COMP 7855

Lab 2: Interactive Android Application

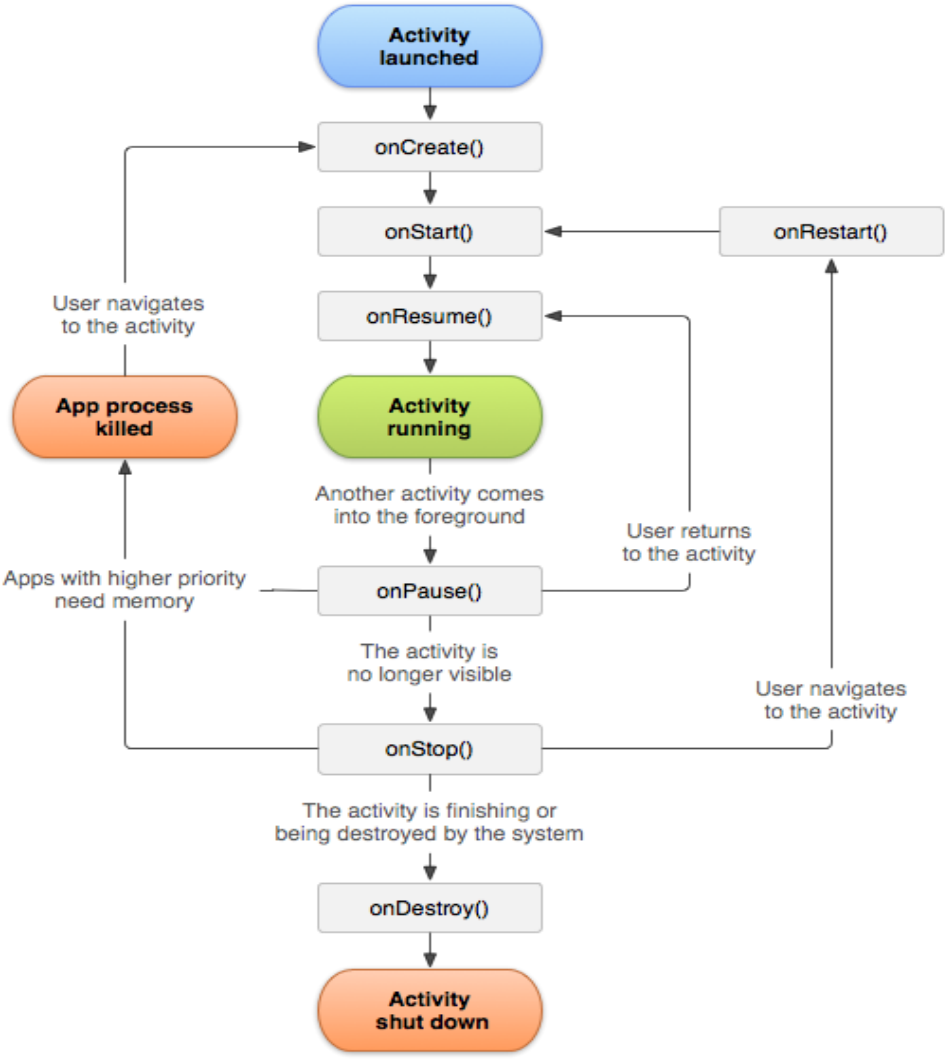
The goal of this lab is to build an interactive Android application with multiple activities with intents and event handling.

In this lab, you will

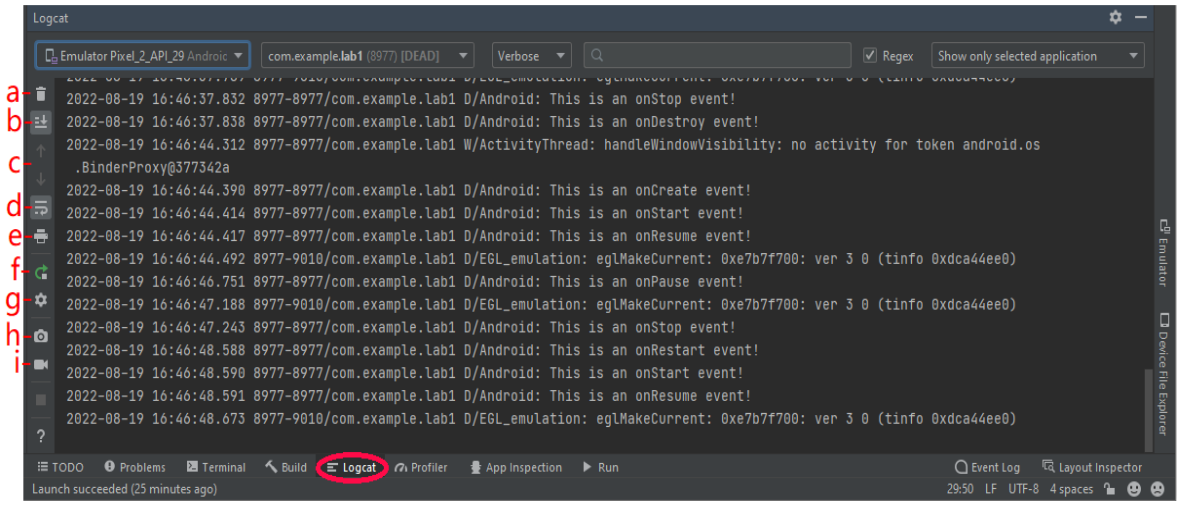
1. Get more familiar with Android Studio.
2. Review the Activity Lifecycle of Android applications.
3. Learn to navigate between activities using intents
4. Learn to handle events using listeners.

I. Activity Lifecycle

First, we are going to explore the sequence of callback methods (events) in the Activity Lifecycle of an Android application, as shown in the following diagram.



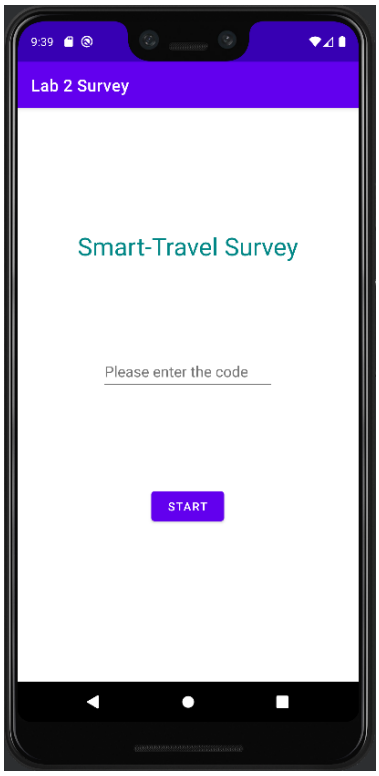
1. Create a new project called “Lab 2 Activity” using the Empty Activity template.
2. In MainActivity.java, import the logging library by adding “import android.util.Log;” on the top.
3. To display system messages in the Logcat window, click on the Logcat option in the bottom line of Android Studio.

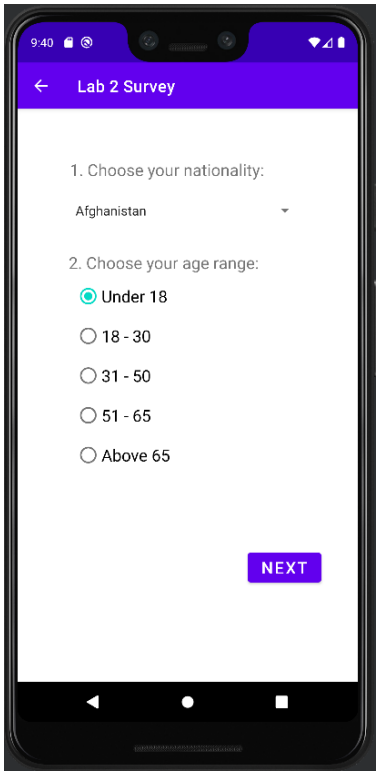


4. The Logcat toolbar on the left provides the following buttons:
   a. Clear logcat: click to clear the visible log.
   b. Scroll to the end: click to jump to the bottom of the log and see the latest log messages.
   c. Up/Down the stack trace: click to navigate up/down the stack traces in the log (available when you have a list of method calls that lead to the exception being thrown).
   d. Use soft wraps: click to enable line wrapping and prevent horizontal scrolling.
   e. Print: click to print the logcat message.
   f. Restart: click to clear the log and restart logcat.
   g. Logcat header: click to open the configuration dialog for customizing the appearance of each message.
   h. Screen capture: click to capture a screenshot.
   i. Screen record: click to record a video of the device (for a maximum of 3 minutes).
5. Now, add a debug message to the onCreate method using the Log.d method. With the logging library imported at step 2, we can create different log messages that will appear in logcat (See more completed option list: https://developer.android.com/reference/android/util/Log). Each log method has two parameters – tag (e.g., “Android”) and message (“This is an onCreate event!”).
6. Run your “Lab 2 Activity” application (your AVD should have been set up in Lab 1). In what situation will the debug message for the onCreate method be shown?
7. Similarly, override the onStart, onResume, onPause, onStop, onRestart and onDestroy methods. Each method should show the corresponding debug message. Note that only the onCreate method has the Bundle as a parameter.
8. Run your “Lab 2 Activity” application again. Try to trigger each debug message and compare the cause with the Activity Lifecycle diagram.

II. Smart-Travel Survey

1. Create a new project called "Lab 2 Survey" using the Empty Activity template.
2. Modify your activity_main.xml to show an initial screen of your app.



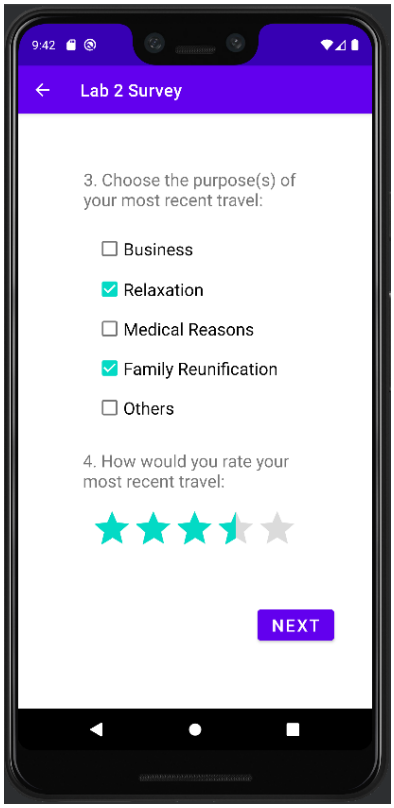


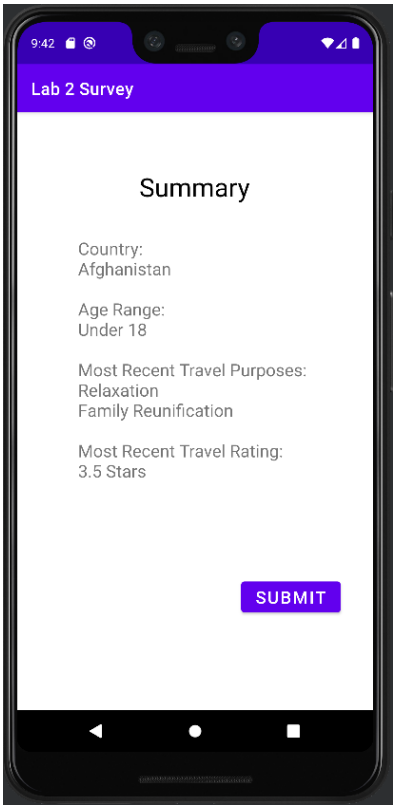
3. Modify your MainActivity.java so that users can start a new Activity called "Page1" by clicking on the "START" button if they enter the code "COMP7855".
4. If the provided code is wrong, a Toast is used to display the message "WRONG CODE".
5. Add a new class called "Page1.java" and a new xml file called "activity_page1.xml".
6. Add a folder called "raw" in the "res" folder and download the "countries.txt" file to the "raw" folder.
7. Modify your activity_page1.xml to show the first page of the survey that contains a spinner and a RadioGroup as follows.
8. The items in the dropdown menu of the spinner should be defined based on the countries specified in the "countries.txt" file, read using the following code.

```
InputStream inputStream =
getBaseContext().getResources().openRawResource(R.raw.countries);
BufferedReader bufferedReader= new BufferedReader(new InputStreamReader(inputStream));
String line = bufferedReader.readLine();
while (line != null) {
        // make use of the line read
      line = bufferedReader.readLine();
}
bufferedReader.close();
inputStream.close();
```

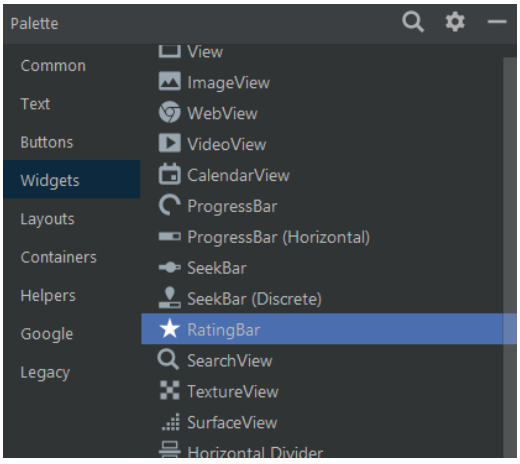
9. For the RadioGroup, make sure the first RadioButton "Under 18" is checked by default.

10. When the user clicks on the "NEXT" button, store the choices of the user in a Bundle and start the next Activity called "Page2" using an Intent with the Bundle.
11. Add a new class called "Page2.java" and a new xml file called "activity_page2.xml".
12. Modify your activity_page2.xml to show the second page of the survey that contains 5 Checkboxes and a RatingBar.





13. RatingBar can be found in widgets, and the getRating() method can be used to derive the rating (double).



14. When the user clicks on the "NEXT" button, store the choices of the user in the same Bundle and start the next Activity called "Page3" using an Intent.
15. Add a new class called "Page3.java" and a new xml file called "activity_page3.xml".

16. Modify your activity_page3.xml to show the third page as the summary of user inputs.
17. When the user clicks on the "Submit" button, start the next Activity called "Page4".
18. Add a new class called "Page4.java" and a new xml file called "activity_page4.xml".
19. Shows a message "Thank you for your participation!" in red color in the middle of the screen.

III. Submission

1. Show your lab instructor your work for Section I and II.
2. Once approved, export your work by clicking on "File → Export → Export to zip file…".
3. Add both zip files (for Section I and II) into another zip file.
4. Rename your file as "Lab2_firstname_lastname_A#.zip".
5. Upload the file to the learning hub (D2L) before leaving the lab.

You Have Completed Lab 2