ITIS/ITCS 4180 Mobile Application Development
Final Exam

Basic Instructions:

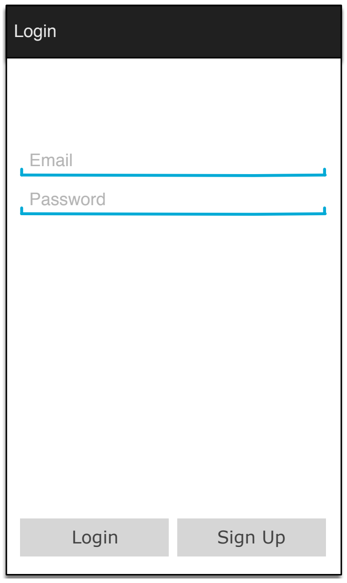
1. This is the Final Exam, which will count for 20% of the total course grade.
2. This Final Exam is an individual effort. Each student is responsible for her/his own Final Exam and its submission.
3. Once you have picked up the exam, you may not discuss it in any way with anyone until the exam period is over.
4. During the exam, you are allowed to use the course videos, slides, and your code from previous home works and in class assignments. You can use the internet to search for answers. You are NOT allowed to use code provided by other students or solicit help from other online persons.
5. Answer all the exam parts, all the parts are required.
6. Please download the support files provided with the Final Exam and use them when implementing your project.
7. Your assignment will be graded for functional requirements and efficiency of your submitted solution. You will loose points if your code is not efficient, does unnecessary processing or blocks the UI thread.
8. Export your Android project and create a zip file which includes all the project folder and any required libraries. The file name is very important and should follow the following format: **800#_Final_Exam.zip.** Submit the exported file using the provided canvas submission link.
9. **Do not try to use any Social Messenger apps, Emails, Or Cloud File Storage services in this exam.**
10. **The required Android Virtual Device (AVD) should have minimum SDK version set to 20 and target SDK at 25.**
11. **Failure to follow the above instructions will result in point deductions.**
12. **Any violation of the rules regarding consultation with others will not be tolerated and will result disciplinary action and failing the course.**

Final Exam (100 Points)

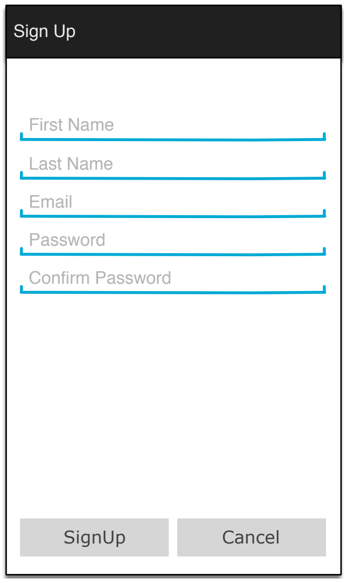
In this final exam assignment you will develop an application that uses the Apple iTunes API and Firebase. The app enables the user to login, sign up, maintain a list of top grossing iTunes apps, and be able to indicated their favorite apps.

Initial Setup

1. You are provided with a skeleton application that includes all the layouts, resources, strings and some partial code.
2. Integrate Firebase Realtime Database and Firebase Authentication into your application.
   a) In the Firebase console enable the Email/Password Sign-In provider.



(a) Login Screen



(b) Sign Up Screen

Figure 1, App wireframe

Part 1: Login (5 points)

This is the launcher screen of the app. The wireframe is shown in Figure 1(a). The requirements are as follows:

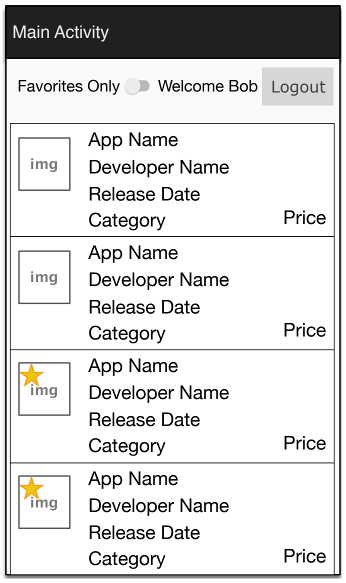
1. The user should provide their email and password. The provided credentials should be used to authenticate the user using Firebase Email/Password login. Clicking the "Login" button should submit the login information to Firebase to verify the user's credentials.
   a) If the user is successfully logged in then start the Main Activity, and finish the Login Screen.

   b) If the user is not successfully logged in, then show a toast message indicating that the login was not successful.
2. Clicking the "Sign Up" button should start the Signup Screen Figure 1(b), and finish the login Screen.

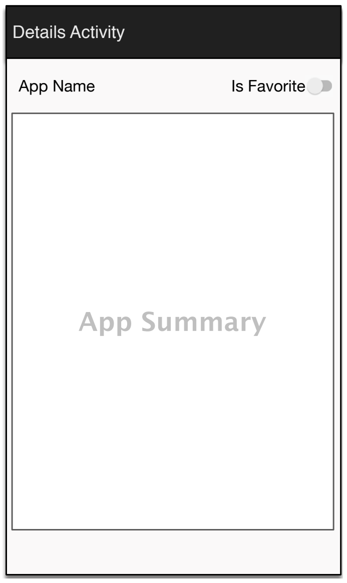
Part 2: SignUp (5 points)

The Sign Up activity is shown in Figure 1(b), with the following requirements:

1. Clicking the "Cancel" button should finish the Signup Screen and start the Login Screen.
2. The user should provide their first name, last name, email, password and password confirmation. Preform the required validation(the given password and the repeated password must match). Clicking the "Sign Up" button should submit the user's information to Firebase signup.
   a) If the signup is not successful display an error message indicating the error message received from Firebase.
   b) If the signup is successful, then display a Toast indicating that the user has been created. Then start the Main Activity and finish the Signup Screen.



(a) Main Activity



(b) Details Activity

Figure 2, App wireframe

Part 1: Main Activity (60 points)

The **Main** activity is responsible for retrieving and displaying the list of top grossing applications in the iTunes store. In addition, it allows the user to maintain a list of apps and to indicate their favorite apps. The requirements are as follows:

1. "Welcome" followed by the name of the currently logged in user should be displayed at the top of the activity as shown in Figure 2(a).

2. When this activity first loads it should retrieve the list of top grossing iTunes applications using the iTunes api.
    a) Use the following api: http://itunes.apple.com/us/rss/topgrossingapplications/limit=30/json
    b) Use a thread pool (or AsyncTask) to communicate with the iTunes api and to parse the JSON. The JSON parsing should be performed and the list of Apps should be retrieved and stored in an ArrayList of App objects.
    c) You are provided with an App class that includes the attributes that should be parsed from the JSON.
3. This application should use Firebase to maintain the list of top grossing applications and to enable users to indicate and maintain their favorite apps. Each user should maintain their own personal list of apps. A straight forward design is to store the retrieved list of apps from the iTunes api to Firebase for the currently logged in user, and to maintain a favorite flag for each app indicating it is favorite or no.
    a) Make sure to avoid duplication to avoid creating multiple app entries for the the same user and the same app. You could use the app id which is retrieved from the JSON to avoid duplication.
    b) You should carefully design you data structure in Firebase to ensure:
        a) Each user has their own list of apps and does not interfere with other users.
        b) For each user the list of apps does not have duplicate apps.
    c) Every time the Main Activity starts it is going to retrieve the list of top grossing iTunes app using the iTunes api, it will also attempt to update the list of apps stored on Firebase, you should make sure to maintain the user previous preferences while updating Firebase, and make sure there are no duplicates.
4. The ListView should display the list of apps stored on Firebase which includes both favorite and non-favorite top grossing apps.
    a) Create a custom adapter to display the apps as indicated in Figure 2(a).
    b) Favorite apps should show a “Star” image above the app’s image as shown in Figure 2(a).
    c) Clicking on a list item should open the “Details” Activity and pass it the required data.
5. The Favorites only switch is used to filter the types of apps displayed in the list view:
    a) If the switch is checked : refresh the list and only display the list of favorite apps.
    b) If the switch is unchecked : refresh the list and display all the apps.
6. Clicking on the “Logout” button should logout the current user, start the “Login” screen and finish the current screen.

<u>**Part 2: Details Activity (30 points)**</u>

The **Details** activity is enables the user to view the app summary and to change the favorite status of a specific app. The requirements are as follows:

1. Upon loading perform the following initializations:
    a) Display the app name and app summary.
    b) The “Is Favorite” switch should be checked if the app is a favorite app, and should be unchecked if otherwise.

2. The “Is Favorite” switch should be used to control the favorite status of the currently displayed app and if the switch status is changed then the data on Firebase should be updated to reflect the new favorite status of the app.
3. Upon returning from this activity the Main Activity will be displayed, and the list should be refreshed to reflect the change if favorite status upon any changes performed in Details Activity.

Task	**Grade**
Sign In	5
Login	5
Main Activity : JSON Retrieval and parsing of top grossing apps from iTunes API	10
Main Activity : Storing App list on Firebase for each user and avoiding duplicates	10
Main Activity : Avoiding duplicates in the app list stored for each user.	10
Main Activity : Custom Adapter and displaying the apps information, images and favorite status correctly.	15
Main Activity : Favorite filtering	10
Main Activity : Display User full name	3
Main Activity : Logout	2
Details Activity : Initialize and display App Name and Summary	2
Details Activity : Initialize and display “Is Favorite” switch correctly	3
Details Activity : Correctly updating App favorite status on Firebase based on the “Is Favorite” switch updates.	15
Details Activity : Correctly refreshes the list in the Main Activity to reflect the favorite status changes.	10
Total	**100**