Homework 2

Layouts, UI Widgets, and Server-Side Programming

Due Tuesday 6/2 @11:59pm (midnight)

Submission Requirement: You only need to submit your apk file (only one per person is allowed for this entire homework) via the Blackboard. See Homework 1 for detailed submission requirements.

Look and Feel (10 points): The quality of your UI will be evaluated for this homework.

Task 1 Activity with Menus (10 points). Starting from this lab, each of you homework itself is an app, which means, you need to develop navigation scheme that allow the TA to go to each of the tasks. TA is not going to read your user manual, even if you have one. Therefore, your navigation needs to be intuitive. We will learn various navigation patterns throughout this course, and will ask you to use them in your homework. In this homework, we will use the menu on the Action Bar for our navigation.

In this task, you need to write an activity that has an Action Bar. Each menu element should correspond to one of the tasks (except Task 1). When a menu item is clicked, its corresponding fragment should be loaded into the Activity.

Task 2 "About Me" Fragment (10 points). This fragment will be the requirement for all future homework. In this fragment, you need to show the following information:

- Your full name
- The class session that you belong to (e.g. CIS 400 or CSE 691, etc.)
- Your picture

You can customize how it looks. Since this fragment will be used again and again in future homework, it is a good idea to make it look nice. You can also improve it in future homework, after you have learned more about UI.

Task 3 `LinearLayout` Fragment (5 points). In this fragment, you need to build the layout shown in Figure 1. You can only use `LinearLayout`. Note, the labels on the button should also be like what are shown in the figure, because their lengths are intentionally made different.

Task 4 `RelativeLayout` Fragment (5 points). In this fragment, you need to build the layout shown in Figure 2. You can only use `RelativeLayout`.

Task 5 Displaying Movie Data (20 points). In this fragment, you need to display the movie information from the files provided by us, including `MovieData.java` and many movie images. Your fragment will have at least a Button. When the user clicks the button (labeled "Load Next Movie"), the data of another

movie will be loaded and displayed within the same fragment. Our movie data contain many fields, and you should display at least the following information:

- Title
- Description
- Image
- Year
- Stars
- Rating (Don't use TextView here; find the most suitable UI widget to display this information)

You are responsible for designing the layout of this fragment. Make it look nice. The 10 points on "look and feel" will mostly be used in here. This layout will be reused in future homework, so make sure you don't keep losing the "look and feel" points, even if you can't get all of it at the first time.

Orientation change: For this movie-data page, you need to prepare two layouts, one designed for the portrait mode, and the other designed for the landscape mode. When the phone's orientation changes, the display of the movie data should be aware of the orientation.

Task 6 SeekBar (10 points). In this fragment, you need to have a SeekBar and an ImageView. The ImageView can display an image of your choice. The mark of the SeekBar should be placed in the middle initially. You need to do the followings:

- When the user slides the SeekBar, the size of the image should change accordingly.
- When the user long-presses the image, the mark of the SeekBar goes to the middle (i.e., the initial position), and the size of the image should change accordingly.

Task 7 Server-side Programming (30 points): For this task, please submit your PHP code and a report with screenshots to demonstrate that you have done this task.

- (10 points) Load your movie data into a browser using this URL: **www.example.com/movies/**. It should be noted that the URL ends with a "/". You need to write your server-side program, but you have to use this URL. Your server-side code should return an array of json objects, just like what you get from http://www.cis.syr.edu/~wedu/Teaching/android/Labs/json/movie.json. You only return the following selected fields in your json array: description, name, image, rating, url, and id.
- (10 points) When a user types this URL: **www.example.com/movies/rating/8.0**, the server should return an array of json objects containing all the movies with the rating larger than 8.0. The number 8.0 should be replaced by whatever is typed, but the prefix **www.example.com/movies/rating/** should be the same. You should write the server-side code.
- (10 points) When a user types this URL: **www.example.com/movies/id/titanic** (there is no "/" at the end of the URL), the details of movie titanic should be returned. Your server-side code should return a single json object, which is like what you get from http://www.cis.syr.edu/~wedu/Teaching/android/Labs/json/titanic.json

Appendix

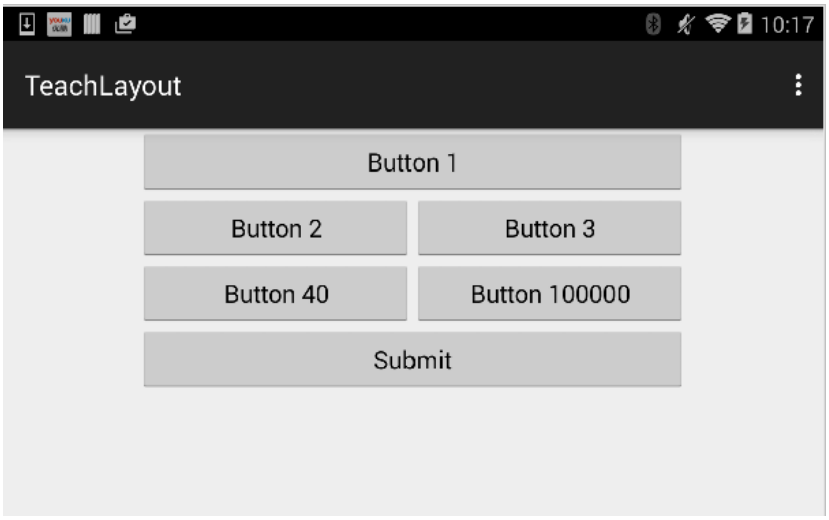

Figure 1

Figure 2