ITIS/ITCS 4180/5180 Mobile Application Development
Homework 6
Date Posted: 03/23/2015 at 20:00
Due Date: 03/30/2015 at 23:55

Basic Instructions:

1. In every file submitted you MUST place the following comments:
    a. Assignment #.
    b. File Name.
    c. Full name of all students in your group.
2. Each group should submit only one assignment. Only the group leader is supposed to submit the assignment on behalf of all the other group members.
3. 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.
4. Please download the support files provided with this assignment and use them when implementing your project.
5. Export your Android project as follows:
    a. From eclipse, choose *"Export…"* from the File menu.
    b. From the Export window, choose *General* then *File System*. Click *Next*.
    c. Make sure that your Android project for this assignment is selected. Make sure that all of its subfolders are also selected.
    d. Choose the location you want to save the exported project directory to. For example, your *Desktop* or *Documents* folder.
    e. When exporting make sure you select *Create directory structure for files*.
    f. Click Finish, and then go to the directory you exported the project to. Make sure the exported directory contains all necessary files, such as the .java and resource files.
6. Submission details:
    a. All the group members should submit the same zip file.
    b. The file name is very important and should follow the following format: **Group#_HW06.zip**
    c. You should submit the assignment through Moodle: Submit the zip file.
7. **Failure to follow the above instructions will result in point deductions.**

Homework 6 (100 Points)

In this assignment you will use google maps to continuously display your location, and track the location of requested users. The application is composed of a single activity.

Important App Requirements:

Points will be deducted if your application does not follow the below requirements:

1. The required Android Virtual Device (AVD) should have **minimum SDK version set to 14 and target SDK at least 19**. The app should display correctly on Nexus 5. Your assignment will not be graded if it does not meet these requirements, and you will not be granted any points on your submission.
2. All API calls, XML parsing and image downloading should be performed using Threads (or AsyncTask) and your code should not block the main thread.
3. **All parse.com communication should be performed using the background mechanisms provided by parse.com and should not block the main thread.**

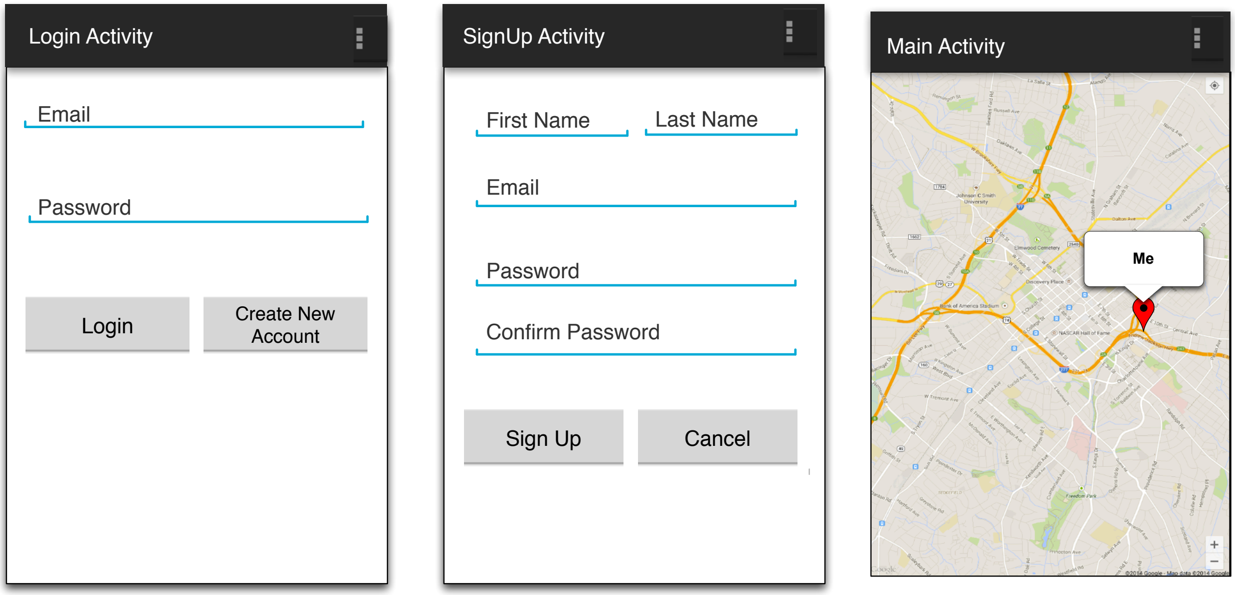


Figure 1: Login, SignUp, and Main Activity

Part 1 (10 Points): User Signup and Login

Your app should implement both login and signup functionalities. You should use parse.com to Store the user's full name, email address and password in the User class. The requirements are as follows:

1. 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 parse provided methods to check if there is a valid current user:
   a) If there is a current valid user, then start the Main 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.

2. Create a Login activity (Figure 1(a)):
   a) The user should provide their email and password.
      - Check the user input, if the email or password field is left empty, print a toast message indicating that these fields are required and don't submit the login information to parse.com.
   b) The provided credentials should be used to authenticate the user using parse.com. Clicking the “Login” button should submit the login information to parse.com to verify the user’s credentials.
      - If the user is successfully logged in then start the Main activity, and finish the Login activity.
      - If the user is not successfully logged in, then show a toast message indicating that the login was not successful.
   c) Clicking the “Create New Account” button should start the Signup activity and finish the Login activity.
3. Create a Signup activity (Figure 1(b)):
   a) Clicking the “Cancel” button should finish the Signup activity and start the Login activity.
   b) The user should provide their first name, last name, email and password. The provided credentials should be stored in the User class in parse.com. Clicking the “Sign Up” button should submit the user’s information to parse.com to verify the user’s credentials.
      - Check the user input, if any of the fields is left empty, or if the password and confirm password do not match, print a toast message indicating the corresponding error message and don’t submit the provided information to parse.com.
      - If an account with the same email already exists, display an error message indicating that the account account was not created and the user should select a different email.

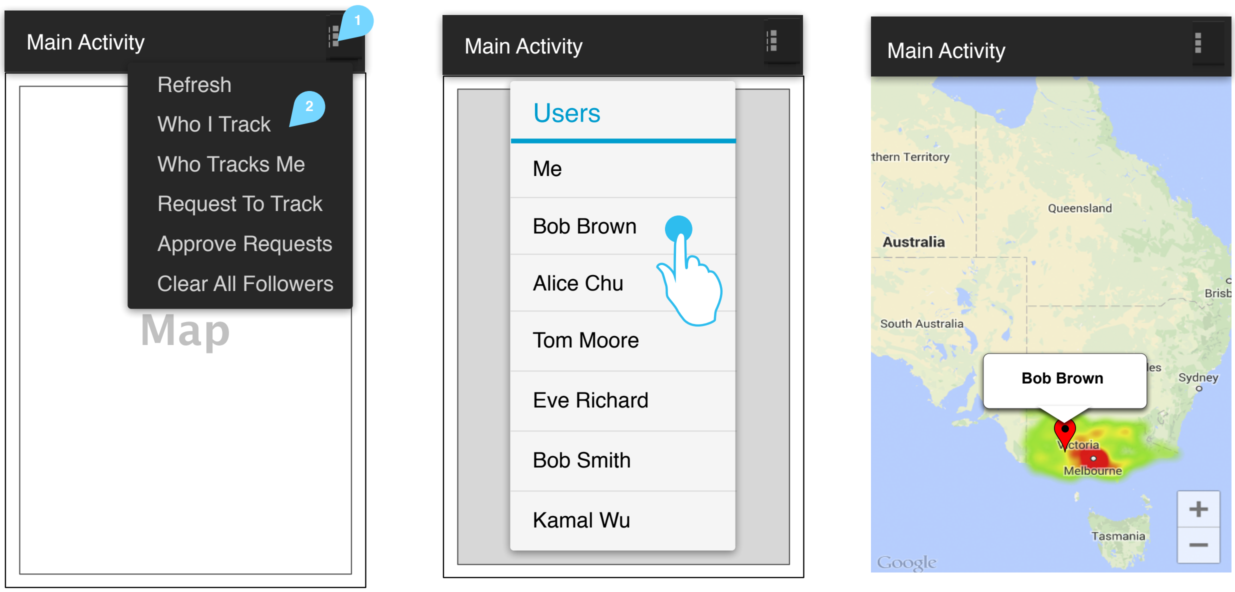
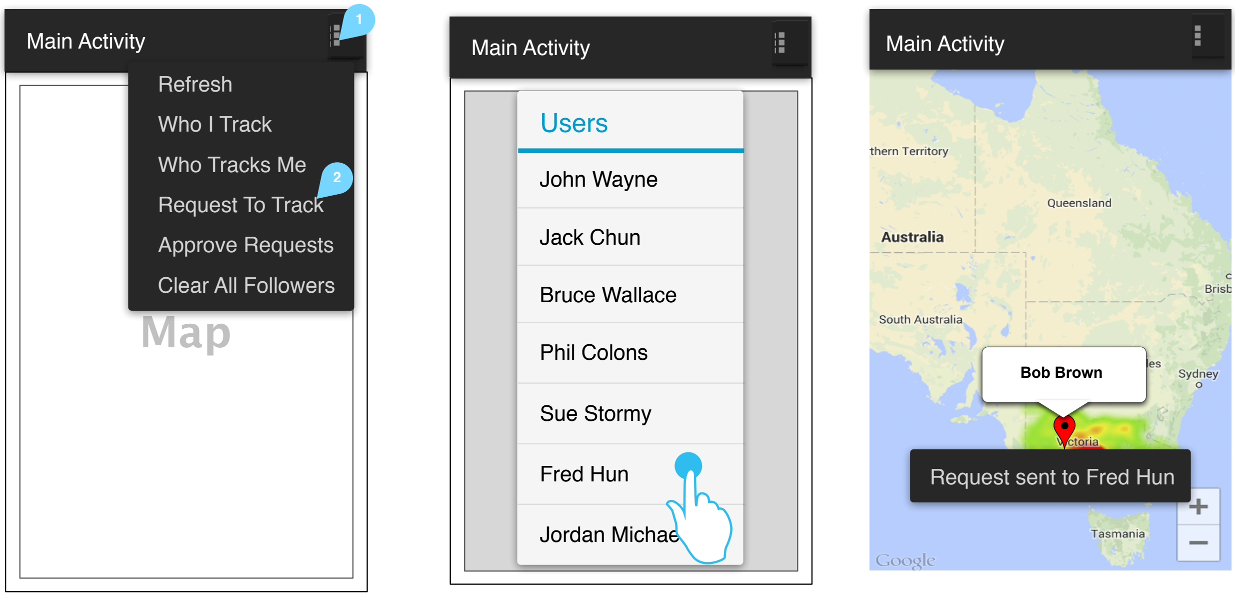


Figure 2: Main Activity, “Who I Track” is selected

- 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 successfully login. Then start the Main activity and finish the Signup activity.
- Note that, the username and email should set to the same value in the user's table.

Part 2 (70 Points): Main Activity

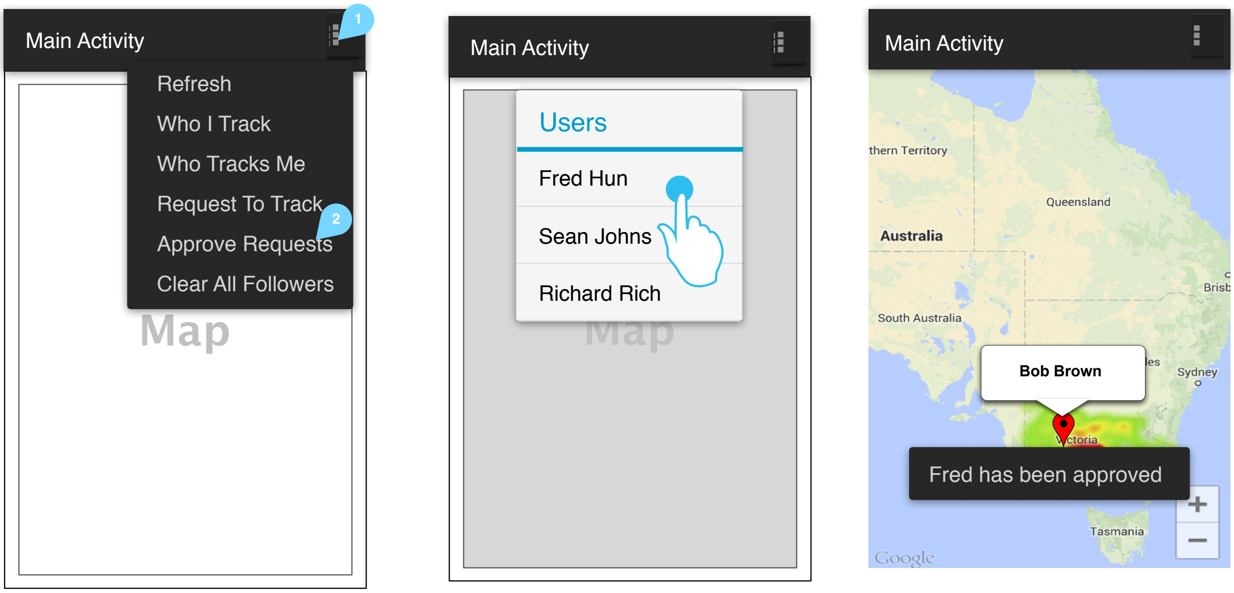
1. Create a locations app on parse.com. There should be two classes: one that maintains a list of all users and their various latitude and longitude (stored as a Parse GeoPoint object for each place), a class that reflects all requests to track location between owner, requester, and approval status.
2. When the Main Activity is opened, a Google Map Fragment should be displayed, and the activity should begin tracking your current location. Every 10 seconds, your name and location should be saved in the locations class in parse.com, and the map should be updated to show your current location with a marker, see Figure 1(c).
3. Clicking on the "Refresh" menu item should reload the map fragment currently being displayed.
4. Clicking on the "Who I Track" menu item should display a custom AlertDialog showing an EditText as shown in Figure 2(b). This will display a list of users retrieved from Parse who I have been enabled to track their location.
   a) When a user is selected from the available list, the map is updated to show a heat map of their reflecting their locations. A single marker should be placed where the most recent location entry is. Clicking on this marker should display an InfoWindow with the user's name. See Figure 2(c).
   b) You can read more about Info Window/Customized on this link: https://developers.google.com/maps/documentation/android/infowindows#showhide_an_info_window



Request sent to Fred Hun

Figure 3: Main Activity, Requesting to track a user

5. Clicking on the “Who Tracks Me” menu item should display a custom AlertDialog showing an EditText. This will simply display a list of users retrieved from Parse who I have approved to track my location.
6. Clicking on the “Request To Track” menu item should display a custom AlertDialog with a list of available contacts. See Figure 3.
   a) When a user is selected from the available contacts, the user is sent a request to track, which must be approved by them before they appear under “Who I Track.”
   b) A Toast message should be displayed indicated that a request was sent.
7. Clicking on the “Approve Requests” menu item should display a custom AlertDialog with a list of users who have requested access to track your location. See Figure 4.
   a) When a user is selected from this list, the user should be updated as an approved request in parse.com, and a Toast message should appear indicating approval.
8. Clicking on the “Clear All Followers” menu item should display a Toast message signaling that all users approved to track you should be removed from parse.com.



Fred has been approved

Figure 4: Main Activity, Approving request to track

Part 3 (15 Points): Demo Video & APK file

1. The demo video should be less than five (5) minutes long, and must include footage that clearly explains all the app features and functionalities through a comprehensive step wise demonstration.
   (a) The video must be uploaded to YouTube or Vimeo, and a link to the video must be provided.
   (b) Lastly, it must not include third-party trademarks, copyrighted music or other material unless the you have acquired permission to use such material.

Part 4 (5 Points): APK file

2. Submit the Android application package (APK) installable file for your app. Submit the credentials or special input data that is required for your app to login if needed.