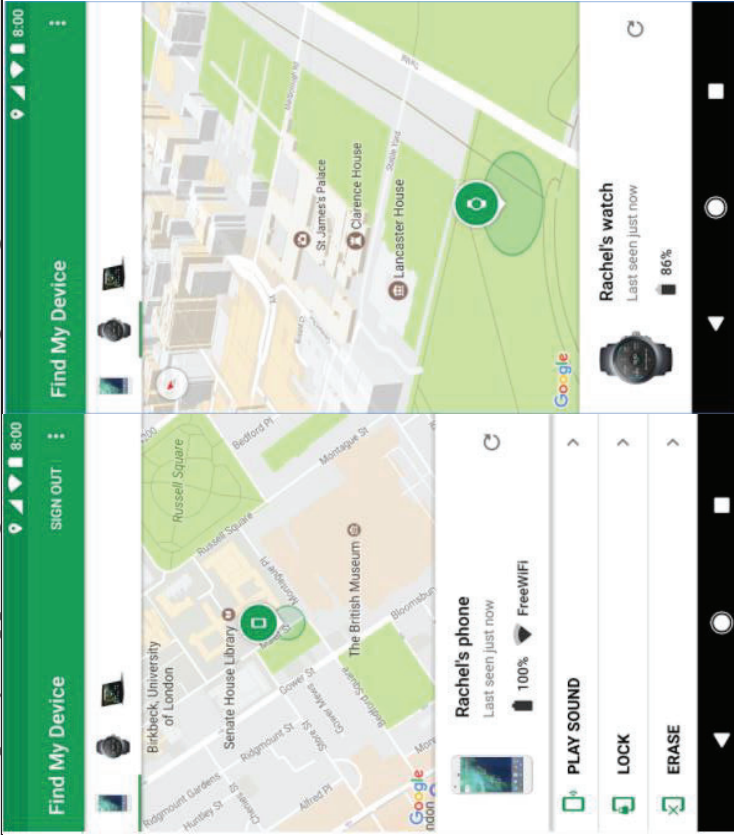
Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| | |
|-------------------------|--|
| <p>US9467838</p> | <p><u>Exemplary Supporting Evidence Regarding Accused Products</u></p> <p>replacement of the first data. The second map data may come from one of many sources and servers. The second map data includes georeferenced data.</p> <p><u>Regarding Google Maps</u>, Google Latitude, Google Plus, Google Hangouts, Google Messages, Google Allo, Google Duo, Google Chrome, and Android Messenger, the user, via the first device, or the device itself receives second map data from a second server. The received second map data occurs responsive to user input (e.g., zoom, drag, pan, change focus, refresh or reload request, change map type, device or symbol selection, another device or user selection, change in position of first device, change in position of a second device). Alternatively, the second map data may be received responsive to an automatic and/or pre-determined control caused by an instruction from within the first device or from the one or more second devices, e.g. a refresh. The received second map data includes an update to the first data or a replacement of the first data. The second map data may come from one of many sources and servers. The second map data includes georeferenced data.</p> <p><u>Exemplary Support for Find My Device:</u></p> |
|-------------------------|--|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products



<https://play.google.com/store/apps/details?id=com.google.android.apps.adm&hl=en>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

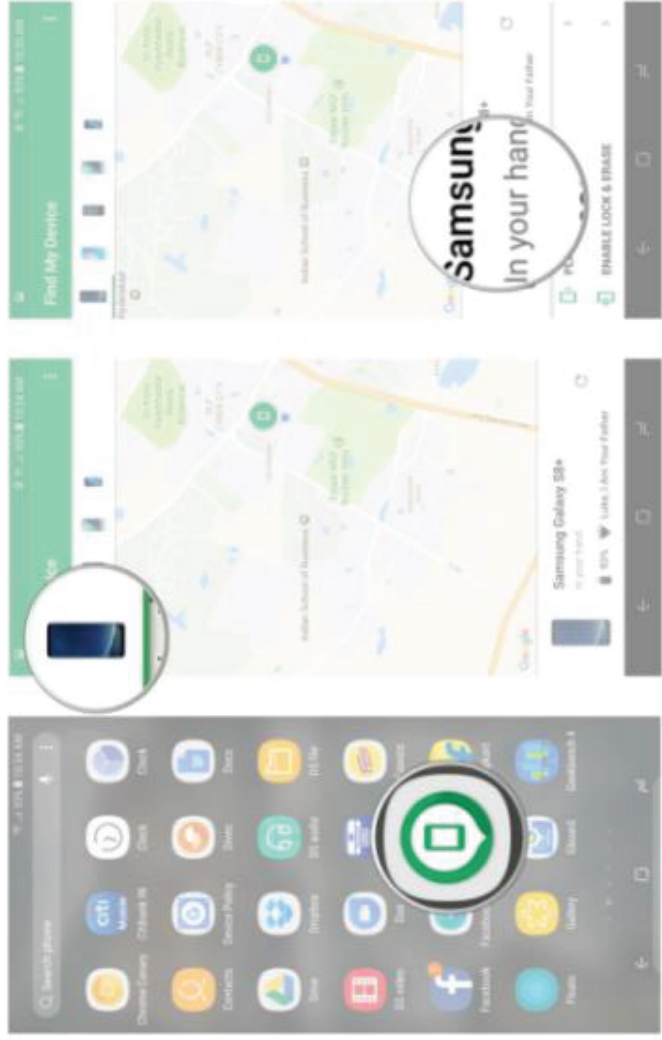
US9467838

Exemplary Supporting Evidence Regarding Accused Products

Once you're signed in to Find My Device, you'll see a map with your current location as well as the make and model of your phone, and two options — Play Sound, and Enable Lock & Erase. Hitting the latter option will allow you to start using the Lock and Erase functions.

If you've signed into more than one phone, you can select a particular device by browsing the list at the top of the screen.

1. Open **Find My Device** from your home screen or app drawer.
2. Select your phone from the **list of devices at the top of the screen**.
3. See if your phone is **discoverable**.



<https://www.androidcentral.com/find-my-device>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

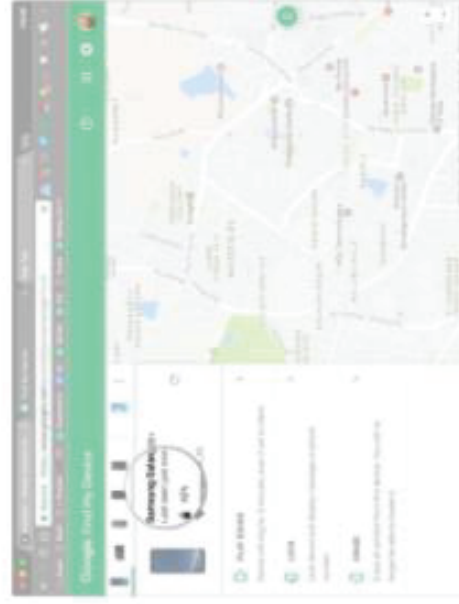
US9467838

Exemplary Supporting Evidence Regarding Accused Products

How to locate your phone over the internet

If you've lost your phone, you can remotely locate it through the Find My Device website. You'll need to sign in to the Google account that was used to set up Find My Device. It takes a few seconds, but the service should be able to track your phone. Alternatively, you can also do a Google search for "find my phone" to locate your handset.

1. Head to the [Find My Device website](#).
2. Sign in to your Google account.
3. Check if your device is visible.



<https://www.androidcentral.com/find-my-device>

Exemplary Support for Google Maps:

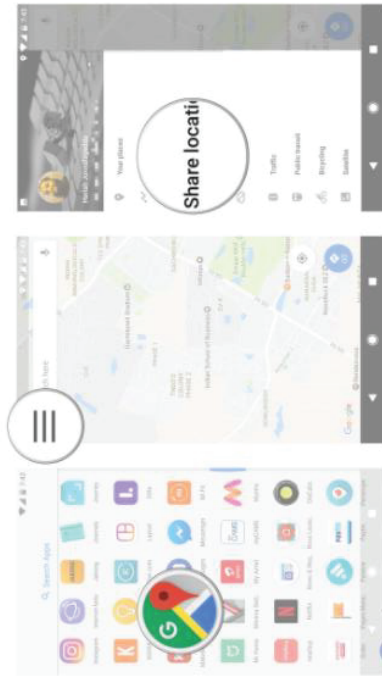
Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

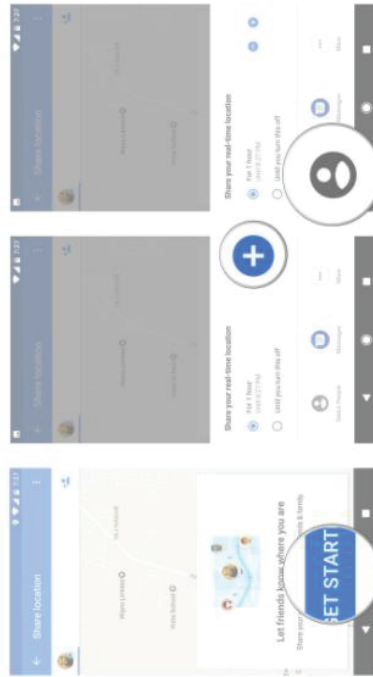
Exemplary Supporting Evidence Regarding Accused Products

How to share your location in Google Maps

1. Open Google Maps from the app drawer or the home screen.
2. Tap the hamburger menu (the three horizontal lines) on the top left corner of the screen.
3. Select Share location.



4. Tap Get Started.
5. Use the + icon to select a time period or select the Until you turn this off setting to share your location indefinitely.
6. Tap Select People.



<https://www.androidcentral.com/how-share-location-google-maps>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

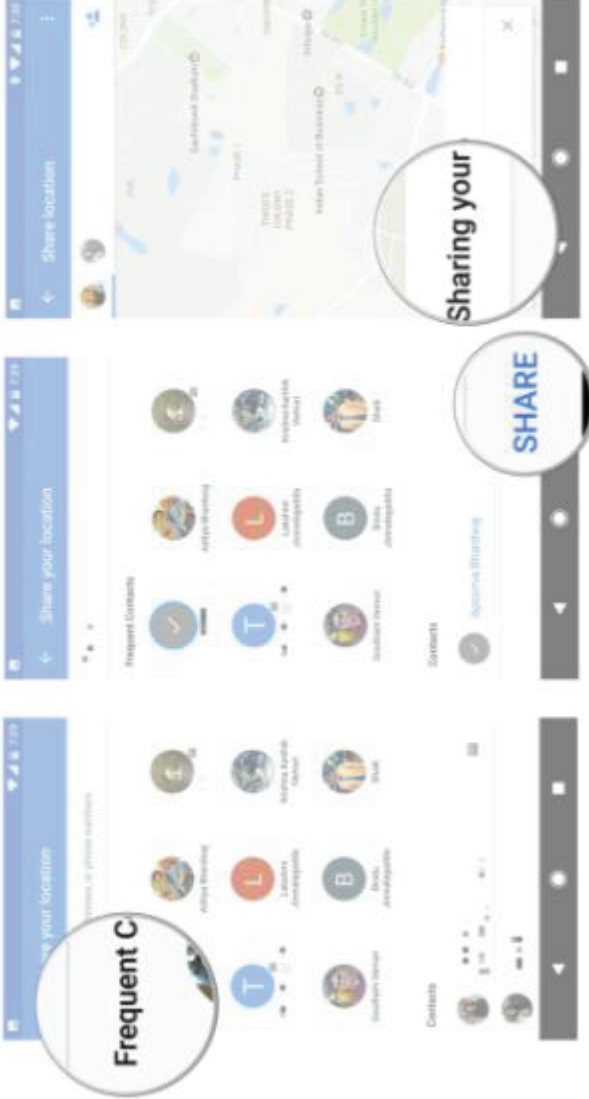
| | | |
|-------------------------|--|---|
| <p>US9467838</p> | <p>Exemplary Supporting Evidence Regarding Accused Products</p> | <p>7. You'll see a list of your frequent contacts at the top, along with a full list of contacts. Pick the contacts by tapping their name.</p> <p>8. Once you've selected the contacts you want to share your location to, tap Share.</p> <p>9. You'll see a message saying that the selected contact can view your location.</p>  <p>https://www.androidcentral.com/how-share-location-google-maps</p> |
|-------------------------|--|---|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

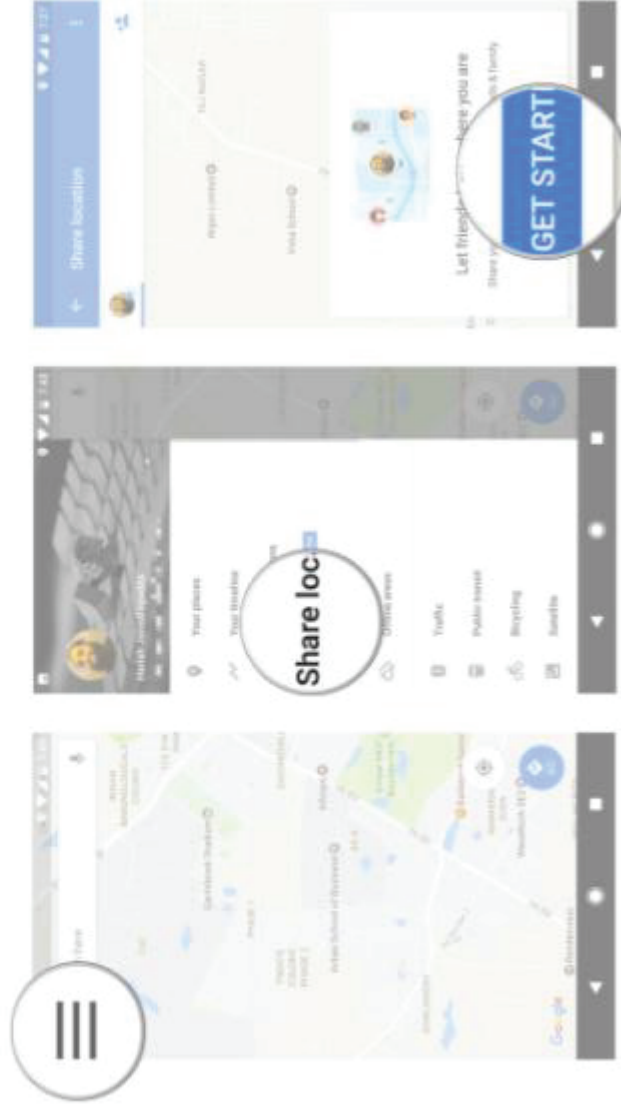
US9467838

Exemplary Supporting Evidence Regarding Accused Products

How to create a shareable link

You can also create a link and use it to share your location easily. Here's how to do it:

1. Tap the hamburger menu on the top left corner of the screen.
2. Select Share location.
3. Tap Get Started.



<https://www.androidcentral.com/how-share-location-google-maps>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| | |
|------------------|---|
| <p>US9467838</p> | <p>Exemplary Supporting Evidence Regarding Accused Products</p> <ol style="list-style-type: none">4. Select the amount of time you want to share your location.5. Tap More.6. Select your app of choice to create and send a unique URL that broadcasts your current location. You can email it, send the link via Messenger, or even tweet it to the intended recipient.  <p>https://www.androidcentral.com/how-share-location-google-maps</p> |
|------------------|---|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

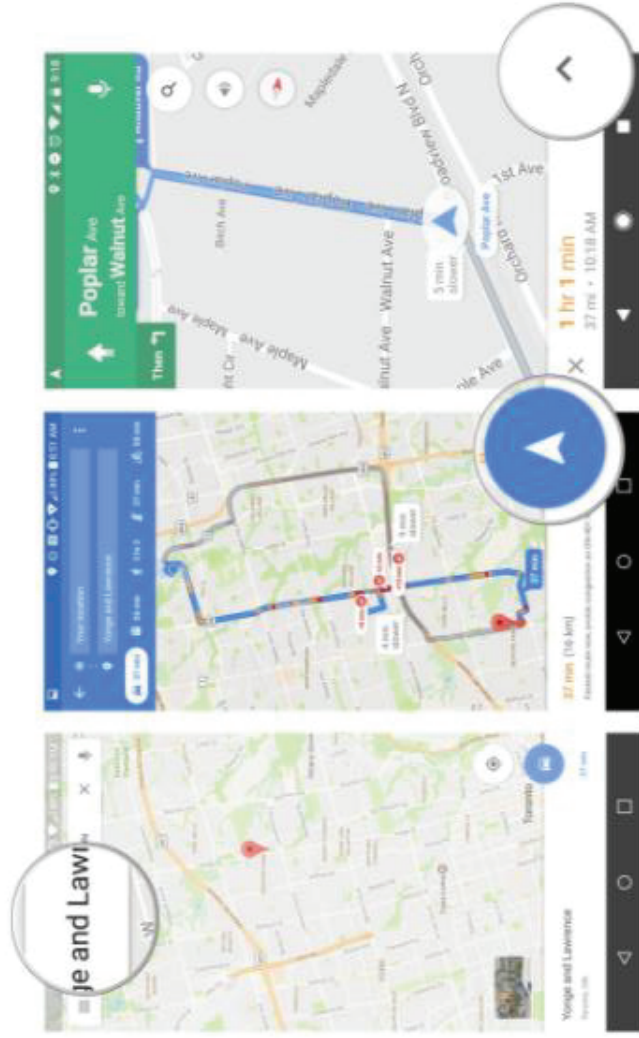
US9467838

Exemplary Supporting Evidence Regarding Accused Products

How to share your navigation directions while you walk, drive or transit

One of the best ways to share your location with someone is during a drive, walk or transit. If you're meeting a friend or family member somewhere, or navigating towards their home, Google Maps lets you share your location with them for the duration of the trip. It's magic!

1. In the **search bar** enter your destination.
2. Pick your navigation type (drive, transit, walk) and press the **blue navigate button**.
3. Tap the **arrow next to the time-to-destination number** at the bottom of the screen.



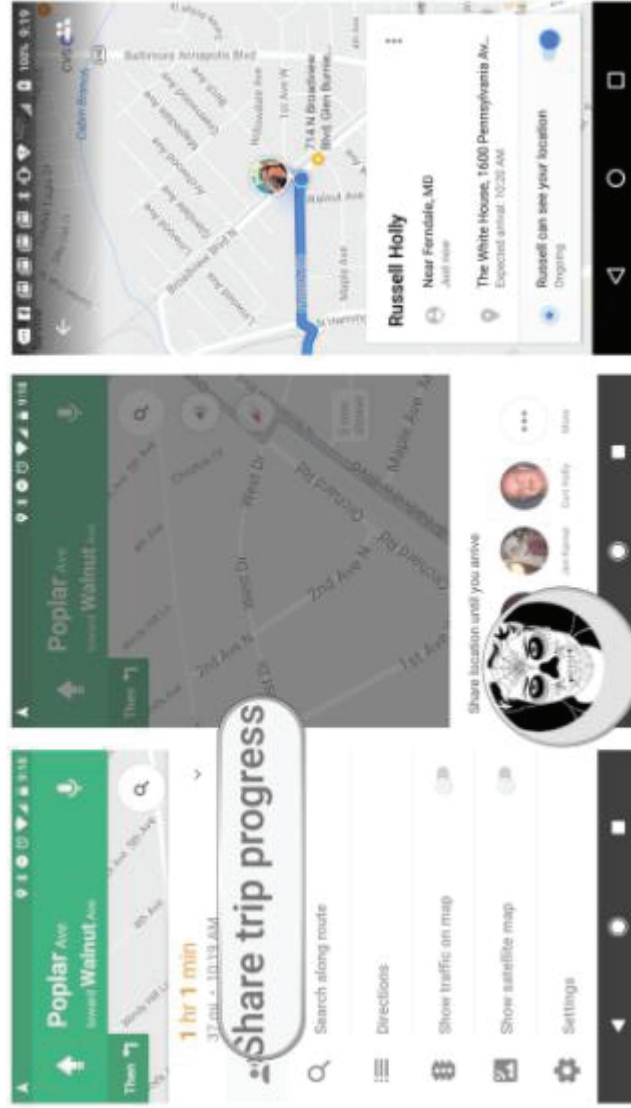
<https://www.androidcentral.com/how-share-location-google-maps>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products

- 4. Tap Share trip progress.
- 5. Choose one or more contacts to share trip progress.



You can also stop sharing your location with someone before a trip ends.
<https://www.androidcentral.com/how-share-location-google-maps>

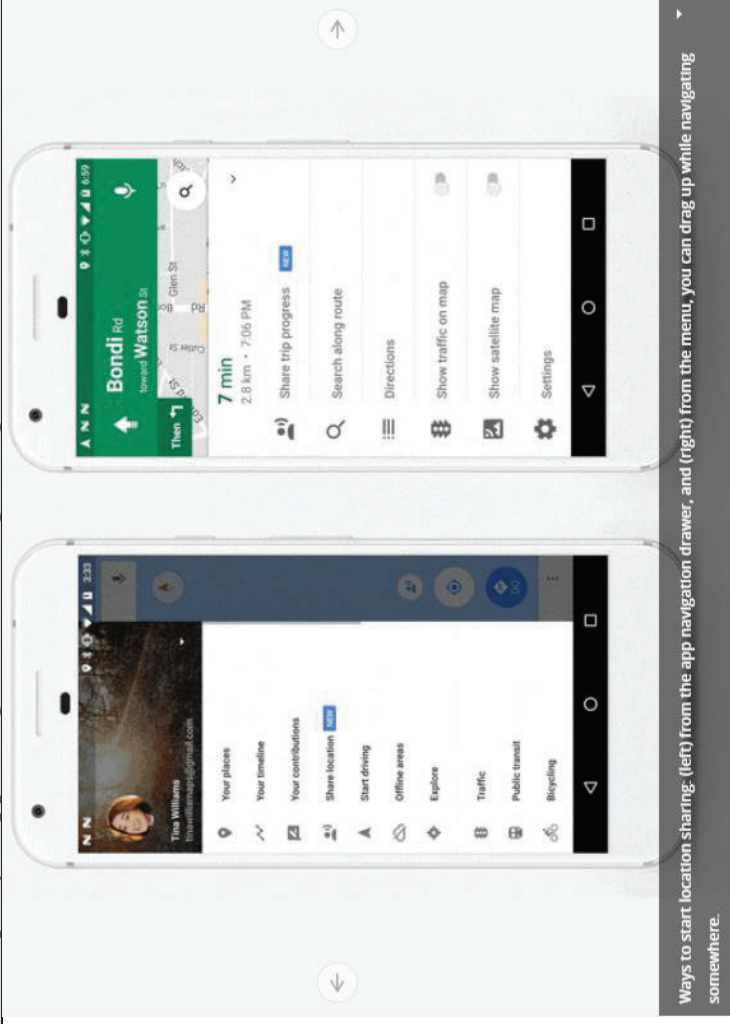
Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| | | |
|------------------|---|---|
| <p>US9467838</p> | <p>Exemplary Supporting Evidence Regarding Accused Products</p> | <p>1. Tap the arrow next to the time-to-destination number at the bottom of the screen.</p> <p>2. Tap Stop sharing.</p>  <p>That's it!</p> <p>Are you excited that location sharing is back in Google Maps? How often do you use the feature? https://www.androidcentral.com/how-share-location-google-maps</p> |
|------------------|---|---|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products



<https://arstechnica.com/gadgets/2017/03/location-sharing-finally-returns-to-google-maps/>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products

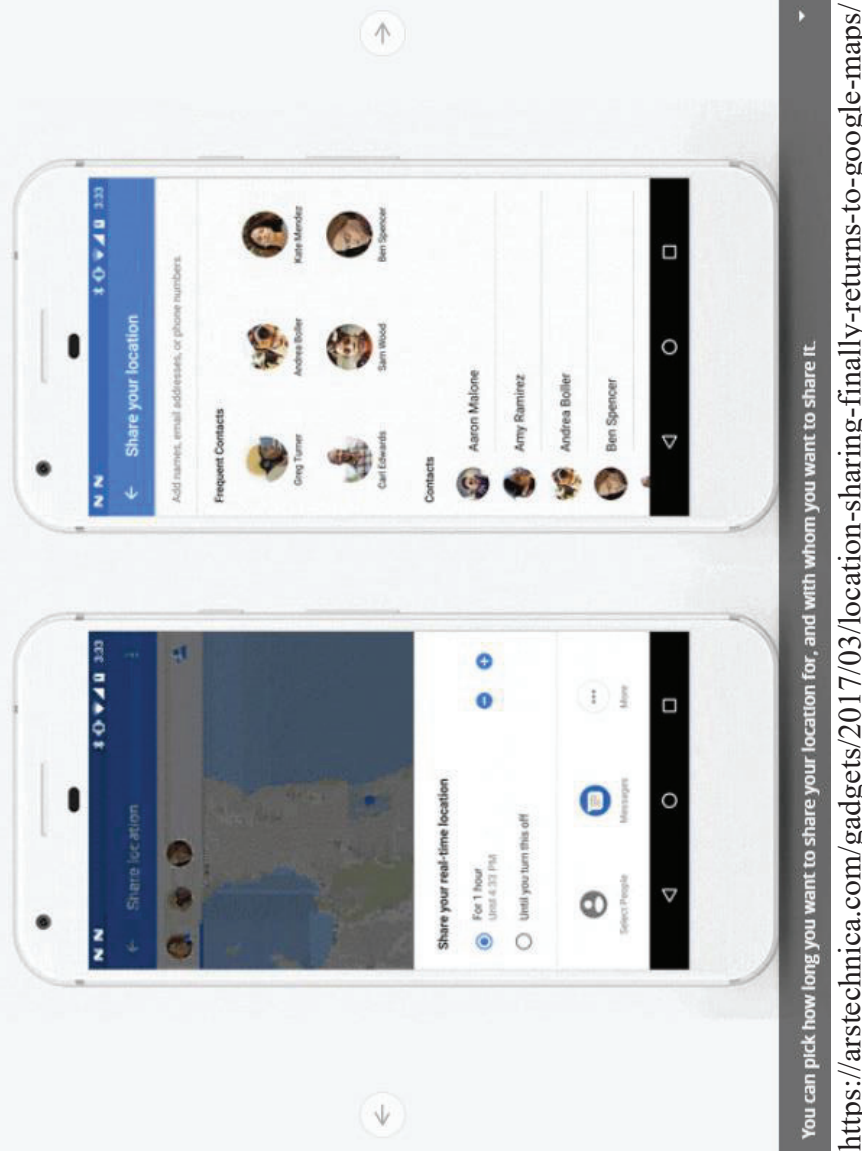
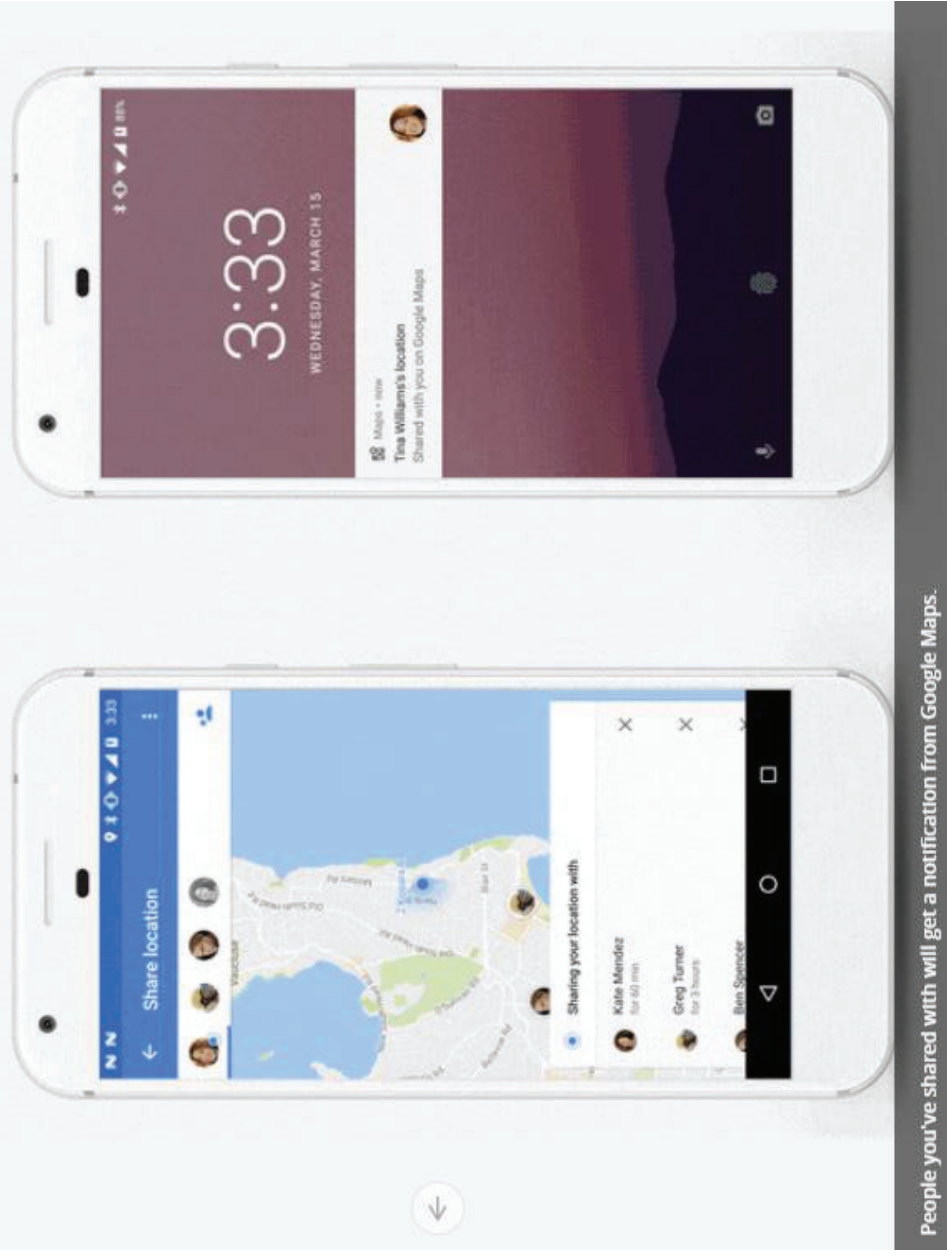


Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products

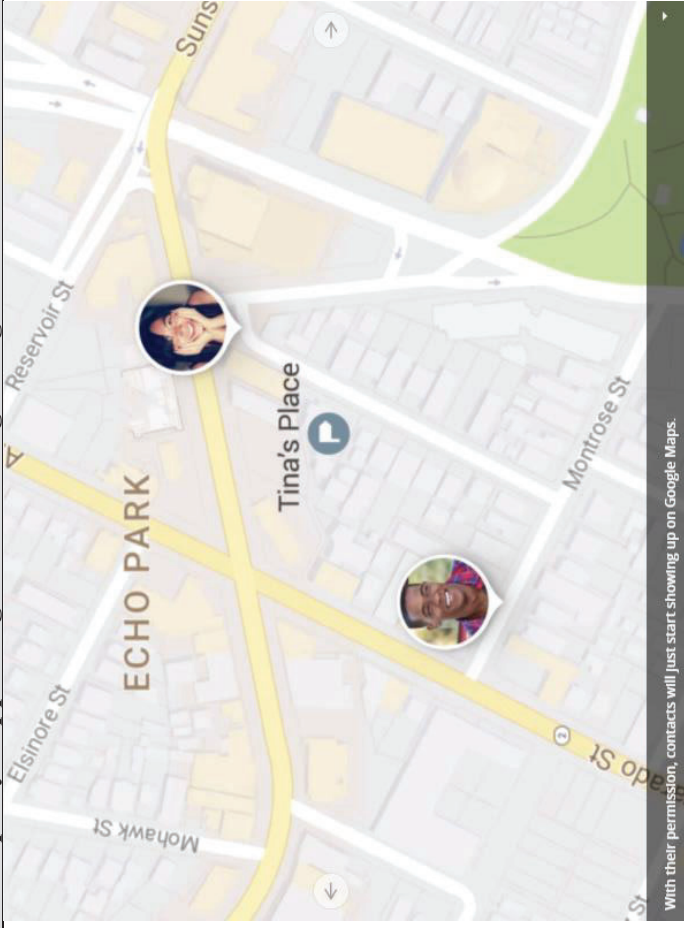


<https://arstechnica.com/gadgets/2017/03/location-sharing-finally-returns-to-google-maps/>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products



<https://arstechnica.com/gadgets/2017/03/location-sharing-finally-returns-to-google-maps/>

Exemplary Screenshots:

See, e.g., 1D and 1E above.

Exemplary Source Code:

The above functionality is performed at least in part by the following publicly available source code and/or source code that invokes or is invoked by the following source code (or a substantially similar copy compiled and loaded onto the Accused Products by ZTE). AGIS reserves the right to supplement these contentions with additional source code as discovery progresses and as additional source code is made available.

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products

```

44 * Class that sends chat message via SMS.
45 *
46 * The interface emulates a blocking sending similar to making an HTTP request.
47 * It calls the SmsManager to send a (potentially multipart) message and waits
48 * on the sent status on each part. The waiting has a timeout so it won't wait
49 * forever. Once the sent status of all parts received, the call returns.
50 * A successful sending requires success status for all parts. Otherwise, we
51 * pick the highest level of failure as the error for the whole message, which
52 * is used to determine if we need to retry the sending.
53 */
54 public class SmsSender {
55     private static final String TAG = LogUtil.BUGLE_TAG;
56
57     public static final String EXTRA_PART_ID = "part_id";
58
59     /*
60      * A map for pending sms messages. The key is the random request UUID.
61      */
62     private static ConcurrentHashMap<Uri, SendResult> sPendingMessageMap =
63         new ConcurrentHashMap<Uri, SendResult>();
64
65     private static final Random RANDOM = new Random();
66
67     // Whether we should send multipart SMS as separate messages
68     private static Boolean sSendMultipartSmsAsSeparateMessages = null;
69

```

<https://android.googlesource.com/platform/packages/apps/Messaging/+nougat-mr1-release/src/com/android/messaging/sms/SmsSender.java>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products

```

253 // Actually sending the message using SmsManager
254 private static void sendInternal(final Context context, final int subId, String dest,
255     final ArrayList<String> messages, final String serviceCenter,
256     final boolean requireDeliveryReport, final Uri messageUri) throws SmsException {
257     Assert.notNull(context);
258     final SmsManager smsManager = PhoneUtils.get(subId).getSmsManager();
259     final int messageCount = messages.size();
260     final ArrayList<PendingIntent> deliveryIntents = new ArrayList<PendingIntent>(messageCount);
261     final ArrayList<PendingIntent> sentIntents = new ArrayList<PendingIntent>(messageCount);
262     for (int i = 0; i < messageCount; i++) {
263         // Make pending intents different for each message part
264         final int partId = (messageCount <= 1 ? 0 : i + 1);
265         if (requireDeliveryReport && (i == (messageCount - 1))) {
266             // TODO we only care about the delivery status of the last part
267             // Shall we have better tracking of delivery status of all parts?
268             deliveryIntents.add((PendingIntent)getBroadcast(
269                 context,
270                 partId,
271                 getSendStatusIntent(context, SendStatusReceiver.MESSAGE_DELIVERED_ACTION,
272                     messageUri, partId, subId),
273                 0/*flag*/));
274         } else {
275             deliveryIntents.add(null);
276         }
277         sentIntents.add((PendingIntent)getBroadcast(
278             context,
279             partId,
280             getSendStatusIntent(context, SendStatusReceiver.MESSAGE_SENT_ACTION,
281                 messageUri, partId, subId),
282             0/*flag*/));
283     }
284     if (sSendMultiPartSmsAsSeparateMessages == null) {
285         sSendMultiPartSmsAsSeparateMessages = MmsConfig.get(subId)
286             .getSendMultiPartSmsAsSeparateMessages();
287     }

```

<https://android.googlesource.com/platform/packages/apps/Messaging/+/nougat-mr1-release/src/com/android/messaging/sms/SmsSender.java>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products

```

288     try {
289         if (sSendMultipartSmsAsSeparateMessages) {
290             // If multipart sms is not supported, send them as separate messages
291             for (int i = 0; i < messageCount; i++) {
292                 smsManager.sendTextMessage(dest,
293                     serviceCenter,
294                     messages.get(i),
295                     sentIntents.get(i),
296                     deliveryIntents.get(i));
297             }
298         } else {
299             smsManager.sendMultipartTextMessage(
300                 dest, serviceCenter, messages, sentIntents, deliveryIntents);
301         }
302     } catch (final Exception e) {
303         throw new SmsException("SmsSender: caught exception in sending " + e);
304     }
305 }

```

<https://android.googlesource.com/platform/packages/apps/Messaging/+/nougat-mr1-release/src/com/android/messaging/sms/SmsSender.java>

```

56 * Class that receives incoming SMS messages through android.provider.Telephony.SMS_RECEIVED
57 *
58 * This class serves two purposes:
59 * - Process phone verification SMS messages
60 * - Handle SMS messages when the user has enabled us to be the default SMS app (Pre-KLP)
61 */
62 public final class SmsReceiver extends BroadcastReceiver {
63     private static final String TAG = LogUtil.BUGLE_TAG;
64
65     private static ArrayList<Pattern> sIgnoreSmsPatterns;
66

```

<https://android.googlesource.com/platform/packages/apps/Messaging/+/nougat-mr1-release/src/com/android/messaging/receiver/SmsReceiver.java>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products

```

200 public static void deliverSmsMessages(final Context context, final int subId,
201     final int errorCode, final android.telephony.SmsMessage[] messages) {
202     final ContentValues messageValues =
203         MmsUtils.parseReceivedSmsMessage(context, messages, errorCode);
204
205     LogUtil.v(TAG, "SmsReceiver.deliverSmsMessages");
206
207     final long nowInMillis = System.currentTimeMillis();
208     final long receivedTimestamps = MmsUtils.getMessageDate(messages[0], nowInMillis);
209
210     messageValues.put(Sms.Inbox.DATE, receivedTimestamps);
211     // Default to unread and unseen for us but ReceiveSmsMessageAction will override
212     // seen for the telephony db.
213     messageValues.put(Sms.Inbox.READ, 0);
214     messageValues.put(Sms.Inbox.SEEN, 0);
215     if (OsUtil.isAtLeastL_MRI()) {
216         messageValues.put(Sms.SUBSCRIPTION_ID, subId);
217     }
218
219     if (messages[0].getMessageClass() == android.telephony.SmsMessage.MessageClass.CLASS_0 ||
220         DebugUtils.isDebugEnabled()) {
221         Factory.get().getUIIntents().launchClassZeroActivity(context, messageValues);
222     } else {
223         final ReceiveSmsMessageAction action = new ReceiveSmsMessageAction(messageValues);
224         action.start();
225     }
226 }

```

<https://android.googlesource.com/platform/packages/apps/Messaging/+nougat-mr1-release/src/com/android/messaging/receiver/SmsReceiver.java>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| US9467838 | Exemplary Supporting Evidence Regarding Accused Products |
|--|--|
| <pre> 228 @Override 229 public void onReceive(final Context context, final Intent intent) { 230 LogUtil.v(TAG, "SmsReceiver.onReceive " + intent); 231 // On KLP+ we only take delivery of SMS messages in SmsDeliverReceiver. 232 if (PhoneUtils.getDefault().isSmsEnabled()) { 233 final String action = intent.getAction(); 234 if (OsUtil.isSecondaryUser() && 235 (Telephony.Sms.Intents.SMS_RECEIVED_ACTION.equals(action) 236 // TODO: update this with the actual constant from Telephony 237 "android.provider.Telephony.MMS_DOWNLOADED".equals(action))) { 238 postNewMessageSecondaryUserNotification(); 239 } else if (!OsUtil.isAtLeastKLP()) { 240 deliverSmsIntent(context, intent); 241 } 242 } 243 } </pre> | <p>https://android.googlesource.com/platform/packages/apps/Messaging/+nougat-mr1-release/src/com/android/messaging/receiver/SmsReceiver.java</p> |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| US9467838 | Exemplary Supporting Evidence Regarding Accused Products |
|-----------|---|
| 52 | * Class that sends chat message via MMS. |
| 53 | * The interface emulates a blocking send similar to making an HTTP request. |
| 54 | * The interface emulates a blocking send similar to making an HTTP request. |
| 55 | */ |
| 56 | public class MmsSender { |
| 57 | private static final String TAG = LogUtil.BUGLE_TAG; |
| 58 | /** |
| 59 | * Send an MMS message. |
| 60 | * @param context Context |
| 61 | * @param messageUri The unique URI of the message for identifying it during sending |
| 62 | * @param sendReq The SendReq PDU of the message |
| 63 | * @throws MmsFailureException |
| 64 | */ |
| 65 | public static void sendMms(final Context context, final int subId, final Uri messageUri, |
| 66 | final SendReq sendReq, final Bundle sentIntentExtras) throws MmsFailureException { |
| 67 | sendMms(context, |
| 68 | subId, |
| 69 | messageUri, |
| 70 | null /* locationUrl */, |
| 71 | sendReq, |
| 72 | true /* responseImportant */, |
| 73 | sentIntentExtras); |
| 74 | } |
| 75 | https://android.googlesource.com/platform/packages/apps/Messaging/+/nougat-mr1- |
| 76 | release/src/com/android/messaging/sms/MmsSender.java |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| US9467838 | Exemplary Supporting Evidence Regarding Accused Products |
|-----------|---|
| 240 | * Download an MMS message. |
| 241 | * |
| 242 | * @param context Context |
| 243 | * @param contentLocation The url of the MMS message |
| 244 | * @throws MmsFailureException |
| 245 | * @throws InvalidHeaderValueException |
| 246 | */ |
| 247 | public static void downloadMms(final Context context, final int subId, |
| 248 | final String contentLocation, Bundle extras) throws MmsFailureException, |
| 249 | InvalidHeaderValueException { |
| 250 | final Uri requestUri = Uri.parse(contentLocation); |
| 251 | final Uri contentUri = MmsFileProvider.buildRawMmsUri(); |
| 252 | |
| 253 | final Intent downloadedIntent = new Intent(SendStatusReceiver.MMS_DOWNLOADED_ACTION, |
| 254 | requestUri, |
| 255 | context, |
| 256 | SendStatusReceiver.class); |
| 257 | downloadedIntent.putExtra(SendMessageAction.EXTRA_CONTENT_URI, contentUri); |
| 258 | if (extras != null) { |
| 259 | downloadedIntent.putExtras(extras); |
| 260 | } |
| 261 | final PendingIntent downloadedPendingIntent = PendingIntent.getBroadcast(context, |
| 262 | 0 /*request code*/, |
| 263 | downloadedIntent, |
| 264 | PendingIntent.FLAG_UPDATE_CURRENT); |
| 265 | |
| 266 | |
| 267 | MmsManager.downloadMultimediaMessage(subId, context, contentLocation, contentUri, |
| 268 | downloadedPendingIntent); |
| 269 | } |

<https://android.googlesource.com/platform/packages/apps/Messaging/+nougat-mr1-release/src/com/android/messaging/sms/MmsSender.java>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| US9467838 | Exemplary Supporting Evidence Regarding Accused Products |
|-----------|---|
| 97 | * Execute an MMS HTTP request, either a POST (sending) or a GET (downloading) |
| 98 | x |
| 99 | * @param urlString The request URL, for sending it is usually the MMSC, and for downloading |
| 100 | * it is the message URL |
| 101 | * @param pdu For POST (sending) only, the PDU to send |
| 102 | * @param method HTTP method, POST for sending and GET for downloading |
| 103 | * @param isProxySet Is there a proxy for the MMSC |
| 104 | * @param proxyHost The proxy host |
| 105 | * @param proxyPort The proxy port |
| 106 | * @param mmsConfig The MMS config to use |
| 107 | * @param userAgent The user agent header value |
| 108 | * @param uaProfUrl The UA Prof URL header value |
| 109 | * @return The HTTP response body |
| 110 | * @throws MmsHttpException For any failures |
| 111 | */ |
| 112 | public byte[] execute(String urlString, byte[] pdu, String method, boolean isProxySet, |
| 113 | String proxyHost, int proxyPort, Bundle mmsConfig, String userAgent, String uaProfUrl) |
| 114 | throws MmsHttpException { |
| 115 | Log.d(MmsService.TAG, "HTTP: " + method + " " + Utils.redactUrlForNonVerbose(urlString) |
| 116 | + (isProxySet ? (" proxy=" + proxyHost + " : " + proxyPort) : "") |
| 117 | + ", PDU size=" + (pdu != null ? pdu.length : 0)); |
| 118 | checkMethod(method); |
| 119 | URLConnection connection = null; |
| 120 | try { |
| 121 | Proxy proxy = Proxy.NO_PROXY; |
| 122 | if (isProxySet) { |
| 123 | proxy = new Proxy(Proxy.Type.HTTP, new InetSocketAddress(proxyHost, proxyPort)); |
| 124 | } |
| 125 | final URL url = new URL(urlString); |
| 126 | // Now get the connection |
| 127 | connection = (URLConnection) url.openConnection(proxy); |
| 128 | connection.setDoInput(true); |
| 129 | connection.setConnectTimeout(|
| 130 | mmsConfig.getInt(CarrierConfigValuesLoader.CONFIG_HTTP_SOCKET_TIMEOUT, |
| 131 | CarrierConfigValuesLoader.CONFIG_HTTP_SOCKET_TIMEOUT_DEFAULT)); |

<https://android.googlesource.com/platform/packages/apps/Messaging/+nougat-mr1-release/src/android/support/v7/mms/MmsHttpClient.java>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products

```

132 // ----- COMMON HEADERS -----
133 // Header: Accept
134 connection.setRequestProperty(HEADER_ACCEPT, HEADER_VALUE_ACCEPT);
135 // Header: Accept-Language
136 connection.setRequestProperty(
137     HEADER_ACCEPT_LANGUAGE, getLocale().getLanguage(Locale.getDefault()));
138 // Header: User-Agent
139 Log.i("MmsService.TAG", "HTTP: User-Agent=" + userAgent);
140 connection.setRequestProperty(HEADER_USER_AGENT, userAgent);
141 // Header: x-wap-profile
142 final String uaProfUrlTagName = mmsConfig.getString(
143     CarrierConfigValuesLoader.CONFIG_UA_PROF_TAG_NAME,
144     CarrierConfigValuesLoader.CONFIG_UA_PROF_TAG_NAME_DEFAULT);
145 if (uaProfUrl != null) {
146     Log.i("MmsService.TAG", "HTTP: UaProfUrl=" + uaProfUrl);
147     connection.setRequestProperty(uaProfUrlTagName, uaProfUrl);
148 }
149 // Add extra headers specified by mms_config.xml's httpparams
150 addExtraHeaders(connection, mmsConfig);
151 // Different stuff for GET and POST
152 if (METHOD_POST.equals(method)) {
153     if (pdu == null || pdu.length < 1) {
154         Log.e("MmsService.TAG", "HTTP: empty pdu");
155         throw new MmsHttpException(0/*statusCode*/, "Sending empty PDU");
156     }
157     connection.setOutput(true);
158     connection.setRequestMethod(METHOD_POST);
159     if (mmsConfig.getBoolean(
160         CarrierConfigValuesLoader.CONFIG_SUPPORT_HTTP_CHARSET_HEADER,
161         CarrierConfigValuesLoader.CONFIG_SUPPORT_HTTP_CHARSET_HEADER_DEFAULT)) {
162         connection.setRequestProperty(HEADER_CONTENT_TYPE,
163             HEADER_VALUE_CONTENT_TYPE_WITH_CHARSET);
164     } else {
165         connection.setRequestProperty(HEADER_CONTENT_TYPE,
166             HEADER_VALUE_CONTENT_TYPE_WITHOUT_CHARSET);

```

<https://android.googlesource.com/platform/packages/apps/Messaging/+nougat-mr1-release/src/android/support/v7/mms/MmsHttpClient.java>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products

```

167     }
168     if (Log.isLoggable(MmsService.TAG, Log.VERBOSE)) {
169         logHttpHeaders(connection.getRequestProperties());
170     }
171     connection.setFixedLengthStreamingMode(pdu.length);
172     // Sending request body
173     final OutputStream out =
174         new BufferedOutputStream(connection.getOutputStream());
175     out.write(pdu);
176     out.flush();
177     out.close();
178     } else if (METHOD_GET.equals(method)) {
179         if (Log.isLoggable(MmsService.TAG, Log.VERBOSE)) {
180             logHttpHeaders(connection.getRequestProperties());
181         }
182         connection.setRequestMethod(METHOD_GET);
183     }
184     // Get response
185     final int responseCode = connection.getResponseCode();
186     final String responseMessage = connection.getResponseMessage();
187     Log.d(MmsService.TAG, "HTTP: " + responseCode + " " + responseMessage);
188     if (Log.isLoggable(MmsService.TAG, Log.VERBOSE)) {
189         logHttpHeaders(connection.getHeaderFields());
190     }
191     if (responseCode / 100 != 2) {
192         throw new MmsHttpException(responseCode, responseMessage);
193     }
194     final InputStream in = new BufferedInputStream(connection.getInputStream());
195     final ByteArrayOutputStream byteOut = new ByteArrayOutputStream();
196     final byte[] buf = new byte[4096];
197     int count = 0;
198     while ((count = in.read(buf)) > 0) {
199         byteOut.write(buf, 0, count);
200     }
201     in.close();
202     final byte[] responseBody = byteOut.toByteArray();
203     Log.d(MmsService.TAG, "HTTP: response size="
204         + (responseBody != null ? responseBody.length : 0));
205     return responseBody;

```

<https://android.googlesource.com/platform/packages/apps/Messaging/+nougat-mr1-release/src/android/support/v7/mms/MmsHttpClient.java>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products

```

38 * Request to send an MMS
39 */
40 class SendRequest extends MmsRequest {
41     // Max send response PDU size in bytes (exceeding this may cause problem with
42     // system intent delivery).
43     private static final int MAX_SEND_RESPONSE_SIZE = 1000 * 1024;
44
45     private byte[] mPduData;
46
47     SendRequest(final String locationUrl, final Uri pduUri, final PendingIntent sentIntent) {
48         super(locationUrl, pduUri, sentIntent);
49     }
50
51     @Override
52     protected boolean loadRequest(final Context context, final Bundle mmsConfig) {
53         mPduData = readPduFromContentUri(
54             context,
55             pduUri,
56             mmsConfig.getInt(
57                 CarrierConfigValuesLoader.CONFIG_MAX_MESSAGE_SIZE,
58                 CarrierConfigValuesLoader.CONFIG_MAX_MESSAGE_SIZE_DEFAULT));
59         return (mPduData != null);
60     }
61
62     @Override
63     protected boolean transferResponse(final Context context, final Intent fillIn,
64         final byte[] response) {
65         // SendConf pdus are always small and can be included in the intent
66         if (response != null && fillIn != null) {
67             if (response.length > MAX_SEND_RESPONSE_SIZE) {
68                 // If the response PDU is too large, it won't be able to fit in
69                 // the PendingIntent to be transferred via system IPC.
70                 return false;
71             }
72             fillIn.putExtra(SmsManager.EXTRA_MMS_DATA, response);
73         }
74         return true;
75     }

```


Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| | |
|------------------|--|
| <p>US9467838</p> | <p>Exemplary Supporting Evidence Regarding Accused Products https://android.googlesource.com/platform/packages/apps/Messaging/+nougat-mr1-release/src/android/support/v7/mms/SendRequest.java</p> <pre>public static LocationRequest create ()</pre> <p>Create a location request with default parameters.</p> <p>Default parameters are for a block accuracy, slowly updated location. It can then be adjusted as required by the applications before passing to the <code>FusedLocationProviderApi</code>.</p> <p>Returns</p> <ul style="list-style-type: none"> • a new location request <p>https://developers.google.com/android/reference/com/google/android/gms/location/LocationRequest</p> |
|------------------|--|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| | |
|-----------|---|
| US9467838 | <p data-bbox="196 716 228 1556">Exemplary Supporting Evidence Regarding Accused Products</p> <p data-bbox="250 800 282 1535">public static final int PRIORITY_BALANCED_POWER_ACCURACY</p> <p data-bbox="315 898 347 1535">Used with <code>setPriority(int)</code> to request "block" level accuracy.</p> <p data-bbox="370 436 435 1535">Block level accuracy is considered to be about 100 meter accuracy. Using a coarse accuracy such as this often consumes less power.</p> <p data-bbox="456 1346 480 1535">Constant Value: 102</p> <p data-bbox="537 968 570 1535">public static final int PRIORITY_HIGH_ACCURACY</p> <p data-bbox="602 737 626 1535">Used with <code>setPriority(int)</code> to request the most accurate locations available.</p> <p data-bbox="659 1115 683 1535">This will return the finest location available.</p> <p data-bbox="708 1346 732 1535">Constant Value: 100</p> <p data-bbox="789 1020 821 1535">public static final int PRIORITY_LOW_POWER</p> <p data-bbox="854 919 878 1535">Used with <code>setPriority(int)</code> to request "city" level accuracy.</p> <p data-bbox="911 338 976 1535">City level accuracy is considered to be about 10km accuracy. Using a coarse accuracy such as this often consumes less power.</p> <p data-bbox="1000 1346 1024 1535">Constant Value: 104</p> <p data-bbox="1049 285 1073 1535">https://developers.google.com/android/reference/com/google/android/gms/location/LocationRequest</p> |
|-----------|---|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| | |
|-----------|---|
| US9467838 | <p data-bbox="185 296 228 1923">Exemplary Supporting Evidence Regarding Accused Products</p> <pre data-bbox="250 323 282 1541">public Task<Location> getLastLocation ()</pre> <p data-bbox="315 323 347 1541">Returns the best most recent location currently available.</p> <p data-bbox="363 323 428 1541">If a location is not available, which should happen very rarely, null will be returned. The best accuracy available while respecting the location permissions will be returned.</p> <p data-bbox="461 323 526 1541">This method provides a simplified way to get location. It is particularly well suited for applications that do not require an accurate location and that do not want to maintain extra logic for location updates.</p> <pre data-bbox="574 323 607 1541">public Task<LocationAvailability> getLocationAvailability ()</pre> <p data-bbox="639 323 704 1541">Returns the availability of location data. When <code>isLocationAvailable()</code> returns true, then the location returned by <code>getLastLocation()</code> will be reasonably up to date within the hints specified by the active <code>LocationRequest</code>s.</p> <p data-bbox="737 323 769 1541">If the client isn't connected to Google Play services and the request times out, null is returned.</p> <p data-bbox="802 323 867 1541">Note it's always possible for <code>getLastLocation()</code> to return null even when this method returns true (e.g. location settings were disabled between calls).</p> <p data-bbox="867 323 925 1541">https://developers.google.com/android/reference/com/google/android/gms/location/FusedLocationProviderClient</p> |
|-----------|---|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| | | | | | | | |
|------------------|--|----------------|---------------------------------------|-----------------|--|---------------|--|
| <p>US9467838</p> | <p>Exemplary Supporting Evidence Regarding Accused Products</p> <p>public Task<Void> requestLocationUpdates (LocationRequest request, LocationCallback callback, Looper looper)</p> <p>Requests location updates with a callback on the specified Looper thread.</p> <p>This method is suited for the foreground use cases. For background use cases, the <code>PendingIntent</code> version of the method is recommended, see <code>requestLocationUpdates(LocationRequest, PendingIntent)</code>.</p> <p>Any previous LocationRequests registered on this LocationListener will be replaced.</p> <p>This call will keep the Google Play services connection active, so make sure to call <code>removeLocationUpdates(LocationCallback)</code> when you no longer need it, otherwise you lose the benefits of the automatic connection management.</p> <p>Callbacks for <code>LocationCallback</code> will be made on the specified thread, which must already be a prepared looper thread.</p> <p>Parameters</p> <table border="1"> <tr> <td>request</td> <td>The location request for the updates.</td> </tr> <tr> <td>callback</td> <td>The callback for the location updates.</td> </tr> <tr> <td>looper</td> <td>The Looper object whose message queue will be used to implement the callback mechanism, or null to make callbacks on the calling thread.</td> </tr> </table> <p>https://developers.google.com/android/reference/com/google/android/gms/location/FusedLocationProviderClient</p> | request | The location request for the updates. | callback | The callback for the location updates. | looper | The Looper object whose message queue will be used to implement the callback mechanism, or null to make callbacks on the calling thread. |
| request | The location request for the updates. | | | | | | |
| callback | The callback for the location updates. | | | | | | |
| looper | The Looper object whose message queue will be used to implement the callback mechanism, or null to make callbacks on the calling thread. | | | | | | |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| | | | | | |
|-----------------------------|--|----------------------|---------------------------------------|-----------------------------|---|
| <p>US9467838</p> | <p>Exemplary Supporting Evidence Regarding Accused Products</p> <pre>public Task<Void> requestLocationUpdates (LocationRequest request, PendingIntent callbackIntent)</pre> <p>Requests location updates with a callback on the specified PendingIntent.</p> <p>This method is suited for the background use cases, more specifically for receiving location updates, even when the app has been killed by the system. In order to do so, use a <code>PendingIntent</code> for a started service. For foreground use cases, the <code>LocationCallback</code> version of the method is recommended, see <code>requestLocationUpdates(LocationRequest, LocationCallback, Looper)</code>.</p> <p>Any previously registered requests that have the same <code>PendingIntent</code> (as defined by <code>equals(Object)</code>) will be replaced by this request.</p> <p>Both <code>LocationResult</code> and <code>LocationAvailability</code> are sent to the given <code>PendingIntent</code>. You can extract data from an <code>Intent</code> using <code>hasResult(Intent)</code>, <code>extractResult(Intent)</code>, <code>hasLocationAvailability(Intent)</code>, and <code>extractLocationAvailability(Intent)</code>.</p> <p>Parameters</p> <table border="1"> <tr> <td><code>request</code></td> <td>The location request for the updates.</td> </tr> <tr> <td><code>callbackIntent</code></td> <td>A pending intent to be sent for each location update.</td> </tr> </table> <p>Returns</p> <ul style="list-style-type: none"> a Task for the call, check <code>isSuccessful()</code> to determine if it was successful. <p>https://developers.google.com/android/reference/com/google/android/gms/location/FusedLocationProviderClient</p> | <code>request</code> | The location request for the updates. | <code>callbackIntent</code> | A pending intent to be sent for each location update. |
| <code>request</code> | The location request for the updates. | | | | |
| <code>callbackIntent</code> | A pending intent to be sent for each location update. | | | | |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| | |
|------------------|--|
| <p>US9467838</p> | <p>Exemplary Supporting Evidence Regarding Accused Products</p> <p>public void onLocationAvailability (LocationAvailability locationAvailability)</p> <p>Called when there is a change in the availability of location data.</p> <p>When <code>isLocationAvailable()</code> returns <code>false</code> you can assume that location will not be returned in <code>onLocationResult(LocationResult)</code> until something changes in the device's settings or environment. Even when <code>isLocationAvailable()</code> returns <code>true</code> the <code>onLocationResult(LocationResult)</code> may not always be called regularly, however the device location is known and both the most recently delivered location and <code>getLastLocation(GoogleApiClient)</code> will be reasonably up to date given the hints specified by the active <code>LocationRequest</code> s.</p> <p>Parameters</p> <p>locationAvailability The current status of location availability.</p> <p>public void onLocationResult (LocationResult result)</p> <p>Called when device location information is available.</p> <p>The most recent location returned by <code>getLastLocation()</code> is not guaranteed to be immediately fresh, but will be reasonably up to date given the hints specified by the active <code>LocationRequest</code> s.</p> <p>Parameters</p> <p>result The latest location result available.</p> <p>https://developers.google.com/android/reference/com/google/android/gms/location/LocationCallback</p> <p>public abstract void onLocationChanged (Location location)</p> <p>Called when the location has changed.</p> <p>Parameters</p> <p>location The updated location.</p> |
|------------------|--|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| | |
|------------------|---|
| <p>US9467838</p> | <p>Exemplary Supporting Evidence Regarding Accused Products https://developers.google.com/android/reference/com/google/android/gms/location/LocationListener</p> <p>Public Constructors</p> <p><code>public MapView (Context context)</code></p> <p><code>public MapView (Context context, AttributeSet attrs)</code></p> <p><code>public MapView (Context context, AttributeSet attrs, int defStyleAttr)</code></p> <p><code>public MapView (Context context, GoogleMapOptions options)</code></p> <p>https://developers.google.com/android/reference/com/google/android/gms/maps/MapView</p> |
|------------------|---|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| | |
|---|---|
| <p>US9467838</p> | <p>Exemplary Supporting Evidence Regarding Accused Products</p> |
| <p>[IG] presenting, via the interactive display of the first device, a second georeferenced map and a second set of one or more user-selectable symbols corresponding to a second set of one or more devices, wherein the second set of symbols</p> | <pre>public void getMapAsync (OnMapReadyCallback callback)</pre> <p>Returns a non-null instance of the <code>GoogleMap</code>, ready to be used.</p> <p>Note that:</p> <ul style="list-style-type: none"> • This method must be called from the main thread. • The callback will be executed in the main thread. • In the case where Google Play services is not installed on the user's device, the callback will not be triggered until the user installs it. • The <code>GoogleMap</code> object provided by the callback is non-null. <p>Parameters</p> <p>callback The callback object that will be triggered when the map is ready to be used.</p> <pre>public final void onCreate (Bundle savedInstanceState)</pre> <p>You must call this method from the parent Activity/Fragment's corresponding method. https://developers.google.com/android/reference/com/google/android/gms/maps/MapView</p> |
| <p>presenting, via the interactive display of the first device, a second georeferenced map and a second set of one or more user-selectable symbols corresponding to a second set of one or more devices, wherein the second set of symbols</p> | <p>ZTE infringes directly and/or indirectly by performing, inducing others to perform, and/or contributing to the performance of: presenting, via the interactive display of the first device, a second georeferenced map and a second set of one or more user-selectable symbols corresponding to a second set of one or more of the second devices, wherein the second set of symbols are positioned on the second georeferenced map at respective positions corresponding to the locations of the second set of second devices, and wherein the second georeferenced map data relate positions on the second georeferenced map to spatial coordinates.</p> <p>Regarding Find My Device and Android Device Manager, the Accused Products present the user with a second and/or updated map display on the display. The second and/or updated map comprises a first symbol positioned on the map and corresponding to the first device. The second and/or updated map comprises one or more second symbols positioned on the map and corresponding to one or more second</p> |

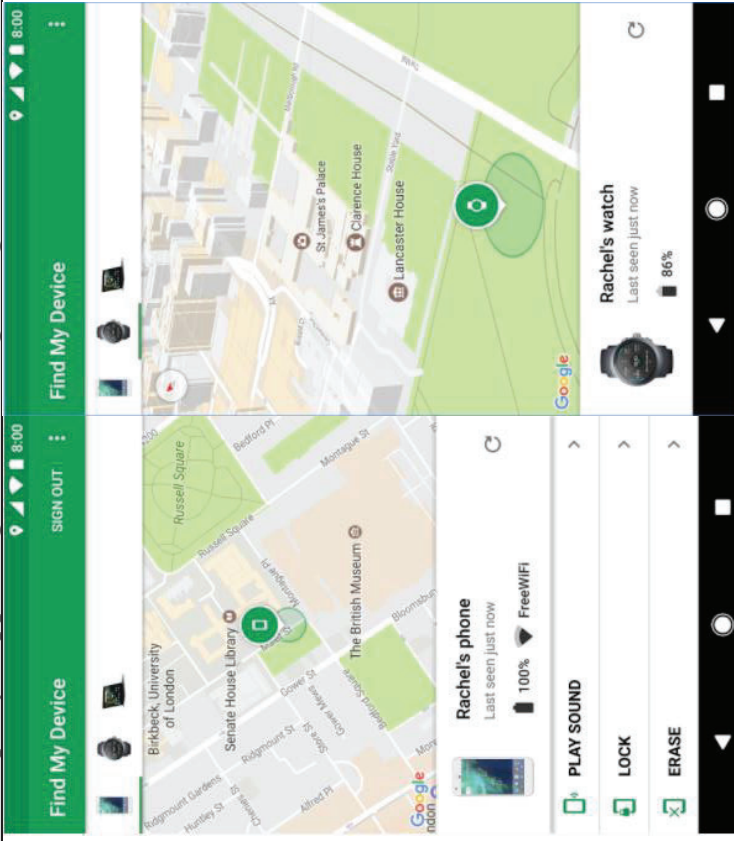
Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| | |
|---|---|
| <p>US9467838</p> <p>are positioned on the second georeferenced map at respective positions corresponding to the locations of the second set of second devices, and wherein the second georeferenced map data relate positions on the second georeferenced map to spatial coordinates;</p> | <p>Exemplary Supporting Evidence Regarding Accused Products</p> <p>devices. The map is interactive because the user may control the display of the map, e.g., pan, zoom, and/or effect change to the map in an otherwise interactive manner. The map is georeferenced for at least the reason that one or more symbols are associated with spatial locations, i.e., coordinates. The symbols are user-selectable because a user may touch the display to select the device associated with the symbol. For example, the user may input a touch selection directly on or near the portion of the display corresponding to the symbol's coordinates on the map to effect a selection of the device. In an alternative example, the user may input a selection event on the display that does not directly correspond to the symbol's coordinates, but which effects a selection of the device because the selection event is otherwise associated with the device corresponding to the symbol.</p> <p>Regarding Google Maps, Google Latitude, Google Plus, Google Hangouts, Google Messages, Google Allo, Google Duo, Google Chrome, and Android Messenger, the Accused Products present the user of a first device with a second and/or updated map display on the display. The second and/or updated map comprises a first symbol positioned on the map and corresponding to the first user or corresponding first device. The second and/or updated map comprises one or more second symbols positioned on the map and corresponding to one or more second users, contacts and/or corresponding second devices. The map is interactive because the user may control the display of the map, e.g., pan, zoom, and/or effect change to the map in an otherwise interactive manner. The map is georeferenced for at least the reason that one or more symbols are associated with spatial locations, i.e., coordinates. The symbols are user-selectable because a user may touch the display to select the user or device associated with the symbol. For example, the user may input a touch selection directly on or near the portion of the display corresponding to the symbol's coordinates on the map to effect a selection of the user or device. In an alternative example, the user may input a selection event on the display that does not directly correspond to the symbol's coordinates, but which effects a selection of the user or device because the selection event is otherwise associated with the user or device corresponding to the symbol.</p> <p>Exemplary Support for Find My Device:</p> |
|---|---|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products



<https://play.google.com/store/apps/details?id=com.google.android.apps.adm&hl=en>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

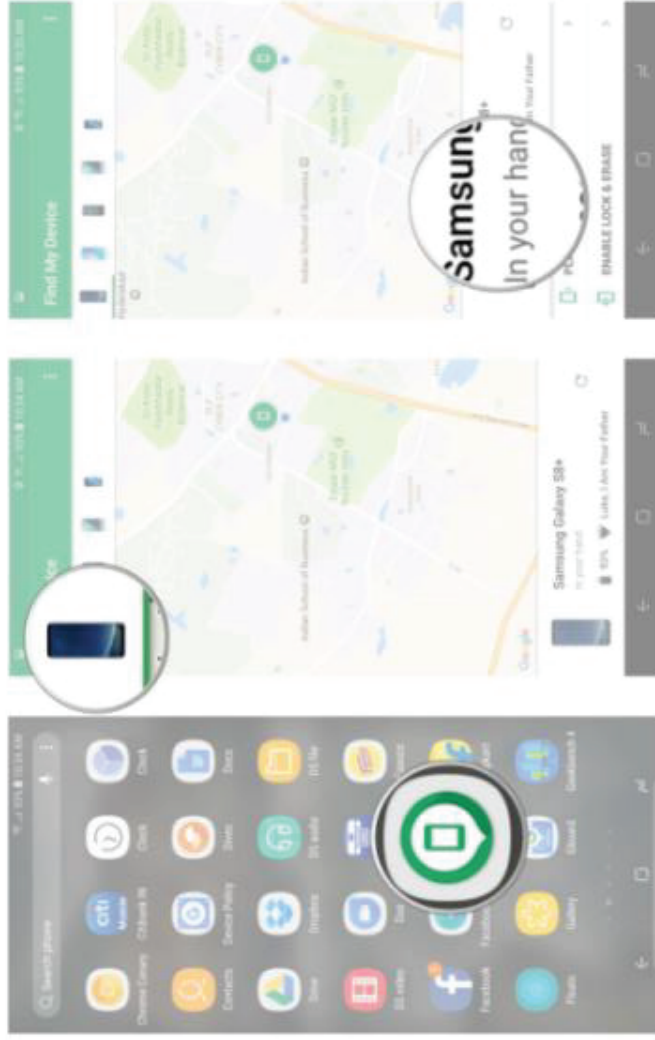
US9467838

Exemplary Supporting Evidence Regarding Accused Products

Once you're signed in to Find My Device, you'll see a map with your current location as well as the make and model of your phone, and two options — Play Sound, and Enable Lock & Erase. Hitting the latter option will allow you to start using the Lock and Erase functions.

If you've signed into more than one phone, you can select a particular device by browsing the list at the top of the screen.

1. Open **Find My Device** from your home screen or app drawer.
2. Select your phone from the **list of devices at the top of the screen**.
3. See if your phone is **discoverable**.



<https://www.androidcentral.com/find-my-device>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

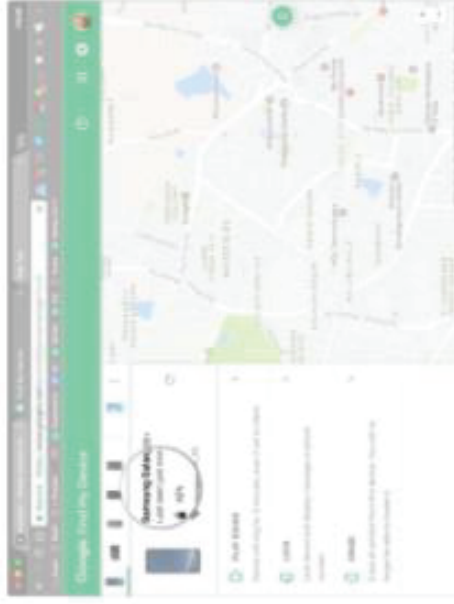
US9467838

Exemplary Supporting Evidence Regarding Accused Products

How to locate your phone over the internet

If you've lost your phone, you can remotely locate it through the Find My Device website. You'll need to sign in to the Google account that was used to set up Find My Device. It takes a few seconds, but the service should be able to track your phone. Alternatively, you can also do a Google search for "find my phone" to locate your handset.

1. Head to the **Find My Device website**.
2. Sign in to your **Google account**.
3. Check if your device is **visible**.



<https://www.androidcentral.com/find-my-device>

Exemplary Support for Google Maps:

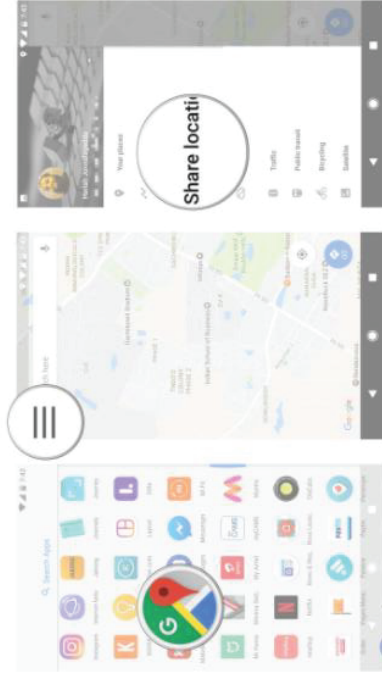
Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

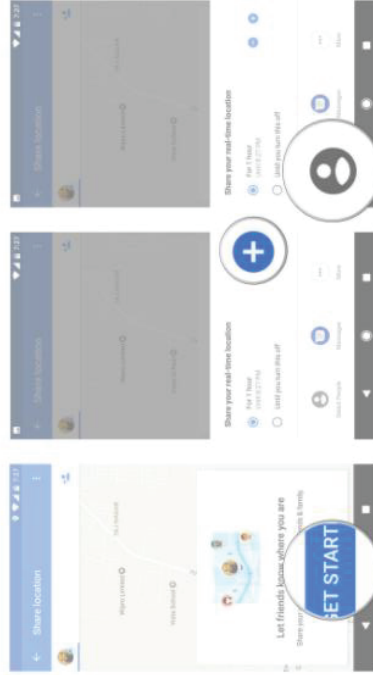
Exemplary Supporting Evidence Regarding Accused Products

How to share your location in Google Maps

1. Open Google Maps from the app drawer or the home screen.
2. Tap the hamburger menu (the three horizontal lines) on the top left corner of the screen.
3. Select Share location.



4. Tap Get Started.
5. Use the + icon to select a time period or select the Until you turn this off setting to share your location indefinitely.
6. Tap Select People.



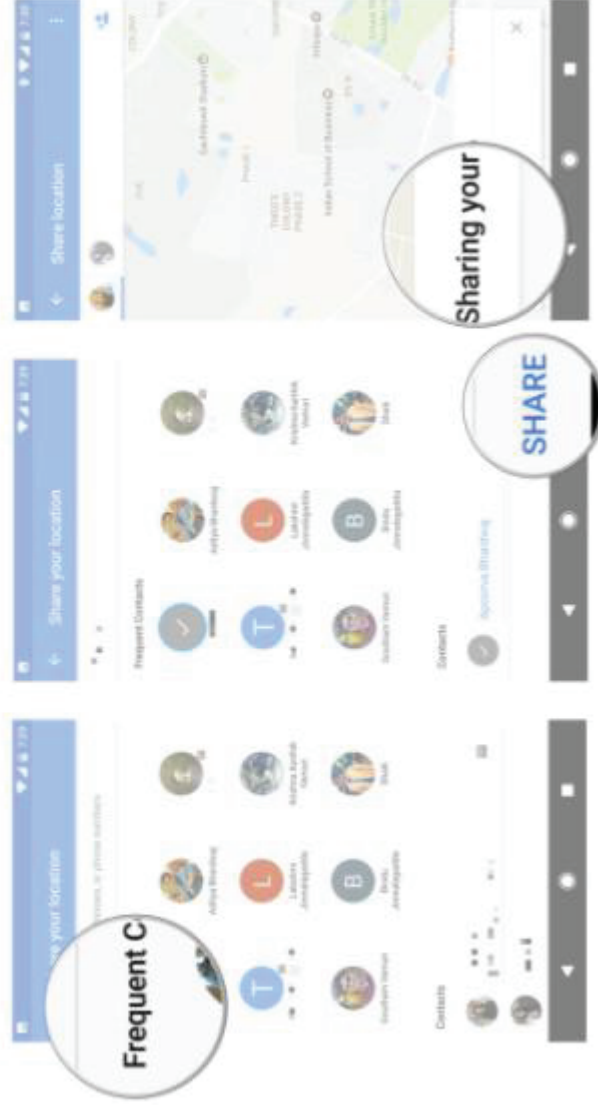
<https://www.androidcentral.com/how-share-location-google-maps>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products

- 7. You'll see a list of your frequent contacts at the top, along with a full list of contacts. Pick the contacts by tapping their name.
- 8. Once you've selected the contacts you want to share your location to, tap Share.
- 9. You'll see a message saying that the selected contact can view your location.



<https://www.androidcentral.com/how-share-location-google-maps>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

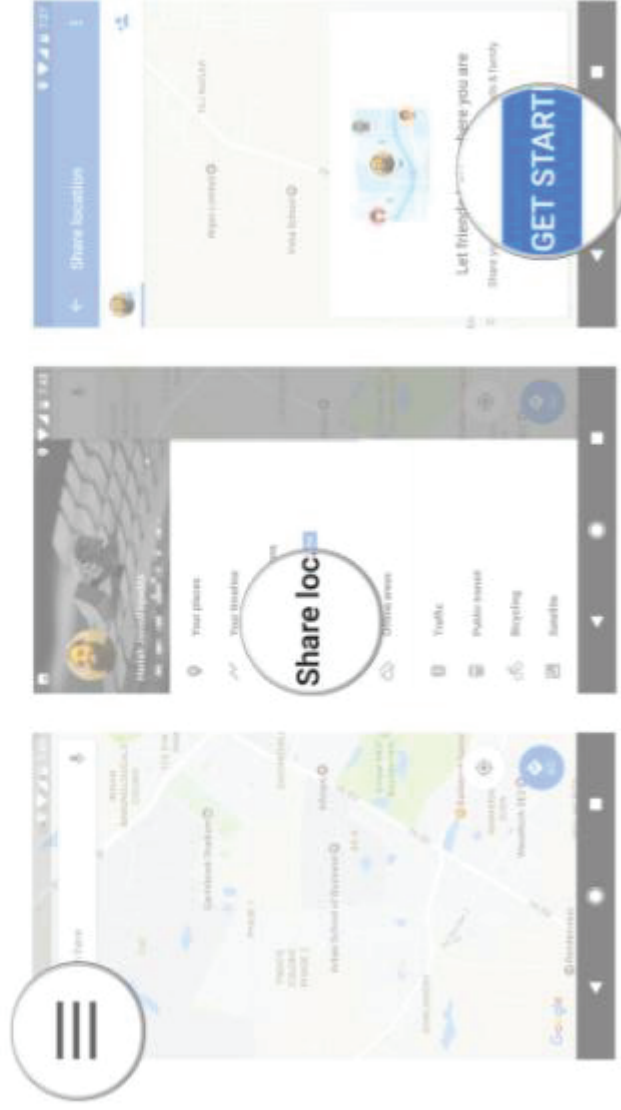
US9467838

Exemplary Supporting Evidence Regarding Accused Products

How to create a shareable link

You can also create a link and use it to share your location easily. Here's how to do it:

1. Tap the hamburger menu on the top left corner of the screen.
2. Select Share location.
3. Tap Get Started.



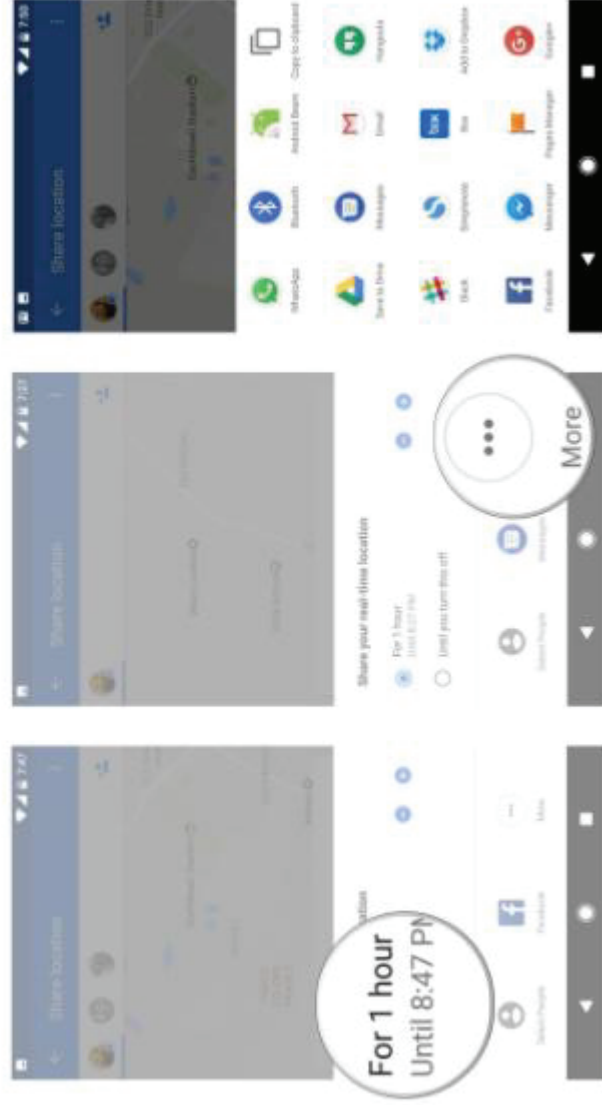
<https://www.androidcentral.com/how-share-location-google-maps>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products

- 4. Select the amount of time you want to share your location.
- 5. Tap **More**.
- 6. Select your **app of choice** to create and send a unique URL that broadcasts your current location. You can email it, send the link via Messenger, or even tweet it to the intended recipient.



<https://www.androidcentral.com/how-share-location-google-maps>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

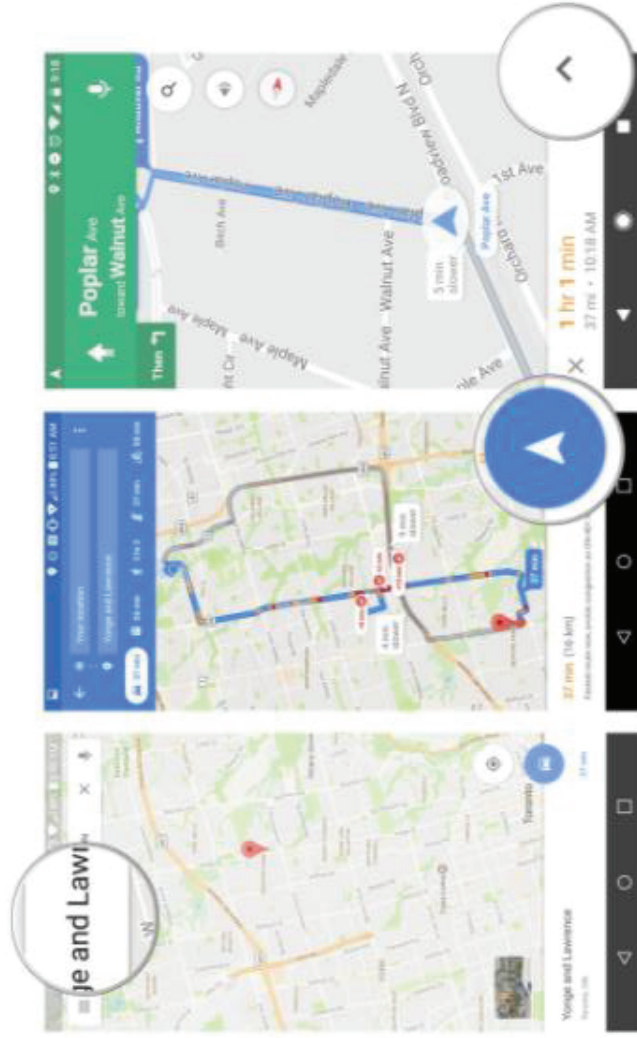
US9467838

Exemplary Supporting Evidence Regarding Accused Products

How to share your navigation directions while you walk, drive or transit

One of the best ways to share your location with someone is during a drive, walk or transit. If you're meeting a friend or family member somewhere, or navigating towards their home, Google Maps lets you share your location with them for the duration of the trip. It's magic!

1. In the **search bar** enter your destination.
2. Pick your navigation type (drive, transit, walk) and press the **blue navigate button**.
3. Tap the arrow next to the **time-to-destination** number at the bottom of the screen.



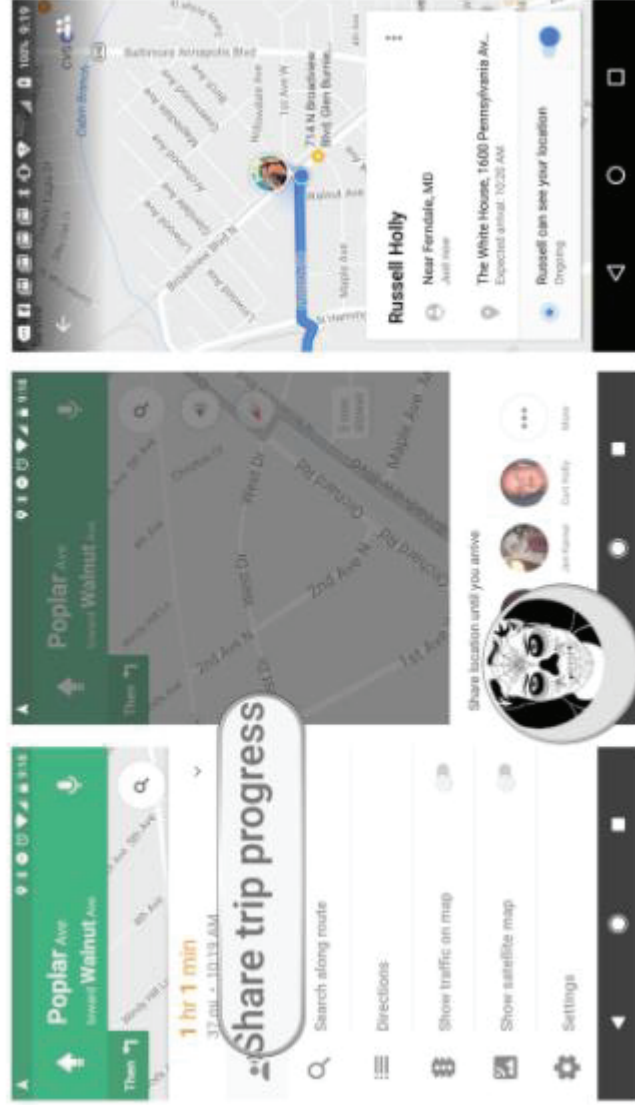
<https://www.androidcentral.com/how-share-location-google-maps>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products

- 4. Tap Share trip progress.
- 5. Choose one or more contacts to share trip progress.



You can also stop sharing your location with someone before a trip ends.
<https://www.androidcentral.com/how-share-location-google-maps>

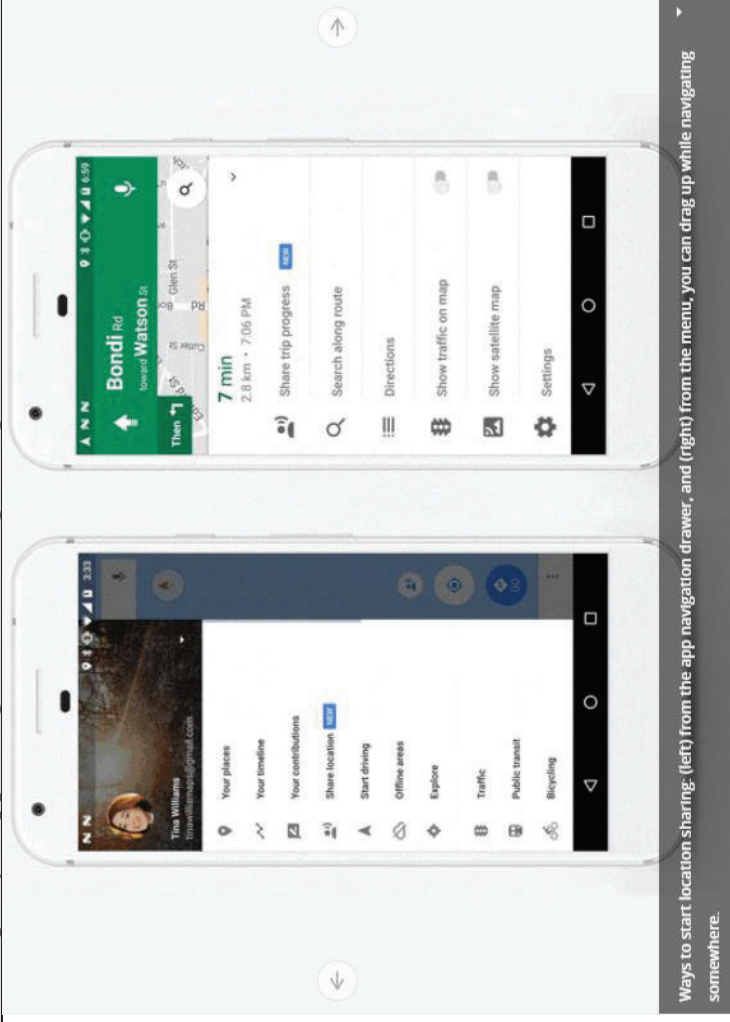
Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| | | |
|------------------|---|---|
| <p>US9467838</p> | <p>Exemplary Supporting Evidence Regarding Accused Products</p> | <p>1. Tap the arrow next to the time-to-destination number at the bottom of the screen.</p> <p>2. Tap Stop sharing.</p>  <p>That's it!</p> <p>Are you excited that location sharing is back in Google Maps? How often do you use the feature? https://www.androidcentral.com/how-share-location-google-maps</p> |
|------------------|---|---|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products



<https://arstechnica.com/gadgets/2017/03/location-sharing-finally-returns-to-google-maps/>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products

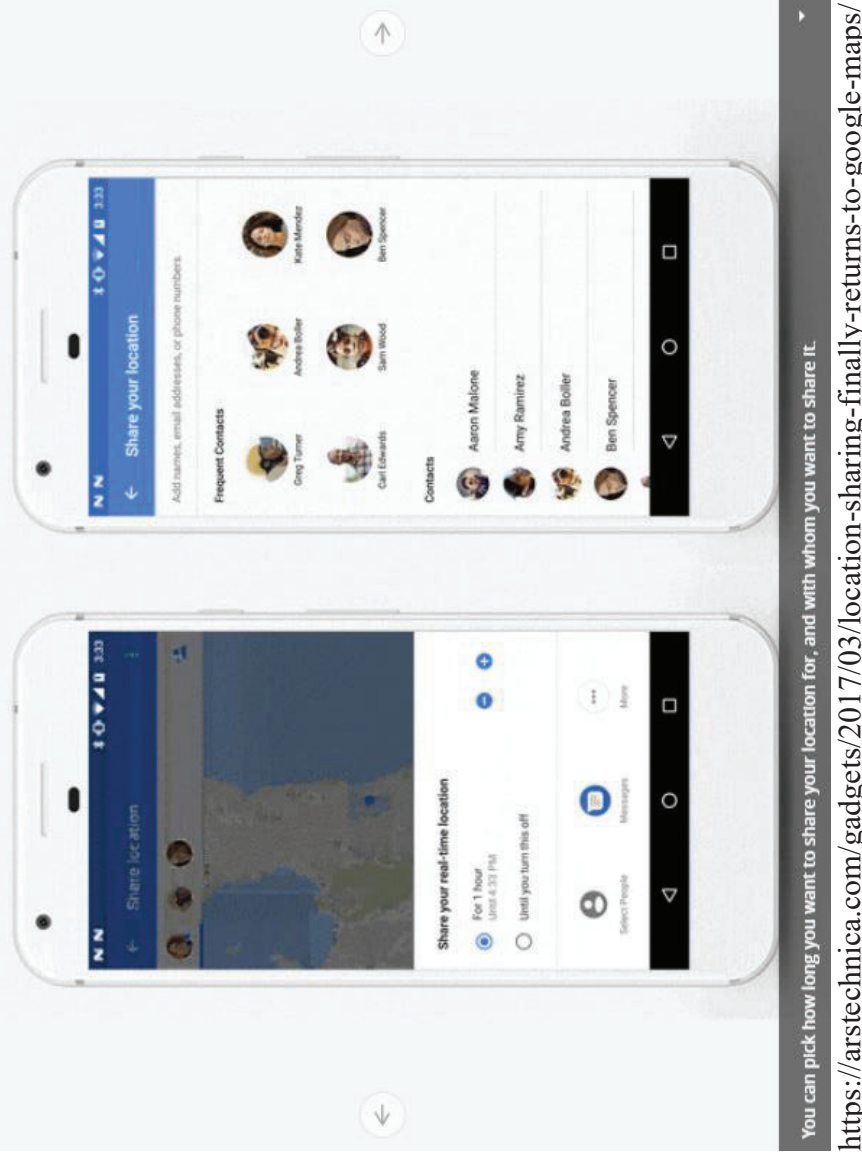
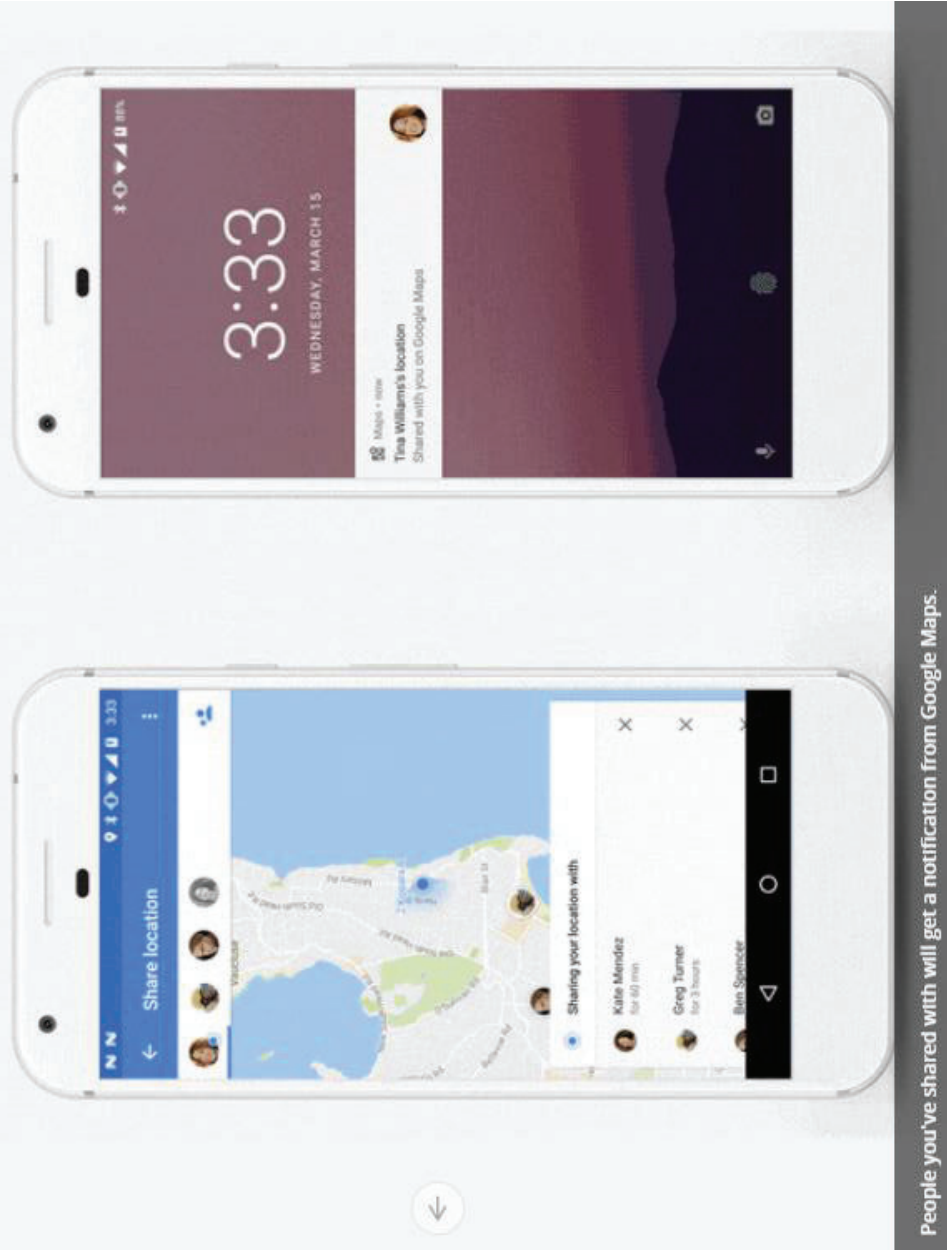


Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products

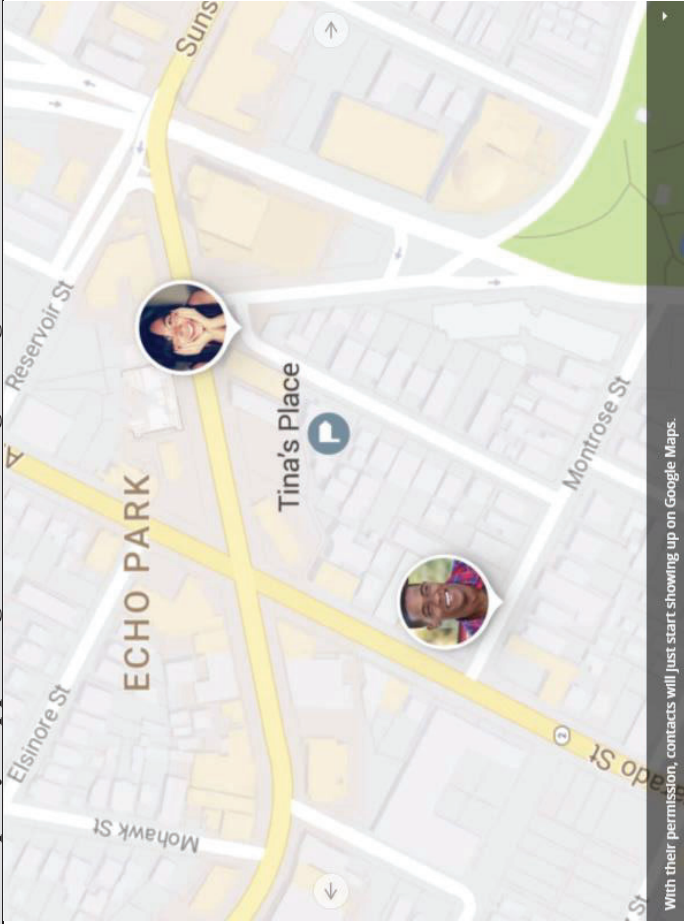


<https://arstechnica.com/gadgets/2017/03/location-sharing-finally-returns-to-google-maps/>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products



<https://arstechnica.com/gadgets/2017/03/location-sharing-finally-returns-to-google-maps/>

Exemplary Find My Device Screenshots:

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

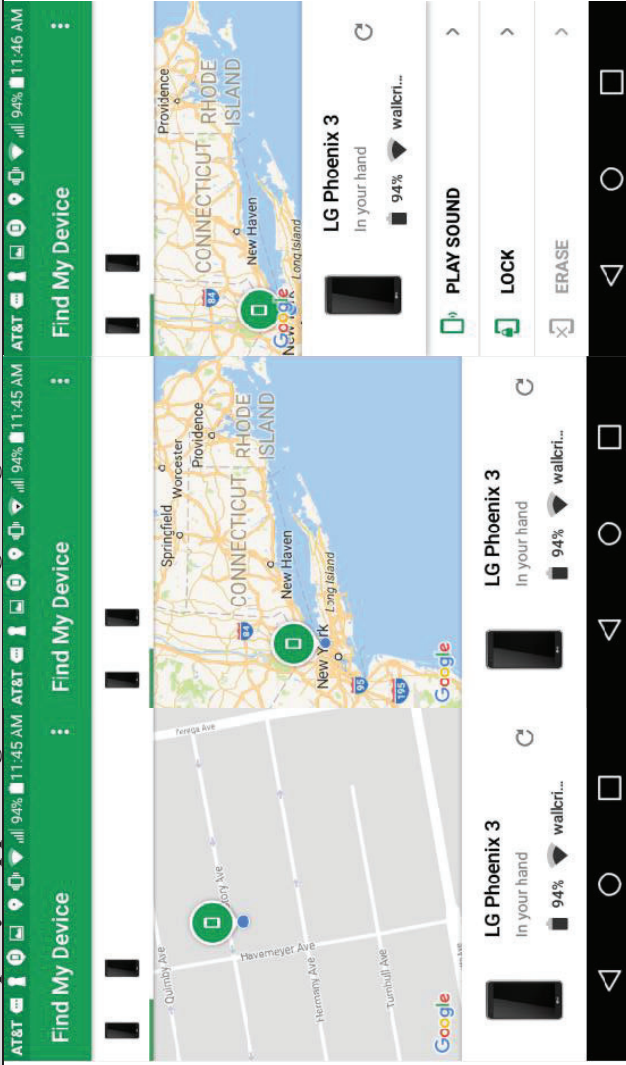
| | |
|-----------|--|
| US9467838 | <p data-bbox="191 709 228 1564">Exemplary Supporting Evidence Regarding Accused Products</p>  |
|-----------|--|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

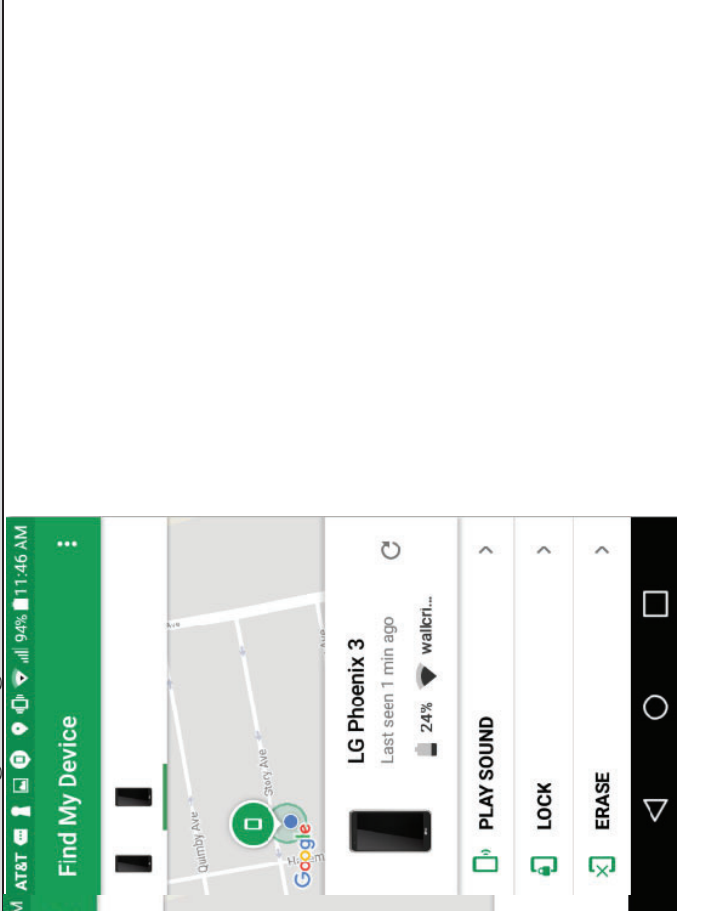
| | |
|------------------|---|
| <p>US9467838</p> | <p>Exemplary Supporting Evidence Regarding Accused Products</p>  |
|------------------|---|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

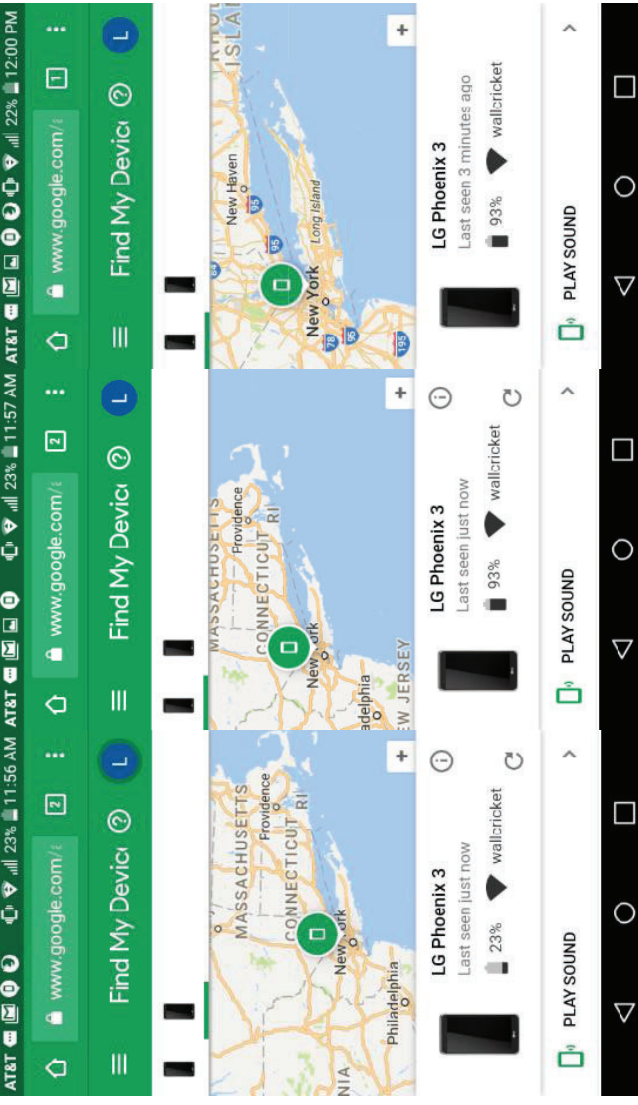
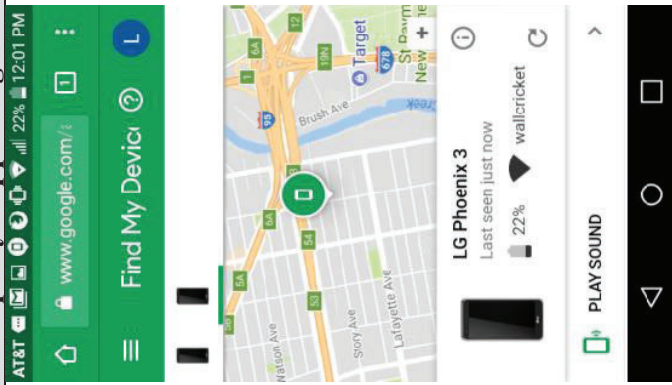
| | |
|-----------|--|
| US9467838 |  |
|-----------|--|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products



Exemplary Google Maps Screenshots:

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

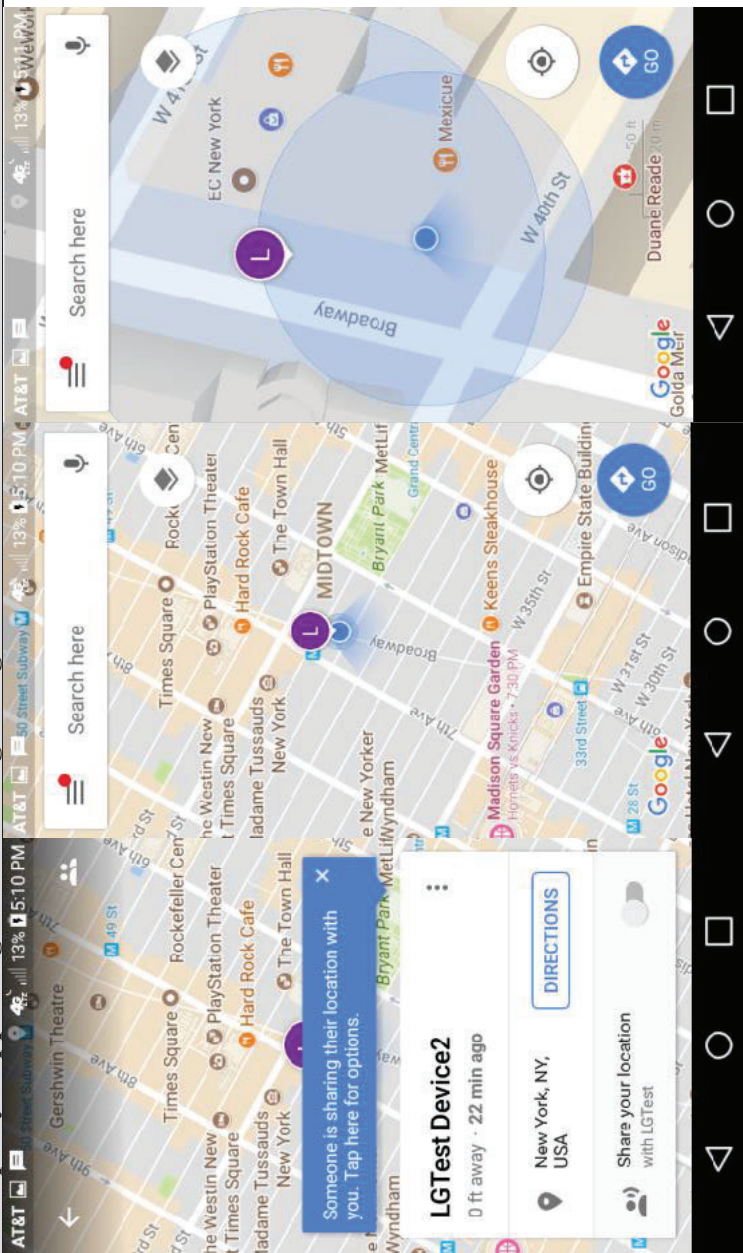
| | |
|-----------|--|
| US9467838 | <p data-bbox="191 709 227 1560">Exemplary Supporting Evidence Regarding Accused Products</p>  |
|-----------|--|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

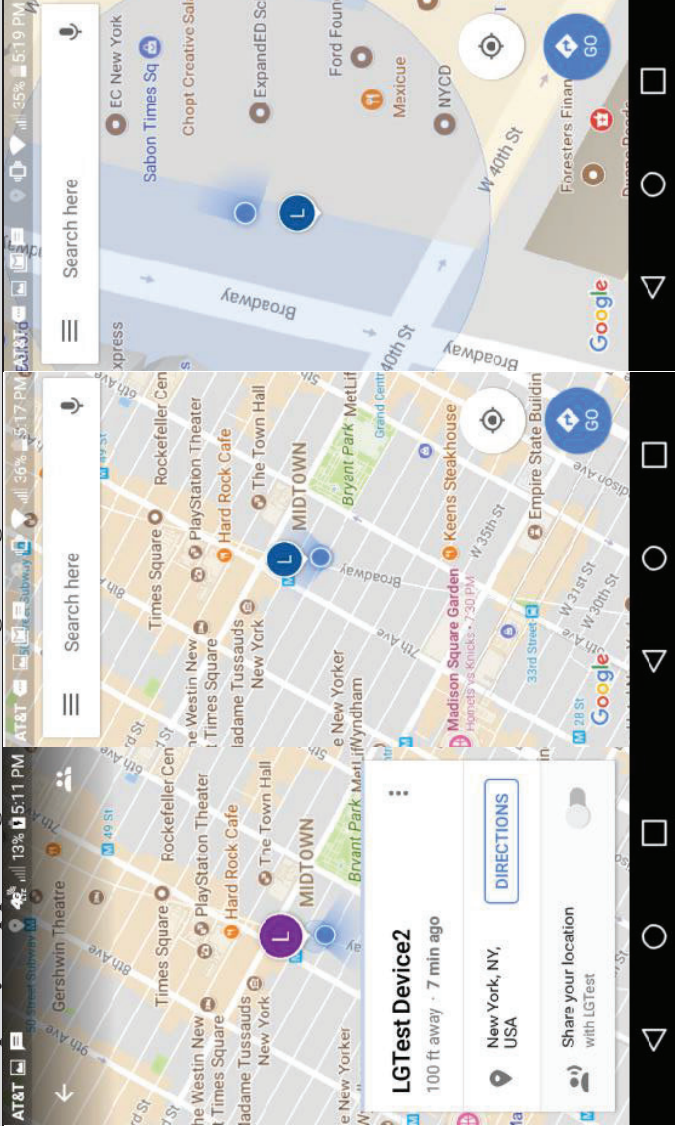
| | |
|------------------|--|
| <p>US9467838</p> | <p>Exemplary Supporting Evidence Regarding Accused Products</p>  |
|------------------|--|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

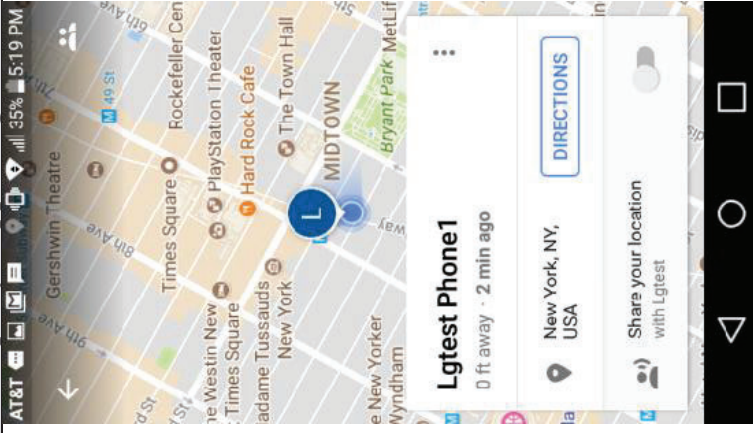
| | |
|-------------------------|--|
| <p>US9467838</p> | <p>Exemplary Supporting Evidence Regarding Accused Products</p> |
| |  <p>Exemplary Source Code:</p> <p>The above functionality is performed at least in part by the following publicly available source code and/or source code that invokes or is invoked by the following source code (or a substantially similar copy compiled and loaded onto the Accused Products by ZTE):</p> |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| | |
|-----------|---|
| US9467838 | <p data-bbox="196 716 228 1556">Exemplary Supporting Evidence Regarding Accused Products</p> <p data-bbox="250 1100 282 1535">public static <code>LocationRequest create ()</code></p> <p data-bbox="315 1052 347 1535">Create a location request with default parameters.</p> <p data-bbox="371 438 436 1535">Default parameters are for a block accuracy, slowly updated location. It can then be adjusted as required by the applications before passing to the <code>FusedLocationProviderApi</code>.</p> <p data-bbox="456 1446 480 1535">Returns</p> <ul data-bbox="501 1262 526 1520" style="list-style-type: none"> • a new location request <p data-bbox="542 281 574 1556">https://developers.google.com/android/reference/com/google/android/gms/location/LocationRequest</p> <p data-bbox="594 800 626 1535">public static final int PRIORITY_BALANCED_POWER_ACCURACY</p> <p data-bbox="659 900 683 1535">Used with <code>setPriority(int)</code> to request "block" level accuracy.</p> <p data-bbox="716 438 781 1535">Block level accuracy is considered to be about 100 meter accuracy. Using a coarse accuracy such as this often consumes less power.</p> <p data-bbox="797 1346 821 1535">Constant Value: 102</p> <p data-bbox="883 974 915 1535">public static final int PRIORITY_HIGH_ACCURACY</p> <p data-bbox="948 743 972 1535">Used with <code>setPriority(int)</code> to request the most accurate locations available.</p> <p data-bbox="1005 1115 1029 1535">This will return the finest location available.</p> <p data-bbox="1062 1346 1086 1535">Constant Value: 100</p> <p data-bbox="1138 1026 1170 1535">public static final int PRIORITY_LOW_POWER</p> <p data-bbox="1203 921 1227 1535">Used with <code>setPriority(int)</code> to request "city" level accuracy.</p> <p data-bbox="1260 344 1325 1535">City level accuracy is considered to be about 10km accuracy. Using a coarse accuracy such as this often consumes less power.</p> <p data-bbox="1341 1346 1365 1535">Constant Value: 104</p> <p data-bbox="1386 281 1419 1556">https://developers.google.com/android/reference/com/google/android/gms/location/LocationRequest</p> |
|-----------|---|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| | |
|-----------|---|
| US9467838 | <p data-bbox="185 296 228 1923">Exemplary Supporting Evidence Regarding Accused Products</p> <pre data-bbox="276 325 316 1543">public Task<Location> getLastLocation ()</pre> <p data-bbox="341 325 381 1543">Returns the best most recent location currently available.</p> <p data-bbox="406 325 462 1543">If a location is not available, which should happen very rarely, null will be returned. The best accuracy available while respecting the location permissions will be returned.</p> <p data-bbox="487 325 560 1543">This method provides a simplified way to get location. It is particularly well suited for applications that do not require an accurate location and that do not want to maintain extra logic for location updates.</p> <pre data-bbox="609 325 649 1543">public Task<LocationAvailability> getLocationAvailability ()</pre> <p data-bbox="673 325 747 1543">Returns the availability of location data. When <code>isLocationAvailable()</code> returns true, then the location returned by <code>getLastLocation()</code> will be reasonably up to date within the hints specified by the active <code>LocationRequest</code>s.</p> <p data-bbox="771 325 803 1543">If the client isn't connected to Google Play services and the request times out, null is returned.</p> <p data-bbox="828 325 893 1543">Note it's always possible for <code>getLastLocation()</code> to return null even when this method returns true (e.g. location settings were disabled between calls).</p> <p data-bbox="901 325 963 1543">https://developers.google.com/android/reference/com/google/android/gms/location/FusedLocationProviderClient</p> |
|-----------|---|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| | | | | | | | |
|-----------------------|---|----------------------|---------------------------------------|-----------------------|--|---------------------|--|
| <p>US9467838</p> | <p>Exemplary Supporting Evidence Regarding Accused Products</p> <pre>public Task<Void> requestLocationUpdates (LocationRequest request, LocationCallback callback, Looper looper)</pre> <p>Requests location updates with a callback on the specified Looper thread.</p> <p>This method is suited for the foreground use cases. For background use cases, the <code>PendingIntent</code> version of the method is recommended, see <code>requestLocationUpdates(LocationRequest, PendingIntent)</code>.</p> <p>Any previous <code>LocationRequests</code> registered on this <code>LocationListener</code> will be replaced.</p> <p>This call will keep the Google Play services connection active, so make sure to call <code>removeLocationUpdates(LocationCallback)</code> when you no longer need it, otherwise you lose the benefits of the automatic connection management.</p> <p>Callbacks for <code>LocationCallback</code> will be made on the specified thread, which must already be a prepared looper thread.</p> <p>Parameters</p> <table border="1"> <tr> <td><code>request</code></td> <td>The location request for the updates.</td> </tr> <tr> <td><code>callback</code></td> <td>The callback for the location updates.</td> </tr> <tr> <td><code>looper</code></td> <td>The Looper object whose message queue will be used to implement the callback mechanism, or null to make callbacks on the calling thread.</td> </tr> </table> <p>https://developers.google.com/android/reference/com/google/android/gms/location/FusedLocationProviderClient</p> | <code>request</code> | The location request for the updates. | <code>callback</code> | The callback for the location updates. | <code>looper</code> | The Looper object whose message queue will be used to implement the callback mechanism, or null to make callbacks on the calling thread. |
| <code>request</code> | The location request for the updates. | | | | | | |
| <code>callback</code> | The callback for the location updates. | | | | | | |
| <code>looper</code> | The Looper object whose message queue will be used to implement the callback mechanism, or null to make callbacks on the calling thread. | | | | | | |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| | | | | | |
|-----------------------------|--|----------------------|---------------------------------------|-----------------------------|---|
| US9467838 | <p>Exemplary Supporting Evidence Regarding Accused Products</p> <pre>public Task<Void> requestLocationUpdates (LocationRequest request, PendingIntent callbackIntent)</pre> <p>Requests location updates with a callback on the specified PendingIntent.</p> <p>This method is suited for the background use cases, more specifically for receiving location updates, even when the app has been killed by the system. In order to do so, use a <code>PendingIntent</code> for a started service. For foreground use cases, the <code>LocationCallback</code> version of the method is recommended, see <code>requestLocationUpdates(LocationRequest, LocationCallback, Looper)</code>.</p> <p>Any previously registered requests that have the same <code>PendingIntent</code> (as defined by <code>equals(Object)</code>) will be replaced by this request.</p> <p>Both <code>LocationResult</code> and <code>LocationAvailability</code> are sent to the given <code>PendingIntent</code>. You can extract data from an <code>Intent</code> using <code>hasResult(Intent)</code>, <code>extractResult(Intent)</code>, <code>hasLocationAvailability(Intent)</code>, and <code>extractLocationAvailability(Intent)</code>.</p> <p>Parameters</p> <table border="1"> <tr> <td><code>request</code></td> <td>The location request for the updates.</td> </tr> <tr> <td><code>callbackIntent</code></td> <td>A pending intent to be sent for each location update.</td> </tr> </table> <p>Returns</p> <ul style="list-style-type: none"> a Task for the call, check <code>isSuccessful()</code> to determine if it was successful. <p>https://developers.google.com/android/reference/com/google/android/gms/location/FusedLocationProviderClient</p> | <code>request</code> | The location request for the updates. | <code>callbackIntent</code> | A pending intent to be sent for each location update. |
| <code>request</code> | The location request for the updates. | | | | |
| <code>callbackIntent</code> | A pending intent to be sent for each location update. | | | | |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| | |
|------------------|--|
| <p>US9467838</p> | <p>Exemplary Supporting Evidence Regarding Accused Products</p> <p>public void onLocationAvailability (LocationAvailability locationAvailability)</p> <p>Called when there is a change in the availability of location data.</p> <p>When <code>isLocationAvailable()</code> returns <code>false</code> you can assume that location will not be returned in <code>onLocationResult(LocationResult)</code> until something changes in the device's settings or environment. Even when <code>isLocationAvailable()</code> returns <code>true</code> the <code>onLocationResult(LocationResult)</code> may not always be called regularly, however the device location is known and both the most recently delivered location and <code>getLastLocation(GoogleApiClient)</code> will be reasonably up to date given the hints specified by the active <code>LocationRequest</code> s.</p> <p>Parameters</p> <p>locationAvailability The current status of location availability.</p> <p>public void onLocationResult (LocationResult result)</p> <p>Called when device location information is available.</p> <p>The most recent location returned by <code>getLastLocation()</code> is not guaranteed to be immediately fresh, but will be reasonably up to date given the hints specified by the active <code>LocationRequest</code> s.</p> <p>Parameters</p> <p>result The latest location result available.</p> <p>https://developers.google.com/android/reference/com/google/android/gms/location/LocationCallback</p> <p>public abstract void onLocationChanged (Location location)</p> <p>Called when the location has changed.</p> <p>Parameters</p> <p>location The updated location.</p> |
|------------------|--|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| | |
|------------------|---|
| <p>US9467838</p> | <p>Exemplary Supporting Evidence Regarding Accused Products https://developers.google.com/android/reference/com/google/android/gms/location/LocationListener</p> <p>Public Constructors</p> <p><code>public MapView (Context context)</code></p> <p><code>public MapView (Context context, AttributeSet attrs)</code></p> <p><code>public MapView (Context context, AttributeSet attrs, int defStyleAttr)</code></p> <p><code>public MapView (Context context, GoogleMapOptions options)</code></p> <p>https://developers.google.com/android/reference/com/google/android/gms/maps/MapView</p> |
|------------------|---|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| | |
|--|--|
| <p>US9467838</p> | <p>Exemplary Supporting Evidence Regarding Accused Products</p> <pre>public void getMapAsync (OnMapReadyCallback callback)</pre> <p>Returns a non-null instance of the <code>GoogleMap</code>, ready to be used.</p> <p>Note that:</p> <ul style="list-style-type: none"> • This method must be called from the main thread. • The callback will be executed in the main thread. • In the case where Google Play services is not installed on the user's device, the callback will not be triggered until the user installs it. • The <code>GoogleMap</code> object provided by the callback is non-null. <p>Parameters</p> <p>callback The callback object that will be triggered when the map is ready to be used.</p> <pre>public final void onCreate (Bundle savedInstanceState)</pre> <p>You must call this method from the parent Activity/Fragment's corresponding method. https://developers.google.com/android/reference/com/google/android/gms/maps/MapView</p> |
| <p>[1H] and identifying user interaction with the interactive display selecting one or more of the second set of user-selectable symbols corresponding to one or more of the second devices and positioned on the second georeferenced map and</p> | <p>ZTE infringes directly and/or indirectly by performing, inducing others to perform, and/or contributing to the performance of: identifying user interaction with the interactive display selecting one or more of the second set of user-selectable symbols corresponding to one or more of the second devices and positioned on the second georeferenced map and user interaction with the display specifying an action and, based thereon, sending third data to the selected one or more second devices via the first server.</p> <p>Regarding Find My Device and Android Device Manager, the Accused Products are configured to allow a user of a first device to interact with the display, to select a device corresponding to a symbol, and to select an action to be performed, such as: play a sound on the second device, put the second device into a lost mode, and erase the second device. Selection of one of the aforementioned actions results in sending data from the first device to a server and then sending data from the server to the second device.</p> |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| | |
|---|--|
| <p>US9467838</p> <p>user interaction with the display specifying an action and, based thereon, sending third data to the selected one or more second devices via the first server.</p> | <p>Exemplary Supporting Evidence Regarding Accused Products</p> <p>Regarding Google Maps, Google Latitude, Google Plus, Google Hangouts, Google Messages, and Android Messenger, the Accused Products are configured to allow a user of the first device to interact with the display, to select a user, contact, or device, and to select an action to be performed, such as: sending a message, initiating a call, initiating a data conference, sharing a location, stop sharing a location, block a user from location sharing, sending a location, requesting a location, or sending other data. When an action is specified, data is sent from the first device to the second device via a server.</p> <p><u>Exemplary Support for Find My Device:</u></p> <p>Introducing Find My Device - the new and improved Android Device Manager. Find My Device helps you easily locate a lost Android device, and keeps your information safe and sound while you look.</p> <p>Locate your phone, tablet or watch. Misplaced your Android Wear device? No problem.</p> <p>Play a sound. Find My Device helps you track down your device when it's close by.</p> <p>Lock, erase or show a message. With Find My Device you can secure your device remotely and help someone get in touch.</p> <p>Permissions Notice</p> <ul style="list-style-type: none"> • Location: Needed to show your device's current location on the map. • Contacts: Needed to access the email address associated with your Google account. <p>https://play.google.com/store/apps/details?id=com.google.android.apps.adm&hl=en</p> |
|---|--|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

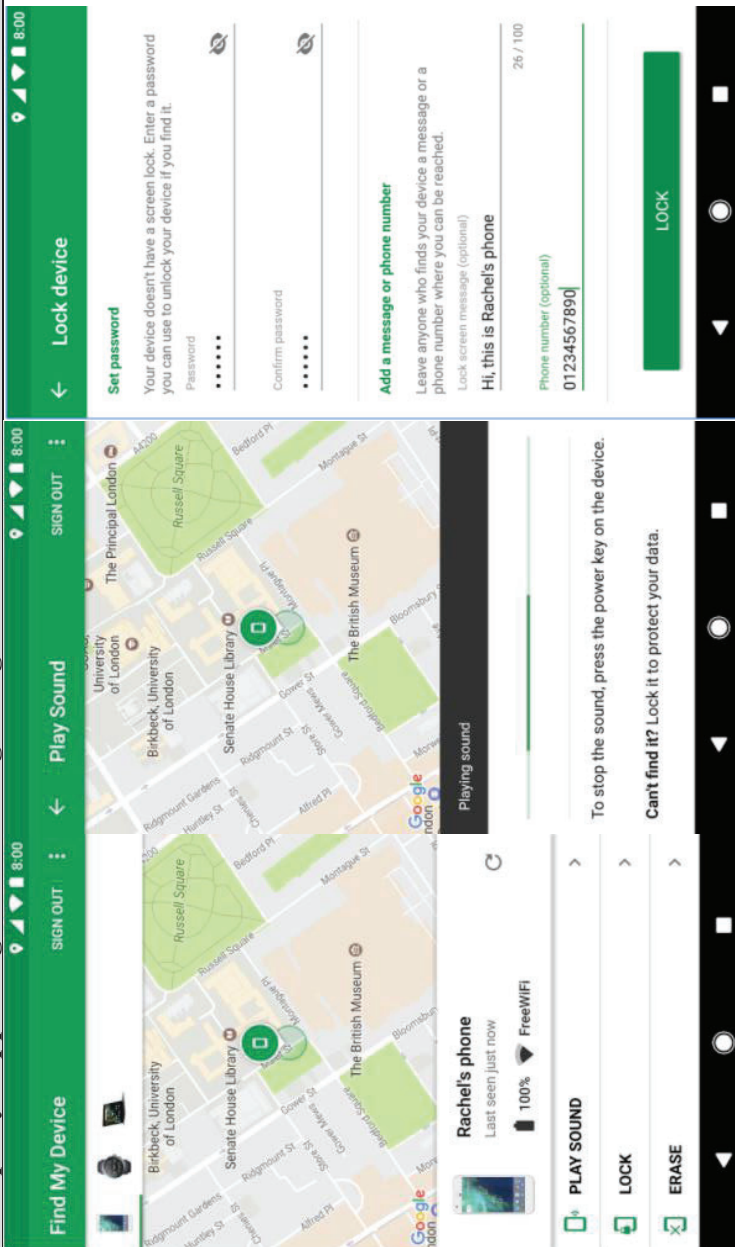
| | |
|------------------|--|
| <p>US9467838</p> | <p>Exemplary Supporting Evidence Regarding Accused Products</p>  <p>The image shows two screenshots of an Android phone interface. The top screenshot displays the 'Find My Device' screen with a map of London and a green 'LOCK' button. The bottom screenshot displays the 'Set password' screen with fields for password, confirm password, and an optional phone number, along with a green 'LOCK' button.</p> <p>https://play.google.com/store/apps/details?id=com.google.android.apps.adm&hl=en</p> |
|------------------|--|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

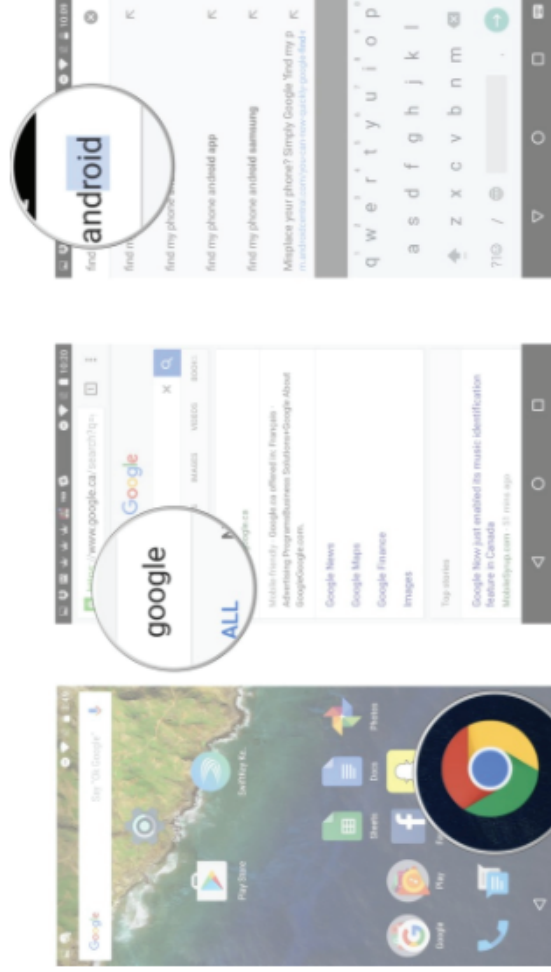
US9467838

Exemplary Supporting Evidence Regarding Accused Products

How to locate your phone with Google

Should you happen to lose your phone, you can locate its whereabouts by logging into your Google account from any computer or even from another phone.

1. Launch a web browser from a phone, tablet, or computer.
2. Navigate to Google if it is not your default search engine or home page.
3. Type find my phone android in the Google search bar.



4. Tap on Find My Device (usually the first option in the search).
5. Enter your email address and password just as though you were checking your email. If you have 2-step verification set up on your Google account (and you most certainly should), you'll need to complete that process as well.

<https://www.androidcentral.com/how-track-android-phone>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

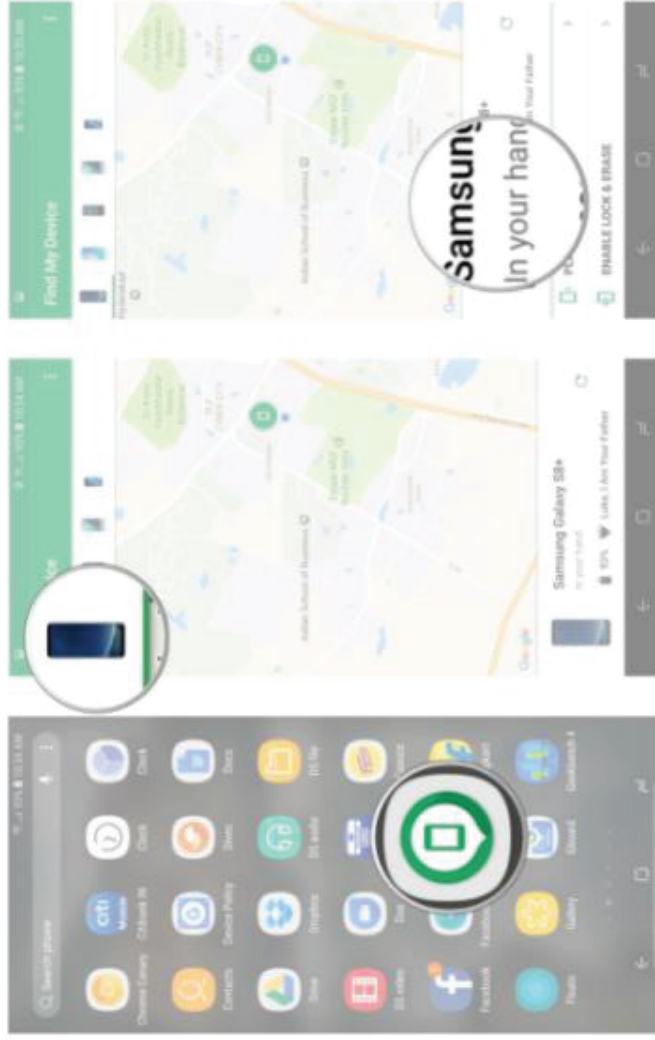
US9467838

Exemplary Supporting Evidence Regarding Accused Products

Once you're signed in to Find My Device, you'll see a map with your current location as well as the make and model of your phone, and two options — Play Sound, and Enable Lock & Erase. Hitting the latter option will allow you to start using the Lock and Erase functions.

If you've signed into more than one phone, you can select a particular device by browsing the list at the top of the screen.

1. Open **Find My Device** from your home screen or app drawer.
2. Select your phone from the **list of devices at the top of the screen**.
3. See if your phone is **discoverable**.

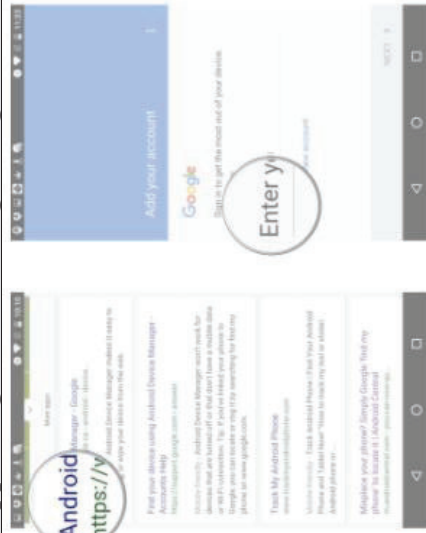


<https://www.androidcentral.com/find-my-device>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

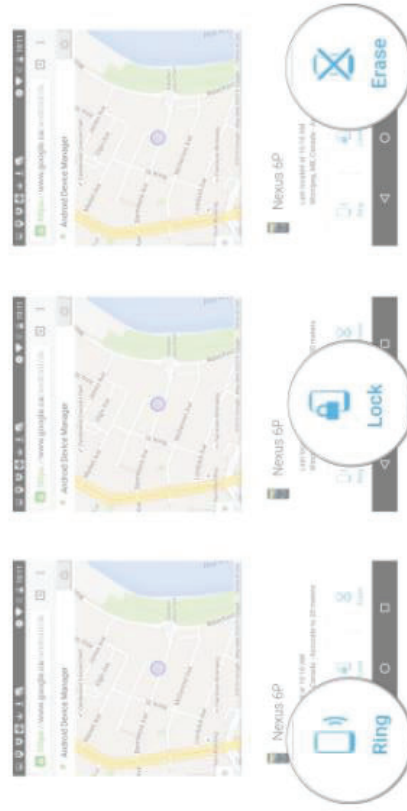
US9467838

Exemplary Supporting Evidence Regarding Accused Products



When your phone is located, you have three options to choose from:

- You can **Ring** your phone so that it makes noise (even if you had it on silent). This feature is helpful if the map indicates that the phone is within earshot and you simply can't see it.
- You can **Lock** your phone so that the finder can't access your home screen. This feature is most helpful if your phone wasn't previously secured with a passcode or a fingerprint sensor.
- You can **Erase** your phone. This is the best option if you know for certain that you aren't likely to retrieve your phone.



<https://www.androidcentral.com/how-track-android-phone>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

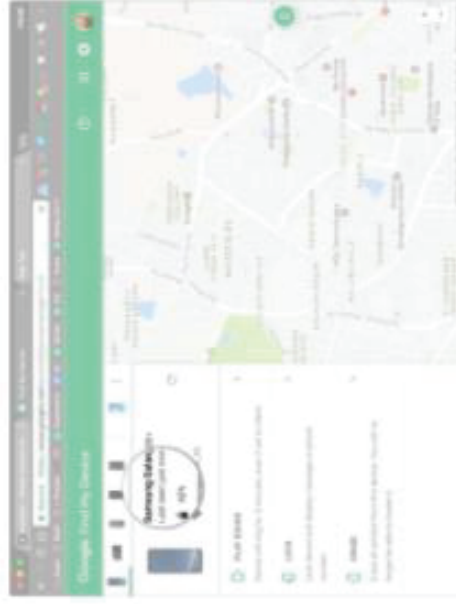
US9467838

Exemplary Supporting Evidence Regarding Accused Products

How to locate your phone over the internet

If you've lost your phone, you can remotely locate it through the Find My Device website. You'll need to sign in to the Google account that was used to set up Find My Device. It takes a few seconds, but the service should be able to track your phone. Alternatively, you can also do a Google search for "find my phone" to locate your handset.

1. Head to the **Find My Device website**.
2. Sign in to your **Google account**.
3. Check if your device is **visible**.



<https://www.androidcentral.com/find-my-device>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

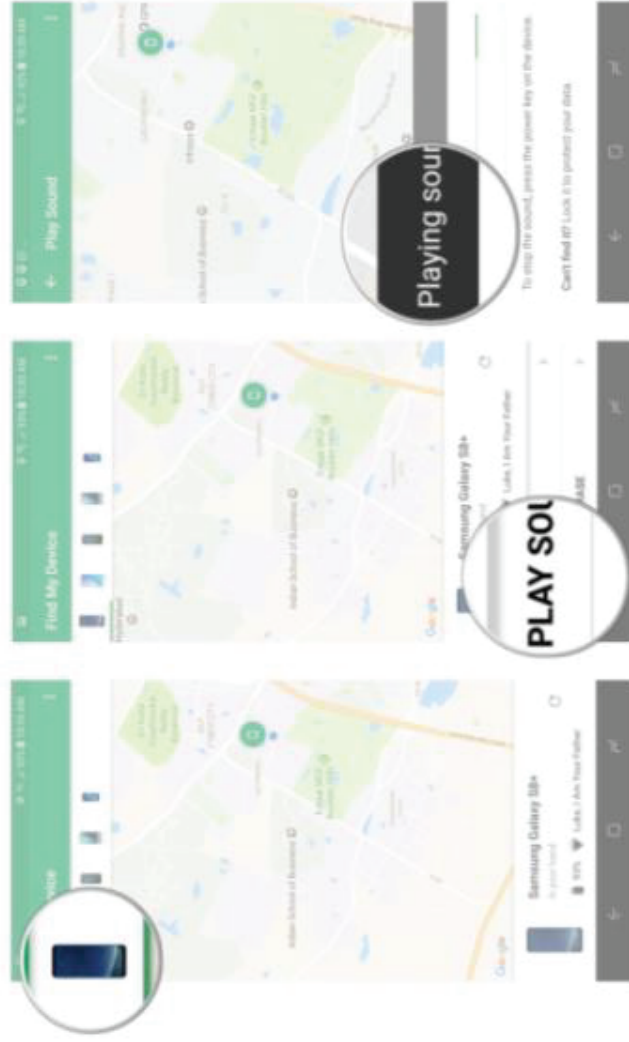
US9467838

Exemplary Supporting Evidence Regarding Accused Products

How to ring your phone with Find My Device

The best part about Find My Device is that it is easily accessible. If you need to locate your phone, just head to the website or log in to the service from another phone. Once you sign in to Find My Device and locate your device, you can use the **Play Sound** option, which plays a loud tone on your phone continuously at full volume for five minutes even if you turned the ringer off. Once you find your phone, you can hit the power button to stop the ringing.

1. Locate your phone on Find My Device.
2. Tap **Play Sound**.
3. Your device will start ringing. You can hit the power button to stop the sound.



<https://www.androidcentral.com/find-my-device>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| | |
|------------------|--|
| <p>US9467838</p> | <p>Exemplary Supporting Evidence Regarding Accused Products</p> <h2>How to lock your phone with Find My Device</h2> <p>There's also a Lock option that lets you set a new password to unlock the phone. You can also display a message over the lock screen and add a button to call back your number so that anyone that comes across your phone can easily get in touch with you.</p> <ol style="list-style-type: none"> 1. Locate your phone on Find My Device. 2. Tap Lock. 3. Enter a message and phone number to display on the lock screen and tap Lock. <p>https://www.androidcentral.com/find-my-device</p> |
|------------------|--|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

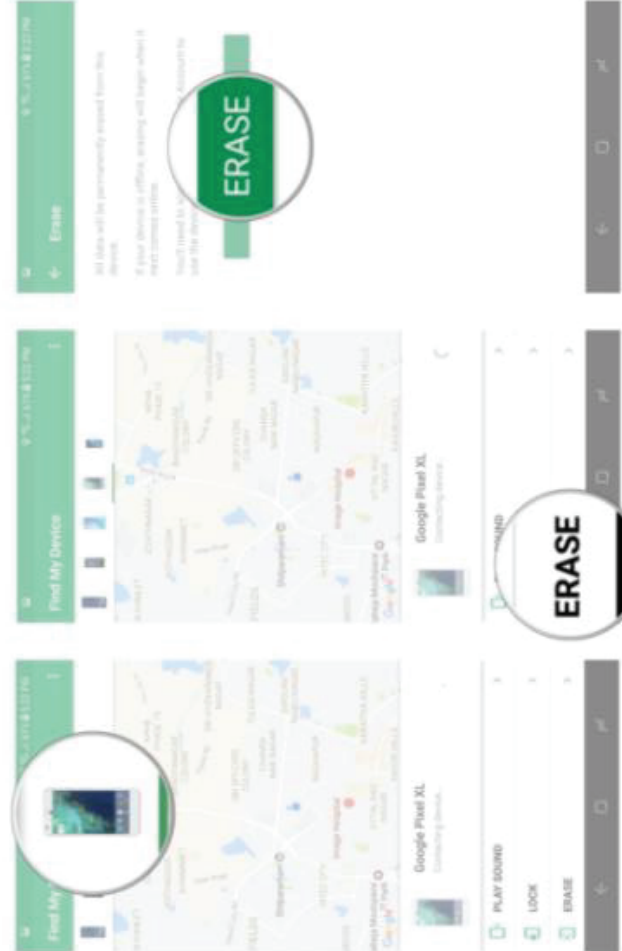
| | |
|------------------|---|
| <p>US9467838</p> | <p>Exemplary Supporting Evidence Regarding Accused Products</p> <p>How to erase your lost phone's data remotely</p> <p>if you're certain that you're not going to see your phone again, there is the nuclear option of erasing the data remotely. Selecting the Erase option deletes all the data on your phone. The service also deletes data from a connected SD card, but there is a chance that it may not be able to, based on the manufacturer and Android platform version. Even if your phone is switched off when you send the Erase command, the factory reset process will be initiated as soon as it goes online.</p> <ol style="list-style-type: none"> 1. Locate your phone on Find My Device. 2. Tap Erase. 3. Confirm deletion of data by hitting the Erase button.  <p>https://www.androidcentral.com/find-my-device</p> <p>Exemplary Support for Google Maps:</p> |
|------------------|---|




Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838





Exemplary Supporting Evidence Regarding Accused Products

COMPUTER ANDROID IPHONE & IPAD


If they have a Google Account

1. If you haven't already, add their Gmail address to your Google Contacts [\[\]](#).
2. On your Android phone or tablet, open the Google Maps app  and sign in. [Learn how to sign in.](#)
3. Tap Menu  > **Location sharing** > Add People .
4. Choose how long you want to share your location.
5. Tap **Select People**.
 - If you're asked about your contacts, give Google Maps access.
6. Choose who you want to share with.
7. Tap **Share**.

If they don't have a Google Account

1. On your Android phone or tablet, open the Google Maps app  and sign in. [Learn how to sign in.](#)
2. Tap Menu  > **Location sharing** > Add People .
3. Tap More  > **Copy to clipboard**. People with this link can see your location for as long as you choose, up to 72 hours.

Share using another app

You can also share through messaging apps. Tap More  > select an app.

Stop sharing

1. Open the Google Maps app .
2. Tap Menu  > **Location sharing**.
3. Next to the person with whom you want to stop sharing, tap Remove .

<https://support.google.com/maps/answer/7326816?co=GENIE.Platform%3DAndroid&oco=1>




Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products

Share your E.T.A

After you start your drive, you can share your destination, estimated arrival time, and where you are on the route.

1. Open the Google Maps app .
 2. Set a driving destination. Learn how to navigate to a place.
 3. After you start navigation, tap More  > **Share trip progress**.
 4. Choose a person from the list.
 5. Tap **Share**.
 6. Location Sharing will stop when you reach your destination or stop navigating.
- To stop sharing before you arrive, tap More  > **Stop sharing**.

See where someone is

If someone shares their location with you, you can see them on the map.

1. Open the Google Maps app .
 2. Tap Menu  > **Location sharing**.
 3. Choose someone.
- To see an updated location, tap on a friend's icon > More  > **Refresh**.

Stop seeing someone's location

1. Open the Google Maps app .
2. On the map, tap their icon.
3. At the bottom, tap More .
4. To temporarily hide someone, tap **Hide from map**. You can stop hiding them at any time.

Note: You can stop someone's location from ever appearing on your map. Learn how to block another person's account.

<https://support.google.com/maps/answer/7326816?co=GENIE.Platform%3DAndroid&oco=1>




Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products

Hide or share lists

Note: You can't share starred places.

1. Open the Google Maps app .
2. Tap Menu  > **Your places** > **Saved**.
3. Next to the list you want to share, tap More  > choose an option:
 - **Hide/Show on your map:** Display or hide your saved places when looking at the map.
 - **Share list:** Allow others to see your saved list.
 - **Sharing options:** You can make your list public, private, or shared. To let anyone with the link see your list, tap **Shared**. To let anyone find and follow your list, tap **Public**.

Follow a list


if you follow a list made by someone else, their saved places will show up in Your Places. The places will also appear as suggested locations in Google Maps.

Follow a list using a link

1. Tap on the link you received to open it.
2. Tap **Follow**. This list will now be added to the group of lists you follow.
3. **Optional:** To unfollow a list someone shared with you, tap the list > **Following**.

See lists made by others

If a user has any Google Maps lists that were made public, you can follow them.

1. Tap on the name of a user whose list you want to follow.
2. Tap **Lists**.
3. Tap on the list you want to follow > **More**  > **Follow**.

https://support.google.com/maps/answer/7280933?hl=en&ref_topic=7301134&co=GENIE.Platform%3DA

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

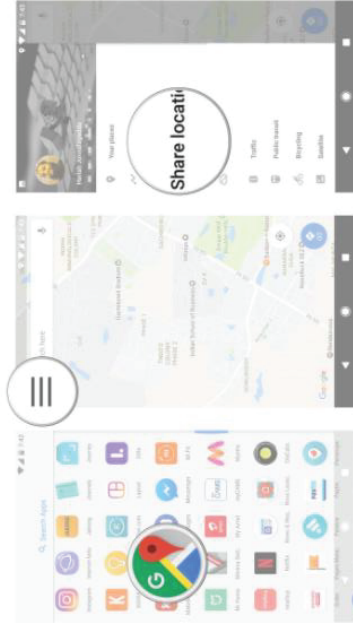
US9467838

Exemplary Supporting Evidence Regarding Accused Products

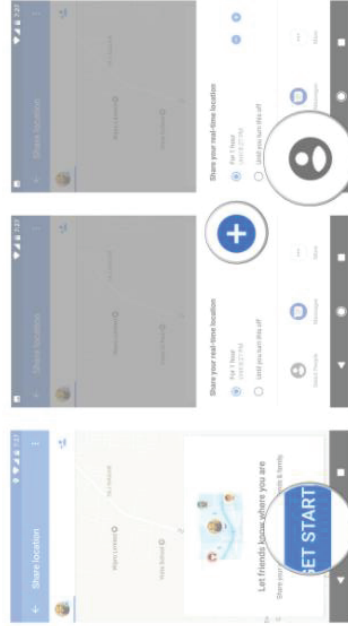
android&oco=1

How to share your location in Google Maps

1. Open Google Maps from the app drawer or the home screen.
2. Tap the hamburger menu (the three horizontal lines) on the top left corner of the screen.
3. Select Share location.



4. Tap Get Started.
5. Use the + icon to select a time period or select the Until you turn this off setting to share your location indefinitely.
6. Tap Select People.



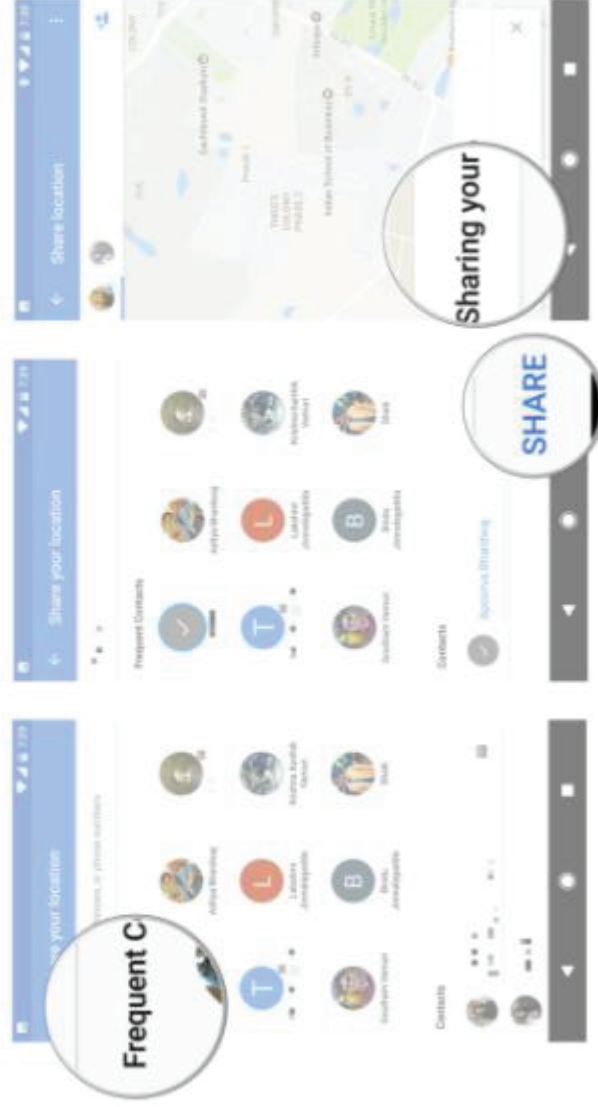
<https://www.androidcentral.com/how-share-location-google-maps>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products

- 7. You'll see a list of your frequent contacts at the top, along with a full list of contacts. Pick the contacts by tapping their name.
- 8. Once you've selected the contacts you want to share your location to, tap Share.
- 9. You'll see a message saying that the selected contact can view your location.



<https://www.androidcentral.com/how-share-location-google-maps>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

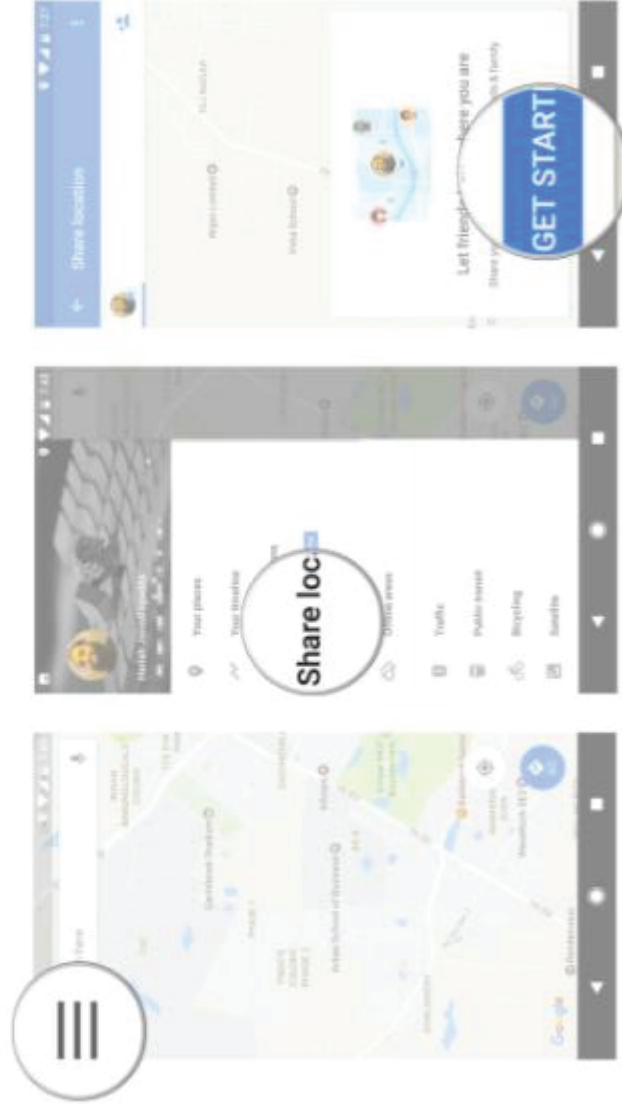
US9467838

Exemplary Supporting Evidence Regarding Accused Products

How to create a shareable link

You can also create a link and use it to share your location easily. Here's how to do it:

1. Tap the hamburger menu on the top left corner of the screen.
2. Select Share location.
3. Tap Get Started.



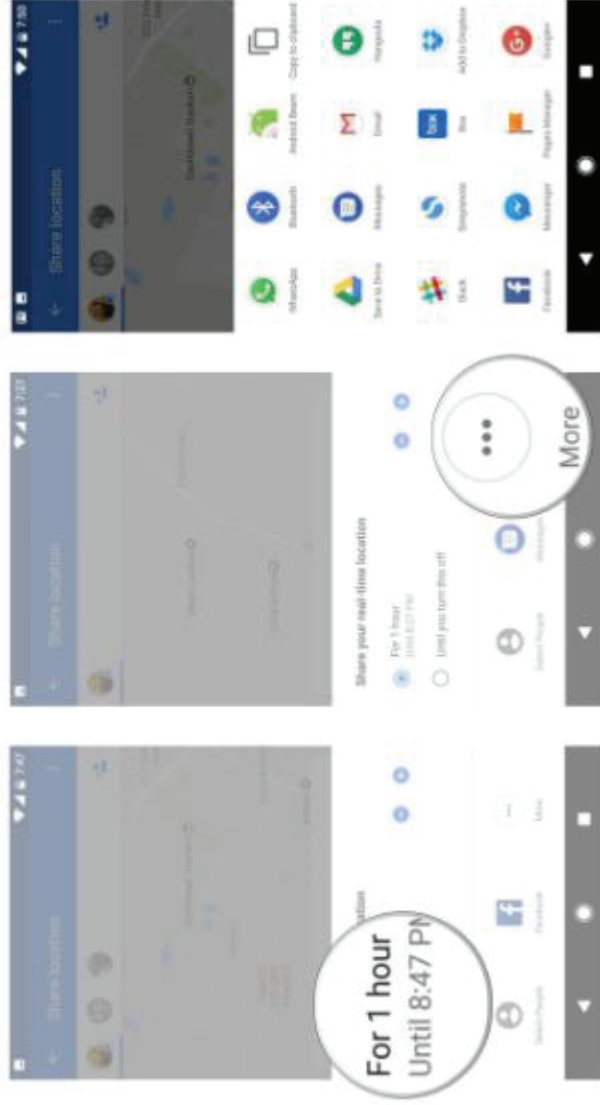
<https://www.androidcentral.com/how-share-location-google-maps>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products

- 4. Select the amount of time you want to share your location.
- 5. Tap **More**.
- 6. Select your **app of choice** to create and send a unique URL that broadcasts your current location. You can email it, send the link via Messenger, or even tweet it to the intended recipient.



<https://www.androidcentral.com/how-share-location-google-maps>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

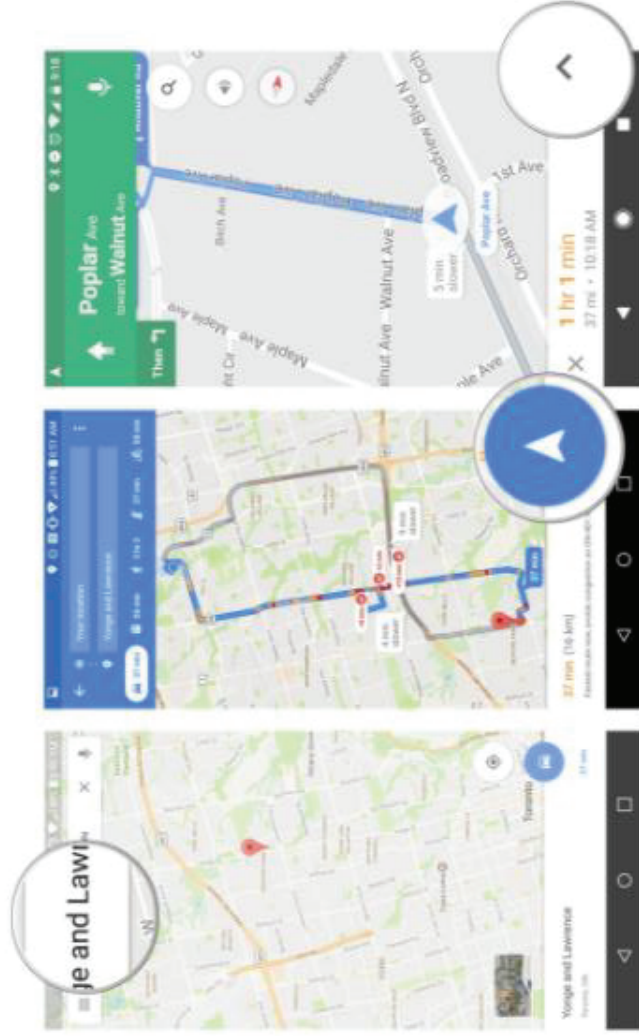
US9467838

Exemplary Supporting Evidence Regarding Accused Products

How to share your navigation directions while you walk, drive or transit

One of the best ways to share your location with someone is during a drive, walk or transit. If you're meeting a friend or family member somewhere, or navigating towards their home, Google Maps lets you share your location with them for the duration of the trip. It's magic!

1. In the search bar enter your destination.
2. Pick your navigation type (drive, transit, walk) and press the blue navigate button.
3. Tap the arrow next to the time-to-destination number at the bottom of the screen.



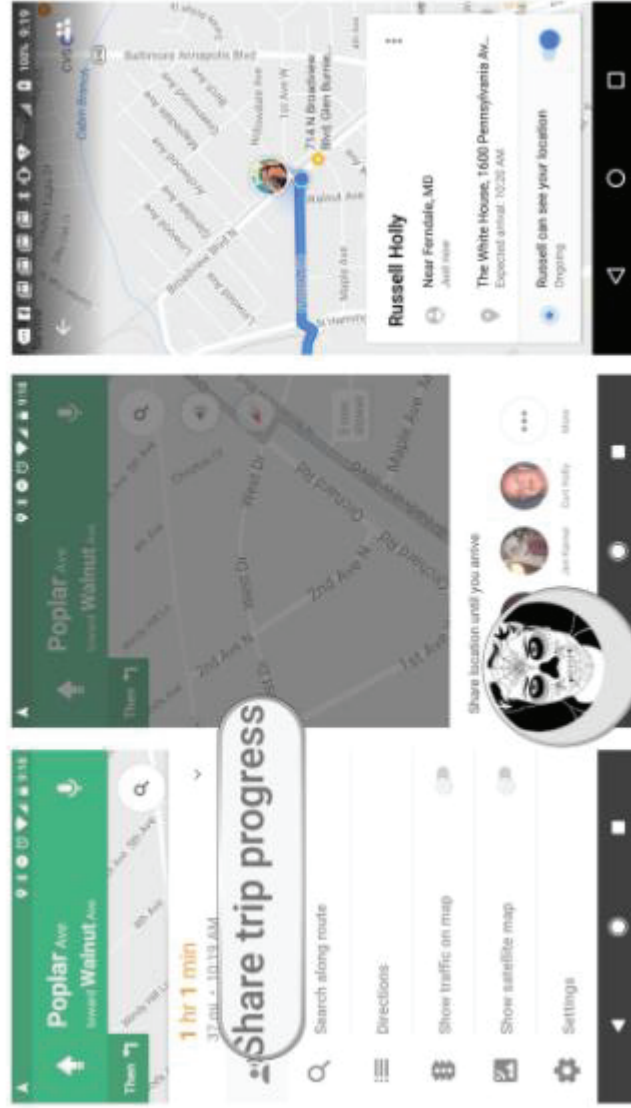
<https://www.androidcentral.com/how-share-location-google-maps>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products

- 4. Tap Share trip progress.
- 5. Choose one or more contacts to share trip progress.



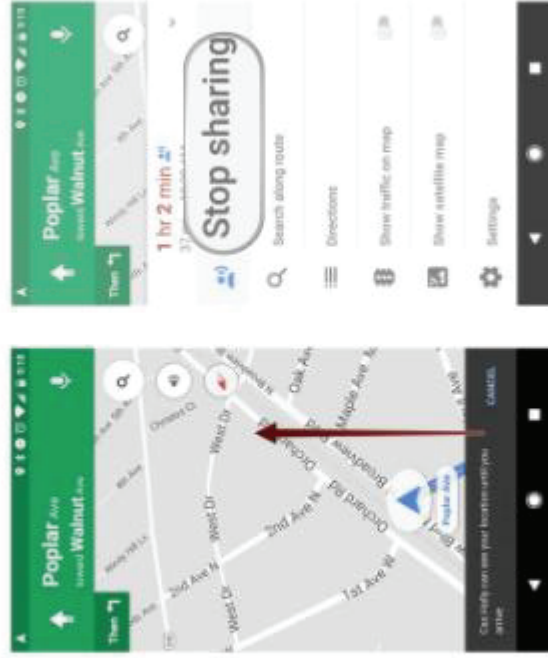
You can also stop sharing your location with someone before a trip ends.
<https://www.androidcentral.com/how-share-location-google-maps>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products

1. Tap the arrow next to the time-to-destination number at the bottom of the screen.
2. Tap Stop sharing.



That's it!

Are you excited that location sharing is back in Google Maps? How often do you use the feature?
<https://www.androidcentral.com/how-share-location-google-maps>



As shown below, a group may also be defined within Google Contacts.

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

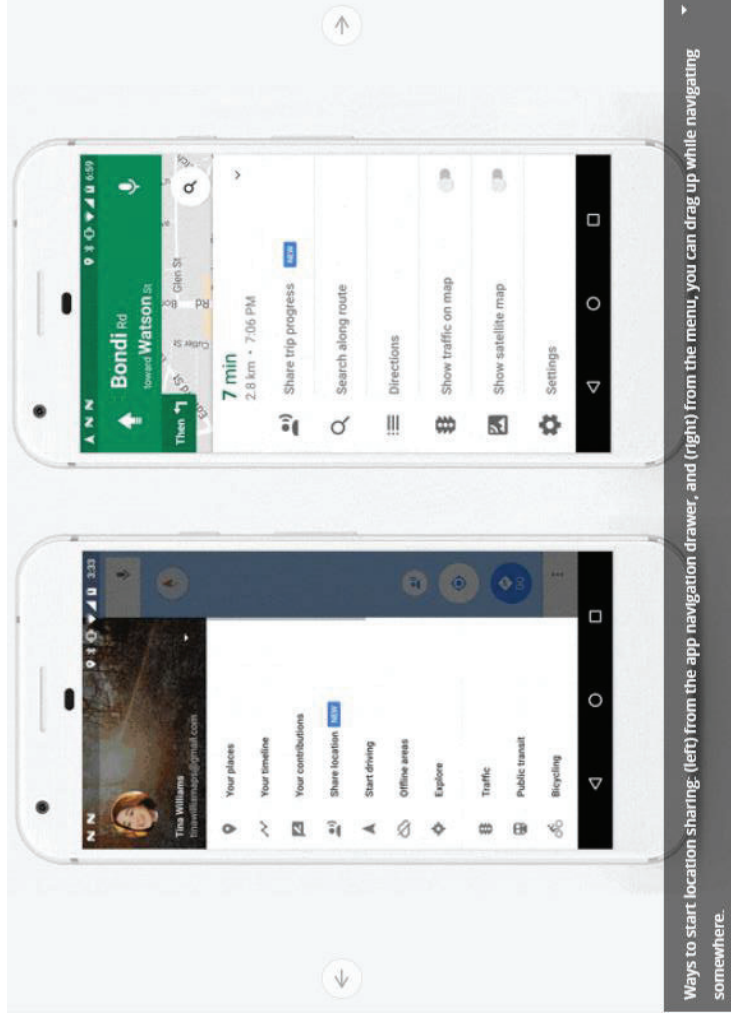
US9467838

Exemplary Supporting Evidence Regarding Accused Products

Share your contacts

1. Open your device's Contacts app .
2. Tap a contact in the list.
3. Tap More  > **Share**.
4. Choose how you want to share the contact.

https://support.google.com/android/answer/6118731?hl=en&ref_topic=6118711



<https://arstechnica.com/gadgets/2017/03/location-sharing-finally-returns-to-google-maps/>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products

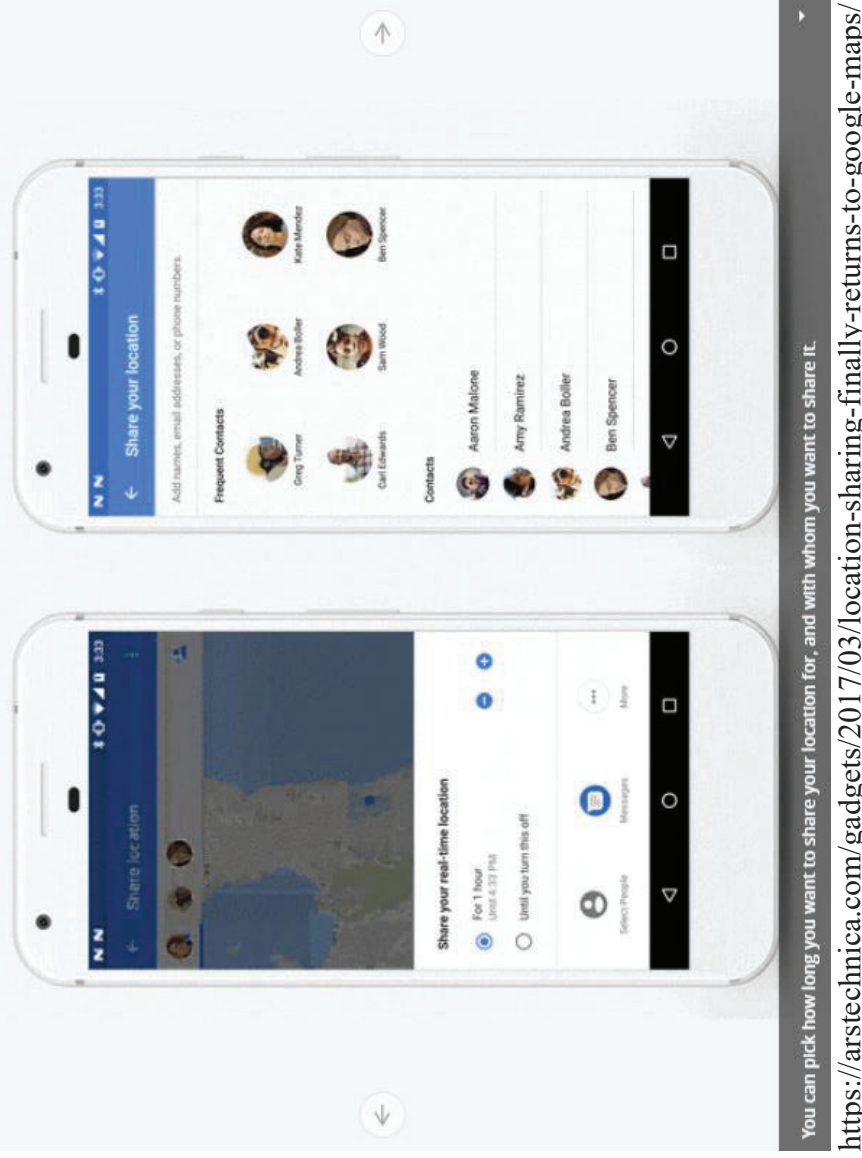


Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products

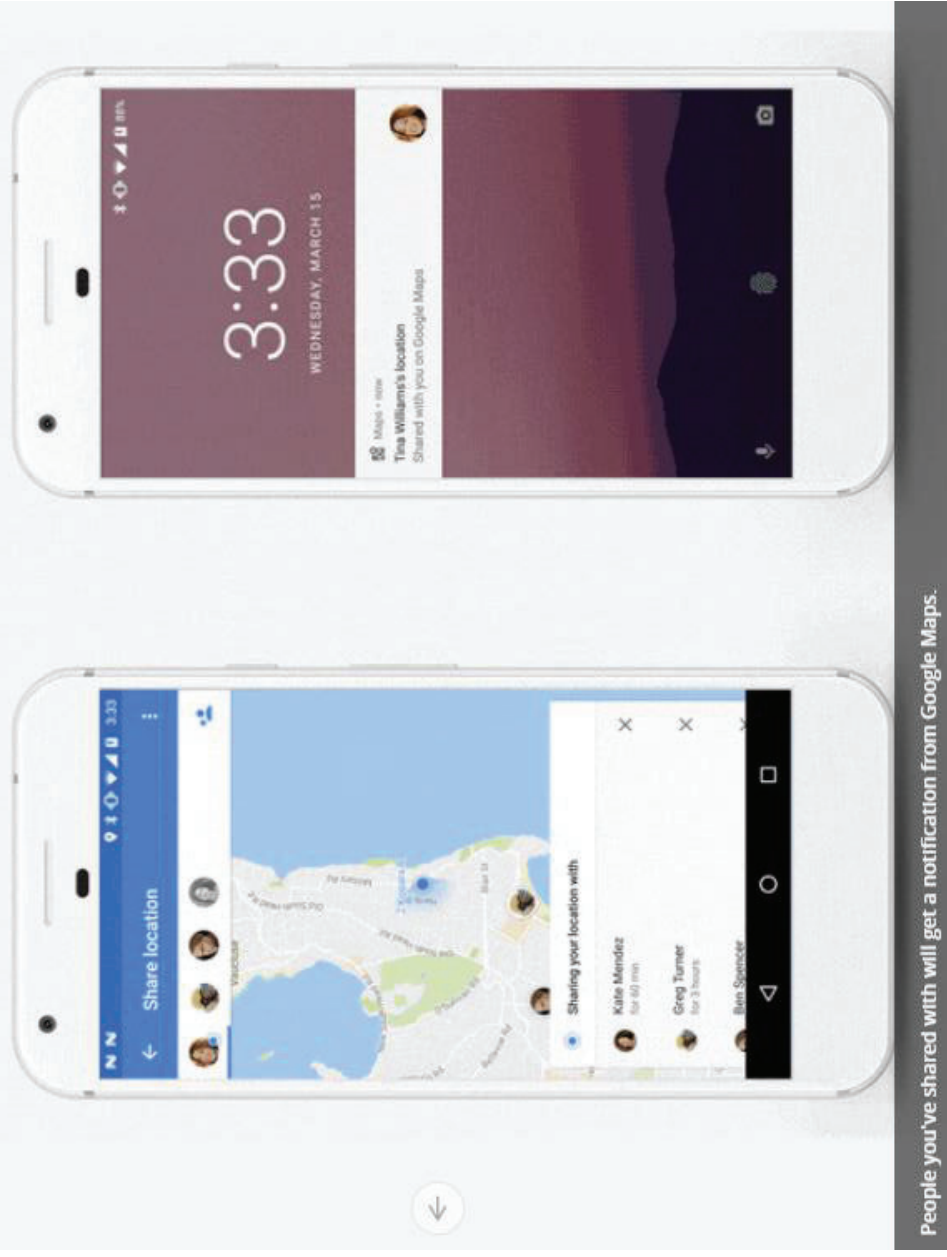


Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

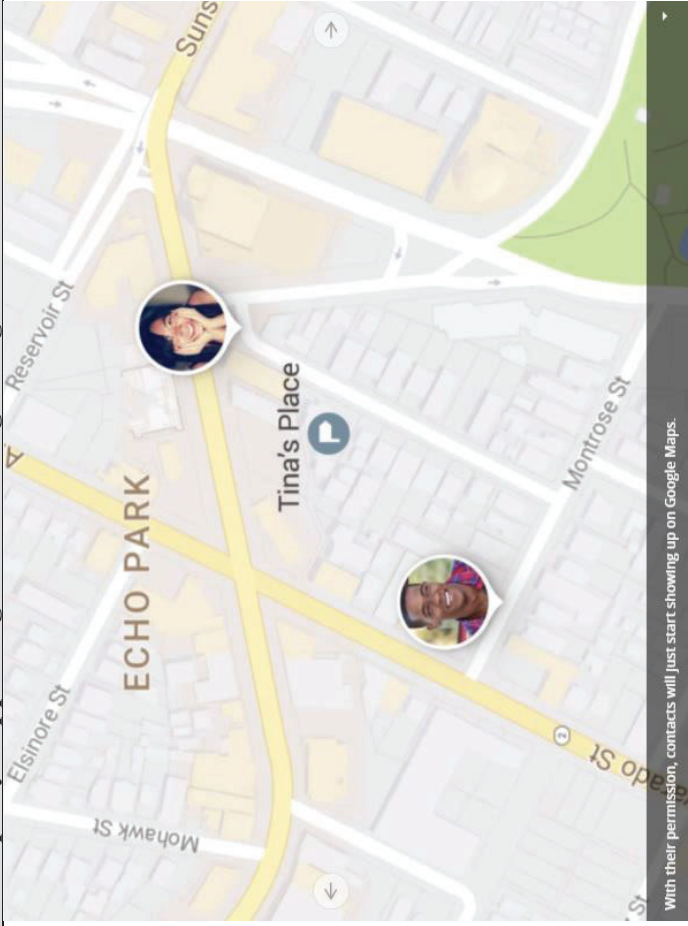
| | |
|-------------------------|--|
| <p>US9467838</p> | <p>Exemplary Supporting Evidence Regarding Accused Products</p>  <p>The screenshot shows a Google Maps view of the Echo Park neighborhood in Los Angeles. Two circular location pins are overlaid on the map, each containing a profile picture of a person. One pin is located on Reservoir St, and the other is on Tina's Place. A blue 'P' icon for a parking spot is also visible on Tina's Place. The map includes street names such as Mohawk St, Elsinore St, Reservoir St, Montrose St, and Echo St. A text overlay at the bottom of the map reads: "With their permission, contacts will just start showing up on Google Maps." Below the map, the URL https://arstechnica.com/gadgets/2017/03/location-sharing-finally-returns-to-google-maps/ is provided.</p> <p><u>Exemplary Find My Device Screenshots:</u></p> |
|-------------------------|--|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| | |
|------------------|--|
| <p>US9467838</p> | <p>Exemplary Supporting Evidence Regarding Accused Products</p> <p>The image displays two sequential screenshots of an Android mobile application interface for finding a device. The top status bar shows 'AT&T', signal strength, Wi-Fi, 94% battery, and the time 11:46 AM. The main header is a green bar with 'Find My Device' and a menu icon. Below this, there are two panels. The left panel shows a map of Rhode Island with a red location pin and the text 'LG Phoenix 3 In your hand'. The right panel shows a map of Quimby Ave with a red location pin and the text 'LG Phoenix 3 Last seen 1 min ago'. Both panels include a battery icon at 94% and a 'wallcri...' notification. At the bottom of each panel are three options: 'PLAY SOUND', 'LOCK', and 'ERASE'. A large green 'LOCK' button is positioned at the bottom right of the right-hand screenshot. The bottom of the screen features a standard Android navigation bar with back, home, and recent apps icons.</p> |
|------------------|--|

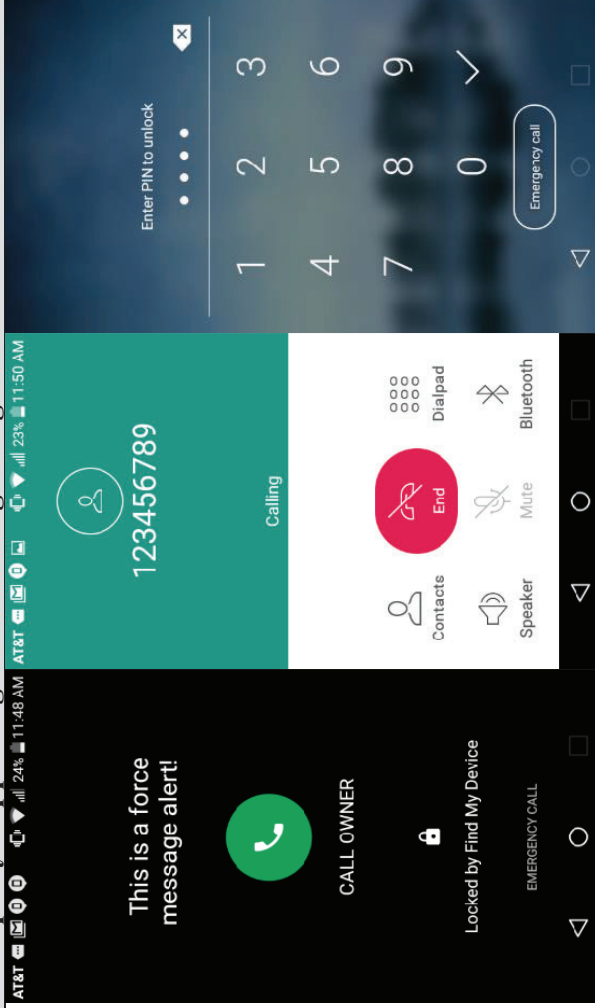
Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| | |
|------------------|--|
| <p>US9467838</p> | <p>Exemplary Supporting Evidence Regarding Accused Products</p> <p>The screenshot shows the 'Find My Device' app interface. At the top, there are two tabs: 'Lock device' and 'Find My Device'. Below the tabs, there is a section titled 'Add a message or phone number' with instructions: 'Leave anyone who finds your device a message or a phone number where you can be reached.' Below this, there is a text input field containing 'This is a force message alert!' and a character count '32 / 100'. A 'Phone number (optional)' field contains '123456789'. A green 'LOCK' button is visible. The bottom part of the screen shows a map with a location marker, a 'Lock requested' status, and details for an 'LG Phoenix 3' device, including 'Last seen 3 mins ago' and '24%' battery. At the bottom, there are three options: 'PLAY SOUND', 'LOCK', and 'ERASE', each with a right-pointing arrow. The Android navigation bar is visible at the very bottom.</p> |
|------------------|--|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products

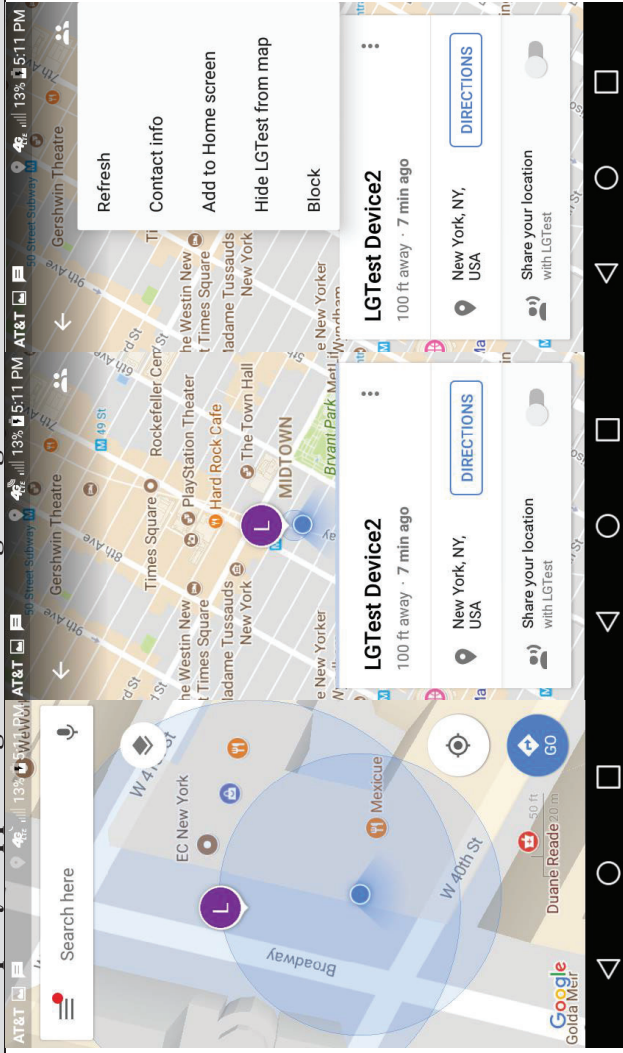


Exemplary Find My Device Screenshots:

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products



Upon information and belief, selection of contact info results in the presentation of actions to be chosen. See, e.g.,

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| | |
|------------------|---|
| <p>US9467838</p> | <p>Exemplary Supporting Evidence Regarding Accused Products</p>  <p>Exemplary Source Code:</p> <p>The above functionality is performed at least in part by the following publicly available source code and/or source code that invokes or is invoked by the following source code (or a substantially similar copy compiled and loaded onto the Accused Products by ZTE). AGIS reserves the right to supplement these contentions with additional source code as discovery progresses and as additional source code is made available. AGIS reserves the right to supplement these contentions with additional source code as discovery progresses and as additional source code is made available.</p> |
|------------------|---|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| US9467838 | Exemplary Supporting Evidence Regarding Accused Products |
|--|--|
| <pre> 44 * Class that sends chat message via SMS. 45 * 46 * The interface emulates a blocking sending similar to making an HTTP request. 47 * It calls the SmsManager to send a (potentially multipart) message and waits 48 * on the sent status on each part. The waiting has a timeout so it won't wait 49 * forever. Once the sent status of all parts received, the call returns. 50 * A successful sending requires success status for all parts. Otherwise, we 51 * pick the highest level of failure as the error for the whole message, which 52 * is used to determine if we need to retry the sending. 53 */ 54 public class SmsSender { 55 private static final String TAG = LogUtil.BUGLE_TAG; 56 57 public static final String EXTRA_PART_ID = "part_id"; 58 59 /* 60 * A map for pending sms messages. The key is the random request UUID. 61 */ 62 private static ConcurrentHashMap<Uri, SendResult> sPendingMessageMap = 63 new ConcurrentHashMap<Uri, SendResult>(); 64 65 private static final Random RANDOM = new Random(); 66 67 // Whether we should send multipart SMS as separate messages 68 private static Boolean sSendMultipartSmsAsSeparateMessages = null; 69 </pre> | <p>https://android.googlesource.com/platform/packages/apps/Messaging/+/nougat-mr1-release/src/com/android/messaging/sms/SmsSender.java</p> |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products

```

253 // Actually sending the message using SmsManager
254 private static void sendInternal(final Context context, final int subId, String dest,
255     final ArrayList<String> messages, final String serviceCenter,
256     final boolean requireDeliveryReport, final Uri messageUri) throws SmsException {
257     Assert.notNull(context);
258     final SmsManager smsManager = PhoneUtils.get(subId).getSmsManager();
259     final int messageCount = messages.size();
260     final ArrayList<PendingIntent> deliveryIntents = new ArrayList<PendingIntent>(messageCount);
261     final ArrayList<PendingIntent> sentIntents = new ArrayList<PendingIntent>(messageCount);
262     for (int i = 0; i < messageCount; i++) {
263         // Make pending intents different for each message part
264         final int partId = (messageCount <= 1 ? 0 : i + 1);
265         if (requireDeliveryReport && (i == (messageCount - 1))) {
266             // TODO we only care about the delivery status of the last part
267             // Shall we have better tracking of delivery status of all parts?
268             deliveryIntents.add((PendingIntent)getBroadcast(
269                 context,
270                 partId,
271                 getSendStatusIntent(context, SendStatusReceiver.MESSAGE_DELIVERED_ACTION,
272                     messageUri, partId, subId),
273                 0/**flag*/));
274         } else {
275             deliveryIntents.add(null);
276         }
277         sentIntents.add((PendingIntent)getBroadcast(
278             context,
279             partId,
280             getSendStatusIntent(context, SendStatusReceiver.MESSAGE_SENT_ACTION,
281                 messageUri, partId, subId),
282             0/**flag*/));
283     }
284     if (sSendMultiPartSmsAsSeparateMessages == null) {
285         sSendMultiPartSmsAsSeparateMessages = MmsConfig.get(subId)
286             .getSendMultiPartSmsAsSeparateMessages();
287     }

```

<https://android.googlesource.com/platform/packages/apps/Messaging/+/nougat-mr1-release/src/com/android/messaging/sms/SmsSender.java>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products

```

288     try {
289         if (sSendMultipartSmsAsSeparateMessages) {
290             // If multipart sms is not supported, send them as separate messages
291             for (int i = 0; i < messageCount; i++) {
292                 smsManager.sendTextMessage(dest,
293                     serviceCenter,
294                     messages.get(i),
295                     sentIntents.get(i),
296                     deliveryIntents.get(i));
297             }
298         } else {
299             smsManager.sendMultipartTextMessage(
300                 dest, serviceCenter, messages, sentIntents, deliveryIntents);
301         }
302     } catch (final Exception e) {
303         throw new SmsException("SmsSender: caught exception in sending " + e);
304     }
305 }

```

<https://android.googlesource.com/platform/packages/apps/Messaging/+/nougat-mr1-release/src/com/android/messaging/sms/SmsSender.java>

```

56 * Class that receives incoming SMS messages through android.provider.Telephony.SMS_RECEIVED
57 *
58 * This class serves two purposes:
59 * - Process phone verification SMS messages
60 * - Handle SMS messages when the user has enabled us to be the default SMS app (Pre-KLP)
61 */
62 public final class SmsReceiver extends BroadcastReceiver {
63     private static final String TAG = LogUtil.BUGLE_TAG;
64
65     private static ArrayList<Pattern> sIgnoreSmsPatterns;
66

```

<https://android.googlesource.com/platform/packages/apps/Messaging/+/nougat-mr1-release/src/com/android/messaging/receiver/SmsReceiver.java>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| US9467838 | Exemplary Supporting Evidence Regarding Accused Products |
|---|--|
| 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 | <pre> public static void deliverSmsMessages(final Context context, final int subId, final int errorCode, final android.telephony.SmsMessage[] messages) { final ContentValues messageValues = MmsUtils.parseReceivedSmsMessage(context, messages, errorCode); LogUtil.v(TAG, "SmsReceiver.deliverSmsMessages"); final long nowInMillis = System.currentTimeMillis(); final long receivedTimestamps = MmsUtils.getMessageDate(messages[0], nowInMillis); messageValues.put(Sms.Inbox.DATE, receivedTimestamps); // Default to unread and unseen for us but ReceiveSmsMessageAction will override // seen for the telephony db. messageValues.put(Sms.Inbox.READ, 0); messageValues.put(Sms.Inbox.SEEN, 0); if (OsUtil.isAtLeastL_MRI()) { messageValues.put(Sms.SUBSCRIPTION_ID, subId); } if (messages[0].getMessageClass() == android.telephony.SmsMessage.MessageClass.CLASS_0 DebugUtils.isDebugEnabled()) { Factory.get().getUIIntents().launchClassZeroActivity(context, messageValues); } else { final ReceiveSmsMessageAction action = new ReceiveSmsMessageAction(messageValues); action.start(); } } </pre> <p>https://android.googlesource.com/platform/packages/apps/Messaging/+nougat-mr1-release/src/com/android/messaging/receiver/SmsReceiver.java</p> |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| US9467838 | Exemplary Supporting Evidence Regarding Accused Products |
|--|--|
| <pre> 228 @Override 229 public void onReceive(final Context context, final Intent intent) { 230 LogUtil.v(TAG, "SmsReceiver.onReceive " + intent); 231 // On KLP+ we only take delivery of SMS messages in SmsDeliverReceiver. 232 if (PhoneUtils.getDefault().isSmsEnabled()) { 233 final String action = intent.getAction(); 234 if (OsUtil.isSecondaryUser() && 235 (Telephony.Sms.Intents.SMS_RECEIVED_ACTION.equals(action) 236 // TODO: update this with the actual constant from Telephony 237 "android.provider.Telephony.MMS_DOWNLOADED".equals(action))) { 238 postNewMessageSecondaryUserNotification(); 239 } else if (!OsUtil.isAtLeastKLP()) { 240 deliverSmsIntent(context, intent); 241 } 242 } 243 } </pre> | <p>https://android.googlesource.com/platform/packages/apps/Messaging/+nougat-mr1-release/src/com/android/messaging/receiver/SmsReceiver.java</p> |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| US9467838 | Exemplary Supporting Evidence Regarding Accused Products |
|-----------|---|
| 52 | * Class that sends chat message via MMS. |
| 53 | * The interface emulates a blocking send similar to making an HTTP request. |
| 54 | * The interface emulates a blocking send similar to making an HTTP request. |
| 55 | */ |
| 56 | public class MmsSender { |
| 57 | private static final String TAG = LogUtil.BUGLE_TAG; |
| 58 | /** |
| 59 | * Send an MMS message. |
| 60 | * @param context Context |
| 61 | * @param messageUri The unique URI of the message for identifying it during sending |
| 62 | * @param sendReq The SendReq PDU of the message |
| 63 | * @throws MmsFailureException |
| 64 | */ |
| 65 | public static void sendMms(final Context context, final int subId, final Uri messageUri, |
| 66 | final SendReq sendReq, final Bundle sentIntentExtras) throws MmsFailureException { |
| 67 | sendMms(context, |
| 68 | subId, |
| 69 | messageUri, |
| 70 | null /* locationUrl */, |
| 71 | sendReq, |
| 72 | true /* responseImportant */, |
| 73 | sentIntentExtras); |
| 74 | } |
| 75 | https://android.googlesource.com/platform/packages/apps/Messaging/+/nougat-mr1- |
| 76 | release/src/com/android/messaging/sms/MmsSender.java |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| US9467838 | Exemplary Supporting Evidence Regarding Accused Products |
|-----------|---|
| 240 | * Download an MMS message. |
| 241 | * |
| 242 | * @param context Context |
| 243 | * @param contentLocation The url of the MMS message |
| 244 | * @throws MmsFailureException |
| 245 | * @throws InvalidHeaderValueException |
| 246 | */ |
| 247 | public static void downloadMms(final Context context, final int subId, |
| 248 | final String contentLocation, Bundle extras) throws MmsFailureException, |
| 249 | InvalidHeaderValueException { |
| 250 | final Uri requestUri = Uri.parse(contentLocation); |
| 251 | final Uri contentUri = MmsFileProvider.buildRawMmsUri(); |
| 252 | |
| 253 | final Intent downloadedIntent = new Intent(SendStatusReceiver.MMS_DOWNLOADED_ACTION, |
| 254 | requestUri, |
| 255 | context, |
| 256 | SendStatusReceiver.class); |
| 257 | downloadedIntent.putExtra(SendMessageAction.EXTRA_CONTENT_URI, contentUri); |
| 258 | if (extras != null) { |
| 259 | downloadedIntent.putExtras(extras); |
| 260 | } |
| 261 | final PendingIntent downloadedPendingIntent = PendingIntent.getBroadcast(context, |
| 262 | 0 /*request code*/, |
| 263 | downloadedIntent, |
| 264 | PendingIntent.FLAG_UPDATE_CURRENT); |
| 265 | |
| 266 | |
| 267 | MmsManager.downloadMultimediaMessage(subId, context, contentLocation, contentUri, |
| 268 | downloadedPendingIntent); |
| 269 | } |
| | https://android.googlesource.com/platform/packages/apps/Messaging/+nougat-mr1-release/src/com/android/messaging/sms/MmsSender.java |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| US9467838 | Exemplary Supporting Evidence Regarding Accused Products |
|-----------|---|
| 97 | * Execute an MMS HTTP request, either a POST (sending) or a GET (downloading) |
| 98 | x |
| 99 | * @param urlString The request URL, for sending it is usually the MMSC, and for downloading |
| 100 | * it is the message URL |
| 101 | * @param pdu For POST (sending) only, the PDU to send |
| 102 | * @param method HTTP method, POST for sending and GET for downloading |
| 103 | * @param isProxySet Is there a proxy for the MMSC |
| 104 | * @param proxyHost The proxy host |
| 105 | * @param proxyPort The proxy port |
| 106 | * @param mmsConfig The MMS config to use |
| 107 | * @param userAgent The user agent header value |
| 108 | * @param uaProfUrl The UA Prof URL header value |
| 109 | * @return The HTTP response body |
| 110 | * @throws MmsHttpException For any failures |
| 111 | */ |
| 112 | public byte[] execute(String urlString, byte[] pdu, String method, boolean isProxySet, |
| 113 | String proxyHost, int proxyPort, Bundle mmsConfig, String userAgent, String uaProfUrl) |
| 114 | throws MmsHttpException { |
| 115 | Log.d(MmsService.TAG, "HTTP: " + method + " " + Utils.redactUrlForNonVerbose(urlString) |
| 116 | + (isProxySet ? (" proxy=" + proxyHost + " : " + proxyPort) : "") |
| 117 | + ", PDU size=" + (pdu != null ? pdu.length : 0)); |
| 118 | checkMethod(method); |
| 119 | URLConnection connection = null; |
| 120 | try { |
| 121 | Proxy proxy = Proxy.NO_PROXY; |
| 122 | if (isProxySet) { |
| 123 | proxy = new Proxy(Proxy.Type.HTTP, new InetSocketAddress(proxyHost, proxyPort)); |
| 124 | } |
| 125 | final URL url = new URL(urlString); |
| 126 | // Now get the connection |
| 127 | connection = (URLConnection) url.openConnection(proxy); |
| 128 | connection.setDoInput(true); |
| 129 | connection.setConnectTimeout(|
| 130 | mmsConfig.getInt(CarrierConfigValuesLoader.CONFIG_HTTP_SOCKET_TIMEOUT, |
| 131 | CarrierConfigValuesLoader.CONFIG_HTTP_SOCKET_TIMEOUT_DEFAULT)); |

<https://android.googlesource.com/platform/packages/apps/Messaging/+nougat-mr1-release/src/android/support/v7/mms/MmsHttpClient.java>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products

```

132 // ----- COMMON HEADERS -----
133 // Header: Accept
134 connection.setRequestProperty(HEADER_ACCEPT, HEADER_VALUE_ACCEPT);
135 // Header: Accept-Language
136 connection.setRequestProperty(
137     HEADER_ACCEPT_LANGUAGE, getLocale().getLanguage(Locale.getDefault()));
138 // Header: User-Agent
139 Log.i("MmsService.TAG", "HTTP: User-Agent=" + userAgent);
140 connection.setRequestProperty(HEADER_USER_AGENT, userAgent);
141 // Header: x-wap-profile
142 final String uaProfileTagName = mmsConfig.getString(
143     CarrierConfigValuesLoader.CONFIG_UA_PROF_TAG_NAME,
144     CarrierConfigValuesLoader.CONFIG_UA_PROF_TAG_NAME_DEFAULT);
145 if (uaProfile != null) {
146     Log.i("MmsService.TAG", "HTTP: UaProfile=" + uaProfile);
147     connection.setRequestProperty(uaProfileTagName, uaProfile);
148 }
149 // Add extra headers specified by mms_config.xml's httpParams
150 addExtraHeaders(connection, mmsConfig);
151 // Different stuff for GET and POST
152 if (METHOD_POST.equals(method)) {
153     if (pdu == null || pdu.length < 1) {
154         Log.e("MmsService.TAG", "HTTP: empty pdu");
155         throw new MmsHttpException(0/*statusCode*/, "Sending empty PDU");
156     }
157     connection.setDoOutput(true);
158     connection.setRequestMethod(METHOD_POST);
159     if (mmsConfig.getBoolean(
160         CarrierConfigValuesLoader.CONFIG_SUPPORT_HTTP_CHARSET_HEADER,
161         CarrierConfigValuesLoader.CONFIG_SUPPORT_HTTP_CHARSET_HEADER_DEFAULT)) {
162         connection.setRequestProperty(HEADER_CONTENT_TYPE,
163             HEADER_VALUE_CONTENT_TYPE_WITH_CHARSET);
164     } else {
165         connection.setRequestProperty(HEADER_CONTENT_TYPE,
166             HEADER_VALUE_CONTENT_TYPE_WITHOUT_CHARSET);

```

<https://android.googlesource.com/platform/packages/apps/Messaging/+nougat-mr1-release/src/android/support/v7/mms/MmsHttpClient.java>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products

```

167     }
168     if (Log.isLoggable(MmsService.TAG, Log.VERBOSE)) {
169         logHttpHeaders(connection.getRequestProperties());
170     }
171     connection.setFixedLengthStreamingMode(pdu.length);
172     // Sending request body
173     final OutputStream out =
174         new BufferedOutputStream(connection.getOutputStream());
175     out.write(pdu);
176     out.flush();
177     out.close();
178     } else if (METHOD_GET.equals(method)) {
179         if (Log.isLoggable(MmsService.TAG, Log.VERBOSE)) {
180             logHttpHeaders(connection.getRequestProperties());
181         }
182         connection.setRequestMethod(METHOD_GET);
183     }
184     // Get response
185     final int responseCode = connection.getResponseCode();
186     final String responseMessage = connection.getResponseMessage();
187     Log.d(MmsService.TAG, "HTTP: " + responseCode + " " + responseMessage);
188     if (Log.isLoggable(MmsService.TAG, Log.VERBOSE)) {
189         logHttpHeaders(connection.getHeaderFields());
190     }
191     if (responseCode / 100 != 2) {
192         throw new MmsHttpException(responseCode, responseMessage);
193     }
194     final InputStream in = new BufferedInputStream(connection.getInputStream());
195     final ByteArrayOutputStream bytesOut = new ByteArrayOutputStream();
196     final byte[] buf = new byte[4096];
197     int count = 0;
198     while ((count = in.read(buf)) > 0) {
199         bytesOut.write(buf, 0, count);
200     }
201     in.close();
202     final byte[] responseBody = bytesOut.toByteArray();
203     Log.d(MmsService.TAG, "HTTP: response size="
204         + (responseBody != null ? responseBody.length : 0));
205     return responseBody;

```

<https://android.googlesource.com/platform/packages/apps/Messaging/+nougat-mr1-release/src/android/support/v7/mms/MmsHttpClient.java>

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

US9467838

Exemplary Supporting Evidence Regarding Accused Products

```

38 * Request to send an MMS
39 */
40 class SendRequest extends MmsRequest {
41     // Max send response PDU size in bytes (exceeding this may cause problem with
42     // system intent delivery).
43     private static final int MAX_SEND_RESPONSE_SIZE = 1000 * 1024;
44
45     private byte[] mPduData;
46
47     SendRequest(final String locationUrl, final Uri pduUri, final PendingIntent sentIntent) {
48         super(locationUrl, pduUri, sentIntent);
49     }
50
51     @Override
52     protected boolean loadRequest(final Context context, final Bundle mmsConfig) {
53         mPduData = readPduFromContentUri(
54             context,
55             pduUri,
56             mmsConfig.getInt(
57                 CarrierConfigValuesLoader.CONFIG_MAX_MESSAGE_SIZE,
58                 CarrierConfigValuesLoader.CONFIG_MAX_MESSAGE_SIZE_DEFAULT));
59         return (mPduData != null);
60     }
61
62     @Override
63     protected boolean transferResponse(final Context context, final Intent fillIn,
64         final byte[] response) {
65         // SendConf pdus are always small and can be included in the intent
66         if (response != null && fillIn != null) {
67             if (response.length > MAX_SEND_RESPONSE_SIZE) {
68                 // If the response PDU is too large, it won't be able to fit in
69                 // the PendingIntent to be transferred via system IPC.
70                 return false;
71             }
72             fillIn.putExtra(SmsManager.EXTRA_MMS_DATA, response);
73         }
74         return true;
75     }

```

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| | |
|------------------|--|
| <p>US9467838</p> | <p>Exemplary Supporting Evidence Regarding Accused Products https://android.googlesource.com/platform/packages/apps/Messaging/+nougat-mr1-release/src/android/support/v7/mms/SendRequest.java</p> <pre>public static LocationRequest create ()</pre> <p>Create a location request with default parameters.</p> <p>Default parameters are for a block accuracy, slowly updated location. It can then be adjusted as required by the applications before passing to the <code>FusedLocationProviderApi</code>.</p> <p>Returns</p> <ul style="list-style-type: none"> • a new location request <p>https://developers.google.com/android/reference/com/google/android/gms/location/LocationRequest</p> |
|------------------|--|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| | |
|-----------|---|
| US9467838 | <p data-bbox="196 716 228 1556">Exemplary Supporting Evidence Regarding Accused Products</p> <p data-bbox="250 800 282 1535">public static final int PRIORITY_BALANCED_POWER_ACCURACY</p> <p data-bbox="315 898 347 1535">Used with <code>setPriority(int)</code> to request "block" level accuracy.</p> <p data-bbox="371 436 436 1535">Block level accuracy is considered to be about 100 meter accuracy. Using a coarse accuracy such as this often consumes less power.</p> <p data-bbox="456 1346 480 1535">Constant Value: 102</p> <p data-bbox="537 968 570 1535">public static final int PRIORITY_HIGH_ACCURACY</p> <p data-bbox="602 737 626 1535">Used with <code>setPriority(int)</code> to request the most accurate locations available.</p> <p data-bbox="659 1115 683 1535">This will return the finest location available.</p> <p data-bbox="708 1346 732 1535">Constant Value: 100</p> <p data-bbox="789 1020 821 1535">public static final int PRIORITY_LOW_POWER</p> <p data-bbox="854 919 886 1535">Used with <code>setPriority(int)</code> to request "city" level accuracy.</p> <p data-bbox="919 338 984 1535">City level accuracy is considered to be about 10km accuracy. Using a coarse accuracy such as this often consumes less power.</p> <p data-bbox="1000 1346 1024 1535">Constant Value: 104</p> <p data-bbox="1049 285 1081 1535">https://developers.google.com/android/reference/com/google/android/gms/location/LocationRequest</p> |
|-----------|---|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| | |
|-----------|--|
| US9467838 | <p data-bbox="185 709 228 1562">Exemplary Supporting Evidence Regarding Accused Products</p> <pre data-bbox="245 1058 280 1541">public Task<Location> getLastLocation ()</pre> <p data-bbox="313 972 341 1551">Returns the best most recent location currently available.</p> <p data-bbox="370 380 431 1551">If a location is not available, which should happen very rarely, null will be returned. The best accuracy available while respecting the location permissions will be returned.</p> <p data-bbox="461 338 522 1551">This method provides a simplified way to get location. It is particularly well suited for applications that do not require an accurate location and that do not want to maintain extra logic for location updates.</p> <pre data-bbox="578 863 613 1541">public Task<LocationAvailability> getLocationAvailability ()</pre> <p data-bbox="646 386 708 1551">Returns the availability of location data. When <code>isLocationAvailable()</code> returns true, then the location returned by <code>getLastLocation()</code> will be reasonably up to date within the hints specified by the active <code>LocationRequest</code>s.</p> <p data-bbox="737 604 764 1551">If the client isn't connected to Google Play services and the request times out, null is returned.</p> <p data-bbox="794 401 855 1551">Note it's always possible for <code>getLastLocation()</code> to return null even when this method returns true (e.g. location settings were disabled between calls).</p> <p data-bbox="862 197 925 1562">https://developers.google.com/android/reference/com/google/android/gms/location/FusedLocationProviderClient</p> |
|-----------|--|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| | | | | | | | |
|------------------|--|----------------|---------------------------------------|-----------------|--|---------------|--|
| <p>US9467838</p> | <p>Exemplary Supporting Evidence Regarding Accused Products</p> <p>public Task<Void> requestLocationUpdates (LocationRequest request, LocationCallback callback, Looper looper)</p> <p>Requests location updates with a callback on the specified Looper thread.</p> <p>This method is suited for the foreground use cases. For background use cases, the <code>PendingIntent</code> version of the method is recommended, see <code>requestLocationUpdates(LocationRequest, PendingIntent)</code>.</p> <p>Any previous LocationRequests registered on this LocationListener will be replaced.</p> <p>This call will keep the Google Play services connection active, so make sure to call <code>removeLocationUpdates(LocationCallback)</code> when you no longer need it, otherwise you lose the benefits of the automatic connection management.</p> <p>Callbacks for <code>LocationCallback</code> will be made on the specified thread, which must already be a prepared looper thread.</p> <p>Parameters</p> <table border="1"> <tr> <td>request</td> <td>The location request for the updates.</td> </tr> <tr> <td>callback</td> <td>The callback for the location updates.</td> </tr> <tr> <td>looper</td> <td>The Looper object whose message queue will be used to implement the callback mechanism, or null to make callbacks on the calling thread.</td> </tr> </table> <p>https://developers.google.com/android/reference/com/google/android/gms/location/FusedLocationProviderClient</p> | request | The location request for the updates. | callback | The callback for the location updates. | looper | The Looper object whose message queue will be used to implement the callback mechanism, or null to make callbacks on the calling thread. |
| request | The location request for the updates. | | | | | | |
| callback | The callback for the location updates. | | | | | | |
| looper | The Looper object whose message queue will be used to implement the callback mechanism, or null to make callbacks on the calling thread. | | | | | | |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| | | | | | |
|-----------------------------|--|----------------------|---------------------------------------|-----------------------------|---|
| <p>US9467838</p> | <p>Exemplary Supporting Evidence Regarding Accused Products</p> <pre>public Task<Void> requestLocationUpdates (LocationRequest request, PendingIntent callbackIntent)</pre> <p>Requests location updates with a callback on the specified PendingIntent.</p> <p>This method is suited for the background use cases, more specifically for receiving location updates, even when the app has been killed by the system. In order to do so, use a <code>PendingIntent</code> for a started service. For foreground use cases, the <code>LocationCallback</code> version of the method is recommended, see <code>requestLocationUpdates(LocationRequest, LocationCallback, Looper)</code>.</p> <p>Any previously registered requests that have the same <code>PendingIntent</code> (as defined by <code>equals(Object)</code>) will be replaced by this request.</p> <p>Both <code>LocationResult</code> and <code>LocationAvailability</code> are sent to the given <code>PendingIntent</code>. You can extract data from an <code>Intent</code> using <code>hasResult(Intent)</code>, <code>extractResult(Intent)</code>, <code>hasLocationAvailability(Intent)</code>, and <code>extractLocationAvailability(Intent)</code>.</p> <p>Parameters</p> <table border="1"> <tr> <td><code>request</code></td> <td>The location request for the updates.</td> </tr> <tr> <td><code>callbackIntent</code></td> <td>A pending intent to be sent for each location update.</td> </tr> </table> <p>Returns</p> <ul style="list-style-type: none"> a Task for the call, check <code>isSuccessful()</code> to determine if it was successful. <p>https://developers.google.com/android/reference/com/google/android/gms/location/FusedLocationProviderClient</p> | <code>request</code> | The location request for the updates. | <code>callbackIntent</code> | A pending intent to be sent for each location update. |
| <code>request</code> | The location request for the updates. | | | | |
| <code>callbackIntent</code> | A pending intent to be sent for each location update. | | | | |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| | |
|-----------|--|
| US9467838 | <p data-bbox="191 247 228 1564">Exemplary Supporting Evidence Regarding Accused Products</p> <p data-bbox="240 247 277 1564">public void onLocationAvailability (LocationAvailability locationAvailability)</p> <p data-bbox="305 247 337 1564">Called when there is a change in the availability of location data.</p> <p data-bbox="365 247 560 1564">When <code>isLocationAvailable()</code> returns <code>false</code> you can assume that location will not be returned in <code>onLocationResult(LocationResult)</code> until something changes in the device's settings or environment. Even when <code>isLocationAvailable()</code> returns <code>true</code> the <code>onLocationResult(LocationResult)</code> may not always be called regularly, however the device location is known and both the most recently delivered location and <code>getLastLocation(GoogleApiClient)</code> will be reasonably up to date given the hints specified by the active <code>LocationRequest</code> s.</p> <p data-bbox="581 247 613 1564">Parameters</p> <p data-bbox="641 247 706 1564">locationAvailability The current status of location availability.</p> <p data-bbox="755 247 787 1564">public void onLocationResult (LocationResult result)</p> <p data-bbox="815 247 847 1564">Called when device location information is available.</p> <p data-bbox="875 247 940 1564">The most recent location returned by <code>getLastLocation()</code> is not guaranteed to be immediately fresh, but will be reasonably up to date given the hints specified by the active <code>LocationRequest</code> s.</p> <p data-bbox="961 247 993 1564">Parameters</p> <p data-bbox="1021 247 1086 1564">result The latest location result available.</p> <p data-bbox="1097 247 1130 1564">https://developers.google.com/android/reference/com/google/android/gms/location/LocationCallback</p> <p data-bbox="1141 247 1174 1564">public abstract void onLocationChanged (Location location)</p> <p data-bbox="1201 247 1234 1564">Called when the location has changed.</p> <p data-bbox="1255 247 1287 1564">Parameters</p> <p data-bbox="1315 247 1380 1564">location The updated location.</p> |
|-----------|--|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| | |
|------------------|---|
| <p>US9467838</p> | <p>Exemplary Supporting Evidence Regarding Accused Products https://developers.google.com/android/reference/com/google/android/gms/location/LocationListener</p> <p>Public Constructors</p> <p>public MapView (Context context)</p> <p>public MapView (Context context, AttributeSet attrs)</p> <p>public MapView (Context context, AttributeSet attrs, int defStyleAttr)</p> <p>public MapView (Context context, GoogleMapOptions options)</p> <p>https://developers.google.com/android/reference/com/google/android/gms/maps/MapView</p> |
|------------------|---|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| US9467838 | Exemplary Supporting Evidence Regarding Accused Products |
|---|---|
| | <pre>public void getMapAsync (OnMapReadyCallback callback)</pre> <p>Returns a non-null instance of the <code>GoogleMap</code>, ready to be used.</p> <p>Note that:</p> <ul style="list-style-type: none"> • This method must be called from the main thread. • The callback will be executed in the main thread. • In the case where Google Play services is not installed on the user's device, the callback will not be triggered until the user installs it. • The <code>GoogleMap</code> object provided by the callback is non-null. <p>Parameters</p> <p>callback The callback object that will be triggered when the map is ready to be used.</p> <pre>public final void onCreate (Bundle savedInstanceState)</pre> <p>You must call this method from the parent Activity/Fragment's corresponding method. https://developers.google.com/android/reference/com/google/android/gms/maps/MapView</p> |
| 2. The method of claim 1, wherein the message is transmitted to the first server. | ZTE infringes directly and/or indirectly by performing, inducing others to perform, and/or contributing to the performance of the method of claim 1, wherein the message is transmitted to the first server. See, e.g. 1B and 1C above. |
| 3. The method of claim 1, wherein the group comprises a plurality of group members permitted to communicate with each | ZTE infringes directly and/or indirectly by performing, inducing others to perform, and/or contributing to the performance of the method of claim 1, wherein the group comprises a plurality of group members permitted to communicate with each other via the communication network. See, e.g. 1B above. |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| US9467838 Exemplary Supporting Evidence Regarding Accused Products | |
|--|--|
| other via the communication network. | |
| 4. The method of claim 1, wherein sending the third data to the selected one or more second devices via the first server comprises using an Internet Protocol to send the third data to the first server. See, e.g. 1H above. For example, contacting via chat, messaging, email, voice, and/or over applications such as Android Device Manager, Find My Phone, Find My Device, Google Latitude, Google Plus, Google Hangouts, Google Maps, Google Assistant, Google Search, Google Messages, Android Messenger, Google Allo, Google Duo, Gmail, and Google Chrome each include sending data over IP. For example, playing a sound and sending a message includes sending data over IP. | ZTE infringes directly and/or indirectly by performing, inducing others to perform, and/or contributing to the performance of the method of claim 1, wherein sending the third data to the selected one or more second devices via the first server comprises using an Internet Protocol to send the third data to the first server. See, e.g. 1H above. For example, contacting via chat, messaging, email, voice, and/or over applications such as Android Device Manager, Find My Phone, Find My Device, Google Latitude, Google Plus, Google Hangouts, Google Maps, Google Assistant, Google Search, Google Messages, Android Messenger, Google Allo, Google Duo, Gmail, and Google Chrome each include sending data over IP. For example, playing a sound and sending a message includes sending data over IP. |
| 5. The method of claim 4, wherein the first device does not have access to respective Internet Protocol addresses of the one or more second devices included in the group. See, e.g. 1H above. Upon information and belief, sending data from a first device contacting via chat, messaging, email, voice, and/or over applications such as Android Device Manager, Find My Phone, Find My Device, Google Latitude, Google Plus, Google Hangouts, Google Maps, Google Assistant, Google Search, Google Messages, Android Messenger, Google Allo, Google Duo, Gmail, and Google Chrome does not require the IP address of the recipient device. | ZTE infringes directly and/or indirectly by performing, inducing others to perform, and/or contributing to the performance of the method of claim 4, wherein the first device does not have access to respective Internet Protocol addresses of the one or more second devices included in the group. See, e.g. 1H above. Upon information and belief, sending data from a first device contacting via chat, messaging, email, voice, and/or over applications such as Android Device Manager, Find My Phone, Find My Device, Google Latitude, Google Plus, Google Hangouts, Google Maps, Google Assistant, Google Search, Google Messages, Android Messenger, Google Allo, Google Duo, Gmail, and Google Chrome does not require the IP address of the recipient device. |
| 6. The method of claim 1, wherein the third data include a short message service message, a text message, an image, or a video. For example, contacting via chat, messaging, email, voice, and/or over applications such as Android Device Manager, Find My Phone, Find My Device, Google Latitude, Google Plus, Google Hangouts, Google Maps, Google Assistant, Google Search, Google Messages, Android Messenger, Google Allo, Google Duo, Gmail, and Google Chrome each include sending data over IP. Upon information and belief, contacting via message includes SMS and text messages. Upon information and belief, contacting via Android Device Manager, Find My Phone, Find My Device, Google Latitude, Google Plus, Google Hangouts, Google Maps, Google Assistant, Google Search, Google Messages, Android Messenger, Google Allo, Google Duo, Gmail, and Google Chrome includes messaging text, images, and video. | ZTE infringes directly and/or indirectly by performing, inducing others to perform, and/or contributing to the performance of the method of claim 1, wherein the third data include a short message service message, a text message, an image, or a video. See, e.g. 1H above. For example, playing a sound and sending a message includes sending the third. For example, contacting via chat, messaging, email, voice, and/or over applications such as Android Device Manager, Find My Phone, Find My Device, Google Latitude, Google Plus, Google Hangouts, Google Maps, Google Assistant, Google Search, Google Messages, Android Messenger, Google Allo, Google Duo, Gmail, and Google Chrome each include sending data over IP. Upon information and belief, contacting via message includes SMS and text messages. Upon information and belief, contacting via Android Device Manager, Find My Phone, Find My Device, Google Latitude, Google Plus, Google Hangouts, Google Maps, Google Assistant, Google Search, Google Messages, Android Messenger, Google Allo, Google Duo, Gmail, and Google Chrome includes messaging text, images, and video. |
| 7. The method of claim | ZTE infringes directly and/or indirectly by performing, inducing others to perform, and/or contributing to |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| Exemplary Supporting Evidence Regarding Accused Products | |
|--|--|
| US9467838 | <p>the performance of the method of claim 6, wherein the video comprises a video clip. See, e.g. 1H above. For example, contact over Android Device Manager, Find My Phone, Find My Device, Google Latitude, Google Plus, Google Hangouts, Google Maps, Google Assistant, Google Search, Google Messages, Android Messenger, Google Allo, Google Duo, Gmail, and Google Chrome includes sending a video clip.</p> |
| <p>6, wherein the video comprises a video clip.</p> | <p>ZTE infringes directly and/or indirectly by performing, inducing others to perform, and/or contributing to the performance of the method of claim 1, wherein the third data include a voice recording. See, e.g. 1H above. For example, contacting via Android Device Manager, Find My Phone, Find My Device, Google Latitude, Google Plus, Google Hangouts, Google Maps, Google Assistant, Google Search, Google Messages, Android Messenger, Google Allo, Google Duo, Gmail, and Google Chrome, include sending a voice recording or audio recording. For example, playing a sound a second device including sending and playing a voice file. For example, Facetime includes sending a recording of a voice. For example, each Accused Product can send a recording.</p> <p>For example, each Accused Product can send a recording.</p> <p>https://www.techrepublic.com/article/how-to-send-audio-clips-via-sms-in-android/</p> |
| <p>8. The method of claim 1, wherein the third data include a voice recording.</p> | <p>ZTE infringes directly and/or indirectly by performing, inducing others to perform, and/or contributing to the performance of the method of claim 1, wherein sending the third data to the selected one or more second devices comprises transmitting a text message to at least one of the selected one or more second devices using an Internet Protocol (IP). See, e.g. 1H above. For example, contacting Android Device Manager, Find My Phone, Find My Device, Google Latitude, Google Plus, Google Hangouts, Google Maps, Google Assistant, Google Search, Google Messages, Android Messenger, Google Allo, Google Duo, Gmail, and Google Chrome includes sending a text message over IP. For example, entering and sending a message to a device includes sending the text message over IP.</p> |
| <p>9. The method of claim 1, wherein sending the third data to the selected one or more second devices comprises transmitting a text message to at least one of the selected one or more second devices using an Internet Protocol (IP).</p> | <p>ZTE infringes directly and/or indirectly by performing, inducing others to perform, and/or contributing to the performance of the method of claim 1, further comprising performing by the first device: identifying user interaction with the interactive display selecting at least one of the second set of user-selectable symbols corresponding to at least one of the second devices and user interaction with the display specifying an action and, based thereon, initiating a phone call or phone conference with the at least one</p> |
| <p>10. The method of claim 1, further comprising performing by the first device: identifying user interaction with the</p> | <p>ZTE infringes directly and/or indirectly by performing, inducing others to perform, and/or contributing to the performance of the method of claim 1, further comprising performing by the first device: identifying user interaction with the interactive display selecting at least one of the second set of user-selectable symbols corresponding to at least one of the second devices and user interaction with the display specifying an action and, based thereon, initiating a phone call or phone conference with the at least one</p> |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| US9467838 | Exemplary Supporting Evidence Regarding Accused Products |
|---|--|
| <p>interactive display selecting at least one of the second set of user-selectable symbols corresponding to at least one of the second devices and user interaction with the display specifying an action and, based thereon, initiating a phone call or phone conference with the at least one second device.</p> <p>11. The method of claim 1, further comprising performing by the first device: identifying user interaction with the interactive display selecting a particular user-selectable symbol corresponding to a particular second device and user interaction with the display specifying an action and, based thereon, initiating voice-over-IP (VOIP) communication with the particular second device. See, e.g. 1H above. For example, Find My Device allows a first device to initiate a phone call or phone conference by sending a phone number to the second device. For example, Google Maps allows a user to call another user by presenting the user of the first device with the option to select the friend's phone number to initiate call and the option to initiate communication. For example, Android Device Manager, Find My Phone, Find My Device, Google Latitude, Google Plus, Google Hangouts, Google Maps, Google Assistant, Google Search, Google Messages, Android Messenger, Google Allo, Google Duo, Gmail, and Google Chrome each provide for initiating a call and/or conference.</p> | <p>second device. See, e.g. 1H above. For example, Find My Device and Android Device Manager allow a first device to initiate a phone call or phone conference by sending a phone number to the second device. For example, Google Maps and its predecessors allow a user to call another user by presenting the user of the first device with the option to select the friend's phone number to initiate call and the option to initiate communication. For example, Android Device Manager, Find My Phone, Find My Device, Google Latitude, Google Plus, Google Hangouts, Google Maps, Google Assistant, Google Search, Google Messages, Android Messenger, Google Allo, Google Duo, Gmail, and Google Chrome each provide for initiating a call and/or conference.</p> |
| <p>11. The method of claim 1, further comprising performing by the first device: identifying user interaction with the interactive display selecting a particular user-selectable symbol corresponding to a particular second device and user interaction with the display specifying an action and, based thereon, initiating voice-over-IP (VOIP) communication with the particular second device. See, e.g. 1H above. For example, Find My Device allows a first device to initiate a phone call or phone conference by sending a phone number to the second device. For example, Google Maps allows a user to call the friend by presenting the user of the first device with the option to select the friend's phone number to initiate call and the option to initiate communication. For example, Android Device Manager, Find My Phone, Find My Device, Google Latitude, Google Plus, Google Hangouts, Google Maps, Google Assistant, Google Search, Google Messages, Android Messenger, Google Allo, Google Duo, Gmail, and Google Chrome each provide for initiating VOIP communication with another device.</p> | <p>ZTE infringes directly and/or indirectly by performing, inducing others to perform, and/or contributing to the performance of the method of claim 1, further comprising performing by the first device: identifying user interaction with the interactive display selecting a particular user-selectable symbol corresponding to a particular second device and user interaction with the display specifying an action and, based thereon, initiating voice-over-IP (VOIP) communication with the particular second device. See, e.g. 1H above. For example, Find My Device allows a first device to initiate a phone call or phone conference by sending a phone number to the second device. For example, Google Maps allows a user to call the friend by presenting the user of the first device with the option to select the friend's phone number to initiate call and the option to initiate communication. For example, Android Device Manager, Find My Phone, Find My Device, Google Latitude, Google Plus, Google Hangouts, Google Maps, Google Assistant, Google Search, Google Messages, Android Messenger, Google Allo, Google Duo, Gmail, and Google Chrome each provide for initiating VOIP communication with another device.</p> |
| <p>12. The method of claim</p> | <p>ZTE infringes directly and/or indirectly by performing, inducing others to perform, and/or contributing to</p> |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| US9467838 | Exemplary Supporting Evidence Regarding Accused Products |
|--|--|
| <p>1, further comprising performing by the first device: identifying user interaction with the particular second device and user interaction with the interactive display selecting a particular user-selectable symbol corresponding to a particular second device and user interaction with the display specifying an action and, based thereon, initiating a data call with the particular second device.</p> | <p>the performance of the method of claim 1, further comprising performing by the first device: identifying user interaction with the interactive display selecting a particular user-selectable symbol corresponding to a particular second device and user interaction with the display specifying an action and, based thereon, initiating a data call with the particular second device. See, e.g. 1H above. For example, Find My Device allows a first device to initiate a phone call or phone conference by sending a phone number to the second device. For example, Google Maps allows a user to call the friend by presenting the user of the first device with the option to select the friend's phone number to initiate call and the option to initiate a Facetime session. For example, Android Device Manager, Find My Phone, Find My Device, Google Latitude, Google Plus, Google Hangouts, Google Maps, Google Assistant, Google Search, Google Messages, Android Messenger, Google Allo, Google Duo, Gmail, and Google Chrome each provide for initiating communication upon selection of a contact.</p> |
| <p>13. The method of claim 1, wherein the first device is a personal digital assistant (PDA) or a personal computer (PC).</p> | <p>ZTE infringes directly and/or indirectly by performing, inducing others to perform, and/or contributing to the performance of the method of claim 1, wherein the first device is a personal digital assistant (PDA) or a personal computer (PC). See, e.g., 1P and 1A above. Upon information and belief, the Accused Products are forms of PDAs in that the functionality of a PDA has been subsumed into smartphones, tablets, and portable media players having the functionalities of a PDA. Upon information and belief, the Accused products are forms of personal computers in that the Accused Products are mobile or portable forms of personal computers. To the extent that it is necessary, AGIS submits that the Accused Products meet the claim limitation "the first device is a personal digital assistant (PDA) or a personal computer (PC)" under the doctrine of equivalents.</p> |
| <p>14. The method of claim 1, wherein the first device is a smart phone.</p> | <p>ZTE infringes directly and/or indirectly by performing, inducing others to perform, and/or contributing to the performance of the method of claim 1, wherein the first device is a smart phone. See, e.g., 1P and 1A above. For example, the Accused Products include a smartphone. To the extent that it is necessary, AGIS submits that tablets within the Accused Products meet the claim limitation "the first device is smartphone" under the doctrine of equivalents.</p> |
| <p>15. The method of claim 1, wherein the display of the first device is a touch</p> | <p>ZTE infringes directly and/or indirectly by performing, inducing others to perform, and/or contributing to the performance of the method of claim 1, wherein the display of the first device is a touch screen display, and wherein the user interaction with the display selecting the one or more user-selectable symbols in the</p> |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| US9467838 | Exemplary Supporting Evidence Regarding Accused Products |
|---|--|
| <p>screen display, and wherein the user interaction with the display selecting the one or more user-selectable symbols in the second set of symbols comprises touching the one or more user-selectable symbols in the second set of symbols.</p> <p>16. The method of claim 1, further comprising performing by the first device: sending updated location information comprising an updated location of the first device, the updated location information being sent based on a predetermined time passage of a predetermined time interval since sending previous location information comprising a previous location of the first device, or both. See, e.g., 1G above. For example, when a first device moves from a first to a second position, the symbol's position on the map is updated to reflect the change. Upon information and belief, the transmission of updated location information is based on a predetermined time passage or a predetermined distance change.</p> <p>For example the Accused Products utilize a location manager service such as LocationServices that monitors, among other things, periods of time that are predetermined, or displacement from a predetermined distance.</p> <p>This method is thus performed at least when a user passes a geographic threshold that results in a location update (e.g., "leaves" notification). For example, Android utilizes the following framework that meets this limitation as implemented on the Accused Products:</p> <p>https://developers.google.com/android/reference/com/google/android/gms/location/LocationServices https://developer.android.com/reference/android/location/LocationManager.html</p> | <p>second set of symbols comprising touching the one or more user-selectable symbols in the second set of symbols. See, e.g., 1P, 1A, and 1H above. For example, the Accused Devices are controlled by touch displays and selection of the symbol occurs on the touch display.</p> |
| <p>16. The method of claim 1, further comprising performing by the first device: sending updated location information comprising an updated location of the first device, the updated location information being sent based on a predetermined time passage of a predetermined time interval since sending previous location information comprising a previous location of the first device, or both. See, e.g., 1G above. For example, when a first device moves from a first to a second position, the symbol's position on the map is updated to reflect the change. Upon information and belief, the transmission of updated location information is based on a predetermined time passage or a predetermined distance change.</p> <p>For example the Accused Products utilize a location manager service such as LocationServices that monitors, among other things, periods of time that are predetermined, or displacement from a predetermined distance.</p> <p>This method is thus performed at least when a user passes a geographic threshold that results in a location update (e.g., "leaves" notification). For example, Android utilizes the following framework that meets this limitation as implemented on the Accused Products:</p> <p>https://developers.google.com/android/reference/com/google/android/gms/location/LocationServices https://developer.android.com/reference/android/location/LocationManager.html</p> | <p>ZTE infringes directly and/or indirectly by performing, inducing others to perform, and/or contributing to the performance of the method of claim 1, further comprising performing by the first device: sending updated location information comprising an updated location of the first device, the updated location information being sent based on a predetermined time interval since sending previous location information comprising a previous location of the first device, displacement of the first device by a predetermined distance relative to a previous location of the first device, or both. See, e.g., 1G above. For example, when a first device moves from a first to a second position, the symbol's position on the map is updated to reflect the change. Upon information and belief, the transmission of updated location information is based on a predetermined time passage or a predetermined distance change.</p> <p>For example the Accused Products utilize a location manager service such as LocationServices that monitors, among other things, periods of time that are predetermined, or displacement from a predetermined distance.</p> <p>This method is thus performed at least when a user passes a geographic threshold that results in a location update (e.g., "leaves" notification). For example, Android utilizes the following framework that meets this limitation as implemented on the Accused Products:</p> <p>https://developers.google.com/android/reference/com/google/android/gms/location/LocationServices https://developer.android.com/reference/android/location/LocationManager.html</p> |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| US9467838 | Exemplary Supporting Evidence Regarding Accused Products |
|---|---|
| Location of the first device, or both. | e.g. requestLocationUpdates(String provider, long minTime, float minDistance, LocationListener listener) Register for location updates using the named provider, and a pending intent. |
| 17. The method of claim 1, further comprising performing by the first device: using a Global Positioning Satellite (GPS) receiver of the first device to obtain data indicative of the location of the first device to obtain information to the first server comprising using an Internet Protocol (IP) to send the first location information to the first server. See, e.g., 1P and 1A above regarding the hardware and software including GPS and A-GPS. See, e.g., 1C and 1D regarding the employment of location services and location sharing to obtain and send locations. For example, Android Device Manager, Find My Phone, Find My Device, Google Latitude, Google Plus, Google Hangouts, Google Maps, Google Assistant, Google Search, Google Messages, Android Messenger, Google Allo, Google Duo, Gmail, and Google Chrome use location services and location sharing to obtain locations and send those locations and transmit locations to a server. | Android also makes use of “geotagging” that invokes time-based updating. ZTE infringes directly and/or indirectly by performing, inducing others to perform, and/or contributing to the performance of the method of claim 1, further comprising performing by the first device: using a Global Positioning Satellite (GPS) receiver of the first device to obtain data indicative of the location of the first device, wherein sending the first location information to the first server comprises using an Internet Protocol (IP) to send the first location information to the first server. See, e.g., 1P and 1A above regarding the hardware and software including GPS and A-GPS. See, e.g., 1C and 1D regarding the employment of location services and location sharing to obtain and send locations. For example, Android Device Manager, Find My Phone, Find My Device, Google Latitude, Google Plus, Google Hangouts, Google Maps, Google Assistant, Google Search, Google Messages, Android Messenger, Google Allo, Google Duo, Gmail, and Google Chrome use location services and location sharing to obtain locations and send those locations and transmit locations to a server. |
| 18. The method of claim 17, wherein sending the first location information to the first server further comprises sending the first location information via the Internet. See, e.g., 1P and 1A above regarding the hardware and software for transmitting data over the Internet. See, e.g., 1C and 1D regarding the employment of location services and location sharing to obtain and send locations over the Internet. For example, Android Device Manager, Find My Phone, Find My Device, Google Latitude, Google Plus, Google Hangouts, Google Maps, Google Assistant, Google Search, Google Messages, Android Messenger, Google Allo, Google Duo, Gmail, and Google Chrome use location services and location sharing to obtain locations and send those locations over the Internet for transmission to recipient devices/users. | ZTE infringes directly and/or indirectly by performing, inducing others to perform, and/or contributing to the performance of the method of claim 17, wherein sending the first location information to the first server further comprises sending the first location information via the Internet. See, e.g., 1P and 1A above regarding the hardware and software for transmitting data over the Internet. See, e.g., 1C and 1D regarding the employment of location services and location sharing to obtain and send locations over the Internet. For example, Android Device Manager, Find My Phone, Find My Device, Google Latitude, Google Plus, Google Hangouts, Google Maps, Google Assistant, Google Search, Google Messages, Android Messenger, Google Allo, Google Duo, Gmail, and Google Chrome use location services and location sharing to obtain locations and send those locations over the Internet for transmission to recipient devices/users. |
| 19. The method of claim 1, wherein participating | ZTE infringes directly and/or indirectly by performing, inducing others to perform, and/or contributing to the performance of the method of claim 1, wherein participating in the group further includes sending first |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| <p>US9467838</p> | <p>Exemplary Supporting Evidence Regarding Accused Products</p> |
|---|--|
| <p>in the group further includes sending first status information to the first server and receiving second status information from the first server, wherein the first status information comprises data indicative of a battery level of the first device, a signal strength of a wireless signal of the first device, or a combination thereof, and wherein the second status information comprises data indicative of one or more battery levels of the respective one or more second devices included in the group, one or more signal strengths of wireless signals of the respective one or more second devices included in the group, or a combination thereof. See, e.g., 1A, 1C, and 1D. For example, Android Device Manager, Find My Phone, Find My Device, Google Latitude, Google Plus, Google Hangouts, Google Maps, Google Assistant, Google Search, Google Messages, Android Messenger, Google Allo, Google Duo, Gmail, and Google Chrome show status indicators for each member of the group, including an indicator of connectivity status of each member (see below). Upon information and belief, status metadata regarding the GPS receiver, wireless signals, device battery, and other status metadata are sent and received between the Accused Products.</p> <p>For example, the Accused Products each the ability to communicate battery level and wireless networks to servers, where that battery level and wireless networks are communicated to the first device.</p> <p>Additionally, the status of the GPS receivers is also communicated within the apps. For example, the second devices communicate the accuracy of their GPS signal, as well as status so that the first device can view both the accuracy of the signal (represented by a circle around the user) as well as the status (online vs. offline).</p> <p>Additionally, status of a GPS receiver may be communicated through notifications that communicate when a device has passed a location threshold.</p> | <p>status information to the first server and receiving second status information from the first server, wherein the first status information comprises data indicative of a battery level of the first device, a signal strength of a wireless signal of the first device, a status of a Global Positioning Satellite (GPS) receiver of the first device, or a combination thereof, and wherein the second status information comprises data indicative of one or more battery levels of the respective one or more second devices included in the group, one or more signal strengths of wireless signals of the respective one or more second devices included in the group, or a combination thereof. See, e.g., 1A, 1C, and 1D. For example, Android Device Manager, Find My Phone, Find My Device, Google Latitude, Google Plus, Google Hangouts, Google Maps, Google Assistant, Google Search, Google Messages, Android Messenger, Google Allo, Google Duo, Gmail, and Google Chrome show status indicators for each member of the group, including an indicator of connectivity status of each member (see below). Upon information and belief, status metadata regarding the GPS receiver, wireless signals, device battery, and other status metadata are sent and received between the Accused Products.</p> <p>For example, the Accused Products each the ability to communicate battery level and wireless networks to servers, where that battery level and wireless networks are communicated to the first device.</p> <p>Additionally, the status of the GPS receivers is also communicated within the apps. For example, the second devices communicate the accuracy of their GPS signal, as well as status so that the first device can view both the accuracy of the signal (represented by a circle around the user) as well as the status (online vs. offline).</p> <p>Additionally, status of a GPS receiver may be communicated through notifications that communicate when a device has passed a location threshold.</p> |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| Exemplary Supporting Evidence Regarding Accused Products | |
|--|---|
| <p>US9467838</p> <p>status of GPS receivers of the second devices included in the group, or a combination thereof.</p> <p>20. The method of claim 1, wherein the second georeferenced map data comprise a satellite image or aerial photograph.</p> | <p>ZTE infringes directly and/or indirectly by performing, inducing others to perform, and/or contributing to the performance of the method of claim 1, wherein the second georeferenced map data comprise a satellite image or aerial photograph. See, e.g., 1D and 1G. For example and shown below, Android Device Manager, Find My Phone, Find My Device, Google Latitude, Google Plus, Google Hangouts, Google Maps, Google Assistant, Google Search, Google Messages, Android Messenger, Google Allo, Google Duo, Gmail, and Google Chrome show maps in satellite and/or aerial formats.</p> |
| <p>21. The method of claim 1, wherein the spatial coordinates comprise latitude and longitude coordinates.</p> | <p>ZTE infringes directly and/or indirectly by performing, inducing others to perform, and/or contributing to the performance of the method of claim 1, wherein the spatial coordinates comprise latitude and longitude coordinates. See, e.g., 1C, 1D, and 1G. For example, the Accused Products present maps with, at least, latitude and longitude coordinates.</p> |
| <p>22[A]. The method of claim 1, further comprising identifying, by the first device, user interaction with the display selecting a particular user-selectable symbol positioned on the second georeferenced map and corresponding to a particular second device, wherein identifying the user interaction selecting the particular user-selectable symbol comprises: detecting user selection of a portion of the interactive display corresponding to a position on the second georeferenced map. See, e.g., 1G and 1H. For example, the user of the first device selects a symbol on the map by touching the display. The selection occurs on a touch display of the first device which includes logic for detecting the touch position and corresponding the touch position to a position on the georeferenced map. See, e.g., 1P an 1A.</p> | <p>ZTE infringes directly and/or indirectly by performing, inducing others to perform, and/or contributing to the performance of the method of claim 1, further comprising identifying, by the first device, user interaction with the display selecting a particular user-selectable symbol positioned on the second georeferenced map and corresponding to a particular second device, wherein identifying the user interaction selecting the particular user-selectable symbol comprises: detecting user selection of a portion of the interactive display corresponding to a position on the second georeferenced map. See, e.g., 1G and 1H. For example, the user of the first device selects a symbol on the map by touching the display. The selection occurs on a touch display of the first device which includes logic for detecting the touch position and corresponding the touch position to a position on the georeferenced map. See, e.g., 1P an 1A.</p> |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| Exemplary Supporting Evidence Regarding Accused Products | |
|--|---|
| <p>US9467838</p> <p>user interaction selecting the particular user-selectable symbol comprises: detecting user selection of a portion of the interactive display corresponding to a position on the second georeferenced map;</p> <p>[22B] based at least in part on coordinates of the selected position on the second georeferenced map and on the second georeferenced map data relating positions on the second georeferenced map to spatial coordinates, determining spatial coordinates of a location represented by the selected position on the second georeferenced map;</p> | <p>ZTE infringes directly and/or indirectly by performing, inducing others to perform, and/or contributing to the performance of, based at least in part on coordinates of the selected position on the second georeferenced map and on the second georeferenced map data relating positions on the second georeferenced map to spatial coordinates, determining spatial coordinates of a location represented by the selected position on the second georeferenced map. See, e.g., 1P, 1A, 1G and 1H. Upon information and belief, the first device determines spatial coordinates representing selected positions on the map.</p> <p>For example, when markers, such as symbols, are added to maps, they are added based on longitude and latitude.</p> <p>Add a map</p> <p>Display a map, using the Google Maps Android API.</p> <p>Add a <fragment> element to your activity's layout file, activity_maps.xml. This element defines a SupportMapFragment to act as a container for the map and to provide access to the GoogleMap object. The tutorial uses the Android support library version of the map fragment, to ensure backward compatibility with earlier versions of the Android framework.</p> <pre style="background-color: #333; color: #fff; padding: 10px;"> <fragment xmlns:android="http://schemas.android.com/apk/res/android" xmlns:tools="http://schemas.android.com/tools" android:id="@+id/map" android:name="com.google.android.gms.maps.SupportMapFragment" </pre> |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| | |
|-----------|---|
| US9467838 | <p data-bbox="191 709 228 1560">Exemplary Supporting Evidence Regarding Accused Products</p> <pre data-bbox="240 709 349 1533"> android:layout_width="match_parent" android:layout_height="match_parent" tools:context="com.example.mapwithmarker.MapsMarkerActivity" /> </pre> <p data-bbox="423 247 493 1560">In your activity's onCreate() method, set the layout file as the content view. Get a handle to the map fragment by calling <code>FragmentManager.findById()</code>. Then use <code>getMapAsync()</code> to register for the map callback:</p> <pre data-bbox="537 426 966 1560"> @Override protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); // Retrieve the content view that renders the map. setContentView(R.layout.activity_maps); // Get the SupportMapFragment and request notification // when the map is ready to be used. SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragmentManager() .findFragmentById(R.id.map); mapFragment.getMapAsync(this); } </pre> <p data-bbox="1003 247 1068 1560">Implement the <code>OnMapReadyCallback</code> interface and override the <code>onMapReady()</code> method, to set up the map when the <code>GoogleMap</code> object is available:</p> <pre data-bbox="1117 800 1421 1560"> public class MapsMarkerActivity extends AppCompatActivity implements OnMapReadyCallback { // Include the onCreate() method here too, as described above. @Override public void onMapReady(GoogleMap googleMap) { // Add a marker in Sydney, Australia, // and move the map's camera to the same location. LatLng sydney = new LatLng(-33.852, 151.211); </pre> |
|-----------|---|

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| US9467838 | Exemplary Supporting Evidence Regarding Accused Products |
|--|---|
| | <pre> googleMap.addMarker(new MarkerOptions().position(sydney) .title("Marker in Sydney")); googleMap.moveCamera(CameraUpdateFactory.newLatLng(sydney)); } } </pre> |
| <p>[22C] and identifying the particular user-selectable symbol based, at least in part, on the spatial coordinates represented by the selected position.</p> | <p>See, also, https://developers.google.com/maps/documentation/android-api/marker</p> <p>ZTE infringes directly and/or indirectly by performing, inducing others to perform, and/or contributing to the performance of identifying the particular user-selectable symbol based, at least in part, on the spatial coordinates represented by the selected position. See, e.g., 1P, 1A, 1G and 1H. Upon information and belief, the first device identifies a symbol based on the determined coordinates.</p> <p>Each marker corresponds to a spatial coordinate (i.e. longitude/latitude) as well as an x/y coordinate on the map display.</p> <p>See, e.g.,</p> <p>https://developers.google.com/maps/documentation/android-api/map-with-marker</p> <p>https://developers.google.com/maps/documentation/android-api/marker</p> |
| <p>23[A]. The method of claim 22, wherein identifying the particular user-selectable symbol based, at least in part, on the spatial coordinates represented by the selected position comprises: searching a database of entities for an entity located nearest to the spatial coordinates represented by the selected position</p> | <p>ZTE infringes directly and/or indirectly by performing, inducing others to perform, and/or contributing to the performance of the method of claim 22, wherein identifying the particular user-selectable symbol based, at least in part, on the spatial coordinates represented by the selected position comprises: searching a database of entities for an entity located nearest to the spatial coordinates represented by the selected position, wherein the entities represented by data in the database include the one or more second devices included in the group, wherein the database data include locations of the respective entities, and wherein the database is searchable by location. See, e.g., 1G, 1H, and 22. Upon information and belief, the Accused Products include a database located at the first device or send a search request to a server for performing the searching methods of this limitation.</p> <p>See, e.g.,</p> |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| Exemplary Supporting Evidence Regarding Accused Products | |
|--|---|
| <p>US9467838</p> <p>selected position, wherein the entities represented by data in the database include the one or more second devices included in the group, wherein the database data include locations of the respective entities, and wherein the database is searchable by location;</p> | <p>Selection with Markers: https://developers.google.com/maps/documentation/android-api/marker</p> <p>Queries with GeoTagging database: https://developers.google.com/android/reference/com/google/android/gms/location/places/GeoDataClient</p> |
| <p>[23B] and based on a result of searching the database, identifying the particular second device as the entity located nearest to the spatial coordinates represented by the selected position, wherein the particular user-selectable symbol corresponds to the particular second device.</p> | <p>ZTE infringes directly and/or indirectly by performing, inducing others to perform, and/or contributing to the performance of, based on a result of searching the database, identifying the particular second device as the entity located nearest to the spatial coordinates represented by the selected position, wherein the particular user-selectable symbol corresponds to the particular second device. See, e.g. 1G and 1H. Upon information and belief, the Accused Devices include a database located either at the first device or send a search request to a server for performing the searching methods of this limitation.</p> <p>See, e.g., Android Maps and Geotagging APIs:</p> <p>Instantiate the Places API clients</p> <p>The following interfaces provide the primary entry points to the Google Places API for Android:</p> <ul style="list-style-type: none"> • The GeoDataClient provides access to Google's database of local place and business information. • The PlaceDetectionClient provides quick access to the device's current place, and offers the opportunity to report the location of the device at a particular place. <p>The LocationServices interface is the main entry point for Android location services.</p> <p>To use the APIs, instantiate GeoDataClient, PlaceDetectionClient, and FusedLocationProviderClient in your fragment's or activity's onCreate() method, as shown in the following code sample:</p> |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| US9467838 | Exemplary Supporting Evidence Regarding Accused Products |
|---|--|
| <p>24[A]. The method of claim 23, wherein the entity is a first entity, and wherein the method further comprises performing by the first device: receiving user input via user interaction with the interactive display of the first device, the user input</p> | <pre> protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity_main); // Construct a GeoDataClient. mGeoDataClient = Places.getGeoDataClient(this, null); // Construct a PlaceDetectionClient. mPlaceDetectionClient = Places.getPlaceDetectionClient(this, null); // Construct a FusedLocationProviderClient. mFusedLocationProviderClient = LocationServices.getFusedLocationProviderClient(this); </pre> <p>See, e.g., https://developers.google.com/maps/documentation/android-api/current-place-tutorial https://developers.google.com/android/reference/com/google/android/gms/location/places/GeoDataClient ZTE infringes directly and/or indirectly by performing, inducing others to perform, and/or contributing to the performance of the method of claim 23, wherein the entity is a first entity, and wherein the method further comprises performing by the first device: receiving user input via user interaction with the interactive display of the first device, the user input specifying a location and a symbol corresponding to a second entity other than the first device and the one or more second devices included in the group. See, e.g., 1G, 1H, 22, and 23. For example, the user of the first device selects a symbol on the map by touching the display. The selection occurs on a touch display of the first device which includes logic for detecting the touch position and corresponding the touch position to a position on the georeferenced map. See, e.g., 1P an 1A. For example, the user of the first device can select other symbols on the touch display.</p> |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| Exemplary Supporting Evidence Regarding Accused Products | |
|--|---|
| <p>US9467838</p> <p>specifying a location and a symbol corresponding to a second entity other than the first device and the one or more second devices included in the group;</p> | <p>[24B] and based on the user input, adding the user-specified symbol to the interactive display at a position on the second georeferenced map corresponding to the user-specified location of the second entity.</p> |
| | <p>ZTE infringes directly and/or indirectly by performing, inducing others to perform, and/or contributing to the performance of, based on the user input, adding the user-specified symbol to the interactive display at a position on the second georeferenced map corresponding to the user-specified location of the second entity. See, e.g., 1G and 1H. For example, the user of the first device selects a symbol on the map by touching the display. The selection occurs on a touch display of the first device which includes logic for detecting the touch position and corresponding the touch position to a position on the georeferenced map. See, e.g., 1P and 1A. For example, the user of the first device can select other symbols on the touch display and a user can drop a marker on a map and/or share a location by specifying another user/device, specifying a method for sharing, and specifying additional data such as text to be sent with the location/marker.</p> |
| <p>25. The method of claim 24, further comprising performing by the first device: transmitting the user-specified symbol and location of the second entity to the one or more second devices or more second devices included in the group for addition of the user-specified symbol to respective interactive displays of the one or more second devices at respective positions on respective georeferenced maps corresponding to the user-specified location of the second entity. See, e.g., 1P, 1A, 22, 23, and 24. For example, the user of the first device can select other symbols on the touch display and a user can drop a marker on a map and/or share a location by specifying another user/device, specifying a method for sharing, and specifying additional data such as text to be sent with the location/marker.</p> <p>See, e.g., Placing a Marker: https://developers.google.com/maps/documentation/android-api/marker based on queries with GeoTagging database: https://developers.google.com/android/reference/com/google/android/gms/location/places/GeoDataClient</p> | <p>ZTE infringes directly and/or indirectly by performing, inducing others to perform, and/or contributing to the performance of the method of claim 24, further comprising performing by the first device: transmitting the user-specified symbol and location of the second entity to the one or more second devices included in the group for addition of the user-specified symbol to respective interactive displays of the one or more second devices at respective positions on respective georeferenced maps corresponding to the user-specified location of the second entity. See, e.g., 1P, 1A, 22, 23, and 24. For example, the user of the first device can select other symbols on the touch display and a user can drop a marker on a map and/or share a location by specifying another user/device, specifying a method for sharing, and specifying additional data such as text to be sent with the location/marker.</p> <p>See, e.g., Placing a Marker: https://developers.google.com/maps/documentation/android-api/marker based on queries with GeoTagging database: https://developers.google.com/android/reference/com/google/android/gms/location/places/GeoDataClient</p> |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| Exemplary Supporting Evidence Regarding Accused Products | |
|---|---|
| US9467838 | Sharing a link: https://support.google.com/maps/answer/144361?co=GENIE.Platform%3DDesktop&hl=en |
| maps corresponding to the user-specified location of the second entity. | ZTE infringes directly and/or indirectly by performing, inducing others to perform, and/or contributing to the performance of the method of claim 25, wherein the user input further specifies information associated with the second entity, and wherein the method further comprises performing by the first device: transmitting the user-specified information associated with the second entity to the one or more second devices included in the group. See, e.g., 1P, 1A, 1B, 22, 23, 24, and 25. For example, the user of the first device can select other symbols on the touch display and a user can drop a marker on a map and/or share a location by specifying another user/device, specifying a method for sharing, and specifying additional data such as text to be sent with the location/marker. |
| 26. The method of claim 25, wherein the user input further specifies information associated with the second entity, and wherein the method further comprises performing by the first device: transmitting the user-specified information associated with the second entity to the one or more second devices included in the group. | ZTE infringes directly and/or indirectly by performing, inducing others to perform, and/or contributing to the performance of the method of claim 26, wherein the information comprises a category of the second entity. See, e.g., 1P, 1A, 22, 23, 24, 25, and 26. For example, the user of the first device can select other symbols on the touch display and a user can drop a marker on a map and/or share a location by specifying another user/device, specifying a method for sharing, and specifying additional data such as text to be sent with the location/marker. |
| 27. The method of claim 26, wherein the information comprises a category of the second entity. | ZTE infringes directly and/or indirectly by performing, inducing others to perform, and/or contributing to the performance of the method of claim 27, wherein the category comprises a vehicle, a person, an event, a site, a building, or a facility. See, e.g., 1P, 1A, 22, 23, 24, 25, and 26. For example, the user of the first device can select other symbols on the touch display and a user can drop a marker on a map and/or share a location by specifying another user/device, specifying a method for sharing, and specifying additional data such as text to be sent with the location/marker. The location can specify a person, an event, or some other location, building, or facility. |
| 28. The method of claim 27, wherein the category comprises a vehicle, a person, an event, a site, a building, or a facility. | ZTE infringes directly and/or indirectly by performing, inducing others to perform, and/or contributing to the performance of the method of claim 27, wherein the category comprises a vehicle, a person, an event, a site, a building, or a facility. See, e.g., 1P, 1A, 22, 23, 24, 25, and 26. For example, the user of the first device can select other symbols on the touch display and a user can drop a marker on a map and/or share a location by specifying another user/device, specifying a method for sharing, and specifying additional data such as text to be sent with the location/marker. The location can specify a person, an event, or some other location, building, or facility. |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| US9467838 | | Exemplary Supporting Evidence Regarding Accused Products |
|--|--|---|
| 29. The method of claim 26, wherein the information comprises an image. | ZTE infringes directly and/or indirectly by performing, inducing others to perform, and/or contributing to the performance of the method of claim 26, wherein the information comprises an image. See, e.g., 1P, 1A, 22, 23, 24, 25, and 26. For example, the user of the first device can select other symbols on the touch display and a user can drop a marker on a map and/or share a location by specifying another user/device, specifying a method for sharing, and specifying additional data such as text to be sent with the location/marker. When information is shared by message, the user may also transmit an image. | |
| 30. The method of claim 26, wherein the information comprises at least one type of information selected from the group consisting of text and video. | ZTE infringes directly and/or indirectly by performing, inducing others to perform, and/or contributing to the performance of the method of claim 26, wherein the information comprises at least one type of information selected from the group consisting of text and video. See, e.g., 1P, 1A, 22, 23, 24, 25, and 26. For example, the user of the first device can select other symbols on the touch display and a user can drop a marker on a map and/or share a location by specifying another user/device, specifying a method for sharing, and specifying additional data such as text to be sent with the location/marker. When information is shared by message, the user may also transmit text and/or video. | |
| 31. The method of claim 26, further comprising performing by the first device: identifying user interaction with the interactive display selecting the symbol corresponding to the second entity, and based thereon, displaying the information associated with the second entity. | ZTE infringes directly and/or indirectly by performing, inducing others to perform, and/or contributing to the performance of the method of claim 26, further comprising performing by the first device: identifying user interaction with the interactive display selecting the symbol corresponding to the second entity, and based thereon, displaying the information associated with the second entity. See, e.g., 1P, 1A, 22, 23, 24, 25, and 26. For example, the user of the first device can select other symbols on the touch display and a user can drop a marker on a map and/or share a location by specifying another user/device, specifying a method for sharing, and specifying additional data such as text to be sent with the location/marker. | |
| 32. The method of claim 31, wherein the first device uses an Internet Protocol to transmit the user-specified symbol, location, and information associated with the second entity. | ZTE infringes directly and/or indirectly by performing, inducing others to perform, and/or contributing to the performance of the method of claim 31, wherein the first device uses an Internet Protocol to transmit the user-specified symbol, location, and information associated with the second entity. See, e.g., 1P, 1A, 22, 23, 24, 25, 26, and 31. For example, the user of the first device can select other symbols on the touch display and a user can drop a marker on a map and/or share a location by specifying another user/device, specifying a method for sharing, and specifying additional data such as text to be sent with the | |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| US9467838 | Exemplary Supporting Evidence Regarding Accused Products |
|--|---|
| <p>information associated with the second entity.</p> | <p>location/marker. See, e.g., 1P and 1A above regarding the hardware and software for transmitting data over the Internet. See, e.g., 1C and 1D regarding the employment of location services and location sharing to obtain and send locations over the Internet. For example, both Find My Friends and Find My Family use location services and location sharing to obtain locations and send those locations to corresponding servers over the Internet for transmission to recipient devices/users. For example, sending messages on the Accused Devices using Android Device Manager, Find My Phone, Find My Device, Google Latitude, Google Plus, Google Hangouts, Google Maps, Google Assistant, Google Search, Google Messages, Android Messenger, Google Allo, Google Duo, Gmail, Google Chrome, and/or associated services includes sending communication over IP.</p> |
| <p>33. The method of claim 26, further comprising performing by the first device: adding data representing the spatial coordinates of the location of the second entity and data representing the second entity to the database.</p> | <p>ZTE infringes directly and/or indirectly by performing, inducing others to perform, and/or contributing to the performance of the method of claim 26, further comprising performing by the first device: adding data representing the spatial coordinates of the location of the second entity and data representing the information associated with the second entity to the database. See, e.g., 1P, 1A, 22, 23, 24, 25, and 26. For example, the user of the first device can select other symbols on the touch display and a user can drop a marker on a map and/or share a location by specifying another user/device, specifying a method for sharing, and specifying additional data such as text to be sent with the location/marker.</p> |
| <p>34[A]. The method of claim 24, wherein the portion of the interactive display is a first portion, wherein the position of the symbol corresponding to the particular second device is a first position, and wherein receiving the user input specifying the location of the second entity comprises: detecting user selection of a second portion of the interactive display corresponding to a second position on the second georeferenced map. See, e.g., 1G and 1H. For example, the user of the first device selects a symbol on the map by touching the display. The selection occurs on a touch display of the first device which includes logic for detecting the touch position and corresponding the touch position to a position on the georeferenced map. See, e.g., 1P an 1A.</p> | <p>ZTE infringes directly and/or indirectly by performing, inducing others to perform, and/or contributing to the performance of the method of claim 24, wherein the portion of the interactive display is a first portion, wherein the position of the symbol corresponding to the particular second device is a first position, and wherein receiving the user input specifying the location of the second entity comprises: detecting user selection of a second portion of the interactive display corresponding to a second position on the second georeferenced map. See, e.g., 1G and 1H. For example, the user of the first device selects a symbol on the map by touching the display. The selection occurs on a touch display of the first device which includes logic for detecting the touch position and corresponding the touch position to a position on the georeferenced map. See, e.g., 1P an 1A.</p> |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| Exemplary Supporting Evidence Regarding Accused Products | |
|---|---|
| <p>US9467838</p> <p>location of the second entity comprises: detecting user selection of a second portion of the interactive display corresponding to a second position on the second georeferenced map;</p> | <p>ZTE infringes directly and/or indirectly by performing, inducing others to perform, and/or contributing to the performance of, based at least in part on coordinates of the second position on the second georeferenced map and on the second georeferenced map data relating positions on the second georeferenced map to spatial coordinates, determining spatial coordinates of a location represented by the second position on the second georeferenced map, wherein the location represented by the second position is the location of the second entity. See, e.g., 1P, 1A, 1G and 1H. Upon information and belief, the first device determines spatial coordinates representing selected positions on the map.</p> |
| <p>[34B] and based at least in part on coordinates of the second position on the second georeferenced map and on the second georeferenced map data relating positions on the second georeferenced map to spatial coordinates, determining spatial coordinates of a location represented by the second position on the second georeferenced map, wherein the location represented by the second position is the location of the second entity.</p> | <p>ZTE infringes directly and/or indirectly by performing, inducing others to perform, and/or contributing to the performance of the method of claim 23, wherein the database is stored on the first device. See, e.g., 1P,</p> |
| <p>35. The method of claim 23, wherein the database</p> | <p>ZTE infringes directly and/or indirectly by performing, inducing others to perform, and/or contributing to the performance of the method of claim 23, wherein the database is stored on the first device. See, e.g., 1P,</p> |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| US9467838 | Exemplary Supporting Evidence Regarding Accused Products |
|--|---|
| is stored on the first device. | 1A, 1G and 1H. Upon information and belief, the first device keeps one or more databases for the purposes of Android Device Manager, Find My Phone, Find My Device, Google Latitude, Google Plus, Google Hangouts, Google Maps, Google Assistant, Google Search, Google Messages, Android Messenger, Google Allo, Google Duo, Gmail, and Google Chrome. |
| 36. The method of claim 23, wherein the database is stored on the first server. | ZTE infringes directly and/or indirectly by performing, inducing others to perform, and/or contributing to the performance of the method of claim 23, wherein the database is stored on the first server. See, e.g., 1P, 1A, 1G and 1H. Upon information and belief, the Accused Products communicate with servers keeping databases for the same purposes discussed in claims 1 and 23, e.g., Android Device Manager, Find My Phone, Find My Device, Google Latitude, Google Plus, Google Hangouts, Google Maps, Google Assistant, Google Search, Google Messages, Android Messenger, Google Allo, Google Duo, Gmail, and Google Chrome. See, e.g., 1E and 1F. |
| 37[A]. The method of claim 1, further comprising performing by the first device: receiving user-specified information transmitted by a particular second device, the user-specified information including a user-specified location and a user-specified symbol corresponding to an entity other than the first device and the one or more second devices included in the group. See, e.g., 1P and 1A above regarding the hardware and software for transmitting data over the Internet. See, e.g., 1C and 1D regarding the use of location services and location sharing to obtain and send locations over the Internet. For example, Android Device Manager, Find My Phone, Find My Device, Google Latitude, Google Plus, Google Hangouts, Google Maps, Google Assistant, Google Search, Google Messages, Android Messenger, Google Allo, Google Duo, Gmail, and Google Chrome use location services and location sharing to obtain locations and send/receive those locations to corresponding servers over the Internet for transmission to recipient devices/users. For example, receiving messages on the Accused Devices using Android Device Manager, Find My Phone, Find My Device, Google Latitude, Google Plus, Google Hangouts, Google Maps, Google Assistant, Google Search, Google Messages, Android Messenger, Google Duo, Gmail, and Google Chrome and associated servers for the same includes sending communication over IP. | |
| [37B] and adding the user-specified symbol to the interactive display at a position on the second | ZTE infringes directly and/or indirectly by performing, inducing others to perform, and/or contributing to the performance of adding the user-specified symbol to the interactive display at a position on the second georeferenced map corresponding to the user-specified location. See, e.g., 1P and 1A above regarding the hardware and software for transmitting data over the Internet. See, e.g., 1C and 1D regarding the use of |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| US9467838 | Exemplary Supporting Evidence Regarding Accused Products |
|---|--|
| <p>georeferenced map corresponding to the user-specified location.</p> | <p>location services and location sharing to obtain and send locations over the Internet. For example, Android Device Manager, Find My Phone, Find My Device, Google Latitude, Google Plus, Google Hangouts, Google Maps, Google Assistant, Google Search, Google Messages, Android Messenger, Google Allo, Google Duo, Gmail, and Google Chrome use location services and location sharing to obtain locations and send/receive those locations to corresponding servers over the Internet for transmission to recipient devices/users. For example, receiving messages on the Accused Devices using Android Device Manager, Find My Phone, Find My Device, Google Latitude, Google Plus, Google Hangouts, Google Maps, Google Assistant, Google Search, Google Messages, Android Messenger, Google Duo, Gmail, and Google Chrome as associated services includes sending communication over IP. For example, a first device receives shared locations and markers from second devices and displays them on a map on the first device.</p> |
| <p>38. The method of claim 37, further comprising performing by the first device: identifying user interaction with the interactive display selecting the user-specified symbol corresponding to the entity, and based thereon, displaying information associated with the entity, wherein information further includes the information associated with the entity. See, e.g., 1P and 1A above regarding the hardware and software for transmitting data over the Internet. See, e.g., 1C and 1D regarding the use of location services and location sharing to obtain and send locations over the Internet. For example, Android Device Manager, Find My Phone, Find My Device, Google Latitude, Google Plus, Google Hangouts, Google Maps, Google Assistant, Google Search, Google Messages, Android Messenger, Google Allo, Google Duo, Gmail, and Google Chrome use location services and location sharing to obtain locations and send/receive those locations to corresponding servers over the Internet for transmission to recipient devices/users. For example, receiving messages on the Accused Devices using Android Device Manager, Find My Phone, Find My Device, Google Latitude, Google Plus, Google Hangouts, Google Maps, Google Assistant, Google Search, Google Messages, Android Messenger, Google Duo, Gmail, and Google Chrome and associated services includes sending communication over IP. For example, a first device receives shared locations and markers from second devices and displays them on a map on the first device. A user of the first device may then interact with the received shared location and/or marker and the map data and specify additional information.</p> | <p>ZTE infringes directly and/or indirectly by performing, inducing others to perform, and/or contributing to the performance of the method of claim 37, further comprising performing by the first device: identifying user interaction with the interactive display selecting the user-specified symbol corresponding to the entity, and based thereon, displaying information associated with the entity, wherein the user-specified information further includes the information associated with the entity. See, e.g., 1P and 1A above regarding the hardware and software for transmitting data over the Internet. See, e.g., 1C and 1D regarding the use of location services and location sharing to obtain and send locations over the Internet. For example, Android Device Manager, Find My Phone, Find My Device, Google Latitude, Google Plus, Google Hangouts, Google Maps, Google Assistant, Google Search, Google Messages, Android Messenger, Google Allo, Google Duo, Gmail, and Google Chrome use location services and location sharing to obtain locations and send/receive those locations to corresponding servers over the Internet for transmission to recipient devices/users. For example, receiving messages on the Accused Devices using Android Device Manager, Find My Phone, Find My Device, Google Latitude, Google Plus, Google Hangouts, Google Maps, Google Assistant, Google Search, Google Messages, Android Messenger, Google Duo, Gmail, and Google Chrome and associated services includes sending communication over IP. For example, a first device receives shared locations and markers from second devices and displays them on a map on the first device. A user of the first device may then interact with the received shared location and/or marker and the map data and specify additional information.</p> |
| <p>39. The method of claim 1, wherein the message including the identifier</p> | <p>ZTE infringes directly and/or indirectly by performing, inducing others to perform, and/or contributing to the performance of the method of claim 1, wherein the message including the identifier corresponding to the group is a first message, and wherein the method further comprises performing by the first device:</p> |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| US9467838 | Exemplary Supporting Evidence Regarding Accused Products |
|---|--|
| <p>corresponding to the group is a first message, and wherein the method further comprises performing by the first device: sending, to a particular second device via the first server, a second message related to remotely controlling the particular second device to perform an action, wherein the particular second device is configured to perform the action based on receiving the second message. See, e.g., 1H. The user of the first device sends an action by message, via corresponding servers, to remotely control the second device upon receiving a message from the server corresponding to the action by message.</p> | <p>sending, to a particular second device via the first server, a second message related to remotely controlling the particular second device to perform an action, wherein the particular second device is configured to perform the action based on receiving the second message. See, e.g., 1H. The user of the first device sends an action by message, via corresponding servers, to remotely control the second device upon receiving a message from the server corresponding to the action by message.</p> |
| <p>40. The method of claim 39, wherein the second message indicates the action to be performed, and wherein the action is selected from the group consisting of playing audio, initiating a phone call, vibrating, converting text to speech, changing sound intensity, and displaying information. See, e.g., 1H. For example, a user may play a sound, initiate a call, and display information. Upon information and belief, the second device vibrates, converts text to speech, and changes sound intensity in accordance with the claim.</p> | <p>ZTE infringes directly and/or indirectly by performing, inducing others to perform, and/or contributing to the performance of the method of claim 39, wherein the second message indicates the action to be performed, and wherein the action is selected from the group consisting of playing audio, initiating a phone call, vibrating, converting text to speech, changing sound intensity, and displaying information. See, e.g., 1H. For example, a user may play a sound, initiate a call, and display information. Upon information and belief, the second device vibrates, converts text to speech, and changes sound intensity in accordance with the claim.</p> |
| <p>41. The method of claim</p> | <p>ZTE infringes directly and/or indirectly by performing, inducing others to perform, and/or contributing to</p> |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| US9467838 | Exemplary Supporting Evidence Regarding Accused Products |
|--|--|
| <p>40, wherein playing audio comprises playing an audio message announcing an emergency.</p> | <p>the performance of the method of claim 40, wherein playing audio comprises playing an audio message announcing an emergency. See, e.g., 1H and 40. For example, a user may play a sound, initiate a call, and display information to announce an emergency, e.g. lost device.</p> |
| <p>42. The method of claim 1, wherein the message including the identifier corresponding to the group is a first message, and wherein the method further comprises performing by the first device: receiving a second message sent by a particular second device; and performing the indicated action. See, e.g., 1H. The first device, because it is running Android Device Manager, Find My Phone, Find My Device, Google Latitude, Google Plus, Google Hangouts, Google Maps, Google Assistant, Google Search, Google Messages, Android Messenger, Google Allo, Google Duo, Gmail, and Google Chrome, receives and performs actions identified in 1H.</p> | <p>ZTE infringes directly and/or indirectly by performing, inducing others to perform, and/or contributing to the performance of the method of claim 1, wherein the message including the identifier corresponding to the group is a first message, and wherein the method further comprises performing by the first device: receiving a second message sent by a particular second device, wherein the second message indicates an action to be performed by the first device; and performing the indicated action. See, e.g., 1H. The first device, because it is running Android Device Manager, Find My Phone, Find My Device, Google Latitude, Google Plus, Google Hangouts, Google Maps, Google Assistant, Google Search, Google Messages, Android Messenger, Google Allo, Google Duo, Gmail, and Google Chrome, receives and performs actions identified in 1H.</p> |
| <p>43. The method of claim 42, wherein the indicated action is selected from the group consisting of playing audio, initiating a phone call, vibrating, converting text to speech, changing sound intensity, and displaying information. See, e.g., 1H. For example, a user may play a sound, initiate a call, and display information. Upon information and belief, the second device vibrates, converts text to speech, and changes sound intensity in accordance with the claim.</p> | <p>ZTE infringes directly and/or indirectly by performing, inducing others to perform, and/or contributing to the performance of the method of claim 42, wherein the indicated action is selected from the group consisting of playing audio, initiating a phone call, vibrating, converting text to speech, changing sound intensity, and displaying information. See, e.g., 1H. For example, a user may play a sound, initiate a call, and display information. Upon information and belief, the second device vibrates, converts text to speech, and changes sound intensity in accordance with the claim.</p> |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| Exemplary Supporting Evidence Regarding Accused Products | |
|---|---|
| <p>US9467838 information.</p> <p>44[A]. The method of claim 1, further comprising performing by the first device: presenting another symbol on the second georeferenced map corresponding to a fixed location and associated with a telephone number;</p> | <p>ZTE infringes directly and/or indirectly by performing, inducing others to perform, and/or contributing to the performance of the method of claim 1, further comprising performing by the first device: presenting another symbol on the second georeferenced map corresponding to a fixed location and associated with a telephone number. See, e.g., 1G and 1H. For example, the first device's map can display icons representing fixed entities which correspond to a location and a telephone number.</p> |
| <p>[44B] and receiving user selection of the other symbol and, based thereon, initiating a telephone call to the telephone number associated with the symbol.</p> | <p>ZTE infringes directly and/or indirectly by performing, inducing others to perform, and/or contributing to the performance of receiving user selection of the other symbol and, based thereon, initiating a telephone call to the telephone number associated with the symbol. See, e.g., 1G and 1H. For example, the user of the first device can select the icon of a fixed entity to retrieve a telephone number and call the telephone number associated with the fixed entity. See, e.g., 1C and 1D for additional map and location data.</p> |
| <p>45. The method of claim 1, further comprising performing, by the first device: presenting a symbol corresponding to a facility, wherein the facility is selected from the group consisting of a hospital, a police station, and a fire station, and wherein the symbol corresponding to the facility is positioned on the second georeferenced map at a position corresponding to a location of the facility. See, e.g. 1H and 44A-B. See, e.g., 1C and 1D for additional map and location data.</p> | <p>ZTE infringes directly and/or indirectly by performing, inducing others to perform, and/or contributing to the performance of the method of claim 1, further comprising performing, by the first device: presenting a symbol corresponding to a facility, wherein the facility is selected from the group consisting of a hospital, a police station, and a fire station, and wherein the symbol corresponding to the facility is positioned on the second georeferenced map at a position corresponding to a location of the facility. See, e.g. 1H and 44A-B. See, e.g., 1C and 1D for additional map and location data.</p> |
| <p>45. The method of claim 1, further comprising performing, by the first device: presenting a symbol corresponding to a facility, wherein the facility is selected from the group consisting of a hospital, a police station, and a fire station, and wherein the symbol corresponding to the</p> | <p>ZTE infringes directly and/or indirectly by performing, inducing others to perform, and/or contributing to the performance of the method of claim 1, further comprising performing, by the first device: presenting a symbol corresponding to a facility, wherein the facility is selected from the group consisting of a hospital, a police station, and a fire station, and wherein the symbol corresponding to the facility is positioned on the second georeferenced map at a position corresponding to a location of the facility. See, e.g. 1H and 44A-B. See, e.g., 1C and 1D for additional map and location data.</p> |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| Exemplary Supporting Evidence Regarding Accused Products | |
|---|--|
| US9467838 facility is positioned on the second georeferenced map at a position corresponding to a location of the facility. | ZTE infringes directly and/or indirectly by performing, inducing others to perform, and/or contributing to the performance of the method of claim 45, further comprising performing, by the first device: identifying user interaction with the interactive display selecting the symbol corresponding to the facility, and based thereon, displaying information associated with the facility. See, e.g., 1G and 1H. For example, the user of the first device can select the icon of a fixed entity to retrieve additional information about the entity, including a telephone number. See, e.g., 1C and 1D for additional map and location data. |
| 46. The method of claim 45, further comprising performing, by the first device: identifying user interaction with the interactive display selecting the symbol corresponding to the facility, and based thereon, displaying information associated with the facility. | ZTE infringes directly and/or indirectly by performing, inducing others to perform, and/or contributing to the performance of the method of claim 46, wherein the information associated with the facility comprises a uniform resource locator (URL) of a web site associated with the facility. See, e.g., 1G and 1H. For example, the user of the first device can select the icon of a fixed entity to retrieve additional information about the entity, including a website's address. See, e.g., 1C and 1D for additional map and location data. |
| 47. The method of claim 46, wherein the information associated with the facility comprises a uniform resource locator (URL) of a web site associated with the facility. | ZTE infringes directly and/or indirectly by performing, inducing others to perform, and/or contributing to the performance of the method of claim 47, wherein the information associated with the facility comprises a uniform resource locator (URL) of a web site associated with the facility. See, e.g., 1G and 1H. For example, the user of the first device can select the icon of a fixed entity to retrieve additional information about the entity, including a website's address. See, e.g., 1C and 1D for additional map and location data. |
| 48. The method of claim 45, further comprising performing, by the first device: identifying user interaction with the interactive display | ZTE infringes directly and/or indirectly by performing, inducing others to perform, and/or contributing to the performance of the method of claim 45, further comprising performing, by the first device: identifying user interaction with the interactive display selecting the symbol corresponding to the facility and user interaction with the display specifying an action, and based thereon, loading a web page associated with the facility. See, e.g., 1G and 1H. For example, the user of the first device can select the icon of a fixed entity to retrieve additional information about the entity, including a website's address for selection. See, e.g., 1C |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| Exemplary Supporting Evidence Regarding Accused Products | |
|--|--|
| US9467838 | and 1D for additional map and location data. |
| selecting the symbol corresponding to the facility and user interaction with the display specifying an action, and based thereon, loading a web page associated with the facility. | ZTE infringes directly and/or indirectly by performing, inducing others to perform, and/or contributing to the performance of the method of claim 1, further comprising performing by the first device: identifying user interaction with the interactive display selecting a subset of the user-selectable symbols corresponding to a subset of the one or more second devices included in the group. See, e.g., 1B. For example, Google Contacts may be grouped by lists. Groups made by linking or associating within Find My Device and Google Maps, taken separately, constitute subsets or subgroups of a group of contacts within the Accused Device. Within the Accused Device, if multiple accounts have been setup for Find My Device, each Account constitutes a subset of second devices. Moreover, groups and subsets may be identified in each of Android Device Manager, Find My Phone, Find My Device, Google Latitude, Google Plus, Google Hangouts, Google Maps, Google Assistant, Google Search, Google Messages, Android Messenger, Google Allo, Google Duo, Gmail, and Google Chrome. |
| [49B] and identifying user interaction with the interactive display specifying an action and, based thereon, assigning the subset of second devices to a sub-net. | ZTE infringes directly and/or indirectly by performing, inducing others to perform, and/or contributing to the performance of the identifying user interaction with the interactive display specifying an action and, based thereon, assigning the subset of second devices to a sub-net. See, e.g., [49A] above. |
| 50[A]. The method of claim 49, further comprising performing | ZTE infringes directly and/or indirectly by performing, inducing others to perform, and/or contributing to the performance of the method of claim 49, further comprising performing by the first device: identifying user interaction with the interactive display selecting the sub-net and user interaction with the display |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| US9467838 | |
|---|--|
| Exemplary Supporting Evidence Regarding Accused Products | specifying an action. See, e.g., 1C and 49A above. For example, a user of a first device can initiate a call with a member or members of the grouped identified. |
| by the first device: identifying user interaction with the interactive display selecting the sub-net and user interaction with the display specifying an action; | ZTE infringes directly and/or indirectly by performing, inducing others to perform, and/or contributing to the performance of, based thereon, sending fourth data to the subset of second devices via the first server or initiating a phone conference with the subset of second devices. See, e.g., 50A above. For example, communication to the recipient devices includes sending data to a server. See, e.g., 1A, 1C, 1D, and 1H above. |
| [50B] and based thereon, sending fourth data to the subset of second devices via the first server or initiating a phone conference with the subset of second devices. | ZTE infringes directly and/or indirectly by performing, inducing others to perform, and/or contributing to the performance of the method of claim 1, wherein the first server is the second server. See, e.g., 1A, 1C, 1D, and 1H above, including a server corresponding to Android Device Manager, Find My Phone, Find My Device, Google Latitude, Google Plus, Google Hangouts, Google Maps, Google Assistant, Google Search, Google Messages, Android Messenger, Google Allo, Google Duo, Gmail, and Google Chrome and associated services.. |
| 51. The method of claim 1, wherein the first server is the second server. | ZTE infringes directly and/or indirectly by performing, inducing others to perform, and/or contributing to the performance of the method of claim 1, wherein the first set of second devices and the second set of second devices are identical. See, e.g., 1A, 1F, and 1G, where the first and second sets of second devices can be the same. |
| 52. The method of claim 1, wherein the first set of second devices and the second set of second devices are identical. | ZTE infringes directly and/or indirectly by performing, inducing others to perform, and/or contributing to the performance of the method of claim 1, wherein the message further includes an identifier corresponding to the first device. See, e.g., 1A and 1B, where the message includes the identifier. Upon information, additional metadata associated with the Accused Device is included within the message. |
| 53. The method of claim 1, wherein the message further includes an identifier corresponding to the first device. | |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| US9467838 | | Exemplary Supporting Evidence Regarding Accused Products | |
|--|---|---|--|
| 54[P]. A system comprising: | ZTE directly and/or indirectly infringes by providing a system [comprising]. | See 1P above. | |
| [54A] a first device programmed to perform operations comprising: | ZTE directly and/or indirectly infringes by a first device programmed to perform operations [of claim 54]. | See 1A above. | |
| [54B] joining a communication network corresponding to a group, wherein joining the communication network comprises transmitting a message including an identifier corresponding to the group; | ZTE directly and/or indirectly infringes by joining a communication network corresponding to a group, wherein joining the communication network comprises transmitting a message including an identifier corresponding to the group. | See 1B above. | |
| [54C] participating in the group, wherein participating in the group includes sending first location information to a first server and receiving second location information from the first server, the first location information comprising a location of the first device, the second location information comprising one or more locations of one or more respective second devices included in the group; | ZTE directly and/or indirectly infringes by: participating in the group, wherein participating in the group includes sending first location information to a first server and receiving second location information from the first server, the first location information comprising a location of the first device, the second location information comprising one or more locations of one or more respective second devices included in the group. | See 1C above. | |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| Exemplary Supporting Evidence Regarding Accused Products | |
|--|---|
| <p>US9467838 one or more locations of one or more respective second devices included in the group;</p> | <p>ZTE infringes directly and/or indirectly by presenting, via an interactive display of the first device, a first interactive, georeferenced map and a first set of one or more user-selectable symbols corresponding to a first set of one or more of the second devices, wherein the first set of symbols are positioned on the first georeferenced map at respective positions corresponding to the locations of the first set of second devices, and wherein first georeferenced map data relate positions on the first georeferenced map to spatial coordinates.</p> <p>See 1D above.</p> |
| <p>[54D] presenting, via an interactive display of the first device, a first interactive, georeferenced map and a first set of one or more user-selectable symbols corresponding to a first set of one or more of the second devices, wherein the first set of symbols are positioned on the first georeferenced map at respective positions corresponding to the locations of the first set of second devices, and wherein first georeferenced map data relate positions on the first georeferenced map to spatial coordinates;</p> | <p>ZTE infringes directly and/or indirectly by: sending, to a second server, a request for second georeferenced map data different from the first georeferenced map data.</p> <p>See 1E above.</p> |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| Exemplary Supporting Evidence Regarding Accused Products | |
|---|--|
| US9467838 | |
| [54F] receiving, from the second server, the second georeferenced map data; | ZTE infringes directly and/or indirectly by receiving, from the second server, the second georeferenced map data. See 1F above. |
| [54G] presenting, via the interactive display of the first device, a second georeferenced map and a second set of one or more user-selectable symbols corresponding to a second set of one or more of the second devices, wherein the second set of symbols are positioned on the second georeferenced map at respective positions corresponding to the locations of the second set of second devices, and wherein the second georeferenced map data relate to spatial coordinates. | ZTE infringes directly and/or indirectly by presenting, via the interactive display of the first device, a second georeferenced map and a second set of one or more user-selectable symbols corresponding to a second set of one or more of the second devices, wherein the second set of symbols are positioned on the second georeferenced map at respective positions corresponding to the locations of the second set of second devices, and wherein the second georeferenced map data relate to spatial coordinates. See 1G above. |
| [54H] and identifying user interaction with the interactive display selecting one or more of the second set of user-selectable symbols corresponding to one or more of the second devices and positioned on the second georeferenced map and user interaction with the display specifying an action and, based thereon, sending third data to the selected one or more second devices via the first server. | ZTE infringes directly and/or indirectly by identifying user interaction with the interactive display selecting one or more of the second set of user-selectable symbols corresponding to one or more of the second devices and positioned on the second georeferenced map and user interaction with the display specifying an action and, based thereon, sending third data to the selected one or more second devices via the first server. |

Exhibit D - Claim Chart for U.S. Patent No. 9,467,838 Against ZTE

| US9467838 | Exemplary Supporting Evidence Regarding Accused Products |
|--|---|
| <p>selectable symbols corresponding to one or more of the second devices and positioned on the second georeferenced map and user interaction with the display specifying an action and, based thereon, sending third data to the selected one or more second devices via the first server.</p> | <p>See 1H above.</p> |