Mobile Application Development
Homework 04

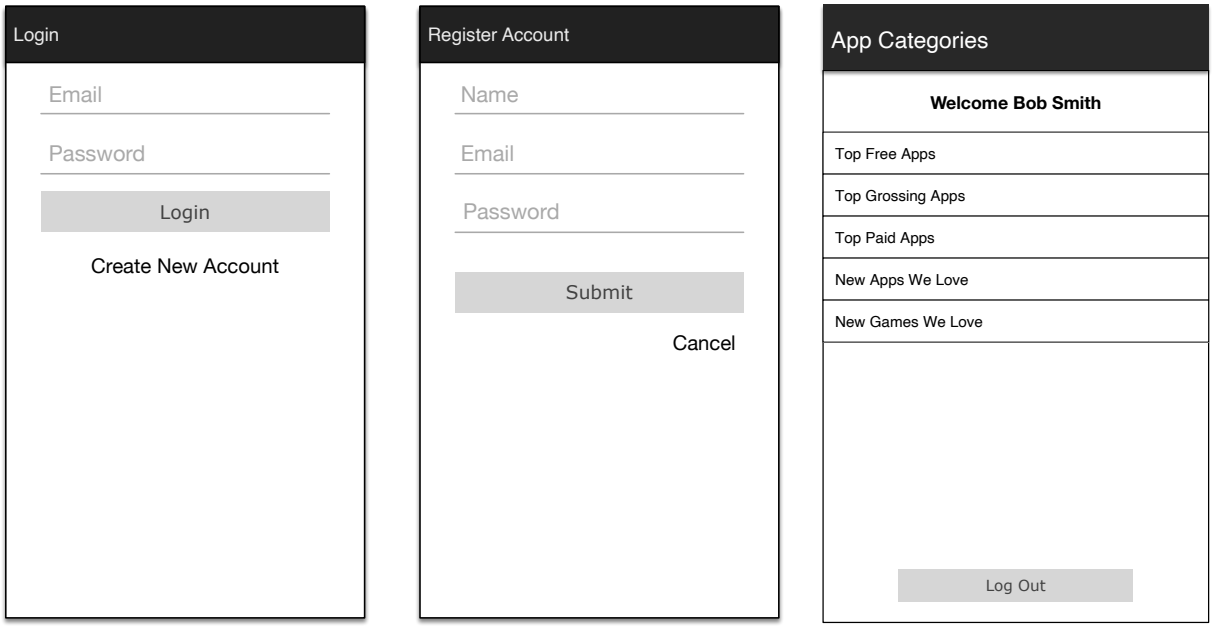
Basic Instructions:

1. In every file submitted you MUST place the following comments:
   a. Assignment #.
   b. File Name.
   c. Full name of the student.
2. Each group is required to submit the assignment on Canvas.
3. Please download the support files which include a Java project to be used for this assignment.
4. **Submit Codes:**
   a. Zip all the project folder to be submitted on canvas.
5. Submission details:
   a. The file name is very important and should follow the following format:
      Group#_HW04.zip
   b. You should submit the assignment through Canvas: Submit the zip file.
6. **Failure to follow the above instructions will result in point deductions.**

Homework 04 (100 points)

In this assignment you will develop “iTune Apps” app, which enables the user to display apps from different categories.You are provided with support files the include data and classes to use for this project. The Data class that includes a dictionary of apps lists for different categories.

- You are provided with a DataServices class which you need to import in your project. The DataServices class should emulate the account and data management functions.
- ***Note that all the DataServices methods should be called in the background and NOT on the main thread. You are free to use either AsyncTasks or Threads for this assignment.***
- ***Note that all UI changes should be performed on the main thread only.***
- In this app you will have only one Activity and 5 fragments, all communication between fragments should be managed by the activity.



(a) Login Fragment (b) Register Fragment (c) App Categories Fragment

Figure 1, Application Wireframe

Part 1: Login Fragment (15 Points)

The interface should be created to match the UI presented in Figure 1(a). The requirements are as follows:

1. Upon entering the email and password, clicking the submit button should:
   a. Clicking “Login” button, if all the inputs are not empty, you should attempt to login the user by calling the DataServices.login() method. ***Note that the DataServices.login() method should be called on a child thread using AsyncTask or Thread.***
   b. If there is missing input, show a Toast indicating missing input.
   c. If the login is successful, then communicate the returned account with the activity and replace the current fragment with the App Categories Fragment.

d. If login is not successful, show a Toast message indicating that the login was not successful.

2. Clicking the “Create New Account” should replace this fragment with the Register Fragment.

Part 2: Register Fragment (15 Points)

The interface should be created to match the UI presented in Figure 1(b). The requirements are as follows:

1. This fragment should allow a user to create a new account. Upon entering the name, email and password, clicking the Submit button should:
   a. If all the inputs are not empty, you should attempt to signup the user by calling the DataServices.register() method. ***Note that the DataServices.register() method should be called on a child thread using AsyncTask or Thread.***
   b. If the registration is successful then communicate the returned account with the activity and replace the current fragment with the App Categories Fragment.
   c. If the registration is not successful, show a Toast message indicating that the login was not successful.
   d. If there is missing input, show a Toast indicating missing input.
2. Clicking “Cancel” should replace this fragment with the Login Fragment.

(a) Apps List (b) App Details

Figure 2, App Wireframe

Part 3: App Categories (20 points):

This screen contains a ListView/RecyclerView of app categories retrieved from the DataServices by calling getAppCategories method. This screen is shown in Figure 1(c). The requirements are as follows:

1. The app categories should be retrieved from DataServices by calling DataServices.getAppCategories() method. Note that you should provide the token retrieved during login/registration in order to retrieve the list of categories. ***Note that the DataServices.getAppCategories() method should be called on a child thread using AsyncTask or Thread.***
2. The list of categories should be displayed using a ListView/RecyclerView as shown in Figure 1(c).
3. The app should also greet the user by showing "Welcome <name>" at the top of the screen, in order to retrieve the account the DataServices.getAccount() method. ***Note that the DataServices.getAccount() method should be called on a child thread using AsyncTask or Thread.***
4. If the user clicks on a category list item:
   a. Replace the current fragment with the App List Fragment, pass the selected app category to the App List Fragment.
   b. Push the current fragment on the back stack.

3. Clicking the "Logout" button should delete the account/token stored in the Main Activity, and replace this fragment with the Login Fragment.

Part 4: Apps List (40 points):

This screen contains a ListView/RecyclerView of apps based on the selected category in the App Categories Fragment as shown in Figure 2(a). The requirements are as follows:

1. The category is passed from the App Categories Fragment should be retrieved, the activity title should be set to the selected category name.
2. The list of apps corresponding to this category should be retrieved by calling the DataServices.getAppsByCategory() method. The list should be used to populate the ListView/RecyclerView as shown in Figure 2(a). Each list item shows the app name, artist name, and release date. ***Note that the DataServices.getAppsByCategory() method should be called on a child thread using AsyncTask or Thread.***
3. If the user clicks on an App list item:
   a. Replace the current fragment with the App Details, pass the selected app to the App Details Fragment.
   b. Push the current fragment on the back stack.

Part 5: App Details (10 points):

This screen displays the app details as shown in Figure 2(b). The requirements are as follows:

1. This screen should receive the app object from the previous Apps List fragment and should show the app details as shown in Figure 2(b).
2. The app details include the app name, artist name, and release date. In addition, the app genres should be displayed in a ListView/RecyclerView as shown in Figure 2(b).