

Hello,

Attached is the source code of the location tracking app **Tracer**. It is specially designed for selection a location on a map and to provide a radius, i.e. define a geofence. Whenever the user enters or exits the geofence, a notification should be shown on the smartphone.

The geographic locations from network (cellular and WiFi) and GPS are used for comparing the location. The wake timer mode uses a combination of AlarmManager and BroadcastReceiver to wake up the phone regularly and compare the location. This mode generates relatively constant recording with trivial variation in intervals, and it is much more battery friendly. In terms of the new doze mode introduced since Android 6.0, the best option is to put the app in the whitelist for battery optimization exemption. There is a `setAndAllowWhileIdle()` method under AlarmManager but, android will not dispatch these alarms more than about every minute or every 15 minutes in low-power idle mode. So it is not a suitable choice for an app like ours that needs constant wake up as often as every 10-15 seconds. Furthermore, the service will stop when the phone is powered off, but will resume after the phone is switched on again.

Obtaining an accurate user location on a mobile device can be difficult. Generally there are three concerns that will challenge you the most:

1. **Multiple location sources** – There is more than one provider from which location is acquired (GPS, WI-FI and Cell-ID) and they all vary very much in accuracy, power consumption and speed.
2. **User movement** – Because of user movement location data must be refreshed on a reasonable time interval.
3. **Varying accuracy** – Location estimations coming from different providers may not be with the same accuracy. Location from newest provider might

be significantly less accurate than the estimation coming from an older provider.

To conclude, the developer has used combination of GPS and Network Provider to get current location and once in every 10 seconds it is matched with the user selected coordinate. The notification will be sent when it is in the range of geofence as well as when the user is out of the range of geofence.

Developer tried to implement “**Exponential Backoff**” algorithm that uses feedback to multiplicatively decrease the rate of some process, in order to gradually find an acceptable rate but due to constrain of time he did not able to implement .

Application Manual: The user install the application and map screen shown in Figure 1.



The user clicks on the map and dialog box provides with input field of radius and location coordinates of the selected place as can be seen in Figure 2.

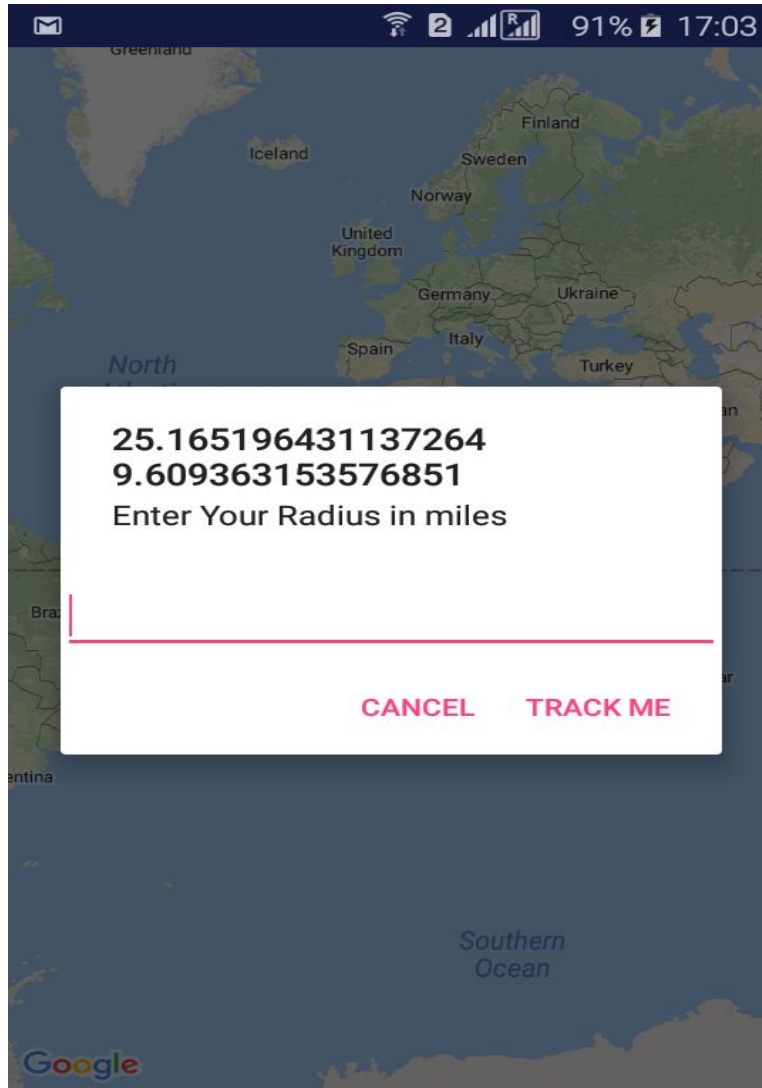


Figure 2 . Selected Destination Coordinates

User needs to input a radius in the text space and click the Track Me button as shown in Figure 3.

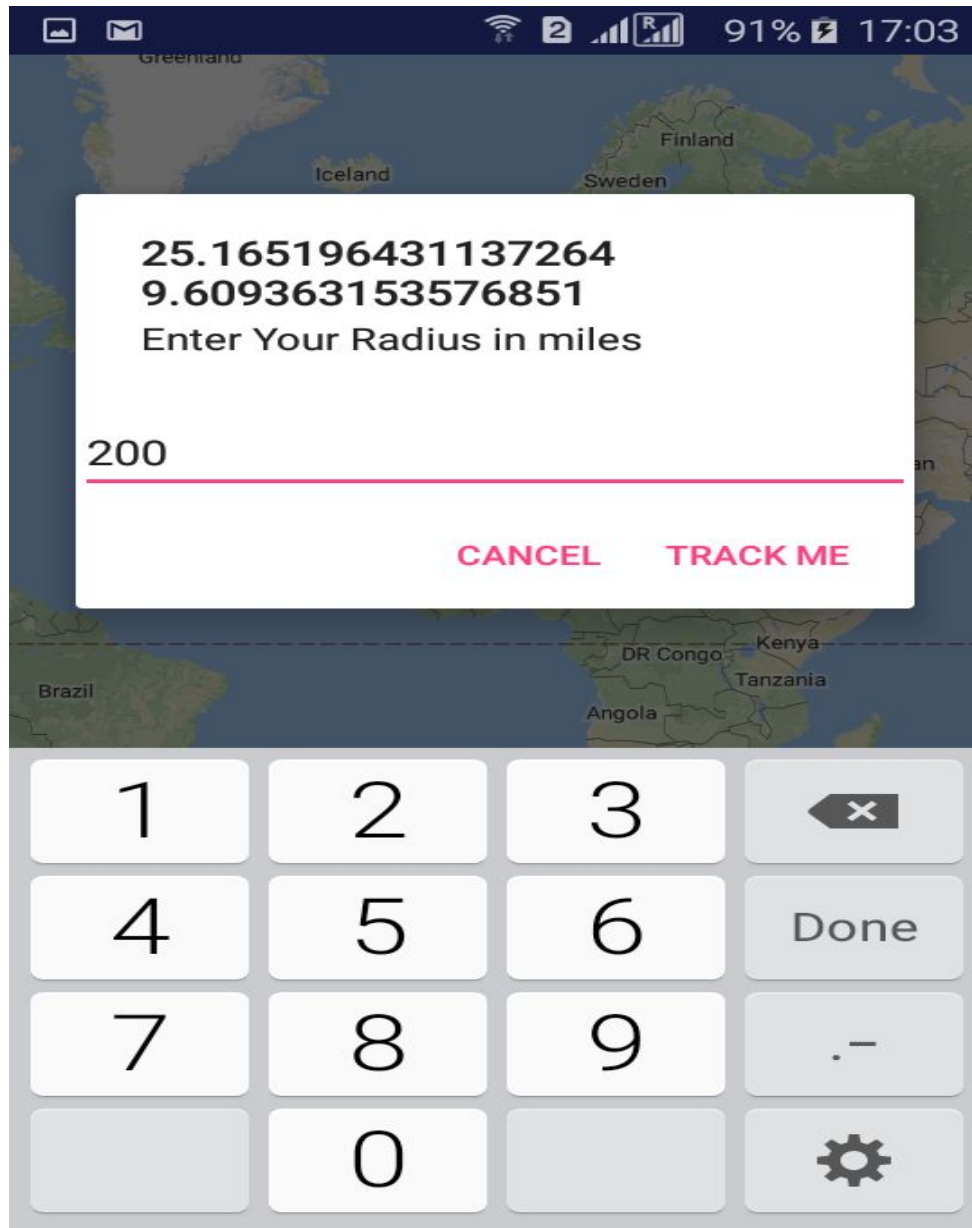


Figure 3. Geofence range was 200 miles

The tracking start notification will be seen in a status bar as soon as user has clicked on TrackMe Button as in Figure 4.

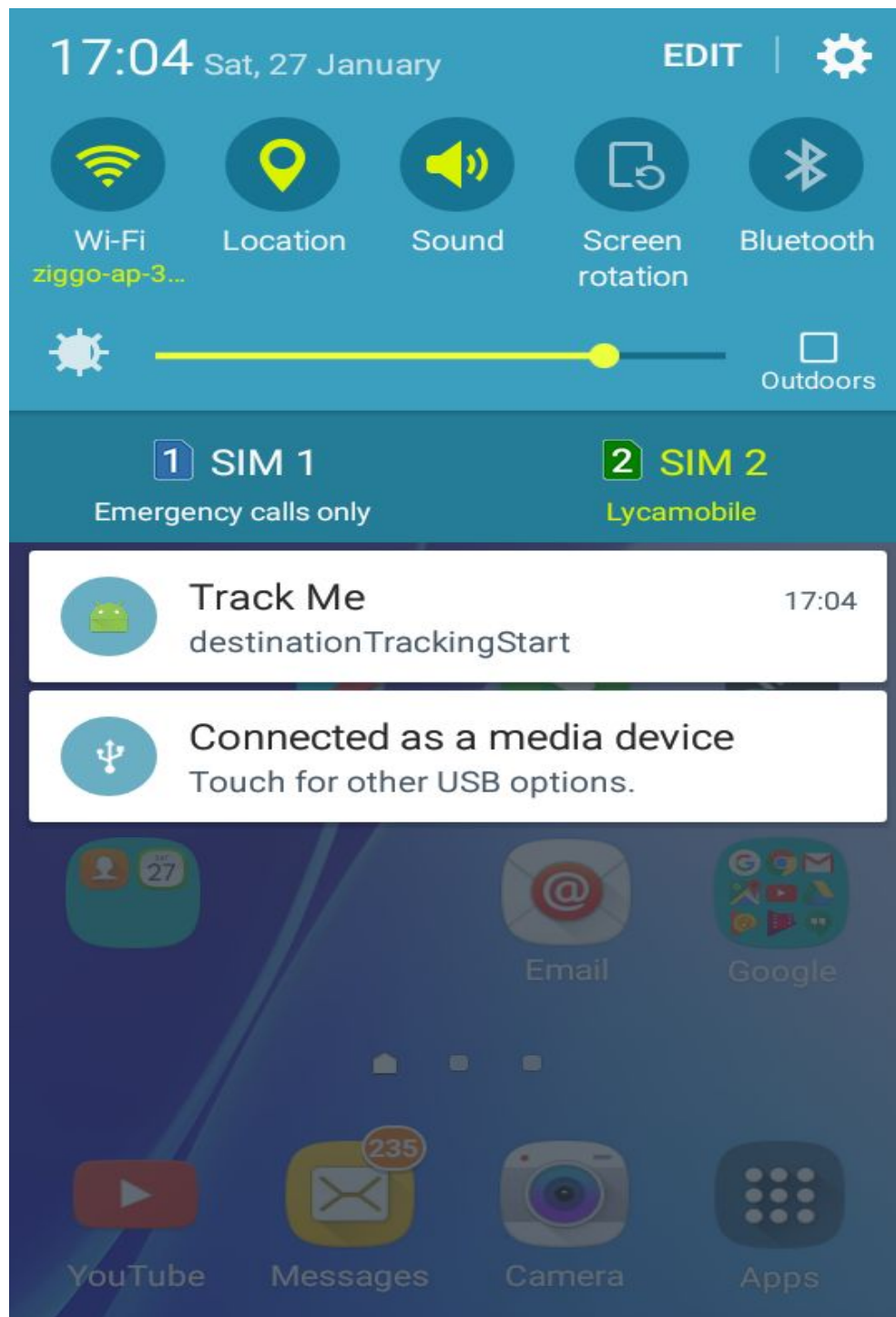


Figure 4. Notification that Tracking Started.

As soon as the user arrives in a destination range, notification will be prompted as can be seen in Figure 5.

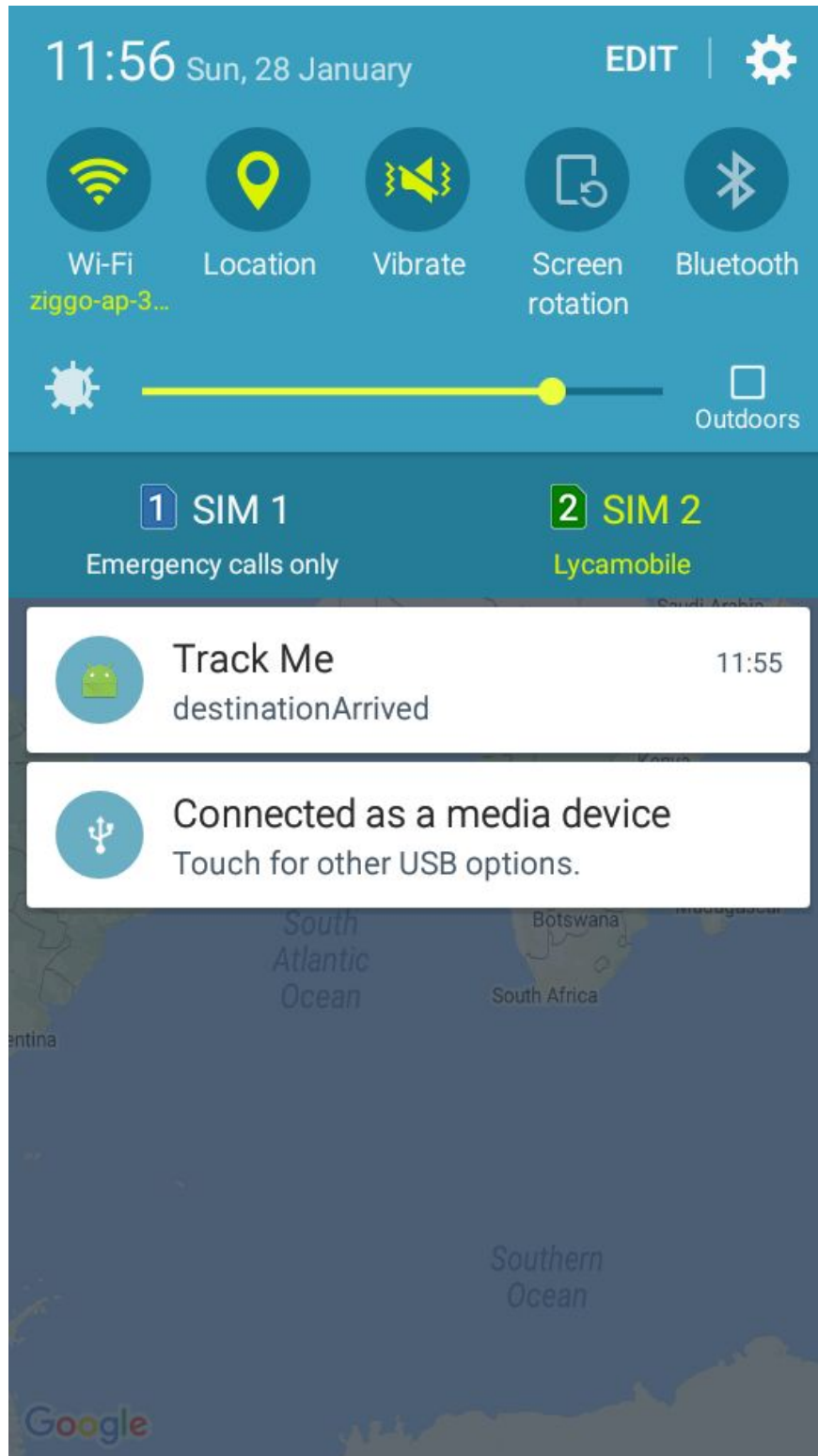


Figure 5. Destination arrived notification.

Once the user exits the range of destination, notification will be sent as seen in Figure 6.

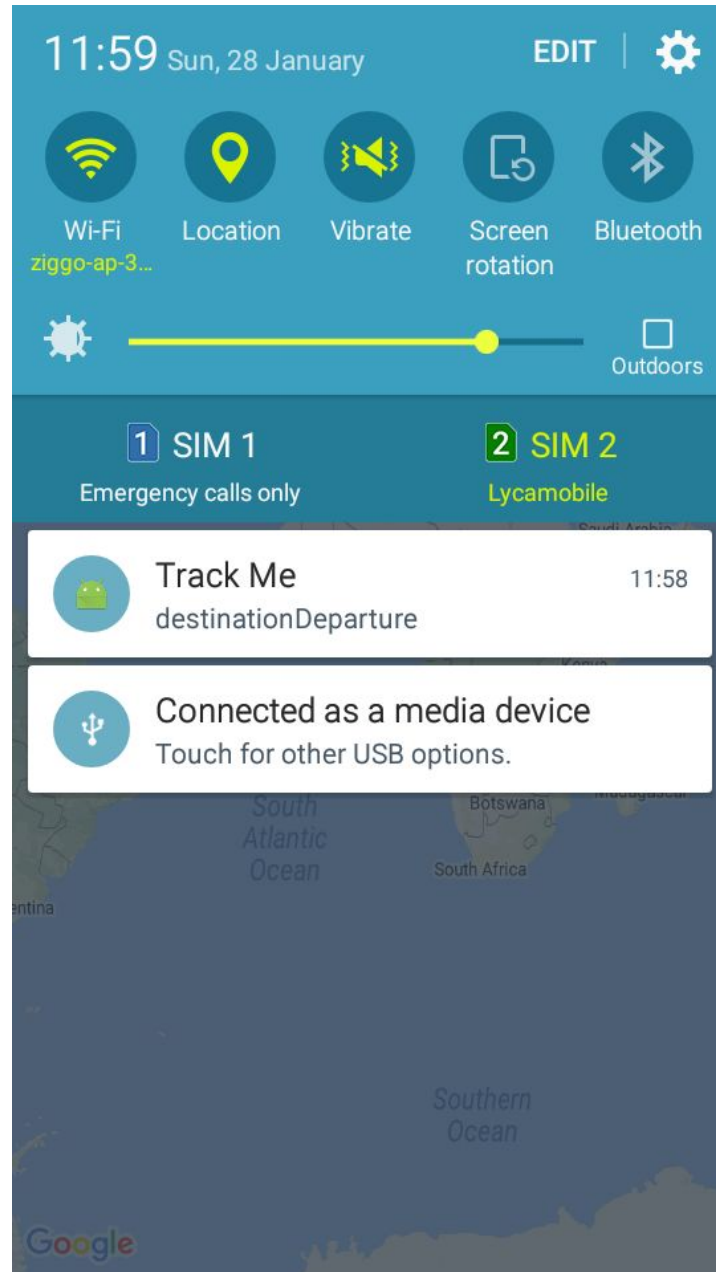


Figure 6. Destination Departure Notification.

Due to the constraint of time, Developer could not focus on followings items :

- Developer did not focus on security which includes reverse engineering, code obfuscation, impediment of dynamic instrumentation framework such as Frida or Xposed.

- If the device is idle, the service should stop by using sensor and resume when it location coordinates change.
- Developer did not focus on several use cases such as input handling, constraint range etc.
- Developer did not able to perform Boundary value Testing, Compatibility Testing, End-to-End testing, Automated Testing, Acceptance Testing. However, he able to performed some of the junit test cases and Happy path Testing.
- Developer did not able to check the application in different models, configurations, versions of devices.

Future Enhancement

- Developer will use **Exponential Backoff** (<https://carlosbecker.com/posts/exponential-backoff-java8/>) implementation to set the interval time for request the location. Let take a scenario to understand better.
 1. The default request interval is 10 second and you get the response of coordinates after 10 second .
 2. Second time, you get the same response or some minor difference .
 3. Third time, you get the same response or some minor difference .
 4. The fourth time you will exponentially increase your interval time to 12 second.
 5. Fifth time, if the result will be same , you will again increase your interval to 15 second.
 6. In this way , we can increase the efficiency of the battery and achieved intelligent job-scheduling .
 7. If I am correct, Gmail and What-Up are using same technique to check the internet connectivity.
- Another option, If the device is idle, the service will stop by using sensor fusion data and resume when it location coordinates change.

Thanks you so much for time ,The information was totally based on my current knowledge and may it vary. Apologize in advance for this.

Hope to discuss more on this.