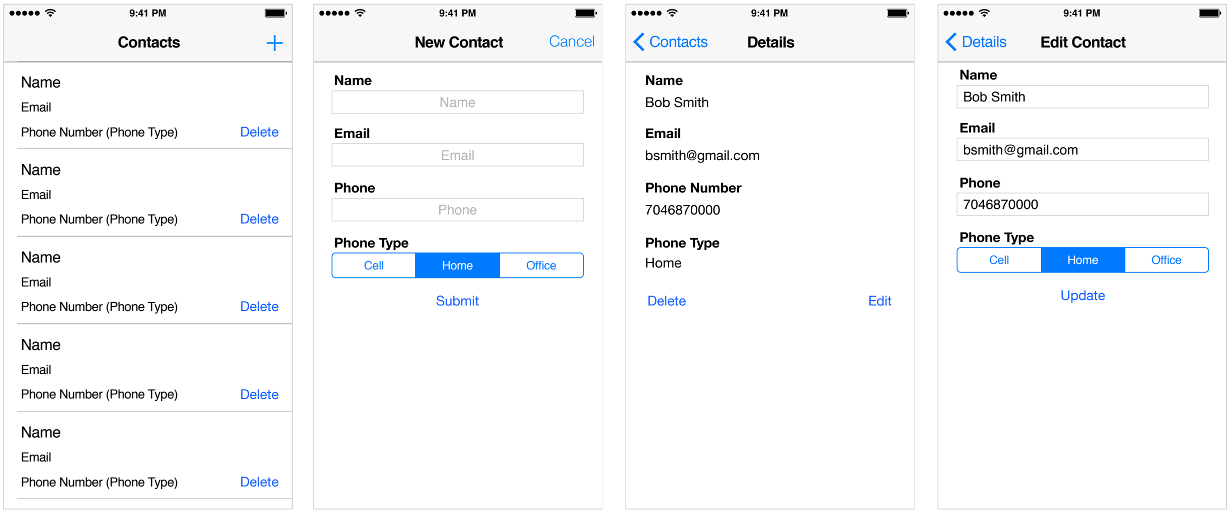
ITIS/CS 4180 – Mobile App Development
Homework 02

Basic Instructions:

1. In every file submitted you MUST place the following comments:
    a. Assignment #.
    b. File Name.
    c. Student Full Name.
2. This is an individual assignment, each student is expected to work alone and submit their own work.
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. Create a zip file which includes all the project folder, any required libraries, and your presentation material.
6. Submission details:
    a. You should submit the assignment through canvas: Submit the zip file.
7. **Failure to follow the above instructions will result in point deductions.**

Homework 02 (100 Points)

In this assignment you will get familiar with iOS TableViews. You will develop a contacts application which enable you to list, create, edit and delete contacts.



(a) Contacts List (b) Contact Details (c) New Contact (d) Edit Contact

Figure 1, Application Wireframe

Part 1 : Contacts List View Controller (30 Points)

The interface should be created to match the UI presented in Figure 1(a). The requirements are as follows:

1. This ViewController should be embedded in a navigation controller to provide the transition to the Apps ViewController.
2. This ViewController should store and maintain an array of contacts to used in the application. The array should be mutable to enable the addition and deletion of contacts. You should create a Contact class to store the contact information.
3. The array of contacts should be displayed in a TableView.
4. Clicking on a row item should segue to the Details ViewController. The Contact List ViewController should pass the required information to the Details ViewController in order to display the contact details.
5. Clicking the “Delete” button should delete the selected contact and should refresh the TableView to show the updated list of contacts.
6. Clicking the “+” button should segue to the New Contact ViewController, you should use the “Present Modally” segue.

Part 2 : New Contact View Controller (20 Points)

The interface should be created to match the UI presented in Figure 1(b). The requirements are as follows:

1. This ViewController should display new contact form as shown in Figure 1(b).
2. This ViewController should be displayed using the “Present Modally” segue.

3. Clicking the “Cancel” button should dismiss this ViewController and return back to the Contacts List ViewController.
4. Clicking the “Submit” button, the app should check if all the fields are entered and display an alert dialog if any of the entries is missing indicating that the missing field is required.
    a. If all the fields are entered, this ViewController should send the Contacts List View Controller the new contact object and then dismiss the current ViewController.
    b. Upon returning to the Contacts List View Controller the TableView should be refreshed and should include the newly created contact.

Part 3 : Contact Details Controller (25 Points)

The interface should be created to match the UI presented in Figure 1(c). The requirements are as follows:

1. This ViewController should display the contact details as shown in Figure 1(c).
2. Clicking the “Delete” button should signal the Contacts List ViewController indicating that this contact should be deleted from the contacts array. Then the current Details View Controller should be dismissed to show the updated Contacts ViewController with this contact deleted.
3. Clicking the “Update” button should segue to the Edit Contact ViewController and should pass the contact object to the Edit Contact ViewController.

Part 4 : Edit Contact View Controller (25 Points)

The interface should be created to match the UI presented in Figure 1(d). The requirements are as follows:

1. This ViewController should display the edit form populated with the details of the contact that was passed by the Details ViewController.
2. Clicking the “Update” button, the app should check if all the fields are entered and display an alert dialog if any of the entries is missing indicating that the missing field is required.
    a. If all the fields are entered, this ViewController should send the Contacts List View Controller the new contact object and then dismiss the current ViewController.
    b. Upon returning to the Contact Details View Controller the contact details should show the updated contact information.
    c. Upon returning to the Contacts List View Controller the TableView should be refreshed and should include the updated contact.