Lab – Permissions

Objectives:

Familiarize yourself with Android Permissions. Create applications that use, define and enforce Android Permissions.

Exercise A: Using Permissions

This exercise uses Permissions so that it can load protected content. The application is called Lab3_Permissions and its main Activity is called ActivityLoaderActivity. This Activity's user interface displays a Button labeled "PhoneStatus Activity".

When the user clicks this Button, the application will start a new Activity called "PhoneStatusActivity." That Activity's user interface is shown below.

This activity presents a TextView that initially displays the words, "Phone Number Goes Here." It also presents a Button labeled, "Get Phone Number" and another Button labeled, "Go To DangerousActivity". When the user presses the "Get Phone Number" Button, the application will open a view prompting the user to allow the PermissionsLab application to make and manage phone calls.
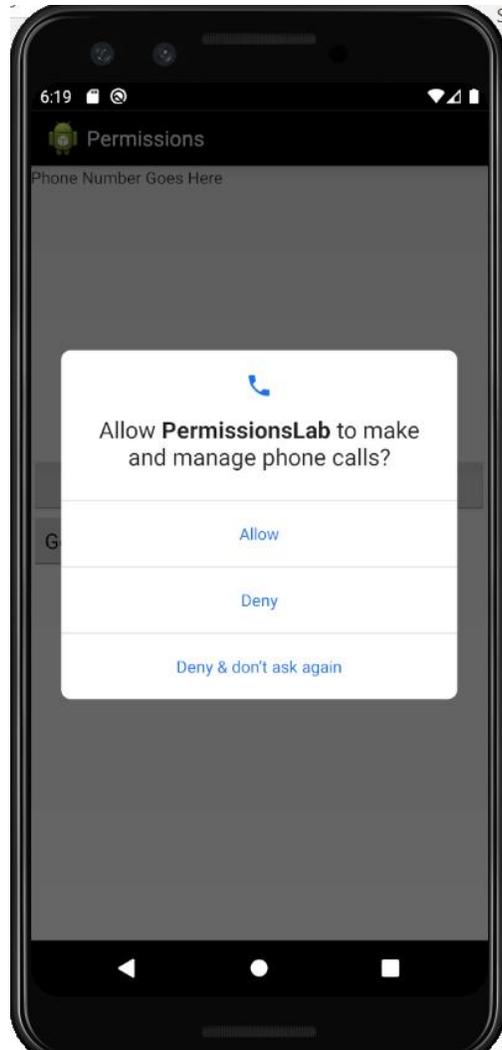
When the user clicks "ALLOW" to grant access to the application, the application then retrieves the device's phone number and then displays it in the TextView, as shown below.

In order to access the device's phone number, your application must have permission to retrieve it. In order to complete this assignment you'll need to find the specific permission you need (https://developer.android.com/reference/android/telephony/TelephonyManager.html).

Exercise B: Defining and Enforcing Custom Permissions

In this exercise, you'll define, enforce and use permissions so that your application can access a separate, permission-protected application, called DangerousApp. You will build your solution to this exercise by extending your solution to Exercise A.

When the user clicks on the Button (shown above) labeled "Go To DangerousActivity", an Activity called "GoToDangerousActivity" will be started. That Activity's user interface appears below.
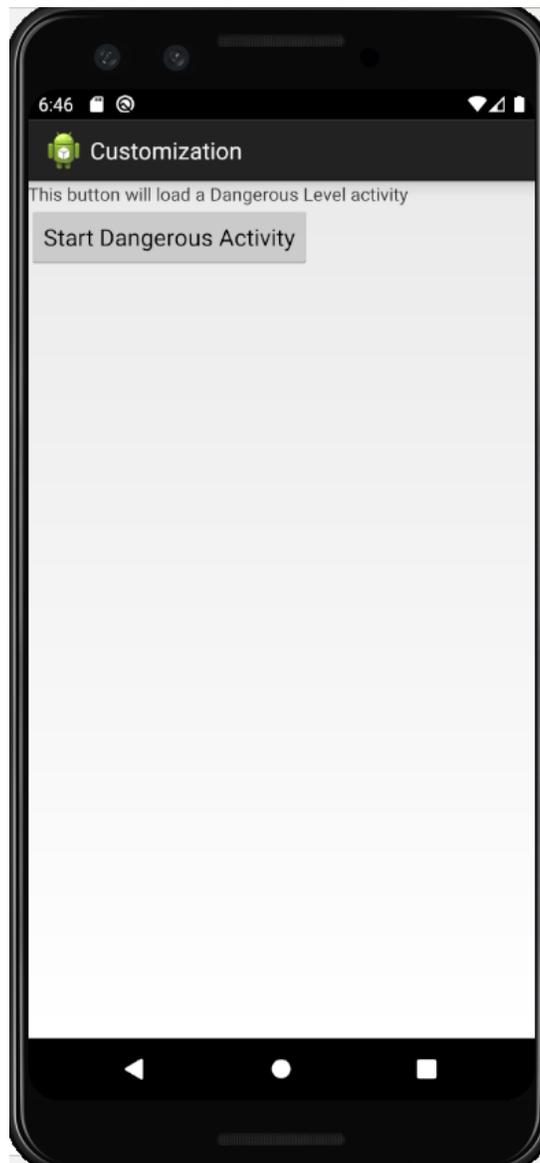
When the user clicks on the "Start Dangerous Activity" Button, this Activity will use an **Implicit Intent** with the **action**, "course.labs.permissions.DANGEROUS_ACTIVITY", to start the "DangerousApp." Just as in the Exercise A, a dialog will open prompting permission from the user to perform an unknow action:
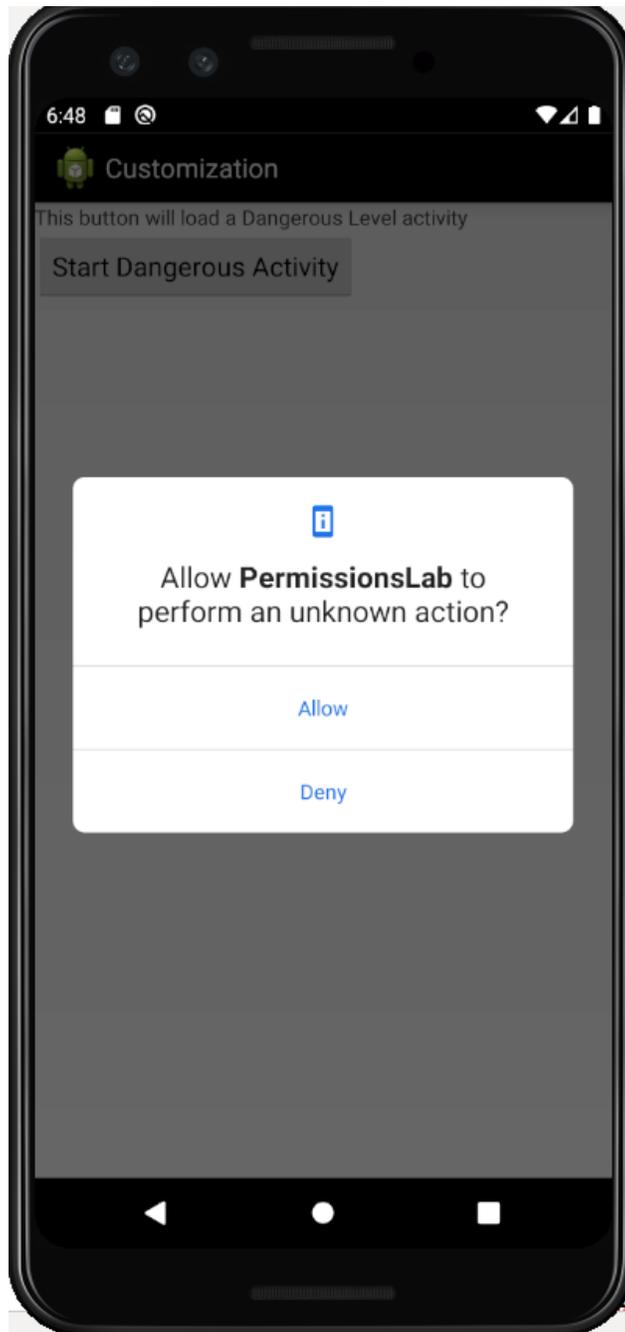
As shown below, after clicking "ALLOW", the app will simply display a TextView, containing the words, "You have opened a dangerous activity."

To implement the Lab3_DangerousApp application, you will need to import and modify a separate Android application project that contains a single Activity called DangerousActivity. The application will define and enforce its own custom permission, "course.labs.permissions.DANGEROUS_ACTIVITY_PERM", which will have a "dangerous" protection level. See http://developer.android.com/guide/topics/manifest/permission-element.html for more information. You will also specify an intent filter for the DangerousActivity of the Lab3_DangerousApp that matches the Implicit Intent that the Lab3_Permissions use to start the DangerousActivity.

Implementation Notes:

1. Run git pull upstream master to retrieve the new android projects for this lab assignment. They will be inside of the labs directory inside of your local repo.
2. For Exercise A:
   a. In the Lab3_Permissions ActivityLoaderActivity.kt, find the comment containing the String TODO:
      i. In the onCreate(), add click listener to the startPhoneStatusButton button to call the startPhoneStatusActivity(),
      ii. In startPhoneStatusActivity() method. Start the PhoneStatusActivity.
   b. In the Lab3_PermissionsLab's PhoneStatusActivity.kt, find the comment containing the String TODO:
      i. In the onCreate(), add click listener to the getPhoneNumButton button to call the loadPhoneNumber(),
      ii. In the onCreate(), add click listener to the goToDangerousActivityButton button to call the startGoToDangerousActivity(),
      iii. In the startGoToDangerousActivity() method. Start the GoToDangerousActivity.
   c. In the Lab3_PermissionsLab's AndroidManifest.xml, find the comments containing a TODO String. Where indicated, add the appropriate uses-permission element so that this application can read the device's phone number.

3. For Exercise B:
   a. In the Lab3_DangerousApp's AndroidManifest.xml, find the comments containing a TODO String. Where indicated, define and enforce a new permission named, "course.labs.permissions.DANGEROUS_ACTIVITY_PERM", that has a dangerous protection level.
   b. In the Lab3_DangerousApp's AndroidManifest.xml, find the comments containing a TODO String. Where indicated, add Intent Filter information so that the DangerousActivity of this application can be started by an implicit Intent, having the Action, "course.labs.permissions.DANGEROUS_ACTIVITY"
   c. In the AndroidManifest.xml file for the Lab3_PermissionsLab, find the comments containing a TODO String. Where indicated, add the appropriate uses-permission element so that this application can start the DangerousApp.

Testing

The test cases for this Lab are in the Lab3_Permissions. You can run all the test cases, by right clicking on the tests folder and then selecting 'Run Tests in ….', or one at a time, by right clicking on an individual test case class and then continuing as before. The test classes are Robotium test cases, but be aware that the test cases have been tested only for the standard AVD described below.
As you implement various steps of the Lab, run the test cases every so often to see if you are making progress toward completion of the 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. Also, when testing, make sure that your device is in Portrait mode with the screen unlocked when the test cases start running.
2. The test cases require that you've installed both the Lab3_ Permissions and the Lab3_dangerousApp applications. Remember - If you change an app, you'll need to reinstall it.
3. Spelling counts. Pay attention to how to specify data values in your xml files. Leave out or misspell a single letter and Android won't understand what you really meant.

As you implement various steps of the lab, run the test cases every so often to see if you are making progress toward completion of the lab.
Once you've passed all the test cases, follow the following instructions to submit your work via GitLab for grading.

Submission

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. Always verify by navigating to your repo on gitlab that your code changes are there.
Note: you must commit your changes locally before you can push them to your remote origin repo.