Mobile Application Development
Homework 1

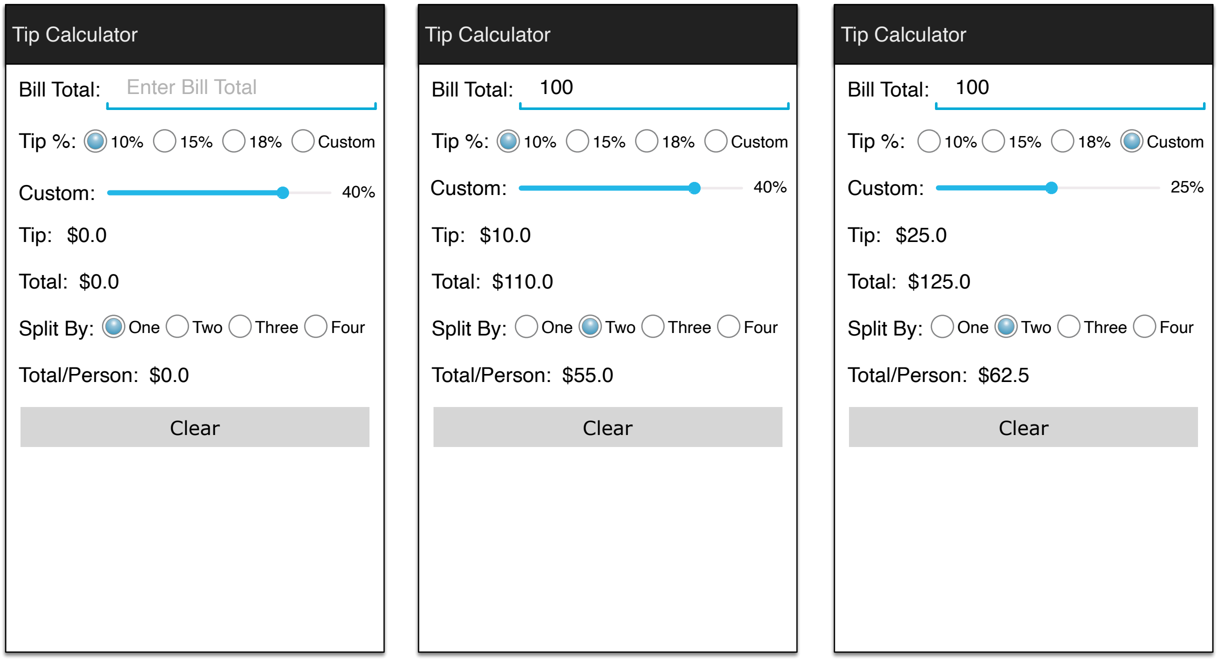
Basic Instructions:

1. In every file submitted you MUST place the following comments:
   a. Assignment #.
   b. File Name.
   c. Full name of the student.
2. Each group is required to submit the assignment on Canvas.
3. Submission details:
   a. Zip all the project folder to be submitted on canvas.
   b. The file name is very important and should follow the following format: **Group#_HW#.zip**
   c. You should submit the assignment through Canvas: Submit the zip file.

4. Failure to follow the above instructions will result in point deductions.

Homework 01 (100 points)

In this assignment you will build an 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. The app also provides the ability to split the overall total over multiple persons.



Tip Calculator

Bill Total: Enter Bill Total

Tip %: ● 10% ○ 15% ○ 18% ○ Custom

Custom: 40%

Tip: $0.0

Total: $0.0

Split By: ● One ○ Two ○ Three ○ Four

Total/Person: $0.0

Clear

Tip Calculator

Bill Total: 100

Tip %: ● 10% ○ 15% ○ 18% ○ Custom

Custom: 40%

Tip: $10.0

Total: $110.0

Split By: ○ One ● Two ○ Three ○ Four

Total/Person: $55.0

Clear

Tip Calculator

Bill Total: 100

Tip %: ○ 10% ○ 15% ○ 18% ● Custom

Custom: 25%

Tip: $25.0

Total: $125.0

Split By: ○ One ● Two ○ Three ○ Four

Total/Person: $62.5

Clear

(a) Initial App Screen

(b) Tip calculated using the 10% option

(c) Tip calculated using the custom option

Figure 1, Application Screen Shots

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 decimal numbers. When the app starts the bill total EditText should be empty, and should display the hint message “Enter Bill Value” as indicated in Figure 1(a).
4. Use a RadioGroup containing RadioButtons to enable the user to pick from the tip options 10%, 15%, 18% and Custom. When the app 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 40%. 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 app starts the tip and total values should be both set to $0.0 as shown in Fig 1(a).
7. The Split By radio buttons enable the user to indicate by how many persons should the Total should be divide by, “One” should be the default selection when the app starts.
8. The “Total/Person” label indicates the total due per person by dividing the Total by the number of persons, when the app starts this should be set to $0.0 as shown in Fig 1(a).
9. Finally, “Clear” button which should reset the activity to the initial states that matches the state shown in Figure 1(a).

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, total, and total/person values should be calculated and update based on the selected tip option, bill total and split by option. See Figure 1. Note that, if the custom tip option is selected, the tip percentage should be retrieved from the custom SeekBar.
2. 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, total, and total/person values should be calculated based on the current custom tip progress.
3. If the user updates or edits the bill total the tip, total, and total/person values should be updated to reflect their new values based on the selected tip and split by options.
4. If the bill total is empty, then tip, total and total/person values should be set to “$0.0” strings.
5. If a split by option is selected the total/person values should be calculated and update based on the selected tip option, bill total and split by option. See Figure 1.
6. The app should be reset to initial state (Fig 1(a)) if the “Clear” button is clicked.