

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 UITableViews to manage users. 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, an Array of User objects should be stored in the Users UIViewController to maintain the list of users added.

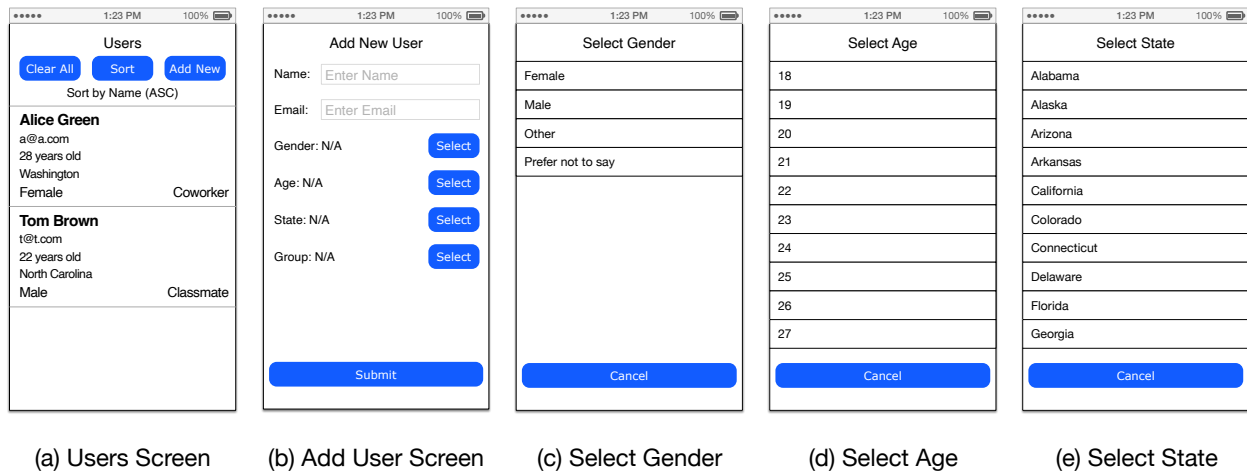


Figure 1, App Wireframe

Part 1, Users UIViewController (30 Points):

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

1. This UIViewController should store and maintain a mutable array of User objects to be used to enable the addition and deletion of users.
2. The Array of Users should be displayed in a UITableView as shown in Figure 1(a).
 - a. Each row item should display the name, email, age, state, gender and group.
 - b. Implement the Delegate and DataSource required by the UITableView.
3. Clicking the “Clear All” button should:
 - a. Remove all the users from the users array.
 - b. Refresh the UITableView to display no users as the users array is empty.
4. Clicking the “Add New” button should segue to the Add User UIViewController, which should be presented modally.
 - a. Use Unwind Segue or Notification Center or Protocol/Delegate to communicate the newly created user back from the Add User UIViewController.
 - b. Upon receiving the new User object, add the User object to the users array, and refresh the UITableView to display the updated list of users.
5. Clicking the “Sort” button should segue to the Sort Selection UIViewController, which should be presented modally.
 - a. Use Unwind Segue or Notification Center or Protocol/Delegate to communicate the selected sort criteria back from the Sort Selection UIViewController.
 - b. Upon receiving the new sort criteria, the users array should be sorted based on the selected criteria, and the UITableView should be refreshed to display the updated sorted list. In addition, update the UILabel above the UITableView to indicate the selected sort criteria.

6. Clicking on the row item should:
 - a. Segue to the User Detail UIViewController which should be presented modally.
 - b. Send the selected User object to the User Detail UIViewController.

Part 2, Add User UIViewController (10 Points):

This UIViewController 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 segue to a UIViewController to allow the selection of a specific attribute. For example there is select gender, age, state and group.
2. Clicking a “Select” button should segue to the corresponding Select UIViewController:
 - a. Use Unwind Segue or Notification Center or Protocol/Delegate to communicate the selected value back from the corresponding Select UIViewController.
 - b. Upon receiving the selected value, the value selected should be displayed as shown in Figure 2(b).
2. Clicking the Submit button should:
 - a. If any of the entries is not entered or selected then show an Alert Dialog message indicating the missing input.
 - b. If all the required data is entered and selected then create a User object which contains all the entered and selected values. Send the created User object back to the Users UIViewController using any of the approaches such as Unwind Segue or Notification Center or Protocol/Delegate. If needed dismiss this UIViewController.

Part 3, Select Gender UIViewController (10 Points):

This UIViewController 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 UITableView as shown in Figure 1(c). The gender selections can be retrieved using the provide Data class.
2. Clicking a list item should:
 - a. Send the selected item back to the Add User UIViewController. Use Unwind Segue or Notification Center or Protocol/Delegate to communicate the selected value back.
 - b. Dismiss this UIViewController if needed.
 - c. Display the selected gender value in the Add User UIViewController as shown in Figure 2(d).
3. Clicking “Cancel” should simply dismiss this UIViewController.

Part 4, Other Select UIViewControllers (30 Points):

These UIViewControllers are the select age, state, and group UIViewControllers 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 UITableView. 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. Clicking a list item should:

- a. Send the selected item back to the Add User UIViewController. Use Unwind Segue or Notification Center or Protocol/Delegate to communicate the selected value back.
 - b. Dismiss this UIViewController if needed.
 - c. Display the selected value (age, state, and group) in the Add User UIViewController as shown in Figure 2(d).
3. Clicking “Cancel” should simply dismiss this UIViewController.

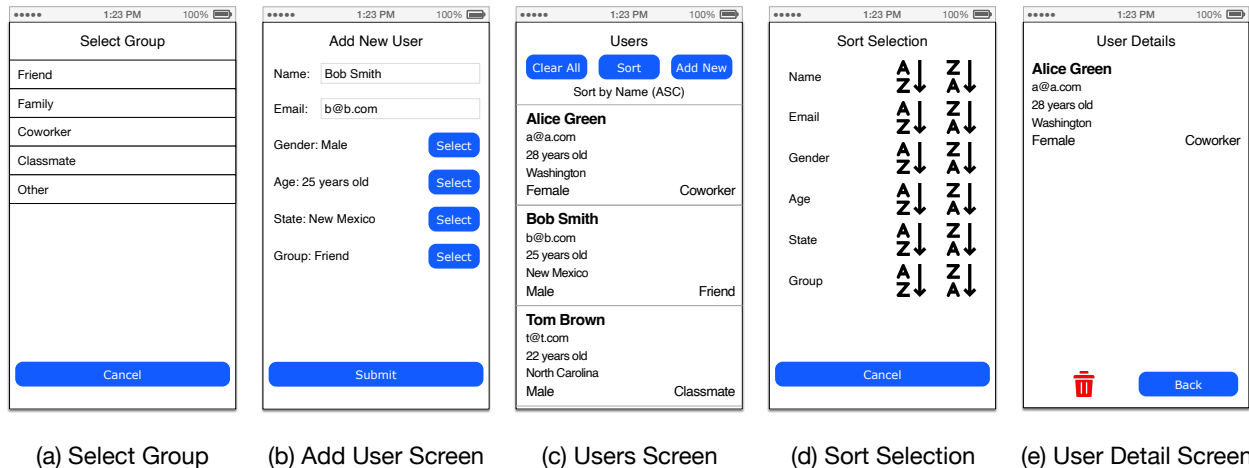


Figure 2, App Wireframe

Part 5, User Detail UIViewController (10 Points):

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

1. This UIViewController should receive a User object from the Users UIViewController
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 User object back to the Users UIViewController using any of the approaches such as Unwind Segue or Notification Center or Protocol/Delegate. If needed dismiss this UIViewController.
 - b. The Users UIViewController should delete the received User object from the users array, and should reload the UITableView to display the updated array.
4. Clicking “Back” should simply dismiss this UIViewController.

Part 6, Sort UIViewController (10 Points):

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

1. This UIViewController allows the user to select the sorting criteria to be used to sort the list presented in the Users UIViewController. The sort criteria includes sort ascending/descending based on name, email, gender, age, state, and group.
2. Clicking on a sort criteria should:
 - a. Send the selected sort criteria back to the Users UIViewController. Use Unwind Segue or Notification Center or Protocol/Delegate to communicate the selected sort criteria back.
 - b. The Users UIViewController should sort the users array based on the selected sort criteria, and should reload the UITableView to display the updated users array.
 - c. The selected sort criteria should also be indicated in the Users UIViewController as shown in Figure 2(c).
3. Clicking "Cancel" should simply dismiss this UIViewController.