

ITIS/ITCS 4180/5180 Mobile Application Development  
In Class Assignment 9

**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. Please download the support files provided with this assignment and use them when implementing your project.
4. Export your 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 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.
5. Submission details:
  - a. When you submit the assignment, compress your exported project into a single zip file. The format of compressed file name is InClassAssignment#.zip
  - b. You should submit the assignment through Moodle: Submit the zip file.
- 6. Failure to follow the above instructions will result in point deductions.**

## **In Class Assignment 9 (100 Points)**

In this assignment you will implement an app to add and display upcoming events. You will use [parse.com](https://parse.com) to store and retrieve user events

### **Important App 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 3.2" QVGA (ADP2) (320x480: mdpi). **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. For this assignment you will be provided with a project folder which includes the activity initial implementations and UI. Import the provided project and add your implementation.
3. **All [parse.com](https://parse.com) communication should be performed using the background mechanisms provided by [parse.com](https://parse.com) and should not block the main thread.**

### **Part A: User Signup and Login (25 Points)**

Your app should implement both login and signup functionalities. You should use [parse.com](https://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 Events 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. The provided credentials should be used to authenticate the user using [parse.com](https://parse.com). Clicking the "Login" button should submit the login information to [parse.com](https://parse.com) to verify the user's credentials.
    - If the user is successfully logged in then start the Events 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.
  - b) 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 name, email and password. The provided credentials should be stored in the User class in [parse.com](https://parse.com). Clicking the "Sign Up" button should submit the user's information to [parse.com](https://parse.com) to verify the user's credentials.

- 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 Events activity and finish the Signup activity.
- Note that, the username and email should set to the same value in the user's table.

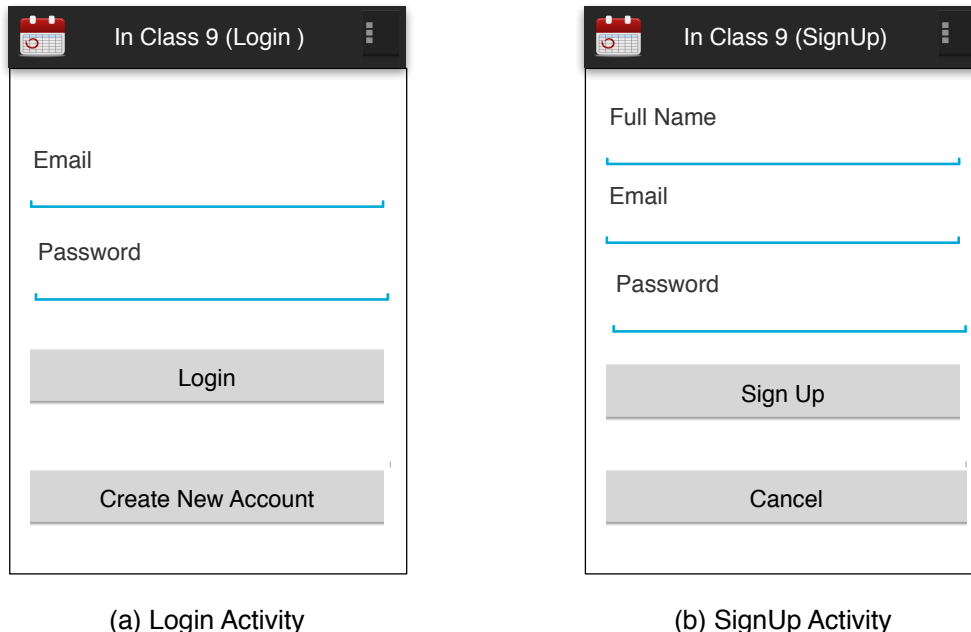


Figure 1, Wireframe for Login and SignUp Activities

### **Part B: Events Activity (25 Points)**

The Events activity should retrieve the events for the currently logged in user that are stored in the Events class on [parse.com](https://parse.com). A ListView should display the event names of the retrieved events, see Figure 2. The requirements are as follows:

1. In [parse.com](https://parse.com) create a new Event parse class to store the event data, which includes the event name, description, location, event owner, and event date.
2. The Events should be retrieved in the background using the Parse provided background query mechanism. The query should retrieve the events belonging to the currently logged in user. Check the documentation provided at [https://parse.com/docs/android\\_guide#queries](https://parse.com/docs/android_guide#queries)
  - a) You can consider using the ParseQueryAdapter class. Check the documentation at [https://parse.com/docs/android\\_guide#ui-queryadapter](https://parse.com/docs/android_guide#ui-queryadapter)
3. Clicking a event in the ListView should start the Event Detail activity.
4. Clicking the “Add Event” image, which is above the ListView, should start the Add Event activity. As shown in Figure 2(b).

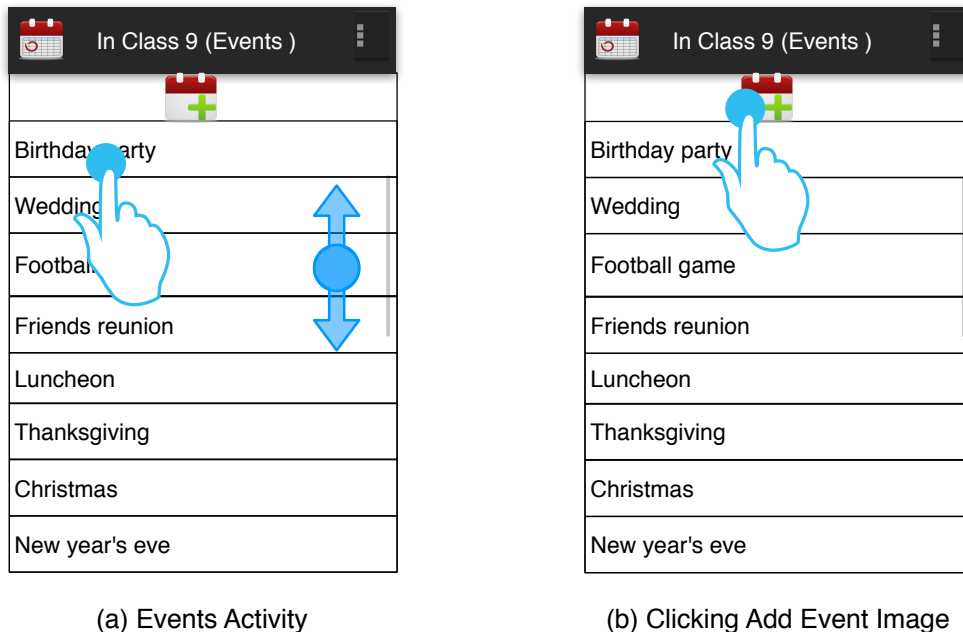


Figure 2, Wireframe for Events Activities

### Part C: Add Event Activity (25 Points)

The Add Event activity should enable the currently logged-in user to create a new event. The user should enter the event name, description, location and date. The wireframe for this activity is shown in Figure 3. The requirements are as follows:

1. Clicking the “Add Event” button:
  - a) Should validate the user’s input and ensure that all the fields are provided.
    - If any field is missing, display a Toast to indicate the a missing field.
    - If all the fields are provided, then save the new event in the Event class on [parse.com](https://parse.com) using Parse’s background storing mechanism. Then the Add Event activity should be finished. Upon returning to the Events activity the list should be updated to display the newly added event.

**Add An Event**

Event Name

Description

Location

Date

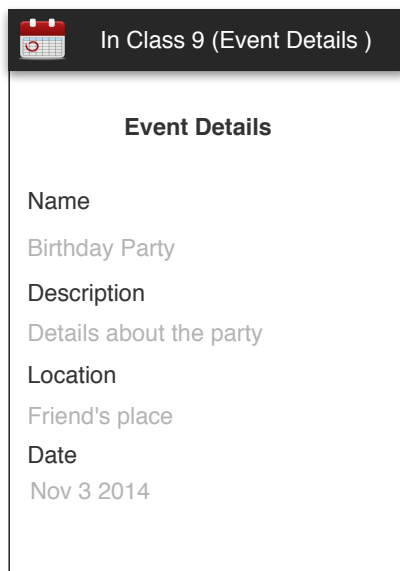
Add Event

Figure 3, Add Event Activity

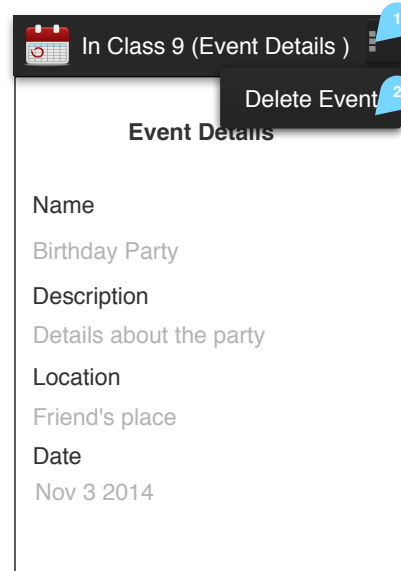
### Part D: Event Details Activity (25 Points)

The Event Details activity shows the event details, which include the event's name, description, location and date, see Figure 4. The requirements are as follows:

1. Clicking the back button should finish the Event Details activity and go back to the Events activity.
2. Clicking the "Delete Event" action bar menu item should delete the currently displayed event, as shown in Figure 4(b).
  - a) If the event is successfully deleted then display a toast message indicating the successful deletion of the event, and finish the event detail activity.
  - b) If the deletion is not successful then display a toast message indicating that there was an error while deleting the event.



(a) Event Details Activity



(b) Deleting Event

Figure 4, Wireframe for Event Details Activity