Inclass Activity on TabBarControllerView:

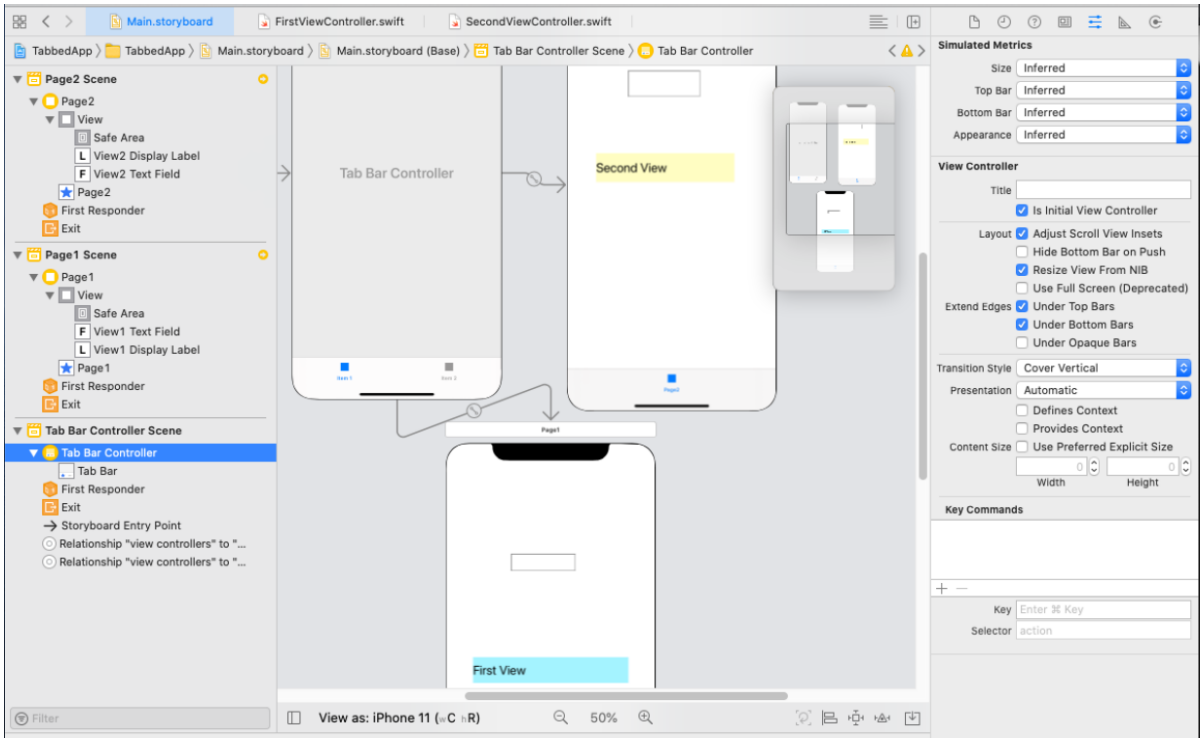
1. Create a new single view iOS app with XCode, and name it as MyTabBarControllerApp.

2. Clear the Storyboard by removing the View Controller object.

3. Add a **Tab Bar Controller** object into Storyboard from Objects library. Select **Tab Bar Controller Scene**, go to Attribute Inspector and check the "**Is Initial View Controller**" checkbox

4. Select the star icon under item 1 Scene, change the title from "item 1" to "Page1". Select the star icon under item 2 Scene, change the title from "item 2" to "Page2". Note: all naming with "item 1" will automatically be changed to "Page1", and "item 2" be changed to "Page2".

5. Add both a TextField and a Label objects on First View Controller and Second View Controller respectively. Use Attributes Inspector for Highlighting the Label with light-blue background color on First View Controller, and with light-yellow background color on Second View Controller, and increase the font size (to 24 points) of the label for both view controllers:



6. Create two new Swift files, and name them as FirstViewController and SecondViewController respectively. Copy the code from ViewController.swift into the two new files, and then rename the class name from **ViewController** to **FirstViewController** and **SecondViewController** respectively.

7. Select the first view controller (Page1 Scene), go to Identity Inspector and select **FirstViewController** class for it. Select the second view controller (Page2 Scene), go to Identity Inspector and select **SecondViewController** class for it.

8. Now connect each View Controller's GUI object as referencing outlets to its associated Swift file (i.e Page1 Scene View Controller connects to FirstViewController.swift; Page2 Scene View Controller connects to SecondViewController.swift). After the connections, we should end up the file contents of both swift files like these:

```
import UIKit

class FirstViewController: UIViewController {

    @IBOutlet weak var view1Textfield: UITextField!
    @IBOutlet weak var view1DisplayLabel: UILabel!

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

```
import UIKit

class SecondViewController: UIViewController {

    @IBOutlet weak var view2Textfield: UITextField!
    @IBOutlet weak var view2DisplayLabel: UILabel!

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

9. Next add an override function at the bottom of the FirstViewController class:

```
override func viewWillDisappear(_ animated: Bool) {
  if let secondTab = self.tabBarController!.viewControllers![1]
as? SecondViewController {
          secondTab.view2DisplayLabel.text = view1Textfield.text!
+ " from Page1!"
      }
   }
```

10. Also add the same override function at the bottom of the SecondViewController class:

```
override func viewWillDisappear(_ animated: Bool) {
   if let firstTab = self.tabBarController!.viewControllers![0] as?
FirstViewController {
          firstTab.view1DisplayLabel.text = view2Textfield.text! + "
from Page2!"
      }
   }
```

11. Run the app.

    When you enter a name (e.g. Johnny) in the text field and touch on the “Page2” tab, the first view disappears, the second view will appear and a message like “Johny from Page1!” will show on the bottom label. When you enter a name (e.g. Peter) in the text field on the second view, and touch on the “Page1” tab, the second view disappears, the first view will appear again with a message like “Peter from Page2!” will show on the bottom label. This shows that both views are able to pass information to each other.

    Take a couple screen shots of the results and submit them.