

CS5520: Mobile Application Development

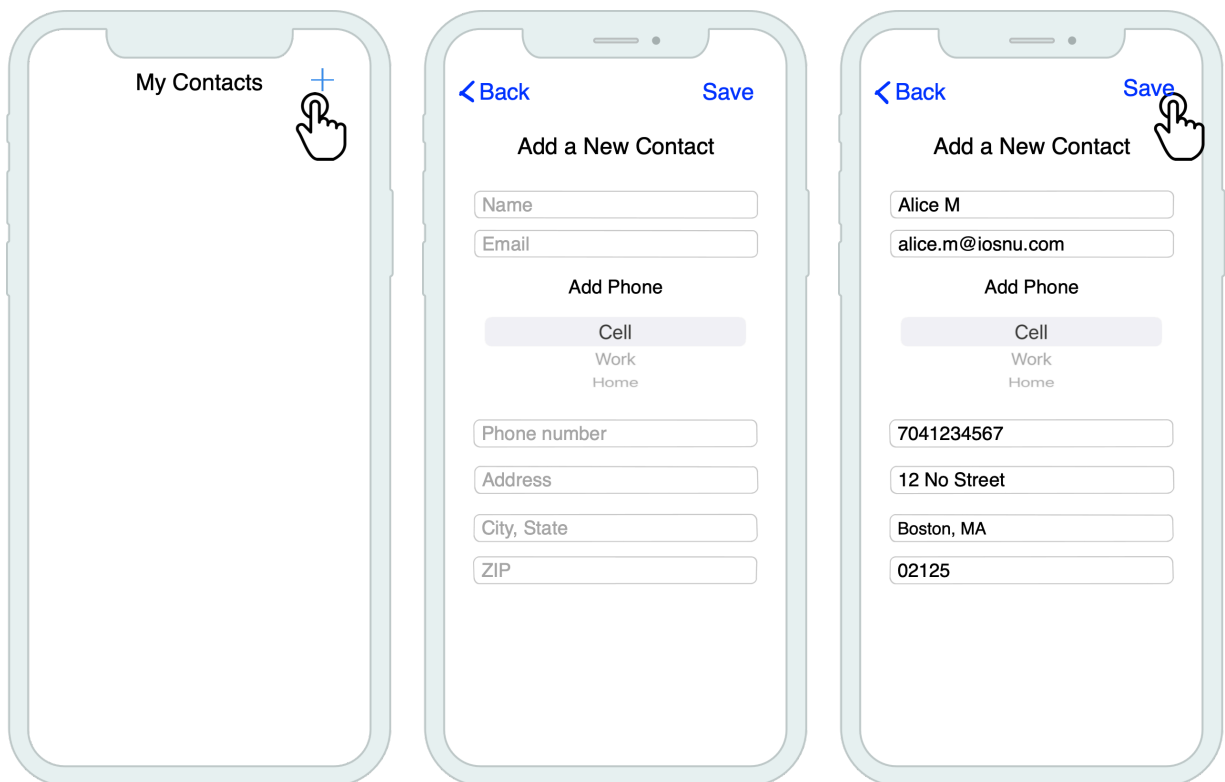
Weekly Assignment 4 (100 points)

Basic Instructions:

1. Submission details:

- The project name should be
WA4_<Your_Last_Name>_<Last_4_digits_of_your_NUID>
- After building the project, right-click on the project from Project Navigator.
- Click on Show in Finder.
- Select the project directory and the .xcodeproj file, and compress them to create a zip file.
- Then, submit the zip file in Canvas.
- (Please read the “How to Submit the Assignments” from Modules -> Quick Links)

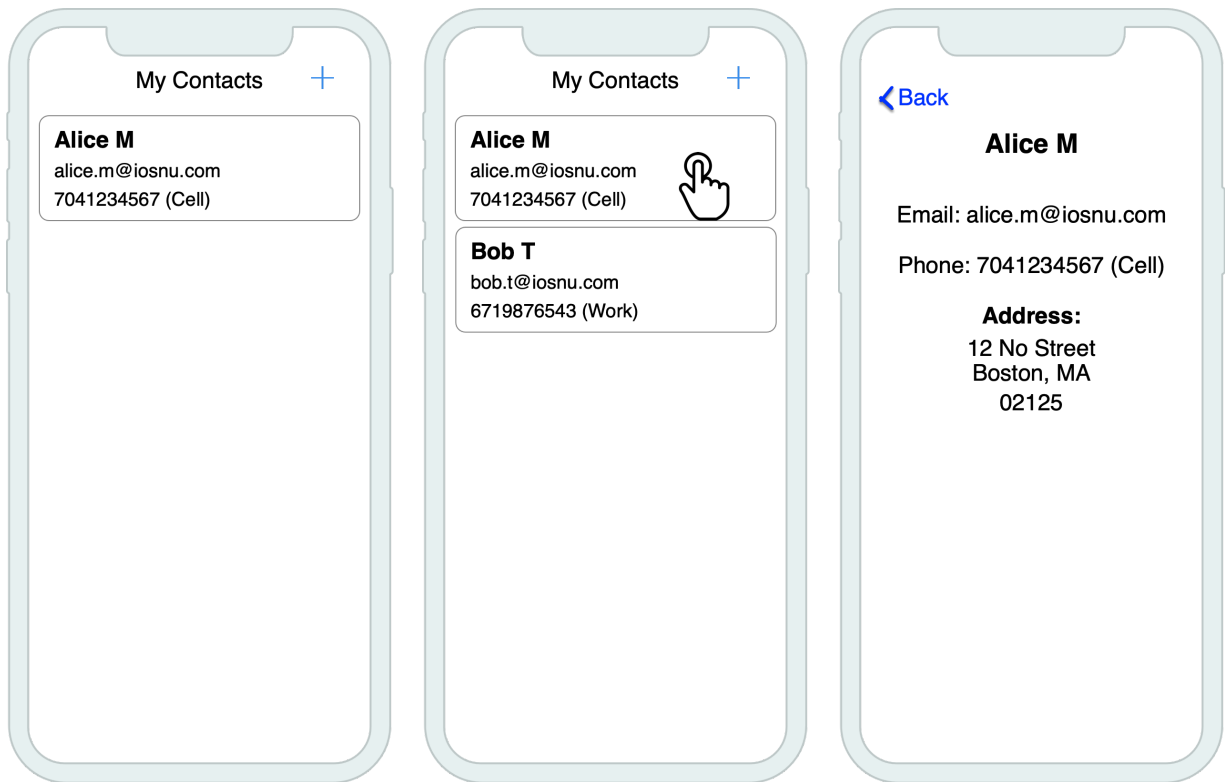
Weekly Assignment 4 (100 points)



a. Main Screen, tap 'Add' right Bar button

b. Add a New Contact Screen

c. Filling up the details and tapping 'Save'



d. Back to the Main Screen with the new contact. Display the contact in a TableView.

e. Selecting one item in the TableView opens the Details Screen.

f. Displaying the details in the Details Screen.

Figure 1: App Wireframes

Requirements

1. You must avoid using storyboards and build the interface with codes.
2. You must keep the front-end and back-end codes separate.
3. There will be three screens in the app.
 - a. **Main Screen** (Figure 1 (a)) will contain the following elements.
 - i. A right Bar Add button (with a plus icon).
 - ii. And there will be a TableView that occupies the whole screen.
 - b. **Add New Contact Screen** (Figure 1 (b and c)) will contain the following elements.
 - i. A 'Save' right Bar Button.

- ii. A Label displaying the text, “Add a New Contact.” Make this text a little bit bigger (play with the font size).
 - iii. Six TextFields will be used to put the name, email, phone number, address, city and state, and zip code.
 - iv. A UIPickerView to choose the type of phone number.
- c. **Details Screen** (Figure 1(f)): will contain the following elements.
- i. A Label on top displays the name of the contact, followed by two Labels displaying the email and phone number.
 - ii. Then, four Labels display the address in the following format:

Address:

1 Street Address

City, State

Zip

4. Main Screen: The main screen should have the following functionalities:

- a. If a user taps the ‘Add’ right Bar button, it should push the Add New Contact Screen on the UINavigationController.
- b. When a user completes adding a new contact and comes back to the main screen, the new contact should be added to the TableView and displayed.
- c. The TableView should display the following:
 - i. The name of the contact (in bold font)
 - 1. You can make a Label’s text bold by writing

```
labelTitle.font =  
UIFont.boldSystemFont(ofSize: 16)
```
 - ii. The email of the contact.
 - iii. And the phone number (with the type).
 - iv. You can try setting the border of the TableViewCell like the following:

1. For border color, use the wrapper view's **layer.borderColor** property. For example:
`wrapperCellView.layer.borderColor = UIColor.gray.cgColor`
 2. For border width, use the wrapper view's **layer.borderWidth** property.
 3. For the border's corner radius, use the wrapper view's **layer.cornerRadius** property.
- d. Tapping on a row on the TableView should open the Details Screen to display the tapped contact.
- i. Instantiate Details Screen.
 - ii. Send the contact to the Details Screen by setting the contact to a variable in the Details Screen.
 - iii. Push the Details Screen's instance on UINavigationController

5. Add New Contact Screen: The Add New Contact Screen should have the following functionalities:

- a. You should validate the inputs when the user taps the 'Save' button for the following:
 - i. No empty values for name, address, and 'city and state'.
 - ii. The email should be a valid email address.
 - iii. The phone number should be an integer.
 - iv. The zip code should be a valid Zip code.
 - v. None of the TextFields should be empty.
- b. The 'Save' right Bar button is a system button like 'Add.' You can add a 'Save' right Bar button by doing something like:

```
UIBarButtonItem(  
    barButtonItemSystemItem: .save, target: self,  
    action: #selector(onSaveBarButtonTapped)  
)
```
- c. If the user taps on the 'Save' Bar Button, It should delegate the tasks back to the main screen and add the contact to the TableView.

6. Details Screen: The Details screen should have the following functionalities:

- a. When the user taps on a row of the TableView, it should push the Details screen on the UINavigationController. Use the tableView() method for 'didSelectRowAt' to get the selected row, then send the corresponding contact to the Details screen.
 - b. The details screen should display the details like Figure 1 (f).
- 7. Hiding Keyboard:** You must hide the software/on-screen keyboard when tapping outside.

Good luck!!!