Basic Instructions:

1. In every file submitted you MUST place the following comments:
    a. Assignment #.
    b. File Name.
    c. Full name
2. Each team is required to submit the assignment on Canvas.
3. Please download the support files provided with this assignment and use them when implementing your project.
4. Submission details:
    a. Compress the contents of your project folder. The file name is very important and should follow the following format: **HW1.zip**
    b. Only one group member is required to submit on behalf of the whole group.
    c. You should submit the assignment through Canvas: Submit the zip file.
5. The required Android Virtual Device (AVD) should have **minimum SDK version set to 16 and target SDK at 25**.
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 Blood Alcohol Content (BAC) Level Calculator application.
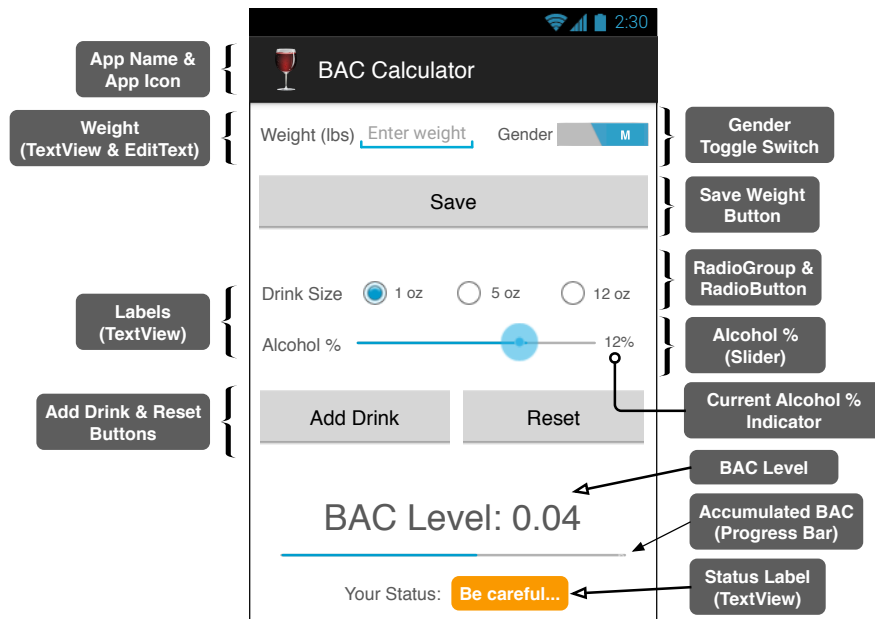

Figure 1, Application User Interface

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 Android Studio. 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 his/her weight in pounds. The EditText component should be setup to limit the weight value to only positive numbers. When the application starts the weight value EditText should be empty, and should display the hint message "Enter Weight" as indicated in Figure 1.
4. Use a Switch widget to allow a user to set his/her gender as either Male or Female.
5. Create a Save button to store the weight and gender; whenever these are entered, all calculations going forward should use the new weight and gender. Note that the text for the button should be retrieved from the string.xml file.
6. Use a RadioGroup containing RadioButtons to enable the user to pick from the drink size options of 1 ounce (a shot), 5 ounces (a glass), or 12 ounces (a can or bottle). When the application starts the 1 ounce choice should be selected.
7. Use the SeekBar to enable the user to pick a custom percentage of alcohol per

volume for the selected drink. The maximum alcohol percentage value should be set to 25%, and the slider should move in increments of 5%. When the application starts the percentage value should be set to 5%. On the right of the SeekBar use a TextView to display the current progress of the SeekBar, which represents the current alcohol percentage.

8. Create an Add Drink button and a Reset button; note that the text for the buttons should be retrieved from the string.xml file.
9. Use TextView components for creating the "BAC Level" and "Your Status" labels. When the application starts the "BAC Level" should be set to "0.00" and the "Your Status" should be set to "You're safe."
10. Use a ProgressBar to indicate the BAC level after each drink. The maximum progress should be set when the BAC reaches .25 or higher. When the application starts, the ProgressBar should be set to 0.
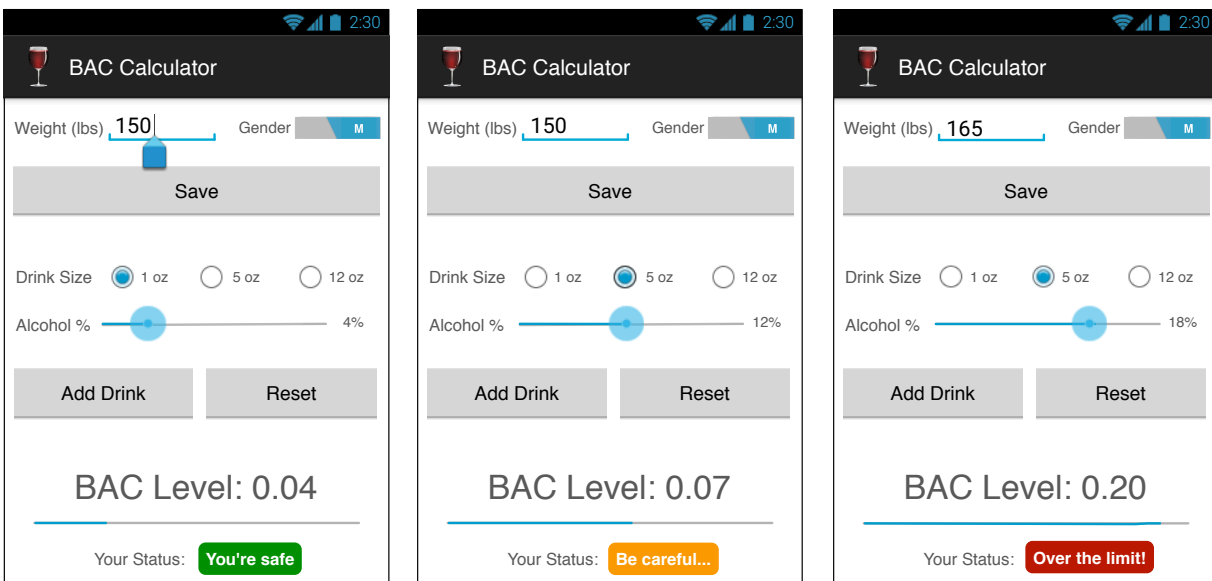
| (a) BAC calculated after the first drink | (b) BAC calculated after a drink is added | (c) BAC calculated with new weight and additional drink |

Figure 2, Application Screen Shots

Part 2 (70 Points): Event Handlers and App Behavior (MainActivity)
In this part you will build the required logic for the BAC calculator app. The requirements are as follows:
1. The BAC level cannot be calculated without a weight and gender value. If a user does not enter in and save a weight value and tries to add a drink, use the setError() method to display an error message informing the user to "Enter the weight in lbs."
2. If the user presses the Add Drink button, you should calculate what the current BAC level is using the currently selected weight, gender, and drink options, based on a simplified version of the "Widmark BAC Formula:" **% BAC = (A x 6.24 / (W x r))**. [Here we are ignoring the passage of time in the formula.] See Figure 2(a).

a. A = liquid ounces of alcohol consumed = ounces * alcohol percentage (i.e. 5 x .15)
b. W = a person's weight in pounds
c. r = a gender constant of alcohol distribution (.68 for men and .55 for women)
d. More information on calculating BAC can found here: http://www.teamdui.com/bac-widmarks-formula/

3. Once the BAC is calculated for the current drink, it should be added to the previous BAC level and the "BAC Level:" label should be updated to reflect the new BAC. Similarly, the ProgressBar should be updated to reflect the BAC, and should reach the max whenever the BAC is equal to .25 or higher.
4. Whenever the weight/gender is changed and the Save button is pressed, the accumulated BAC level should be recalculated based on the new weight/gender.
5. When the BAC level is less than or equal to .08, the "Your Status" label should read "You're safe." See Figure 2(b). When it is greater than .08 but less than .20, it should read "Be careful…" and when .20 or greater, it reads "Over the limit!" See Figure 2(c).
6. Whenever the BAC level reaches .25, the all buttons should be disabled, **except** for the Reset button. This should be the case, even if BAC was under .25 given a previous weight/gender, but reaches .25 or greater when recalculated with a new weight/gender. Display a Toast that says "No more drinks for you."
7. If the user presses the Reset button, this should reset the drink size and alcohol % parameters to their default values, and clear the saved weight and gender values. Any disabled buttons should be enabled again, if applicable.