ITIS/ITCS 4180/5180 Mobile Application Development
In Class Assignment 12

Basic Instructions:

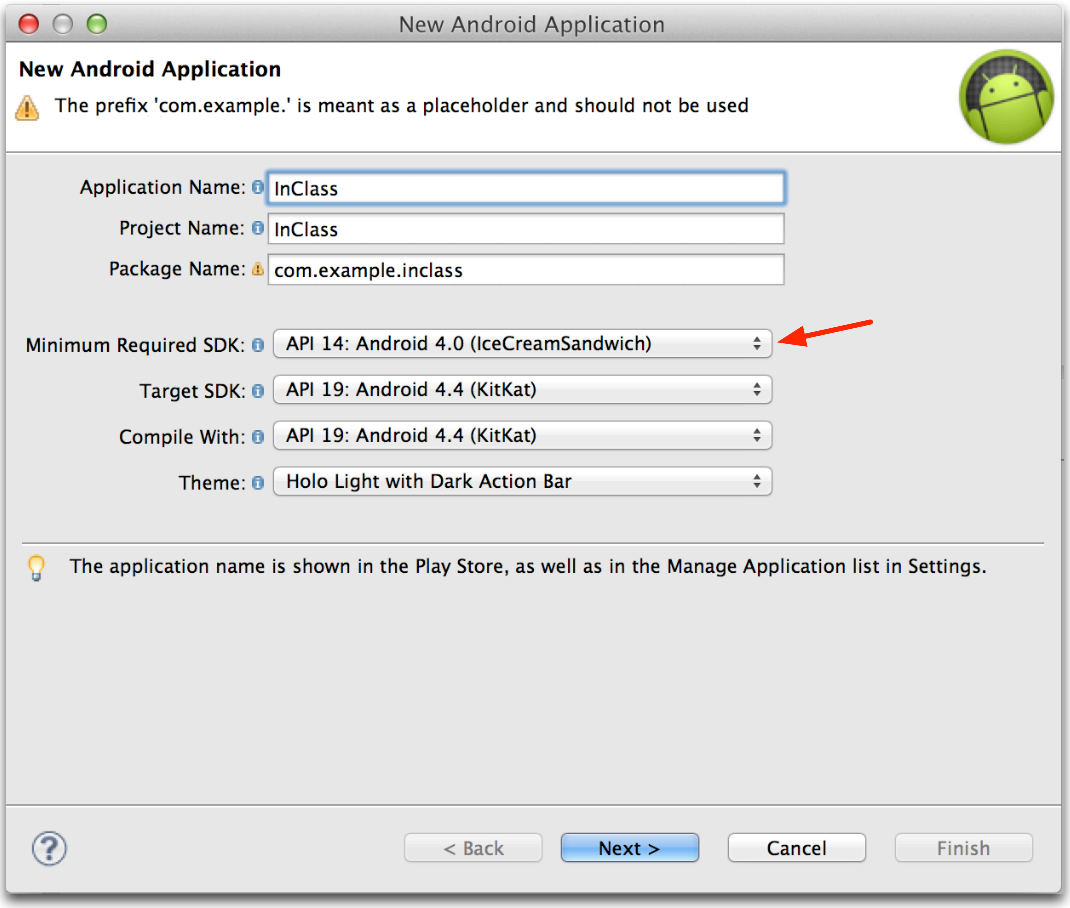
1. In every file submitted you MUST place the following comments:
    a. Assignment #.
    b. File Name.
    c. Full name of all students in your group.
2. Each student in the group is required to submit the assignment on moodle.
3. Please download the support files provided with this assignment and 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. Only one group member is required to submit the project zip file.
    b. The file name is very important and should follow the following format: **Group#_InClass12.zip** For example, Group2A_InClass12.zip
    c. 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 12 (100 Points)

This assignment is an extension to the In Class 08 assignment. In this assignment you will use the new RecyclerView. In this assignment you will implement an app to add, update, delete and retrieve items from a ToDo list. You will use parse.com to store and retrieve the ToDo list.

Important App Requirements:

1. Create a new android project called “In Class 12”.
2. 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.
3. You will be using layout files, and strings.xml to create the required user interfaces. The layout XML file can be modified through the raw xml, or through the GUI tools provided within eclipse.
4. All parse.com communication should be performed using the background mechanisms provided by parse.com and should not block the main thread.



New Android Application

New Android Application

⚠ The prefix 'com.example.' is meant as a placeholder and should not be used

Application Name: InClass

Project Name: InClass

Package Name: com.example.inclass

Minimum Required SDK: API 14: Android 4.0 (IceCreamSandwich)

Target SDK: API 19: Android 4.4 (KitKat)

Compile With: API 19: Android 4.4 (KitKat)

Theme: Holo Light with Dark Action Bar

The application name is shown in the Play Store, as well as in the Manage Application list in Settings.

Figure 1. Choosing Minimum Required SDK to 14

Part A: User Signup and Login (30 Points)

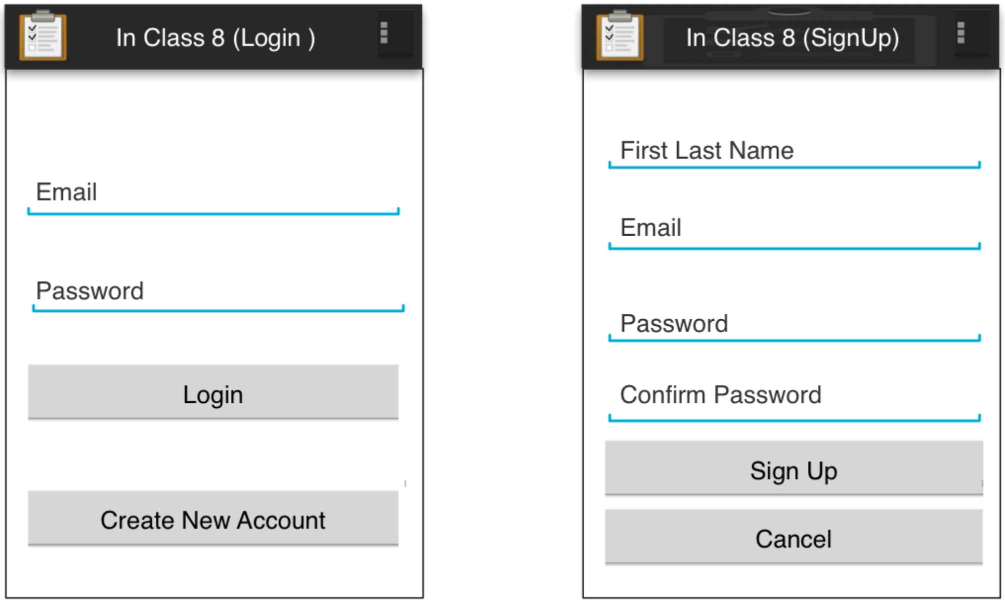
Your app should implement both login and signup functionalities. You should use parse.com to store the user name, email address and password in the User class. The requirements are as follows:

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 ToDo 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 2(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 ToDo 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 2(b)):
   a) Clicking the “Cancel” button should finish the Signup activity and start the Login activity.
   b) The user should provide their user 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 ToDo activity and finish the Signup activity.

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



(a) Login Activity

(b) SignUp Activity

Figure 2, Wireframe for Login and SignUp Activities

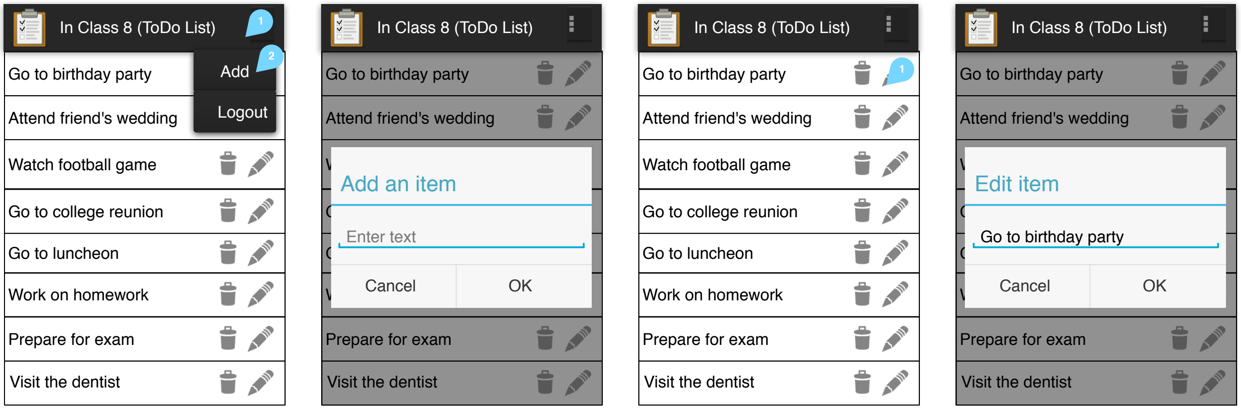
Part B: ToDo Activity (70 Points)

The ToDo activity should retrieve the items in the ToDo list for the currently logged in user that are stored in the ToDo class on parse.com. A ListView should display the items in the ToDo list, see Figure 3a. The requirements are as follows:

1. In parse.com create a new ToDo parse class to store the ToDo list item text.
2. The ToDo list items should be retrieved in the background using the Parse provided background query mechanism. The query should retrieve the ToDo list items belonging to the currently logged in user. Check the documentation provided at https://parse.com/docs/android_guide#queries
3. You are required to use the RecycleView to implement the ToDo list. https://developer.android.com/training/material/lists-cards.html#RecyclerView
4. Clicking the "Add" action bar menu item should show an alertDialog on the ToDo Activity as shown in Figure 3b to enable the currently logged-in user to enter the item description of a new item. To create a custom layout for AlertDialog, check the documentation at http://developer.android.com/guide/topics/ui/dialogs.html. The requirements are as follows:
   a) 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.
      - If the input is provided, then save the new ToDo item in the ToDo class on parse.com using Parse's background storing mechanism, and display a toast message indicating the successful addition of the item

- Upon closing the alertDialog, the list should be updated to display the newly added item.

4. For each item in the list there is a delete and edit icon.
   a) Clicking the delete icon should delete ToDo list item from parse.com and should refresh the list to reflect this change.
   b) Clicking the edit icon should present an alert dialog as shown in Figure 3d. The alert dialog's textview should be populated with todo item's description. Once the user clicks OK:
      - If no input is provided, display a Toast to indicate the a missing description.
      - If the input is provided, then update the ToDo item in the ToDo class on parse.com using Parse's background storing mechanism, and display a toast message indicating the successful update.
      - Upon closing the alertDialog, the list should be updated to display the newly updated item.
5. Clicking the "Logout" action bar menu item should logout the current user, start the login activity and finish the ToDo activity.



(a) To do list (b) Adding item (c) Editing item (d) Editing item alert

Figure 3, Wireframe for ToDo List Activity