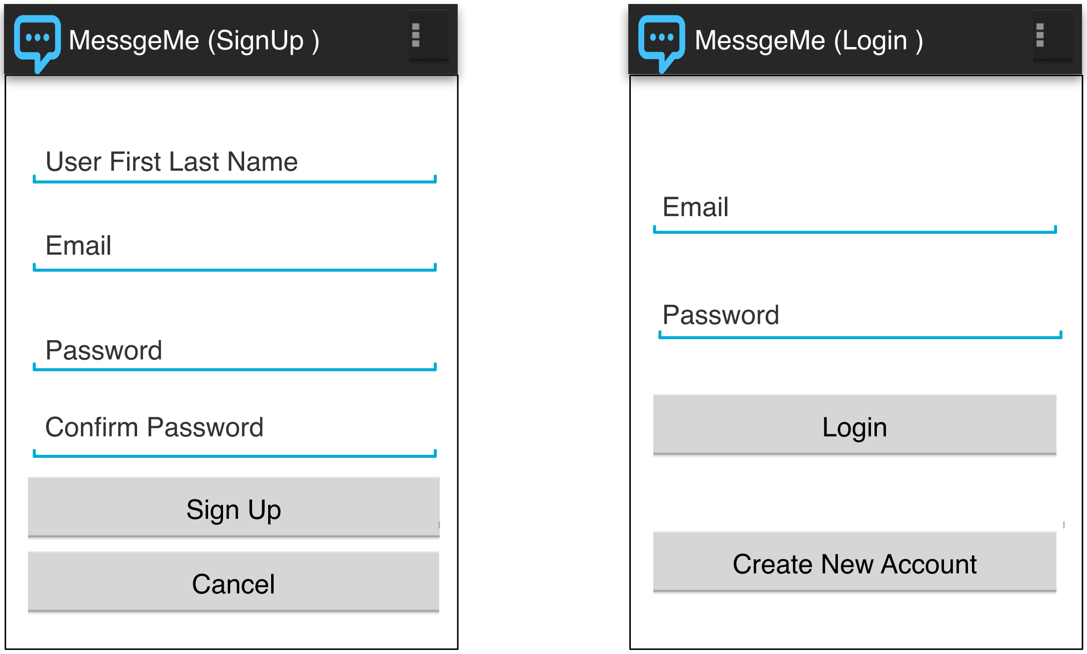
ITIS/ITCS 4180/5180 Mobile Application Development
In Class Assignment 8

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 and create a zip file which includes all the project folder and any required libraries.
6. Submission details:
   a. Only a single group member is required to submit on moodle for each group.
   b. The file name is very important and should follow the following format: **Group#_InClass08.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.**

In Class Assignment 8 (100 Points)

In this assignment you will implement an app to enable users to share public messages with other users. You will use parse.com to store and retrieve the messages. All parse.com communication should be performed using the background mechanisms provided by parse.com and should not block the main thread.



(a) Login Activity

(b) SignUp Activity

Figure 1, Wireframe for Login and SignUp Activities

Part A: User Signup and Login (30 Points)

Your app should implement both login and signup functions. You should use parse.com to store the user first name, last name, email address and password in the Parse 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 Messages 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 Messages 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.
   - 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 Messages activity and finish the Signup activity.
   - Note that, the username and email should set to the same value in the user's table.

Part B: Messages Activity (50 Points)

The Messages activity should retrieve all the messages that are stored in the Message class on parse.com. A ListView should display the items in the messages list, see Figure 2(a). The requirements are as follows:

1. In parse.com create a new Messages parse class to store the messages, these are public messages that should be viewable by any logged in user.
2. The messages list items should be retrieved in the background using the Parse provided background query mechanism. The query should retrieve the Messages list items belonging to the currently logged in user. Check the documentation provided at https://parse.com/docs/android/guide#queries

   a) Clicking the "Refresh" should execute the query again and refresh the list, to show new messages received.

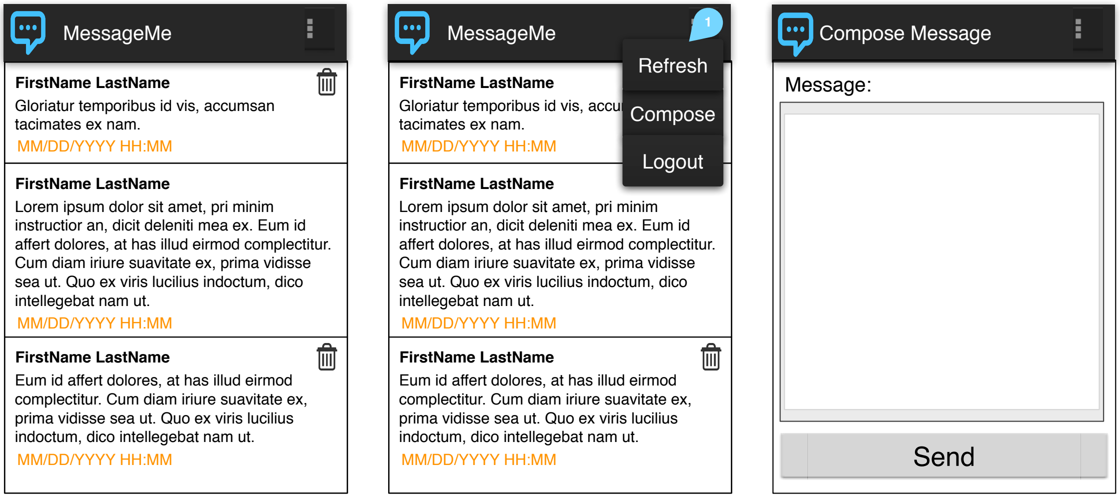
3. The delete icon should be displayed beside messages that were created by the currently logged in user. Tapping the delete icon in the list should do the following:

   - Delete the current message from the Messages table. Check the documentation at https://parse.com/docs/android/guide#objects-deleting-objects

4. Clicking the “Compose” action bar menu item should show start new activity **Compose Activity** ,as shown in Figure 2(b) to enable the currently logged-in user to send messages.
5. Clicking the “Logout” action bar menu item should logout the current user, start the Login Activity and finish the Messages Activity.

Part C: Compose Activity (30 Points)

1. This activity enables the user to compose a new public message. In Parse, you should create the Messages table (class) to hold the message text and a pointer to the user which created the message. You should consider using pointers to connect the user which created the message to the message, check https://parse.com/docs/android/guide#relations-using-pointers
2. In the Compose Activity, the message text should have a maximum of 120 characters. User should be able to enter the message body. You should validate the user’s input and ensure that the text is provided. If no input is provided, display a Toast to indicate the a missing description.
3. Upon tapping the send button, save the new Message item in the Messages class on parse.com using Parse’s background storing mechanism, and display a toast message indicating the successful addition of the item. Then navigate back to the Messages activity.



FirstName LastName
Gloriatur temporibus id vis, accumsan tacimates ex nam.
MM/DD/YYYY HH:MM

FirstName LastName
Lorem ipsum dolor sit amet, pri minim instructior an, dicit deleniti mea ex. Eum id affert dolores, at has illud eirmod complectitur. Cum diam iriure suavitate ex, prima vidisse sea ut. Quo ex viris lucilius indoctum, dico intellegebat nam ut.
MM/DD/YYYY HH:MM

FirstName LastName
Eum id affert dolores, at has illud eirmod complectitur. Cum diam iriure suavitate ex, prima vidisse sea ut. Quo ex viris lucilius indoctum, dico intellegebat nam ut.
MM/DD/YYYY HH:MM

(a) Messages Activity (b) Menu Items (c) Compose Activity

Figure 2, Wireframe for Messages and Compose Activity