

ITIS/ITCS 4180/5180 Mobile Application Development

Homework 1

Date Posted: 05/28/2014 at 11:00

Due Date: 05/31/2014 at 23:55

Basic Instructions:

1. In every file submitted you MUST place the following comments:
 - a. Assignment #.
 - b. File Name.
 - c. Full name of all students in your group.
2. Each group should submit only one assignment. Only the group leader is supposed to submit the assignment on behalf of all the other group members.
3. Please download the support files provided with this assignment and use them when implementing your project.
4. Export your Android project as follows:
 - a. From eclipse, choose "*Export...*" from the File menu.
 - b. From the Export window, choose *General* then *File System*. Click *Next*.
 - c. Make sure that your Android project for this assignment is selected. Make sure that all of its subfolders are also selected.
 - d. Choose the location you want to save the exported project directory to. For example, your *Desktop* or *Documents* folder.
 - e. When exporting make sure you select *Create directory structure for files*.
 - f. Click Finish, and then go to the directory you exported the project to. Make sure the exported directory contains all necessary files, such as the .java and resource files.
5. Submission details:
 - a. When you submit the assignment, compress your exported Android project into a single zip file. The format of compressed file name is HW#.zip
 - b. You should submit the assignment through Moodle: Submit the zip file.
- 6. Failure to follow the above instructions will result in point deductions.**

Homework 1 (100 Points)

In this assignment you will build your first Android application. You will get familiar with common Android components and how to interact with them. You will build a single activity Tip Calculator application.

Notes:

1. The recommended Android Virtual Device (AVD) should have a Target **at least** Android 4.1, API Level 16, and should display correctly on 3.2" QVGA (ADP2) (320x480: mdpi).

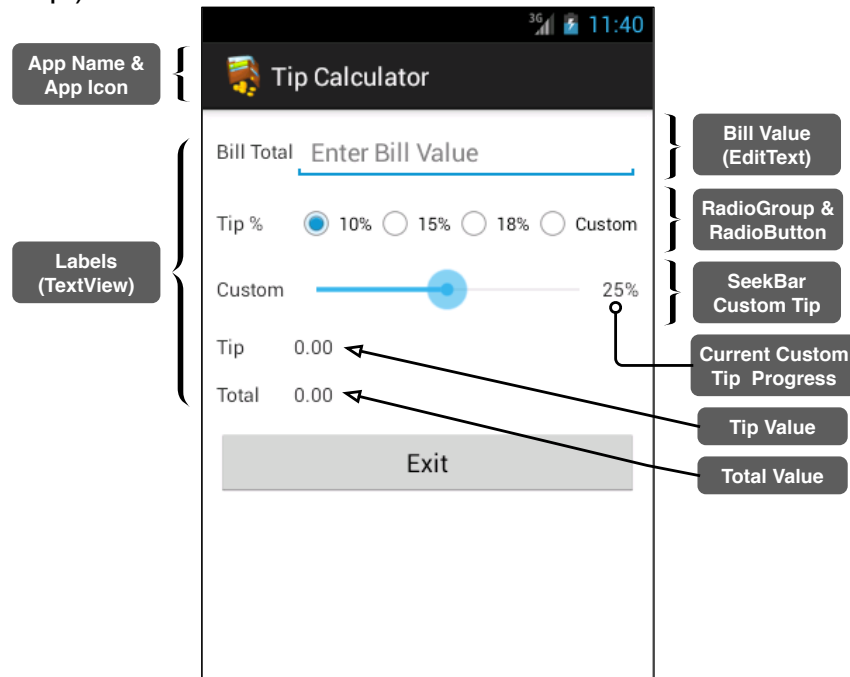


Figure 1, Application User Interface

Part 1 (30 Points): Building the Interface

The interface should be created to match the user interface (UI) presented in Figure 1. You will be using layout files, strings.xml, and drawable files to create the user interface. The layout XML file can be modified through the raw xml, or through the GUI tools provided within eclipse. To build the UI, please follow the following tasks:

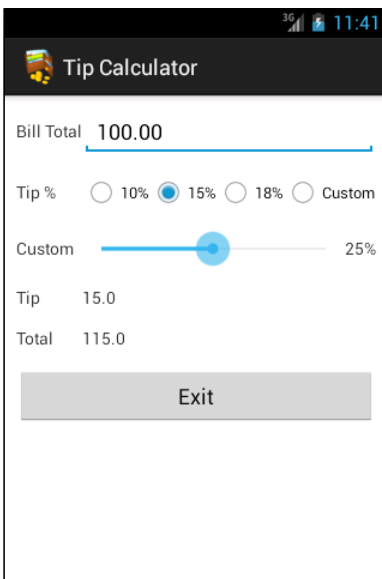
1. Your application should have an application launcher icon, please select your launcher icon to represent your app.
2. The string values used for the text labels, button labels and hints should be read from the strings.xml file and should not be hardwired in the layout file.
3. Use an EditText component for the user to enter the bill total value. The EditText component should be setup to limit the bill total value to only positive numbers. When the application starts the bill total EditText should be empty, and should display the hint message "Enter Bill Value" as indicated in Figure 1.
4. Use a RadioGroup containing RadioButtons to enable the user to pick from the tip options 10%, 15%, 18% and Custom. When the application starts the 10% tip choice should be selected.

5. Use the SeekBar to enable the user to pick a custom tip value, the maximum custom tip value should be set to 50%. When the application starts the custom tip value should be set to 25%. On the right of the SeekBar use a TextView to display the current progress of the SeekBar, which represents the current custom tip value.
6. Use TextView components for creating the tip and total values. When the application starts the tip and total values should be both set to 0.00.
7. Finally, create the Exit button, note that the text for the button should be retrieved from the string.xml file.

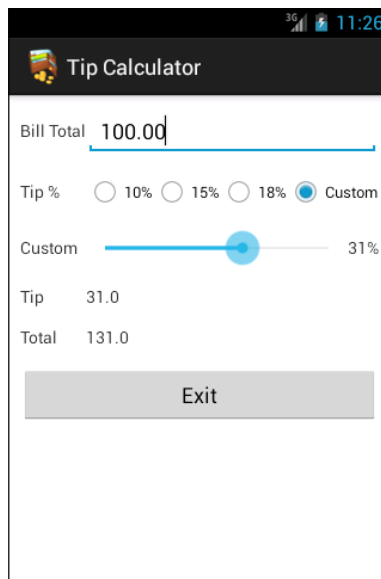
Part 2 (70 Points): Event Handlers and App Behavior (MainActivity)

In this part you will build the required logic for the tip calculator app. The requirements are as follows:

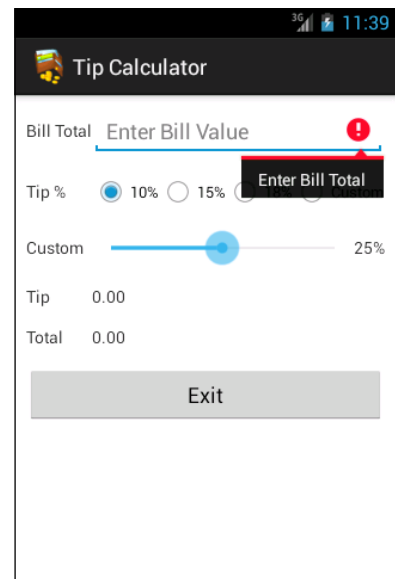
1. The tip radio buttons should enable the user to select one of tip options. If a tip option is selected the tip and total values should be calculated and update based on the selected tip option and the bill total. See Figure 2(a).
2. If the user updates or edits the bill total the tip and total values should be updated to reflect their new values.
3. If the bill total is empty, then tip and total values should be set to 0.00. In addition, use the setError() method to display an error message informing the user to “Enter Bill Total”. See Figure 2(c)
4. When the custom SeekBar is changed the current custom tip progress should be updated to show the SeekBar progress. If the custom tip option is selected, then the tip and total values should be calculated based on the current custom tip progress.
5. The application should be finished if the Exit button is clicked.



(a) Tip calculated using the 15% option



(b) Tip calculated using the custom option



(c) Error message when bill total is empty.

Figure 2, Application Screen Shots