

Mobile Application Development
Assignment 05 (100 Points)

Basic Instructions:

1. In every file submitted you MUST place the following comments:
 - a. Assignment #.
 - b. File Name.
 - c. Full name of the student.
2. Each group is required to submit the assignment on Canvas.
3. **Submit Codes:**
 - a. Zip all the project folder to be submitted on canvas.
4. Submission details:
 - a. The file name is very important and should follow the following format:
Assignment#.zip
 - b. You should submit the assignment through Canvas: Submit the zip file.
5. **Failure to follow the above instructions will result in point deductions.**

In this assignment you will be building an application that uses ListViews to manage users. This app is composed of one activity (Main Activity) and multiple fragments. The app requirements are as follows:

1. You are provided with a skeleton app that includes all the required UI and classes needed to complete this assignment.
2. Note that, the ArrayList of User objects should be stored in the Main Activity to maintain the list of users added.
3. All communication between fragments should be performed through the Main Activity.
4. When the Main Activity starts it should load the “Users” Fragment as the initial fragment.



Figure 1, App Wireframe

Part 1, Users Fragment (30 Points):

The users fragment displays the ListView of users as shown in Figure 1(a). Please follow the steps:

1. This fragment should receive the ArrayList of users from the Main Activity, and should display the list of users as shown in Figure 1(a).
 - a. You should create and use an interface to request the list of users from the Main Activity.
 - b. Use a ListView and you should extend the ArrayAdapter and should be implemented inside the Users Fragment.
 - c. Each row item should display the name, email, age, state, gender and group.
2. Clicking the “Clear All” button should communicate with the Main Activity to:
 - a. Remove all the users from the users ArrayList.
 - b. Reload the ListView to display no users as the users ArrayList is empty.
3. Clicking the “Add New” button should communicate with the Main Activity to:
 - a. Replace the current fragment with the Add User fragment.
 - b. Push the current fragment on the back stack.
 - c. Upon returning from the Add User fragment this fragment should request the updated ArrayList of users and refresh the ListView to show the list of users which contains the newly added user as shown in Figure 2(c).

4. Clicking the “Sort” button should communicate with the Main Activity to:
 - a. Replace the current fragment with the Sort Selection fragment.
 - b. Push the current fragment on the back stack.
 - c. Upon returning from the Sort Selection fragment this fragment should receive the selected sort criteria and should refresh the ListView to display the list of users based on the selected sort criteria.
5. Clicking on the row item should communicate with the Main Activity to:
 - a. Replace the current fragment with the User Detail fragment, send the selected User object to the User Detail fragment.
 - b. Push the current fragment on the back stack.

Part 2, Add User Fragment (10 Points):

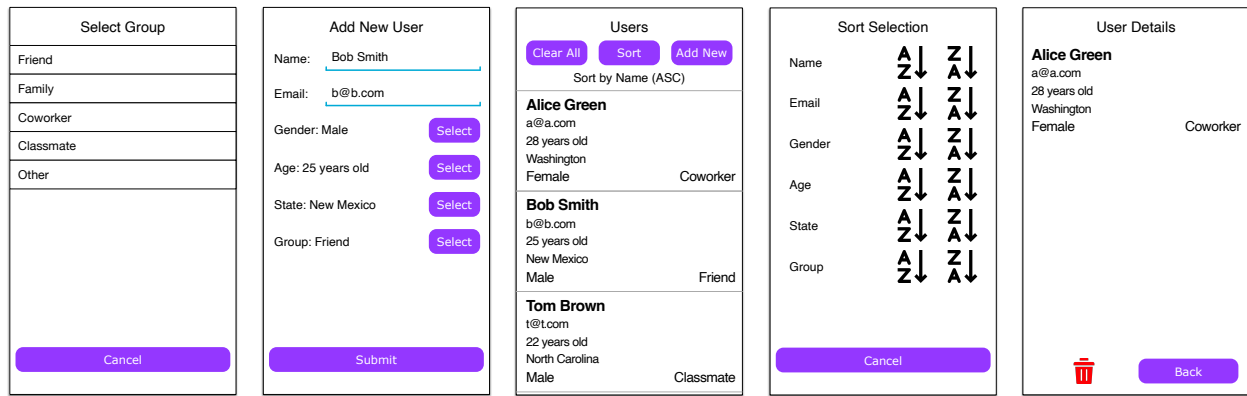
This fragment is used to add a new user. The requirements are listed below:

1. There are several “Select” buttons that should be used by the user to transition to a fragment to allow the selection of a specific attribute. For example there is select gender, age, state and group.
2. Clicking on the “Select” should communicate with the Main Activity to:
 - c. Replace the current fragment with the corresponding Select fragment.
 - d. Push the current fragment on the back stack.
 - e. Upon returning from the Select fragment the selected value should be received through the Main Activity, sent to the Add User fragment and should be displayed see Fig 2(b).
2. Clicking on the Submit button should:
 - a. If any of the entries is not entered or selected then show a toast message indicating the missing input.
 - b. If all the required data is entered then create a User object which contains all the entered and selected values. Send the created User object to the Main Activity which should:
 - i. Add the new user to the users ArrayList hosted in the Main Activity.
 - ii. Pop the back stack which should go back to the Users fragment.

Part 3, Select Gender Fragment (10 Points):

This fragment allows the user to select a gender value as shown in Fig 1(c). The requirements are listed below:

1. List the different gender selections in a ListView as shown in Figure 1(c). The gender selections can be retrieved using the provide Data class.
2. You should use the simple ArrayAdapter as you are only displaying a ListView of string values.
3. Clicking a list item should communicate with the Main Activity to:
 - a. Find the “Add User” fragment by tag and send it the selected gender value.
 - b. Pop the back stack which should display the Add User fragment and display the selected gender value as shown in Figure 2(d).
4. Clicking “Cancel” should simply communicate with the Main Activity to:
 - a. Pop the back stack which should go back to the Add User fragment.



(a) Select Group (b) Add User Screen (c) Users Screen (d) Sort Selection (e) User Detail Screen

Figure 2, App Wireframe

Part 4, Other Select Fragments (30 Points):

These fragments are the select age, state, and group fragments shown in Figures 1(d), 1(e) and 2(a). The requirements are listed below:

1. List the different item (age, state, and group) selections in a ListView. The state and group selections can be retrieved using the provide Data class. The age selections should be created by looping from age 18 to age 100.
2. You should use the simple ArrayAdapter as you are only displaying a ListView of string values.
3. Clicking a list item should communicate with the Main Activity to:
 - a. Find the “Add User” fragment by tag and send it the selected item value (age, state, and group).
 - b. Pop the back stack which should display the Add User fragment and display the selected item value (age, state, and group) as shown in Figure 2(d).
4. Clicking “Cancel” should simply communicate with the Main Activity to:
 - a. Pop the back stack which should go back to the Add User fragment.

Part 5, User Detail Fragment (10 Points):

This fragment displays the selected user details as shown in Figure 2(e). The requirements are listed below:

1. This fragment should receive a User object from the Users fragment through the Main Activity.
2. Display the name, email, gender, age, state, and group as shown in Figure 2(e).
3. Clicking “Delete” should simply communicate with the Main Activity to:
 - a. Send the current User object to the Main Activity, and delete that User object from users ArrayList that is stored in the Main Activity.
 - b. Pop the back stack which should go back to the Users fragment which should show the new list of users not containing the deleted user.
4. Clicking “Back” should simply communicate with the Main Activity to:
 - a. Pop the back stack which should go back to the Users fragment.

Part 6, Sort Fragment (10 Points):

This fragment displays the sort features as shown in Figure 2(d). The requirements are listed below:

1. This fragment allows the user to select the sorting criteria to be used to sort the list presented in the Users Fragment. The sort criteria includes sort ascending/descending based on name, email, gender, age, state, and group.
2. Clicking on a sort criteria should communicate with the Main Activity to:
 - a. Find the “Users” fragment by tag and send it the selected sort criteria value.
 - b. Pop the back stack which should display the Users fragment and refresh the ListView of Users sorted based on the selected criteria.
 - c. The selected sort criteria should also be indicated in the Users Fragment as shown in Figure 2(c).
3. Clicking “Cancel” should simply communicate with the Main Activity to:
 - a. Pop the back stack which should display the Users fragment.