| **MAD2**<br>Diploma in IT<br>Year 2 (2018/19) Semester 4 | Week **4**<br>**3 hours** |
|---|---|
| **Practical 4: iOS Storyboard** | |

## Objectives

At the end of this practical exercise, the students should be able to:

- Design an iOS user interface using Storyboard
- Write and compile an iOS app, Friendster, using Storyboard

**IMPORTANT**

- You will be starting on a new project this week.
- At the end of the session, copy this folder (all your work) to MAD2 network folder so that your tutor may assess your work.
- The path of MAD2 network folder is **`\\ictspace.ict.np.edu.sg\MAD2`**

## Example 1 Part 1 – Using the iOS Storyboard

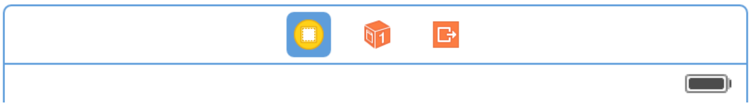
1. Launch Xcode and select File\New\New Project … and choose "Single View Application" template.

2. Enter the following
    - Product Name: Friendster
    - Company identifier: com.ngeeannappsuite
    - Device Family: iPhone

3. Save the project to your work folder containing your past exercises.

4. Select the file `Main.storyboard`.

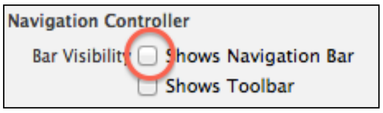
5. Click the Dock of the ViewController. It turns blue when selected.



6. From the main toolbar, select Editor\Embed in\Navigation Controller. This will embed the ViewController in a Navigation Controller.

7. You may zoom the storyboard in and out by double-clicking the left mouse button.

8. Click and select the Navigation Controller. (Its outline turns blue). From the Attributes inspector, uncheck “Shows Navigation Bar”.

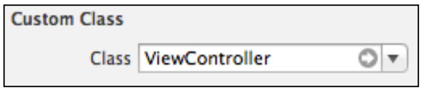


Navigation Controller

Bar Visibility ☐ Shows Navigation Bar

☐ Shows Toolbar

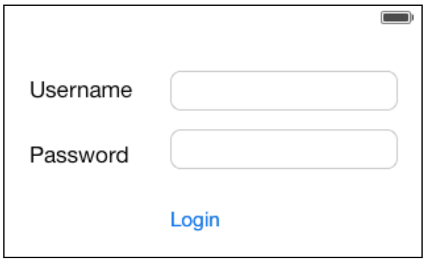
9. Click and select the ViewController Dock. (Its outline must turn blue). From the Identity inspector, select “ViewController” beside Class. This associates the current ViewController with the class files ViewController.swift



Custom Class

Class ViewController

10. Design the following login interface for the ViewController.

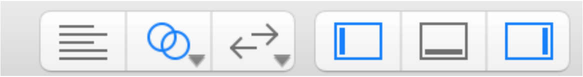


Username

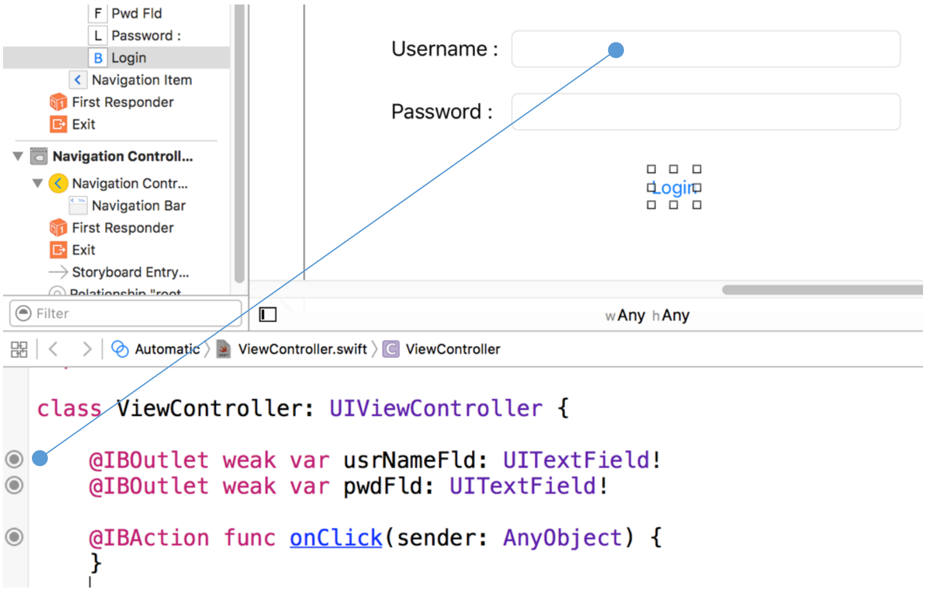
Password

Login

11. Click the Assistant editor on the top right of Xcode:



12. Control-click the username TextField and drag to ViewController.swift as shown below:



```
class ViewController: UIViewController {

    @IBOutlet weak var usrNameFld: UITextField!
    @IBOutlet weak var pwdFld: UITextField!

    @IBAction func onClick(sender: AnyObject) {
    }
```

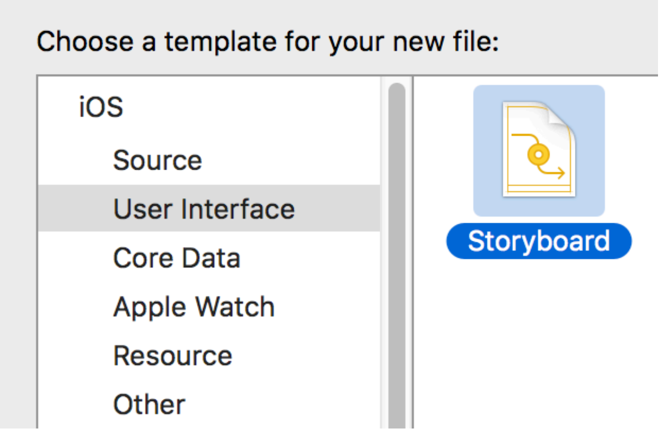
On the pop-up dialog, type `usrNameFld` as the instance variable name. Do the same with the password TextField, using `pwdFld` as the instance variable name.

13. Similarly, for the Login button "onclick" as IBAction.

14. Click Run.

## Example 1 Part 2 – Using the iOS Storyboard

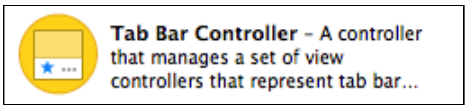
1. Select File\New\File…. Click User Interface and select Storyboard.



Give the storyboard the name "`AppStoryboard`".

2. Select the file `AppStoryboard.storyboard`.

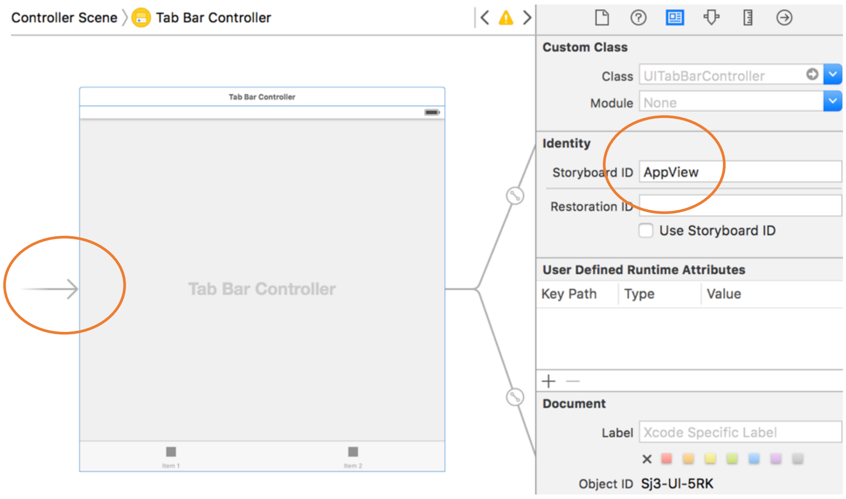
3. Click-drag a Tab Bar Controller to the Storyboard.



**Tab Bar Controller** – A controller that manages a set of view controllers that represent tab bar…

4. Click the top ViewController. From the toolbar, select Editor\Embed in\Navigation Controller. Do the same for the bottom ViewController.

5. Add the Storyboard ID as AppView and make sure that “is initial View Controller” checkbox is checked.



Custom Class
Class UITabBarController
Module None

Identity
Storyboard ID AppView
Restoration ID
□ Use Storyboard ID

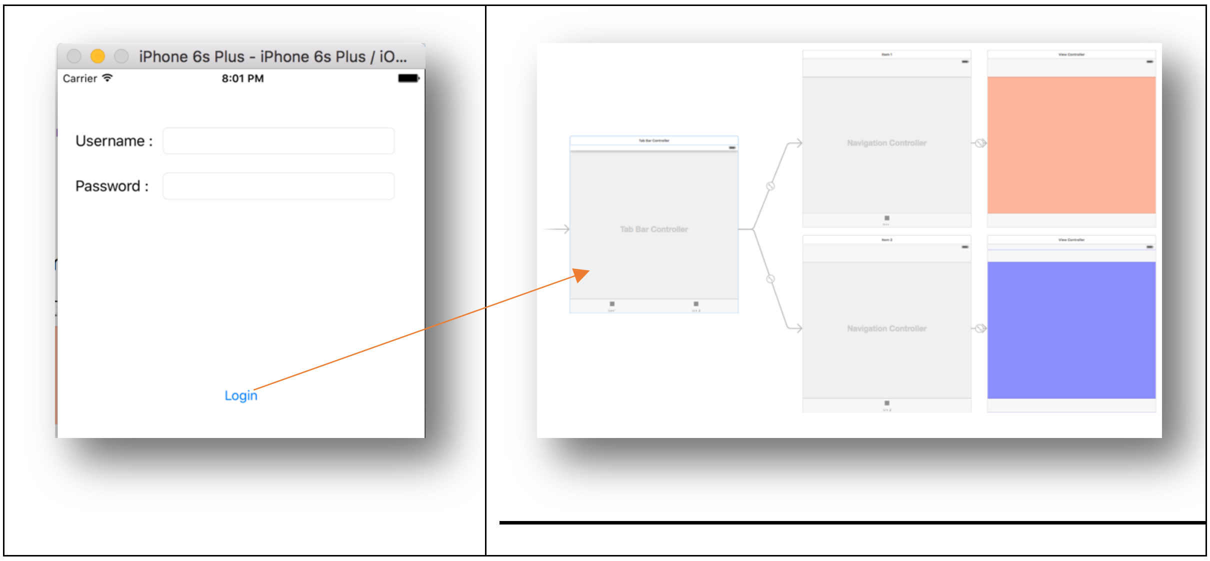
User Defined Runtime Attributes
Key Path | Type | Value

Document
Label Xcode Specific Label
Object ID Sj3-UI-5RK

6. Modify the button handler in ViewController.swift as shown below:

```
@IBAction func onClick(_ sender: AnyObject) {

    let storyboard = UIStoryboard(name: "AppStoryboard", bundle: nil)

    let vc = storyboard.instantiateViewController(withIdentifier: "AppView") as UIViewController

    present(vc, animated: true, completion: nil)

}
```

7. Click Run and observe the result. Note Item1 and Item2 bar buttons at the bottom of the Simulator.

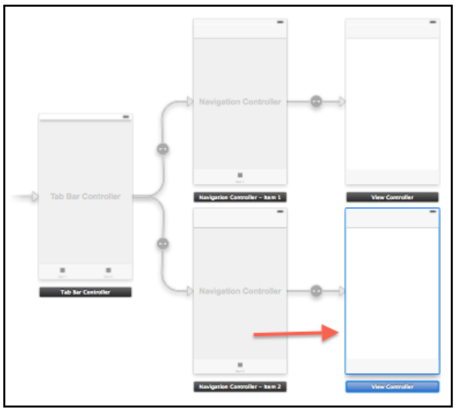


## Example 2 – Configuring the Tab Bar controller

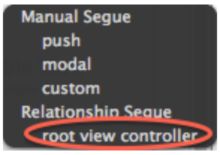
1. Continuing on, select the bottom ViewController (it must have blue outline)

2. Delete it.

3. Drag a “Table View Controller” to the vacant space.

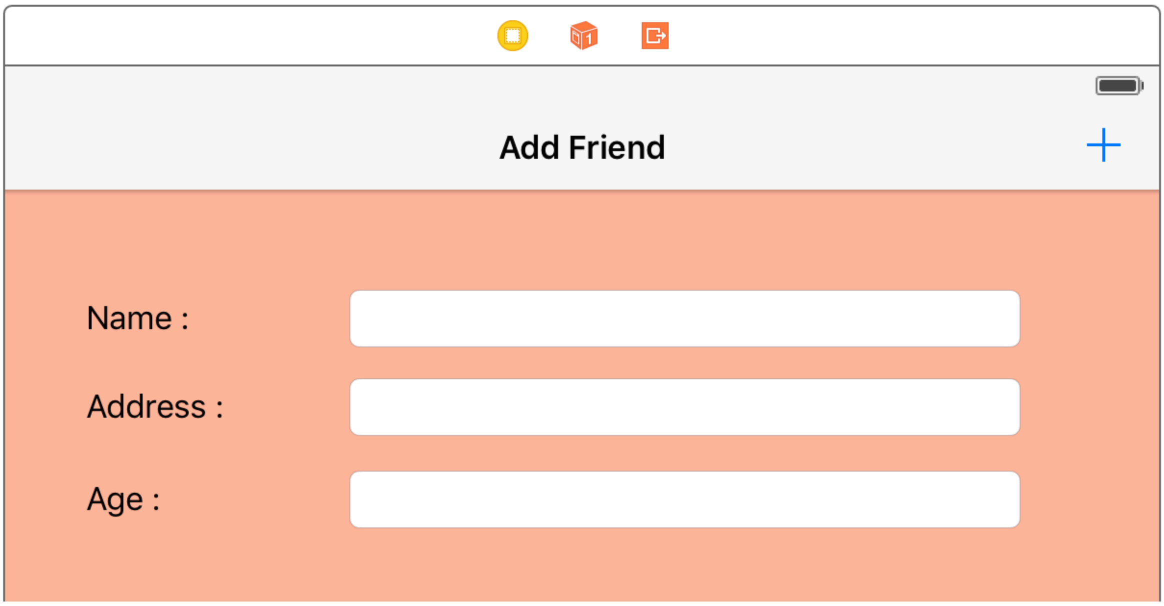


4. Ctrl-click the Navigation Controller and drag to the “Table View Controller”. Choose as shown below:

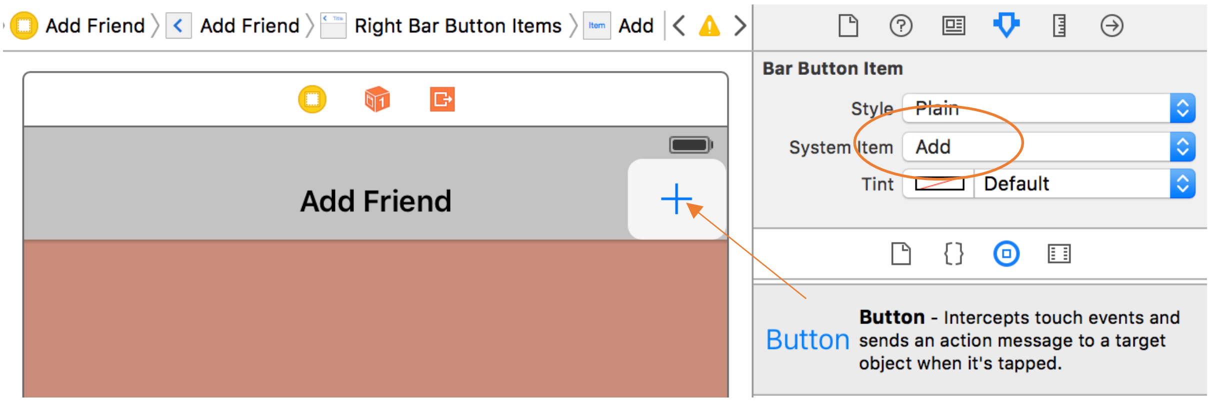


5. From the Xcode main menu, select File\New\File... create a new swift file called `AddFriend`.

6. Implement the following interface for the top View controller.



7. Double-click the Navigation bar to change the title. Select the Bar Button Item you just added. Select the Attributes inspector and change its Identifier.


Add Friend
Add Friend
Right Bar Button Items
Add
Add Friend
+

Bar Button Item

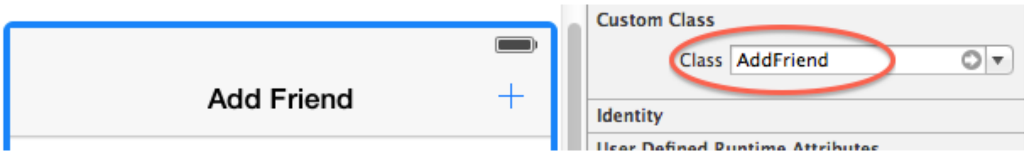
Style Plain

System Item Add

Tint Default

Button
Button - Intercepts touch events and sends an action message to a target object when it's tapped.

8. Link the above View Controller (blue outline) to the `AddFriend` subclass.

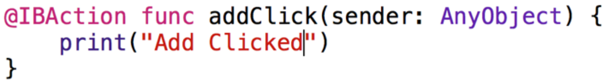


Class AddFriend

9. Wire up the 3 UITextFields to 3 instance variables.  Use the names shown below:

```
@IBOutlet weak var nameTxtFld: UITextField!
@IBOutlet weak var addressTxtFld: UITextField!
@IBOutlet weak var ageTxtFld: UITextField!
```

10. Ctrl-click-drag the '+' button to AddFriend.swift and name the IBAction as shown below:



```
@IBAction func addClick(sender: AnyObject) {
    print("Add Clicked")
}
```

11. Change the title of the Table View controller to "Friend list".

12. Click Run and observe the result.

## Example 3 – Adding a friend

1. Add `Friend.swift` (from Week2 project) to the current project.

2. Add the following code to AppDelegate.swift

```
var window: UIWindow?
var friendsList:[Friend] = []
```

3. Copy and paste/replace the following lines in AppDelegate.swift

```
func application(application: UIApplication, didFinishLaunchingWithOptions
    launchOptions: [NSObject: AnyObject]?) -> Bool {
    // Override point for customization after application launch.

    friendsList = [
        Friend(firstname: "Apple", lastname: "Seed", age: 17),
        Friend(firstname: "Blue", lastname: "Berry", age: 18),
        Friend(firstname: "Corn", lastname: "Syrup", age: 20),
        Friend(firstname: "Date", lastname: "Wheat", age: 16),
        Friend(firstname: "Egg", lastname: "Plant", age: 13)
    ]


    return true
}
```

4. Add the following lines to the top of AddFriend.swift

6. Add the following lines to `@IBAction func addClick(sender: AnyObject)` in AddFriend.swift

```
        let appDelegate = UIApplication.shared.delegate
                             as! AppDelegate
        print(String(appDelegate.friendsList.count))
```

You can now access the Array `friendsList` through the `appDelegate` object.

7. Click Run. Then, click the Add button and observe the print output.

Exercise 1 Modify the `onAdd:` handler to add a Friend object based on the information provided by the user. Verify that the Friend object is indeed added.

## Example 4 – show friend list in a TableView

1. From the Xcode main menu, select File\New\File... and add a UITableViewController subclass called `ShowFriendsController`.

2. Go back to the `AppStoryboard.storyboard` and link the "Friend list" scene to the `ShowFriendsController` class.Refer to step 8 from Example 2.

3. Click on the single row in the table to highlight it in blue. Change its Attributes as shown below:

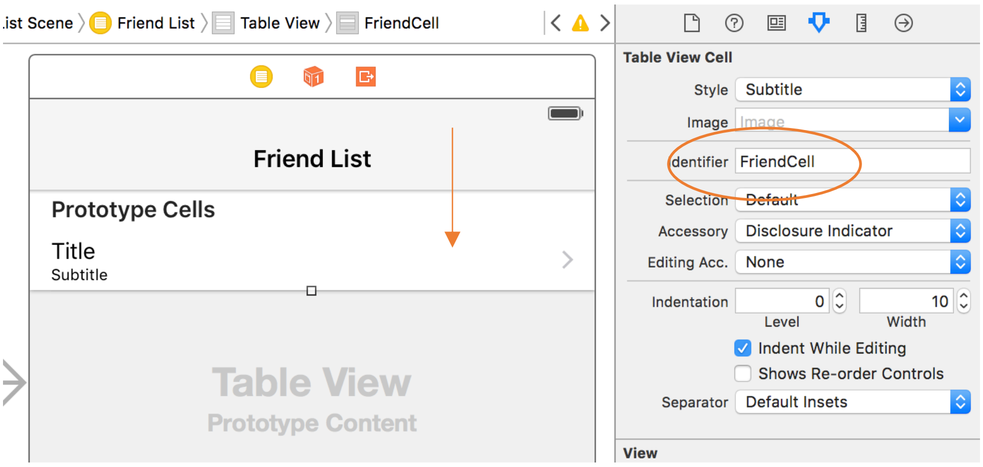


Table View Cell

Style: Subtitle

Image: Image

dentifier: FriendCell

Selection: Default

Accessory: Disclosure Indicator

Editing Acc.: None

Indentation: 0 (Level) 10 (Width)

☑ Indent While Editing

□ Shows Re-order Controls

Separator: Default Insets

View

4. Modify ShowFriendsController.swift as shown below:

```
class ShowFriendsController: UITableViewController {

    var appDelegate = AppDelegate()
```

5. Modify ShowFriendsController.swift as shown below:

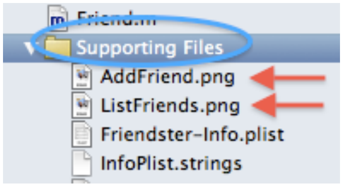
```
override func viewDidLoad() {
    super.viewDidLoad()
    // Do any additional setup after loading the view, typically from a nib.

    appDelegate = UIApplication.shared.delegate as! AppDelegate
    self.tableView.reloadData() //refresh data

}

override func viewDidAppear(_ animated: Bool) {
    self.tableView.reloadData()

}

override func numberOfSections(in tableView: UITableView) -> Int {
    return 1
}

override func tableView(_ tableView: UITableView, numberOfRowsInSection
    section: Int) -> Int {
    return appDelegate.friendsList.count
}

override func tableView(_ tableView: UITableView, cellForRowAt indexPath:
    IndexPath) -> UITableViewCell {
    let cell = self.tableView.dequeueReusableCell(withIdentifier:
        "FriendCell",for: indexPath)
    let f = appDelegate.friendsList[(indexPath as NSIndexPath).row]
    cell.textLabel?.text = f.firstName
    cell.detailTextLabel?.text = f.address
    return cell
}
```

Question: Where else did you encounter "`FriendCell`"?

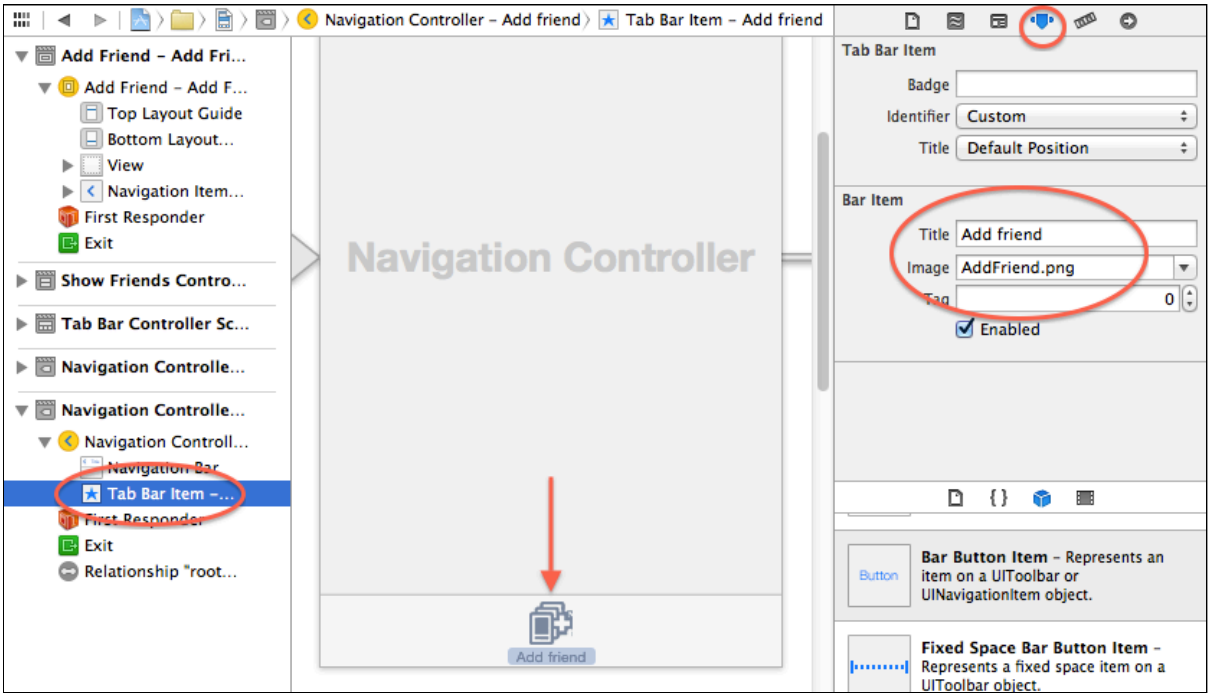
6. Click Run and observe the result. Add a new Friend and ensure that the new object is displayed in the Friend list scene.

7. Download these images files from MeL: `AddFriend.png` and `ListFriends.png`

8. Drag both files to the "Supporting Files" folder on the left of Xcode.



9. Go back to the `AppStoryboard.storyboard` and click the “Add Friend” tab. Assign an image for this tab, as shown below:



Tab Bar Item

Badge:

Identifier: Custom

Title: Default Position

Bar Item

Title: Add friend

Image: AddFriend.png

Tag: 0

☑ Enabled

11. Assign the next image for the “List friends” tab.

Exercise 2 Add a suitable image for the Login button in `Main.storyboard`.

Exercise 3 Modify the `ShowFriendsController` subclass to allow the user to delete an existing friend from his list. Refer to the lecture notes for more information.

Exercise 4 Enhance the Friendster project to allow the user to update a friend’s details, after the friend has been selected in `ShowFriendsController`.