# **CST2335 Lab 7**

This assignment relates to the following Course Learning Requirements:

* CLR 1: Write a graphical user interface-based program, given design documents. Techniques used will be object-oriented programming, structured programming, top-down coding, and event-driven coding.
* CLR 4: Debug program problems using manual and programmatic methods.
* CLR 7: Create GUI programs that support internationalization as well as sustainability.

**Objective of this Assignment:**

# The following is an exercise to help you implement what you have learned in the module. It will allow you to familiarize yourself with the module content as well as practice the skills required to develop software in the Android Studio. The goal of this lab is to become familiar with Fragments. We will create a layout so that it uses an activity with fragments.

# **Pre-Assignment Instructions:**

1. Review the Module 7 content in Brightspace.

# **Assignment Tasks:**

1. Create a new branch in Git. Start by selecting the “Git” menu in Android Studio and select “Branches”. In the resulting dialog click “New branch” and call it “lab\_7”. This lab will create an encyclopedia for Stars Wars information
2. Delete any activities created in previous labs and create a new MainActivity starting from an empty activity. Make sure in res -> values -> themes -> themes.xml, the parent theme is using an ActionBar if that was changed in the previous lab. You can also remove the extra “use-features” from your Android manifest that were added in the previous lab.
3. In the Android project view on the left side of Android Studio, make sure that AndroidStudio is using the “Project Files” view of your project so that you can see if there is a “layout-sw720dp” folder under the “res” folder. If there is not one, then create the folder. You’re going to start by creating two layouts:
   1. In the regular “layout” folder, create a new layout with a ListView that takes up the entire size of the layout.
   2. In the “layout-sw720dp” folder, start with the above layout as a base, but set the ListView width to be 300dp instead of match\_parent. Add a “FrameLayout” to the right of ListView that takes up the remaining space.
4. Right-click on the “App” folder in your project window. On the menu, select “New” -> “Fragment” -> “Fragment (Blank)”. On the next window, give the name “DetailsFragment”. This will generate a java class that extends Fragment, and an xml file that is the layout. The onCreateView function will already be generated that just inflates a layout and returns it, the rest of the generated functions can be removed.
5. Modify the generated fragment layout so that it looks like this:  
   Shape

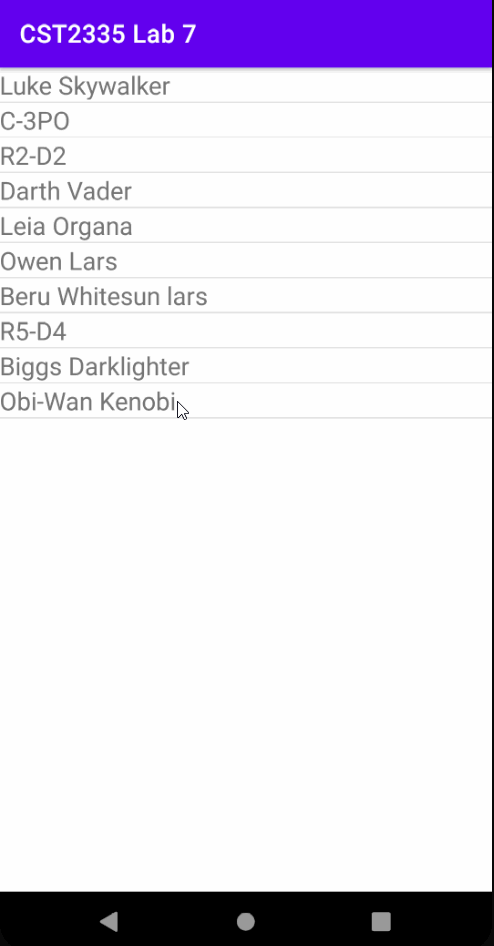
   Description automatically generated  
   Everything is a TextView with a textSize of 20, but the titles also have a textStyle=”bold”
6. In DetailsFragment.java, use getArguments() to get the bundle passed when the Fragment was created, and populate the “fill\_me” TextViews in the layout with the correct values from the bundle.
7. Add an AsyncTask to your MainActivity, it should query the Star Wars API for the list of characters, this endpoint is <https://swapi.dev/api/people/?format=json>. You will want to save all the information returned so it can be passed to the fragment to be displayed.
8. Add a class that extends BaseAdapter to run the ListView. After the AsyncTask completes it should populate the ListView with all the names returned.  
     
   Graphical user interface, text, application

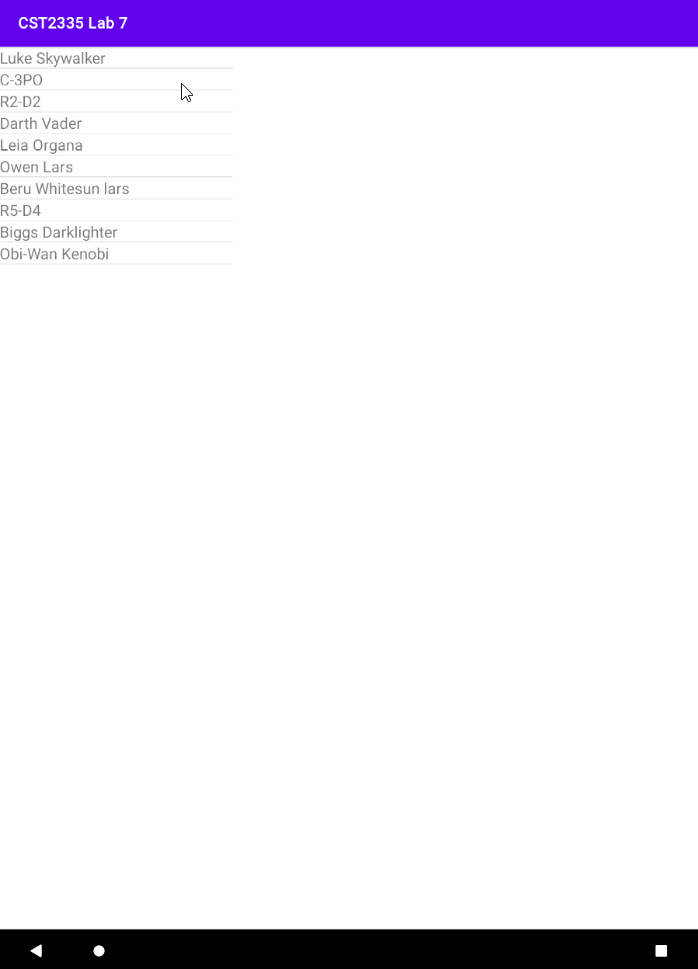
   Description automatically generated
9. In AndroidStudio, create a new Activity called EmptyActivity by using the menus “File” -> “New” -> “Activity” -> “Empty Activity”. It should generate the java and xml files for your new activity. Open the newly created layout file and select all the xml code and delete it. Copy and paste the <FrameLayout> tags that you created from step 3b and paste it into the file. It is important that FrameLayout is now the root tag.
10. In the EmptyActivity, write the code to create the DetailsFragment and use the Fragment Manager to replace the FrameLayout with the fragment.
11. In the MainActivity, add a setOnItemClick listener to the ListView, and use findViewById() to look for the id of the FrameLayout. If it returns null you are on a phone, otherwise you are on a tablet.
    1. If you are on a phone, pass the relevant information in a bundle to the EmptyActivity where the fragment will be displayed.
    2. If you are on a tablet, invoke the Fragment Manager directly to replace the FrameLayout with the DetailsFragment like in step 9.

# **Hints:**

* Examine the payload returned from the Star Wars API, all of the info in stored in a list inside the “results” key. You can get access to this list with a snippet like:  
    
  JSONArray characters = new JSONObject(responseText).getJSONArray("results");  
    
  You can get the length with “characters.length()”, or access a specific entry with “characters.getJSONObject(index)”, and then retrieve the specific keys you need.
* You can not make network calls from the UI thread, so you’ll need to use an AsyncTask to call the API. Remember to notify your ListView adapter when the data has been retrieved.

Here’s a full recording of how the app should function on a phone:



Here’s a full recording of how the app should function on a tablet:

# **Rubric**

|  |  |
| --- | --- |
| **Criteria** | **Points** |
| The app compiles and runs without crashing | 2 |
| When using a tablet at least 720 pixels wide, the detailed information of the selected item appears in a fragment beside the ListView | 3 |
| When using a phone, selecting an item from the ListView shows details info on a fragment in a new activity that takes up the whole screen | 3 |
| Information is correctly queried and displayed from the Star Wars API. | 2 |
| **Total** | **10** |