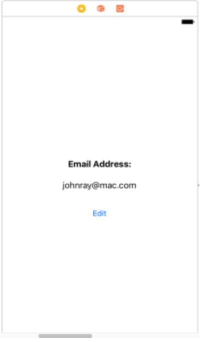
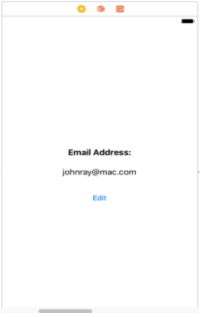
In-class activity on ModalEditor:

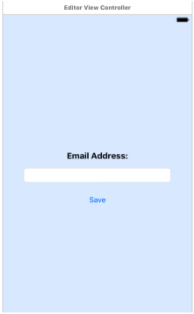
1. Create a single view app with XCode, and name it as MyModalEditor. Set up the intial View Controller to look like this below with the first Label (with text: Email Address:), the second Label (with text: johnray@mac.com) and one Button (Edit):



Select “View Controller Scene” in the Document outline panel, use Attributes Inspector to change the title to “First Controller”.

2. Add a new View Controller with a View of blue background, one label (with text: Email Address:), one TextField (empty text string) and one Button (Save). Then select the second “View Controller Scene” in the Document outline panel, use Attributes Inspector to change the title to “Edit Controller”.





- Right-click on MyModalEditor project folder → New file → Select “Swift File” → Next → Save As: EditViewController → Create

- Copy the contents of ViewController.swift class and paste it into the new Swift file. Then change the class name to “EditViewController”.

- Click on Main.storyboard icon → Edit Controller Scene → the “Show the identity inspector” icon on the top right panel → Choose “EditViewController” class . Now “Edit Controller” links to EditViewController.swift file. Please note that “First Controller” already linked to ViewController.swift file (i.e. associated with ViewController class) as soon as we created this single view app.

3. Connect the First Controller’s second label as referencing outlet to the ViewController class file.

```
@IBOutlet weak var emailLabel: UILabel!
```

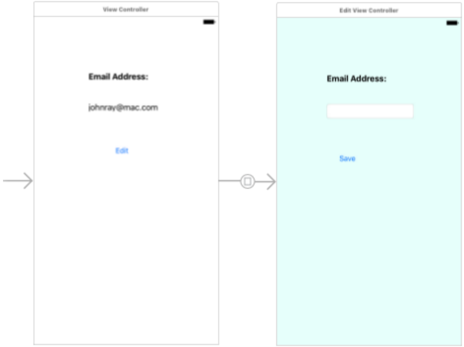
Add an @IBAction func to be called when it returns from the Edit Controller:

```
 // This function will be linked programmatically from the “Unwind segue” to be
//  created later in the EditViewController layout.
 @IBAction func exitToHere(_ sender: UIStoryboardSegue) {
         // No code needed!
 }
```

ViewController class file should look like this:

```
import UIKit

class ViewController: UIViewController {
    @IBOutlet weak var emailLabel: UILabel!

    // This function will be linked programmatically from the “Unwind segue” to be
    //  created later in the EditViewController layout.
    @IBAction func exitToHere(_ sender: UIStoryboardSegue) {
        // No code needed!
    }

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view.
    }
}
```

4. Next create a “Segue” from First Controller to switch to Edit Controller by right-clicking on the Edit button → Triggered Segues: action → Drag to EditViewController and released → Choose **Present Modally**.



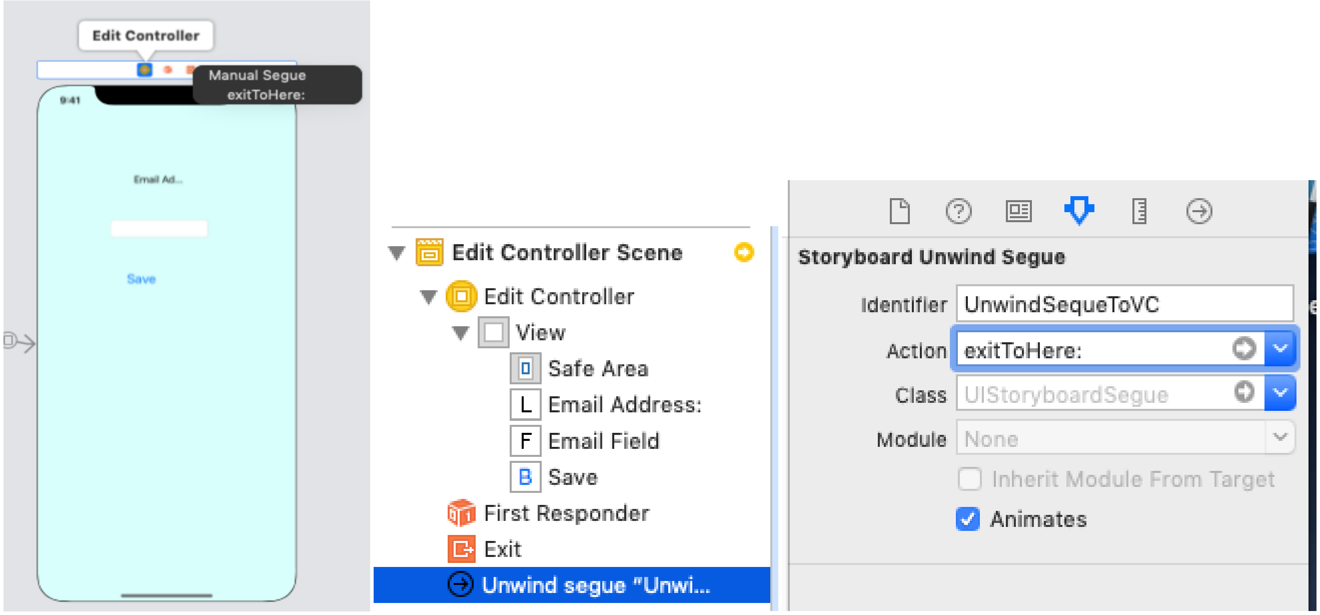
5. Connect the Edit Controller’s Textfield as referencing outlet to the EditViewController class, and sent an event like “Touch inside up” from the Save button to call a method “updateEditor()”.

```
@IBOutlet weak var emailField: UITextField!
 @IBAction func updateEditor(_ sender: Any) {
        (presentingViewController as! ViewController).emailLabel.text =  emailField.text
         /* We''ll use "Triggered Segue" action connect at the end. But if we don't use "Triggered Segue"
            action connection, then we need to call "performSegue function" like this:
         */
         //  performSegue(withIdentifier: "UnwindSegueToVC", sender: self)
       }
```

6. Add a couple more functions to support required actions. The EditViewController class file should end up like this:

```
import UIKit

class EditViewController: UIViewController {
    @IBOutlet weak var emailField: UITextField!
    @IBAction func updateEditor(_ sender: Any) {
        (presentingViewController as! ViewController).emailLabel.text =  emailField.text
         /* We''ll use "Triggered Segue" action connect at the end. But if we don't use "Triggered Segue"
            action connection, then we need to call "performSegue function" like this:
         */
         //  performSegue(withIdentifier: "UnwindSegueToVC", sender: self)
    }

    // This function will be called by an action event " when the "Done" or "Enter"
    //  key is hitted for the emailField.
    @IBAction func hideKeyboard(_ sender: AnyObject) {
        emailField.resignFirstResponder()
    }

    override func viewWillAppear(_ animated: Bool) {
        emailField.text = (presentingViewController as! ViewController).emailLabel.text
        super.viewWillAppear(animated)
    }

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view.
    }
}
```

7. Create an "Unwind segue" by Control- dragging the Edit Controller icon into the Exit icon at the top of the Edit Controller layout (and make sure that you **select** the function exitToHere:). Next select the Unwind segue and provide an identifier (eg.UnwindSegueToVC):



Storyboard Unwind Segue

Identifier: UnwindSequeToVC

Action: exitToHere:

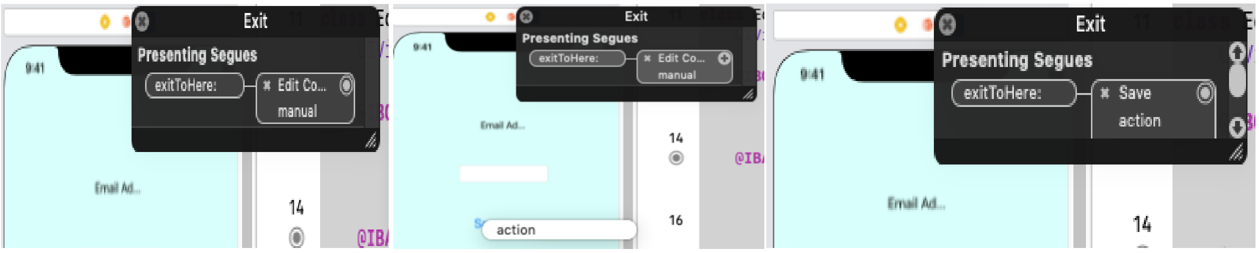
Class: UIStoryboardSegue

Module: None

☐ Inherit Module From Target

☑ Animates

8. When the **Save** button is pressed, it triggers the updateEditor function in which we can call a function like: performSegue (withIdentifier: "UnwindSegueToVC", sender: self)
   OR
   We can directly connect “Triggered Segues: action” to the **Save** button like this:
   Right-click on the **Exit** icon → Drag the dot circle to the **Save** button → click on the displayed “action” alert button→ done!



9. Run the app! Take a couple screen shots to show the email address has been edited and saved on the first screen. Submit your screen shots.