ITIS/ITCS 5180 Mobile Application Development
In Class Assignment 6

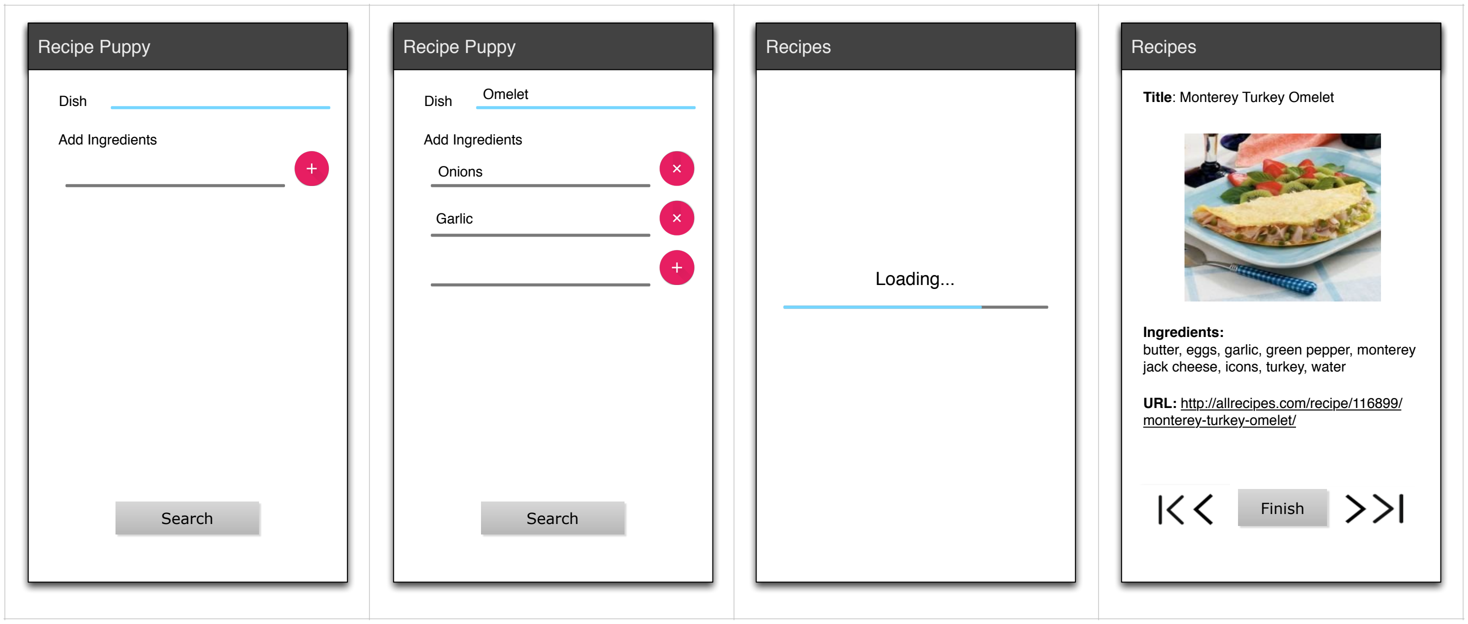
Basic Instructions:

1. This is an individual assignment.
2. In every file submitted you MUST place the following comments:
   a) Assignment #.
   b) Full names of the student.
3. Your assignment will be graded for functional requirements and efficiency of your submitted solution. You will lose points if your code is not efficient, does unnecessary processing or blocks the UI thread.
4. Please download the support files provided with this assignment and use them when implementing your project.
5. Submission details:
   a) Export your Android project and create a zip file which includes all the project folder and any required libraries.
   b) The file name is very important and should follow the following format: **InClass06_800#.zip.** You should submit the assignment through Canvas: Submit the zip file.
6. The Min SDK version should be 20, and the Target SDK version should be 25.
7. **Failure to follow the above instructions will result in point deductions.**

In Class Assignment 06 (100 Points)

In this assignment you will develop the same “RecipePuppy recipe finder” app using the XML API. The App displays recipes from Recipe Puppy (http://www.recipepuppy.com/). This allows you to search the recipes of the dish you are interested in, with proper ingredients. In this assignment, you will learn how to parse JSON.

The base URL of the API is: http://www.recipepuppy.com/api/?format=xml&i=<INGREDIENTS LIST, COMMA SEPARATED>&q=<DISH NAME>. An example URL is: http://www.recipepuppy.com/api/?format=xml&i=onions,garlic&q=omelet .



Title: Monterey Turkey Omelet

Ingredients:
butter, eggs, garlic, green pepper, monterey jack cheese, icons, turkey, water

URL: http://allrecipes.com/recipe/116899/monterey-turkey-omelet/

(a)Search Screen

(b) Added a list of ingredients

(c) Loading recipes

(d) Display recipes

Figure 1, App Wireframe

In this assignment we will build two activities. The first activity is to search, and the second activity is to display the results of the recipe search.

Search Activity (40 Points)

You need to implement the following:

1. An EditText to put the dish name.
2. A scrollable add ingredients panel which can **add up to 5 ingredients** in total.
3. The panel to add ingredients should be implemented using Dynamic Layout. It should follow the same pattern you did in the In Class 05, see figure 1 (b).

4. It should display an empty EditText, and a floating action Add button. Once, you add one ingredient, the floating action button should be changed from Add to Remove button. Follow figure 1(b).
5. Then you need to create the URL as: http://www.recipepuppy.com/api/?i=<INGREDIENTS LIST, COMMA SEPARATED>&q=<DISH NAME>.
6. There should be a Search button. Clicking on that button should take you to the Display recipes activity where it displays a list of recipes.

Recipes Activity (60 Points)

1. Use either XML PullParser or SAXParser to parse XML.
2. Use an AsyncTask or Thread pool to communicate with the RecipePuppy api and to parse the results. Do not use the main thread to parse them.
3. In our example, we wanted to search for a Omelet recipe having two key ingredients: Onions, and Garlic.
4. In received XML, you will find that there are two levels of the hierarchy. Inside “**<recipes>**”, you will find 10 “**<recipe>**”s. You need to **parse all the items in results**.
5. Use AsyncTask or Thread pool to retrieve and parse it.
6. You need to implement a ProgressBar to display the progress while it is parsing, see figure 1(c).
7. Finally display your result. Please follow the instructions:
    1. You have four things to display:
        1. Title (<title> in XML)
        2. Recipe image (**Display the image from the link: https://c1.staticflickr.com/5/4286/35513985750_2690303c8b_z.jpg**).
        3. Ingredients (<ingredients> in XML)
        4. URL (<href> in XML)
    2. You need to use a separate AsyncTask when you load the image. Alternatively, you can use Picasso library to load the image. (http://square.github.io/picasso/).
    3. The URL should be clickable, use implicit intents to open it in browsers.
    4. Next you need to implement a navigation pane. It will have five buttons.
        1. The leftmost button is the indicator of first recipe.
        2. The second button indicates the previous recipe.
        3. The Finish button is to close the app.
        4. The fourth button indicates the next recipe.
        5. The rightmost button indicates the last recipe.
    5. Corner Cases:

   1. Last recipe: clicking on the next button should not take you anywhere. Toast to confirm that it is the last recipe.
   2. First recipe: clicking on the previous button should toast to display that it is the first recipe.
   3. If no recipe is found, return to the main activity and toast to report that there were no recipes found.

6. Clicking on Finish button should Finish the second activity and return you to the main activity.

Teamwork (10 Points)

Good Luck!