

Android Assignments 1- 6.

We will have a series of assignments to get some experience with some of the basic Android functionality. You maybe tempted to spend a lot of time polishing your interfaces for these applications. While you are welcome to do so, the goal of these assignments is to get experience with the Android API. We will focus on polishing GUIs later in the course.

Note: Teams are not allowed on these assignments. You are to do these assignments by yourself. People turning in duplicate code will be treated as having cheated on the assignment.

1. Tip Calculator. Due Sept 25 23:55. Write a tip calculator activity. A User input the percent they wish to tip via a spinner. They type into a text field the amount that they wish to tip on. When they click on a "Calculate" button the amount of the tip and the total amount (tip plus original amount) are displayed. Goal of assignment: create an activities, get some experience using Android views.

Grading for Assignment 1

Working code that meets the specification	90%
Quality of Code	10%

2. Restaurant App. October 2 23:55. The application start screen shows a user sees a listing the of restaurants currently in the database. The list needs to scroll if it does not fit on one screen. The list only displays enough information about each restaurant (name?) so the user can identify it. When a user selects a restaurant, a second activity (restaurant detail view) is called to display all the information about the restaurant. This information includes the restaurant's name, address and phone number. The application start screen also has a menu item that allows the user add a new restaurant to the list. That menu item brings the user to an activity that allows the user to enter the information for a new restaurant. The information about the restaurants is to be stored in the Android database. Goal of assignment: use multiple activities, pass information between activities, use the Android database, more experience with views.

Grading for Assignment 2

Working code that meets the specification	75%
Quality of Code	10%
Unit tests	15%

3. Restaurant Map. October 9 23:55. In the restaurant app add a button to the (restaurant detail view) that when clicked opens a map view showing the location of the restaurant and the location of the user. Goal of assignment: use the location services and maps. For this assignment you need to sign up for an android maps API key with Google. See <http://code.google.com/android/add-ons/google-apis/mapkey.html> for more information.

Grading for Assignment 3

Working code that meets the specification	90%
Quality of Code	10%

4. Restaurant Widget. October 16 23:55. Create a restaurant widget which displays one restaurant. In your restaurant application restaurant detail view add a display button. When the user clicks the button a broadcast is sent to all running restaurant widgets on the phone to update themselves to display that information. Goal of assignment: learn how to create widgets and handle updates.
5. Drawing Circle App. October 30 23:55. When the application start up the first screen seen is a bitmap image of your construction. When the user clicks on the screen they go to another screen. Whenever they touch/click on the screen a circle is drawn centered at that location. Goal of assignment: dealing with bitmaps and 2d drawing.
6. Network App. Due Nov 14, 23:55. When the application starts it shows a list of name of photos that can be displayed. When the user clicks on a name the application starts a new activity that displays the photo selected by the user.

The application will obtain data from a photo server. The server is "bismarck.sdsu.edu" running on port 8009. The server responds to three commands: list, recent and photo.

All commands to the server end in a semicolon. The semicolon is required. It is used by the server to determine the end of your request. The sever closes the connection after responding to each request. So you have to open a new connection to the server for each request. The server returns a JSON array in response to a list or recent command unless there is an error. In case of an error (improper syntax of the request or some other problem) the server will return an error message as non-JSON string. When you try to parse the string as a JSON object you will get a JSONException.

The list command returns a list of photos currently on the server. To send the list command open a socket to the server and send the string:

```
list;
```

The semicolon at the end of the command is required. There are no characters after the semicolon. The server will respond with the list of photos in JSON format. The list is a JSON array containing objects. Each object has an id and a name. The name is for displaying to the user and the id is used to fetch the photo from the server. A sample list response from the server is:

```
[{"id":1,"name":"Cat"}, {"id":2,"name":"Dog"}]
```

After returning the string the server closes the connection (which means you can read to the end of file marker.)

The photo command returns a single photo from the server. To send the list command open a socket to the server and send the string:

```
photo:3;
```

The semicolon at the end of the command is required. There are no characters after the semicolon. The colon is required. The number after the colon is the id of the photo to be returned. The server returns a sequence of bytes that make up the jpeg photo. Do not read the bytes with a reader as it will treat the binary bytes as characters, just use streams. If there is an error the server does not return any bytes, it just closes the connection. After returning the photo the server closes the connection, so you can read until the stream returns a -1.

The command recent returns all photos posted to the server after a given date. This provides the application a method of finding out what photos were posted after the last time it checked with the server. A sample recent command is:

```
recent:11.2.2009;
```

The format is the string **recent** followed by a colon, followed by a date, followed by a semicolon. The format of the date is month (one or two digits) followed by a period, followed by the day of the month (one or two digits), followed by a period, followed by the year (four digits). The server responds with a JSON array with information of all the photos posted to the server after the date listed. The response may be an array with 0 or more elements. A sample response is:

```
[{"id":3,"name":"Moon"}]
```

How To Turn in Your Assignments

In Eclipse select all the Android projects that make up your assignment and export them using the "Export ..." item in the File menu. Export them as a single zip file using the "Archive File" option under "General" in the export wizard. Submit the generated file using the class portal.