ITIS/ITCS 4180/5180 Mobile Application Development
Makeup Midterm

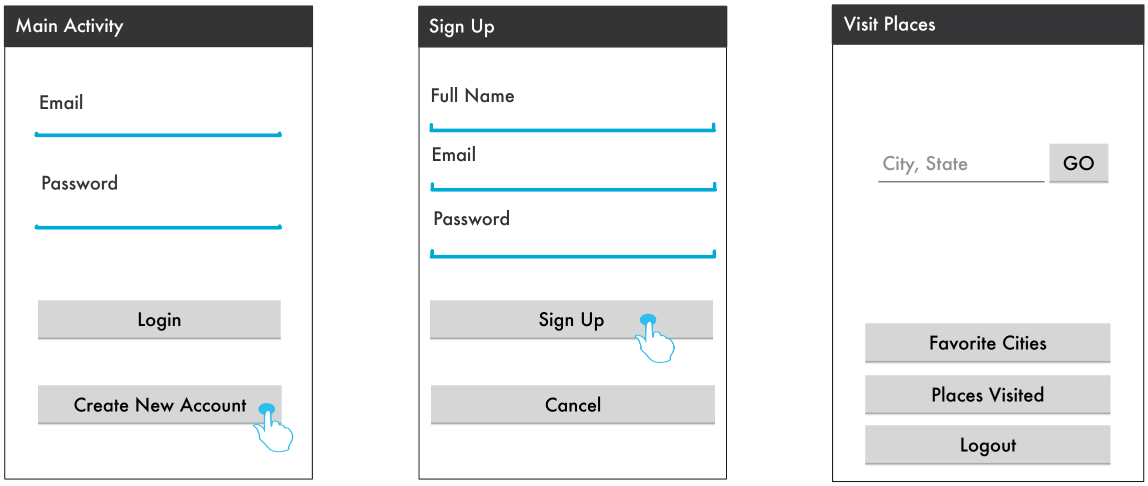
Basic Instructions:

1. This is the Midterm Makeup Exam, which will count for 20% of the total course grade.
2. In every file submitted you MUST place the following comments:
   a. Your Full Name.
3. This Midterm is an individual effort. Each student is responsible for her/his own Midterm and its submission.
4. Once you have picked up the exam, you may not discuss it in any way with anyone until the exam period is over.
5. During the exam, you are allowed to use the course videos, slides, and your code from previous home works and in class assignments. You are NOT allowed to use code provided by other students or from other sources.
6. Answer all the exam parts, all the parts are required.
7. Please download the support files provided with the Midterm and use them when implementing your project.
8. 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.
9. 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: **Lastname_Midterm.zip.** Submit the exported file using the provided canvas submission link.
10. Once you choose to take and submit the Makeup Midterm, no marks from the previous Midterm exam will be considered. We will replace the previous grade with the grade you get from this exam.
11. **Failure to do the above instructions will result in loss of points.**
12. **Any violation of the rules regarding consultation with others will not be tolerated and will result disciplinary action and failing the course.**

Makeup Midterm (100 points)

In this assignment, you will develop an app combining FourSquare API with Firebase. The application should be able to load different venues. You can see all the activities and attractions in a particular venue in details. Your app should be able to save your visited places and favorite cities. You will use Firebase for providing data storage. Below is the API Setup:

1. Go to https://foursquare.com, and create a new account
2. Go to https://developer.foursquare.com, and then click “Get Started”.
3. Click on create an app. Fill all the info needed, and then click “Save changes”
4. This will create 2 parameters **Client ID** & **Client Secret**. These parameters will be used later in the search API.



(a) Login Screen (b) Sign Up Screen (c) Visit Places

Figure 1: Sign In and Sign Up using Firebase.

Part 1: User SignUp and Login (10 points)

You should implement both login and signup functionalities. You should use Firebase to register a user with an email and password, and also store the user’s full name, email address, and password in the User object. The requirements are as follows:

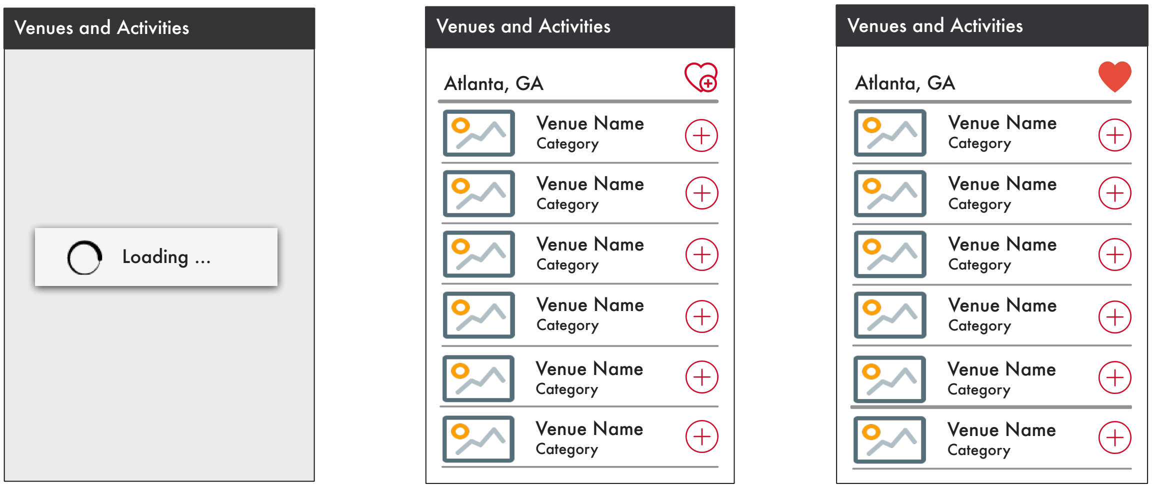
1. In Firebase make sure to enable Email & Password Authentication.
2. The launcher activity should be set to the Login activity. When the app first starts, the Login activity should check if there is a current user session, by using the Firebase provided methods to check if there is a valid current user.
   a) If there is a current valid user, then start the Visit Places activity, and finish the Login activity.
   b) If there is no current valid user, then the Login activity should be used to provide user login.
3. Create a Login Activity (Figure 1a):
   a. The user should provide their email and password. The provided credentials should be used to authenticate the user using Firebase. Clicking the “Login” button should submit the login information to Firebase to verify the user’s credentials.
      i. If the user is successfully logged in then start the Visit Places activity, and finish the Login activity.
      ii. If the user is not successfully logged in, then show a toast message indicating that the login was not successful.
   b. Clicking the “Create New Account” button should start the Signup activity and finish the login activity.
4. Create a Sign Up activity (Figure 1b):
   a) Clicking the “Cancel” button should finish the Signup activity and start the Login activity.

b) The user should provide their Full name, email and password. The provided credentials should be added as a registered user on Firebase. Clicking the "Sign Up" button should submit the user's information to Firebase to verify the user's credentials.
   i. If an account with the same email already exists, display an error message indicating that the account was not created and the user should select a different email.
   ii. If an account with the provided credentials does not already exist, then store the new account information and display a Toast indicating that the user has been created. Then start the Login activity and finish the Signup activity.
   iii. Note that, the user data should be stored in Firebase Users object because the user authentication for registered users will not take Full name in Firebase.

Part 2: Visit Places (5 Points):

This activity displays a search box where you can search a place where you want to visit. The requirements are as follows:

1. The EditText (search box) is where the user should enter the city and state to visit (for example, "Atlanta, GA"). If the user taps the "Go" button, should take the user to the "Venues and Activities" Screen. The app should check if the user has provided a correctly formatted city and state.
2. Tapping the "Favorite Cities", should take the user to the "Favorite Cities" activity.
3. Tapping the "Places Visited", should take the user to the "Places Visited" activity.
4. Tapping the "Logout", should logout the user, open to the Login activity, and finish the current activity.



(a) Calling API and loading (b) Venues in Atlanta, GA (c) Adding city to favorites

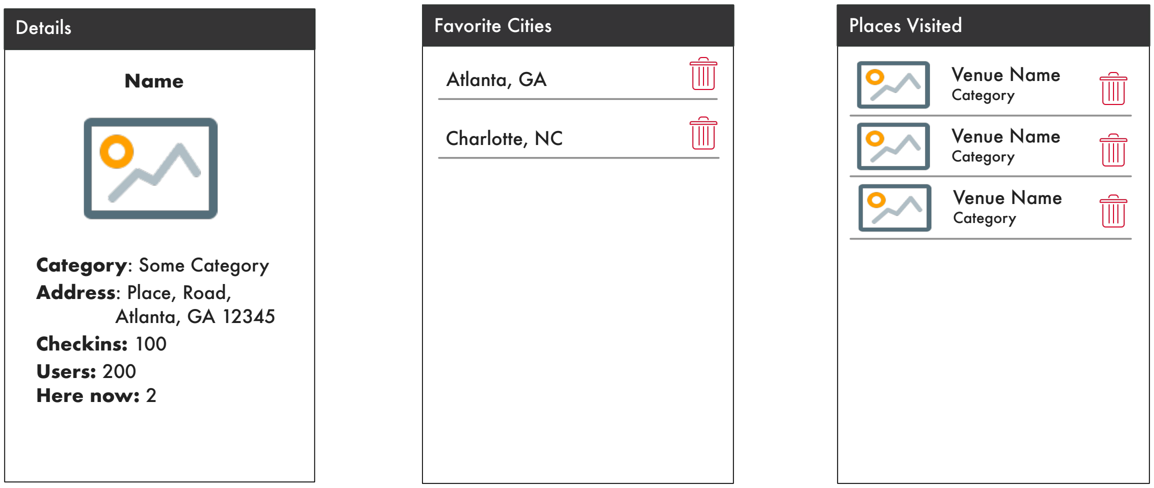
Figure 2: Visit Places

Part 3: Venues and Activities (65 Points):

This activity displays the list of venues at the provided city. The requirements are as follows:

1. This activity should receive the city and state from the "Visit Places" activity.
2. Display the name of the city at the top of the screen see Fig 2(b). The favorite icon is displayed beside the city name as follows:
   a) If the city is not in user's favorite cities on Firebase then show the empty heart with plus sign as shown in Fig 2(b). Clicking the icon should add the city to the favorite cities on Firebase and should update the icon to reflect this change.
   b) If the city is in user's favorite cities on Firebase then show the red heart as shown in Fig 2(c). Clicking the icon should remove the city from the favorite cities on Firebase and should update the icon to reflect this change.
3. Show a progress dialog while the data is being loaded from Foursquare, see Fig 2(a).
4. To access the venues search api, call: https://api.foursquare.com/v2/venues/search?client_id=**CLIENT_ID**&client_secret=**CLIENT_SECRET**&v=**20140806**&near=**CITY,STATE** .

   a) **client_id & client_secret**: use the values from the generated keys on the website
   b) **v**: this parameter represents the date, i.e. **YYYYMMDD**. Use today's date.
   c) **near**: is the city, state value provided from the previous activity
5. The API response is the list of venues near the city provided. You need to retrieve venue id, name, icon url, category name, address, checkins count, users count, and here now.
6. The list of items displayed is the list of venues near the specified city that the user **has not visited so far**. Firebase should be used to manage the storage of cities visited by the user. You can either use ListView or RecyclerView to display the list (Figure 2). For each list item you should display the venue name, icon, the category and a "plus" icon.
7. If the user taps on the "plus" icon for a row item, this should add the selected item to the list of visited places on Firebase, and should reload the list to display only the venues that are not in the user's visited places.
8. If the user taps on a row item, display the Details activity which shows the details of the selected venue, see Figure 3(a).



Category: Some Category
Address: Place, Road,
Atlanta, GA 12345
Checkins: 100
Users: 200
Here now: 2

(a) Details View of a place. (b) Favorite Cities (c) Favorite Places

Figure 3: Other Wireframes

Part 4: Favorite Cities and Places Visited (20 Points):

Implement the following requirements:

1. The "Favorite Cities" activity should display the list of favorite cities retrieved from Firebase, see Figure 3(b).
   a) Tapping on the trash icon of a row item should delete the selected city from the favorite cities on Firebase and should refresh the displayed list to show the updated list of favorite cities.
   b) Tapping on a row item, should display the "Venues and Activities" for the selected city.
2. The "Places Visited" activity should display the list of places visited list retrieved from Firebase, see Figure 3(c).
   a) Tapping on the trash icon of a row item should delete the selected venue from the visited venues on Firebase and should refresh the displayed list to show the updated list of visited venues.
   b) Tapping on a row item, should display the display the Details activity which shows the details of the selected venue, see Figure 3(a).