

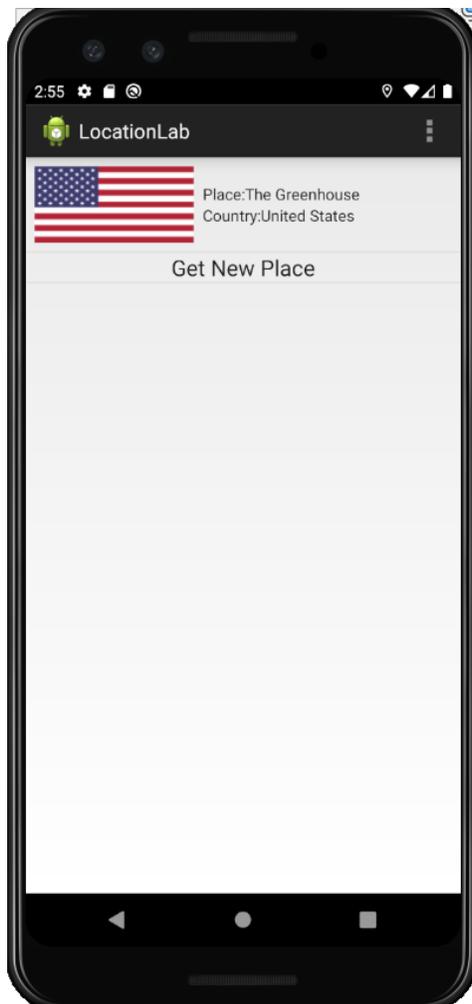
Location Lab

Use Location information within your app.

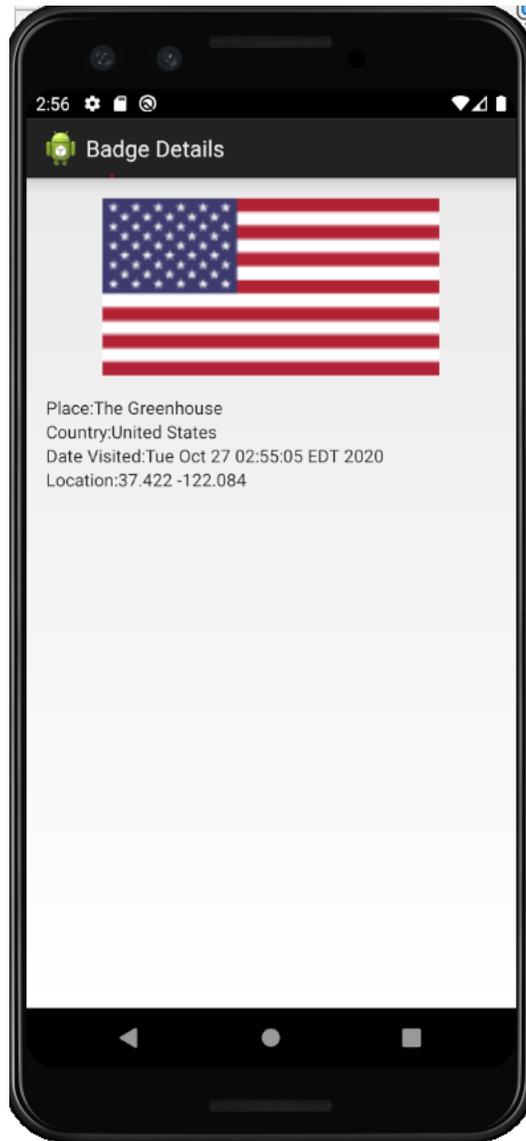
This week's lab will help you learn how to use location information in your Android applications. Upon completion of this lab, you should have a better understanding of how to listen for and respond to Location measurements.

This application displays a ListView containing a set of Place Badges and you get the place's coordinates from the phone's Location measurements.

You will build the application such that it receives locations from the device itself. To implement this functionality, you will need to acquire location readings from Android. Exactly how you implement this is up to you, however, your app will need to listen for location updates from the NETWORK_PROVIDER (which we will control for testing purposes) and the GPS_PROVIDER (so you can enter locations via the emulator). You can also inject locations into the app, by clicking on the menu and selecting one of the menu items, specifically "Place One", "Place Two," or "Place No Country."



In addition, when the user clicks on a Place Badge, the app will open a detail view showing additional information about the place.



ToDo's:

1. The application screenshots shown above require a working network connection. You will also need to create an account at <http://www.geonames.org/login>. Your username will need to be updated in `PlaceDownloaderTask.kt`. In `PlaceDownloaderTask.kt`, we use geonames api services to fetch nearby place names and flag images. Feel free to read this file to see how we do it.

2. Implement the Todos in PlaceViewActivity.kt : more details on Todos explained in the file. Please refer to demo_location_lab.mp4 to know the flow of the app.
3. Implement PlaceBadgeDetailActivity.kt to show the place badge detail as shown in the screenshot above. You need to get the intent data from PlaceViewActivity and create a PlaceRecord using it. Refer to PlaceRecord.kt implementation to see how to access the members.

Testing:

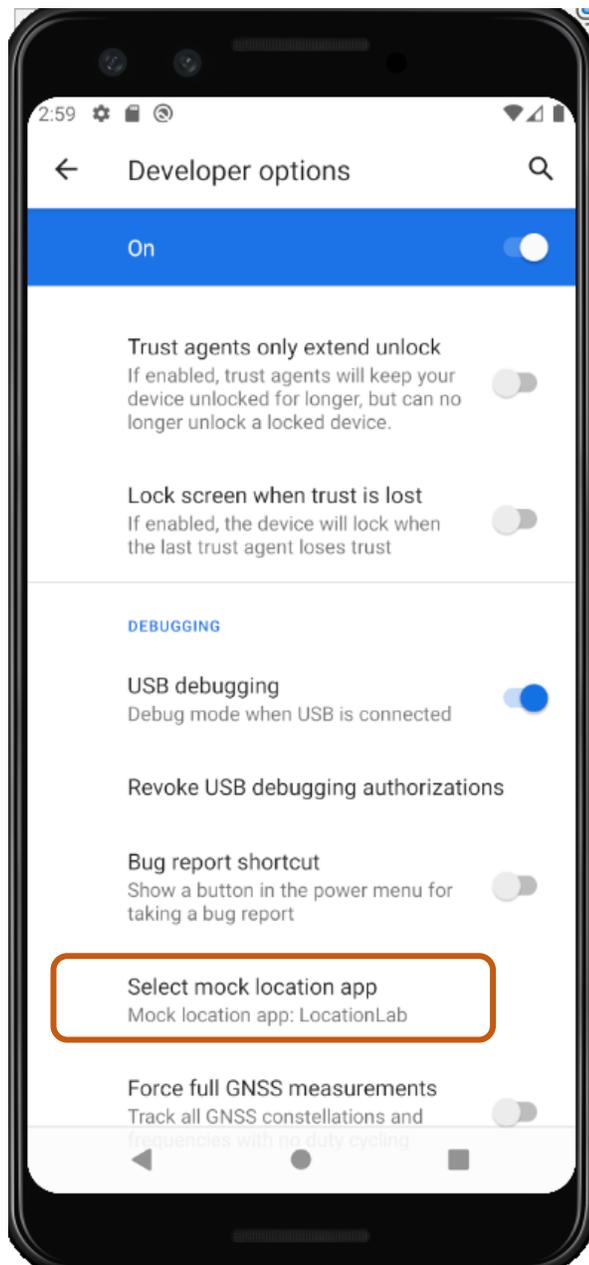
The test cases for this Lab are in the Lab11_LocationLab project. You can run the test cases either all at once, by right clicking the test package and then selecting Run 'Tests in 'course.lab...', or one at a time, by right-clicking on an individual test case class (e.g., TestNoCountryLocation.kt) and then continuing as before. The test classes are Robotium test cases.

Rubric:

All tests are equally weighted for this lab.

Warnings:

1. We have done our testing on an emulator using a Pixel 3 AVD with API level 29. To limit configuration problems, you should test your app against a similar AVD.
2. You will need to go into your device's Developer Options. On Android 4.1 and lower, the Developer options screen is available by default. On Android 4.2 and higher, you must enable this screen as follows:
 - a. Open the Settings app.
 - b. Scroll to the bottom and select About Emulated Device.
 - c. Scroll to the bottom and tap Build number 7 times.
 - d. Return to the previous screen to find System, click on System->Advanced->Developer Options.
 - e. Developer options near the bottom.
3. You will need to make sure that you've chosen "Select mock location app" and select LocationLab as the testing app from the Developer Options.



Submission

As usual, to submit your work, you will need to commit your solution to your repo on GitLab by running the following command: `git push origin master`.

Note: if you have not already pushed this branch to your repo on GitLab you will need to make a slight modification for this first time and run this instead: `git push -u origin master`. This sets up tracking between your local branch and a branch with the same name on your repo in GitLab.