

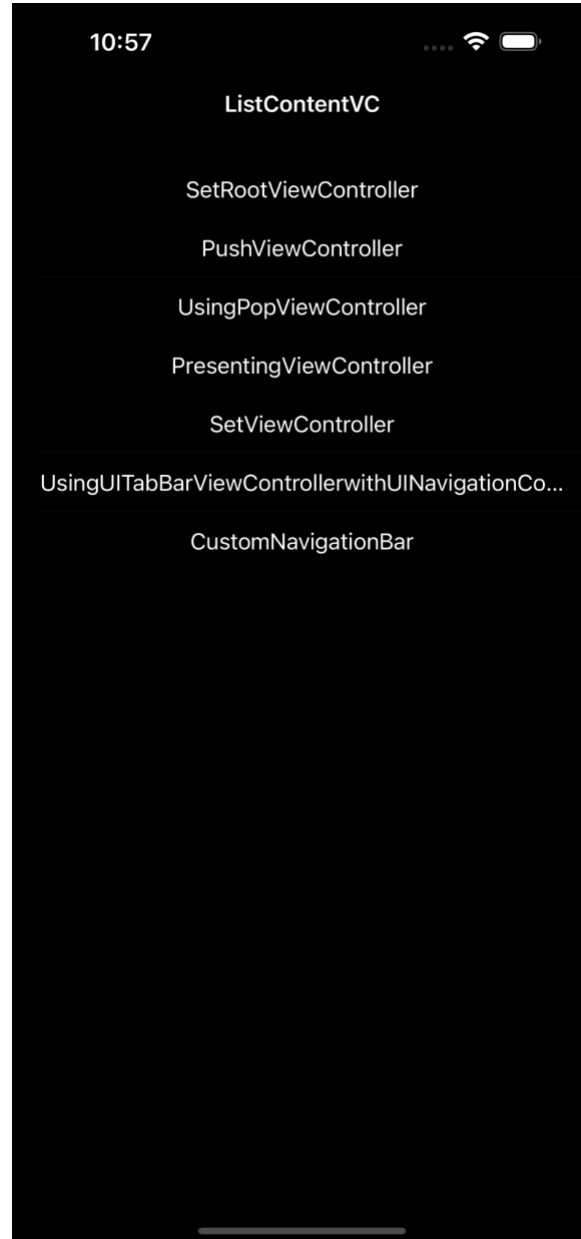
Content

1. Introduction
2. Sample project: ListContent -> ListContentSB
 - a. SetRootViewController
 - b. PushViewController
 - c. UsingPopViewController
 - i. PopViewController
 - ii. PopToViewController
 - iii. PopToRootViewController
 - d. PresentingViewController
 - e. SetViewController
 - i. Append VC
 - ii. Remove VC
 - iii. Insert VC
 - f. Using UITabBarController with UINavigationController
 - i. UITabBarController with UIViewController
 - ii. UITabBarController with Container View
 - g. Custom NavigationBar
 - i. NavigationBar with StatusBar: Normal Navigation & Large Title Navigation
 - ii. NavigationBarItem
 - iii. CustomNavigation

1. Introduction

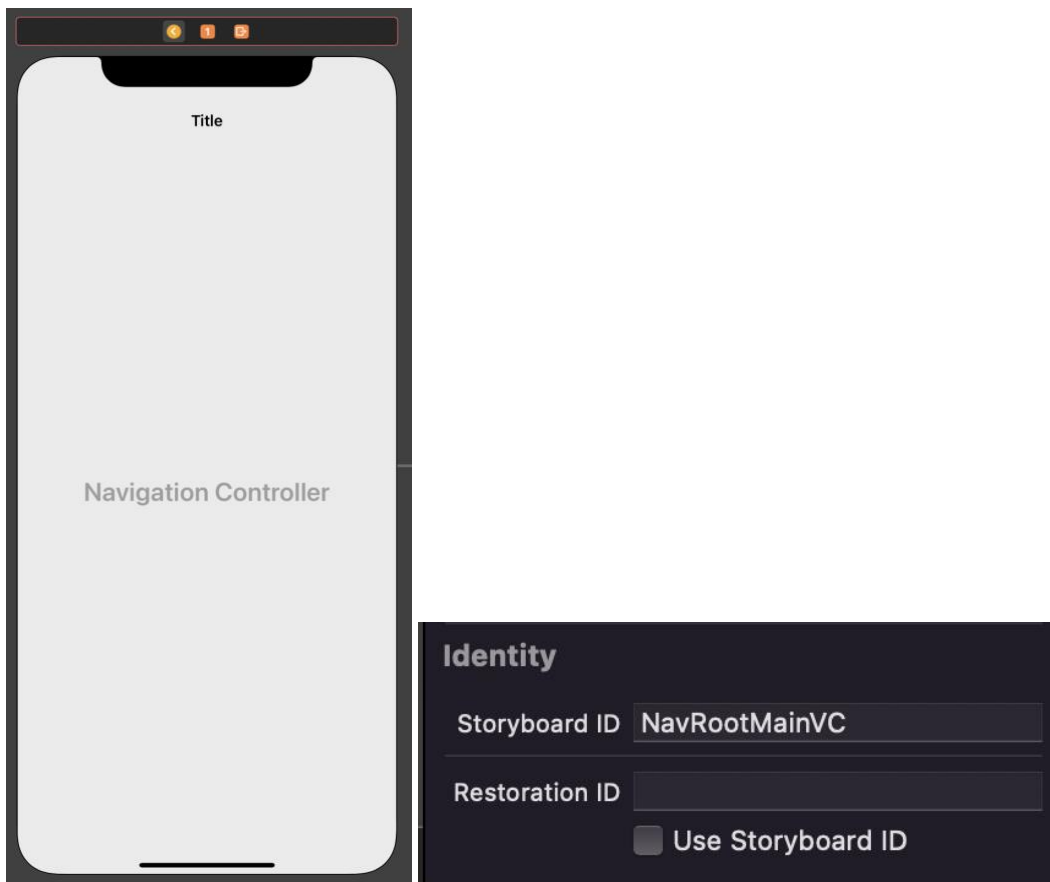
A container view controller that defines a stack-based scheme for navigating hierarchical content.

2. Sample project: ListContent → ListContentSB



a. SetRootViewController

- In RootSB set Storyboard ID to **NavRootMainVC**

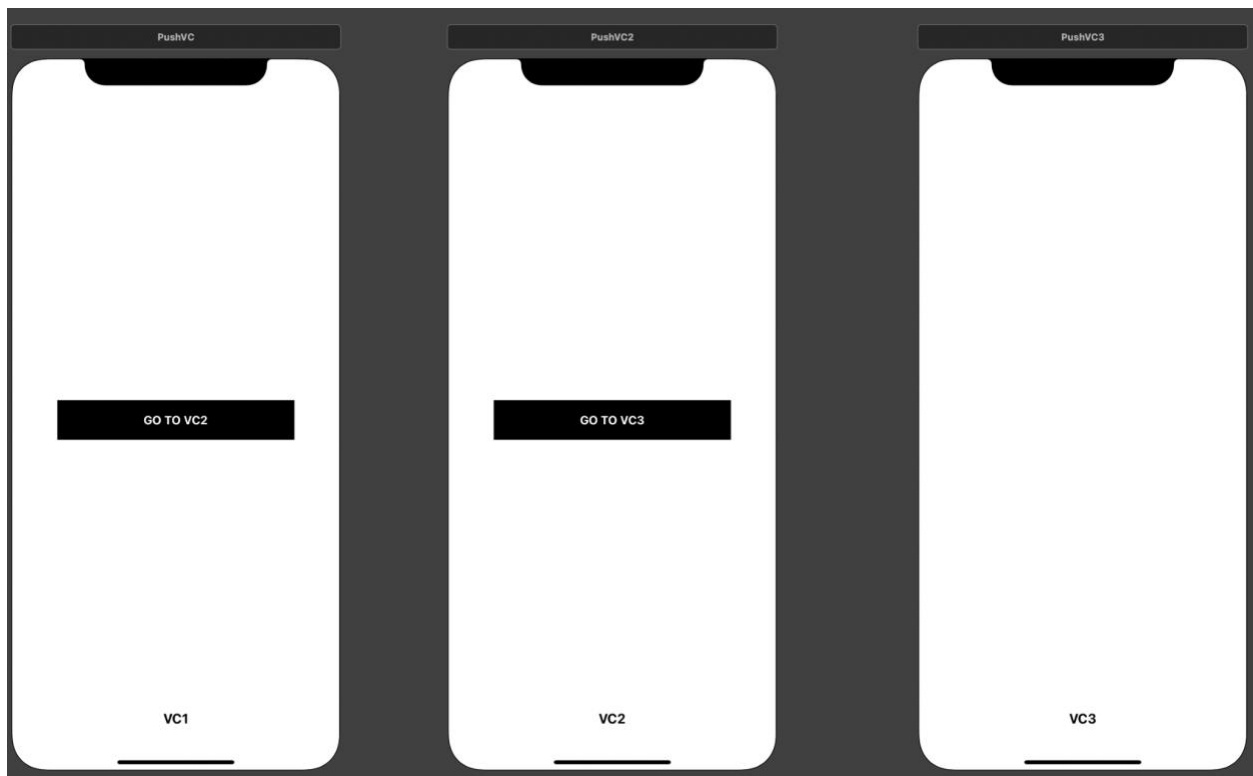


- On **btnLoginClick** write code below to set root

```
func setRoot() {  
  
    // Get SB name  
    let sbName = UIStoryboard(name: "RootSB", bundle: nil)  
  
    // Get VC name  
    let navVC = sbName.instantiateViewController(withIdentifier: "NavRootMainVC")  
  
    // Check NavVC is UINavigationController type  
    if let navViewController = navVC as? UINavigationController {  
  
        // Set root to Navigation of RootMainVC  
        if let window = UIApplication.shared.keyWindow {  
            window.rootViewController = navViewController  
        }  
    }  
}
```

b. PushViewController

- In PushSB have VC1, VC2 & VC3

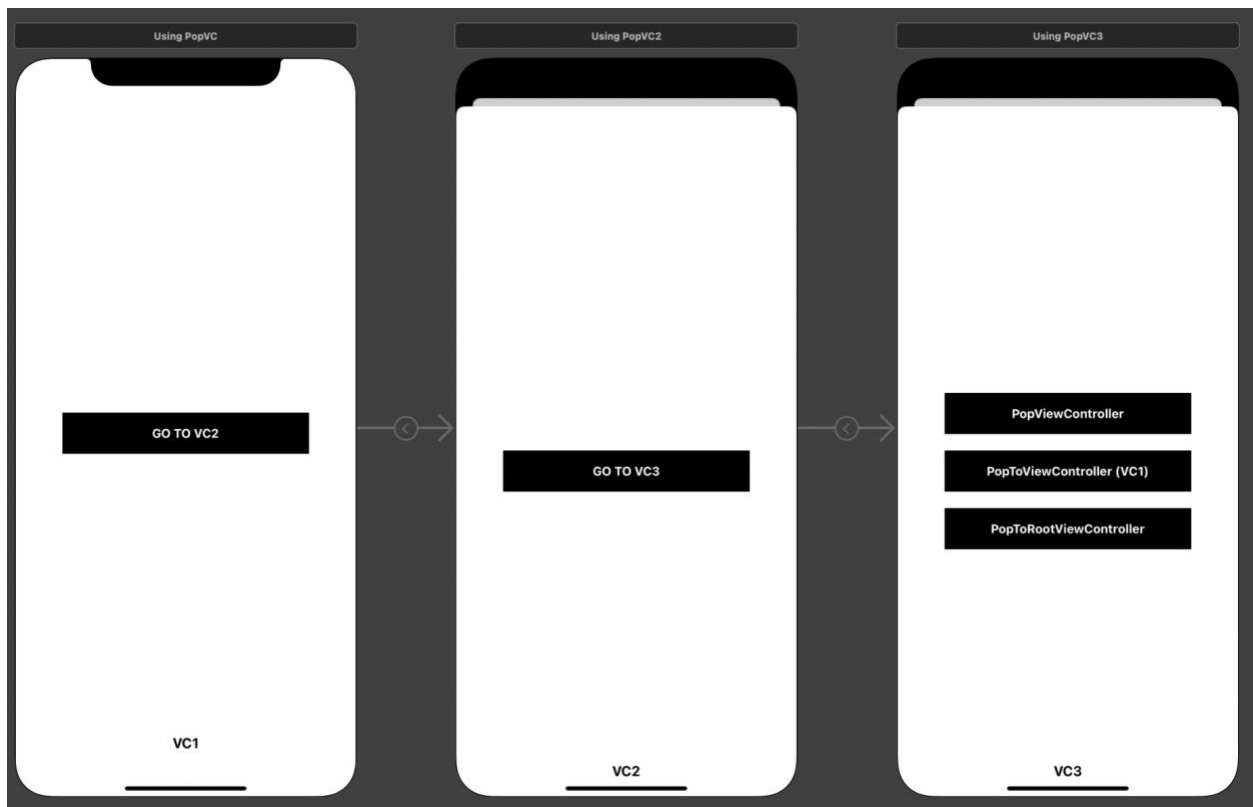


- In PushVC when click `btnGoToVC2` call function `pushNavigation`

```
@IBAction func btnGoToVC2(_ sender: Any) {  
    pushNavigation(storyboard: "PushSB", identifier: "PushVC2")  
}  
  
func pushNavigation(storyboard: String, identifier: String) {  
    let vc = UIStoryboard(name: storyboard, bundle: nil).instantiateViewController(withIdentifier: identifier)  
    navigationController?.pushViewController(vc, animated: true)  
}
```

c. UsingPopViewController

- In UsingPopSB have VC1, VC2 & VC3



i. PopViewController

- In UsingPopVC3 when click **btnPopVCClick**, it will go to VC2

```
@IBAction func btnPopVCClick(_ sender: Any) {
    self.navigationController?.popViewController(animated: true)
}
```

ii. PopToViewController

- In UsingPopVC3 when click **btnPopToVCClick**, it will go to VC1

```
@IBAction func btnPopToVCClick(_ sender: Any) {
    if let vc = self.navigationController?.viewControllers.filter({ $0 is UsingPopVC }).first as? UsingPopVC {
        self.navigationController?.popToViewController(vc, animated: true)
    }
}
```

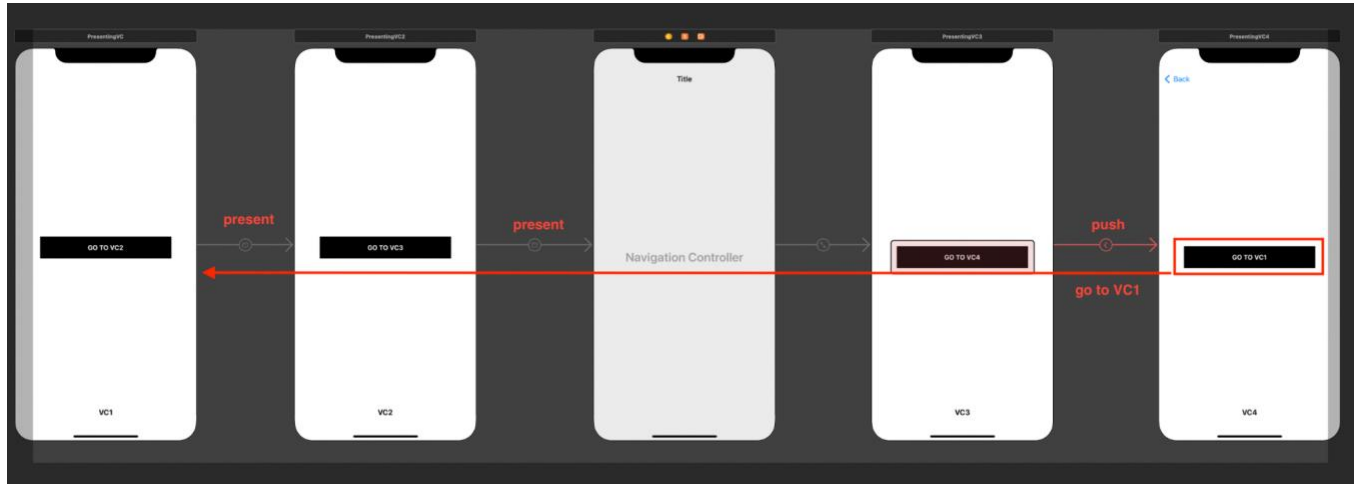
iii. PopToRootViewController

- In UsingPopVC3 when click **btnPopToRootVCClick**, it will go to ListContentVC

```
@IBAction func btnPopToRootVCClick(_ sender: Any) {
    self.navigationController?.popToRootViewController(animated: true)
}
```

d. PresentingViewController

- In PresentingSB have VC1, VC2, VC3, VC4, GO TO VC2 & GO TO VC3 use **present** but GO TO VC4 use **push**



- In PresentingVC4 when click **btnGoToVC1Click**, it will go to VC1 by using code below

```
@IBAction func btnGoToVC1Click(_ sender: Any) {
    self.presentingViewController?.presentingViewController?.dismiss(animated: true, completion: nil)
}
```

e. SetViewController

- In SetSB have VC1, VC2 , VC3 & VC4



i. Append VC

- In SetVC when click **btnAppendVC2ToVC4Click**, it will append VC2, VC3 & VC4 by using code below

```
@IBAction func btnAppendVC2ToVC4Click(_ sender: Any) {
    // Declare navigationController
    var controllers = navigationController?.viewControllers

    // Check VC is available
    guard let secondVC = storyboard?.instantiateViewController(withIdentifier: "SetVC2") else {return}
    guard let thirdVC = storyboard?.instantiateViewController(withIdentifier: "SetVC3") else {return}
    guard let fourthVC = storyboard?.instantiateViewController(withIdentifier: "SetVC4") else {return}

    // Append VC into navigationController
    controllers?.append(secondVC)
    controllers?.append(thirdVC)
    controllers?.append(fourthVC)

    if let controllers = controllers {
        // This method will push or present depending on whether the new top view controller was previously in the stack.
        navigationController?.setViewControllers(controllers, animated: true)
    }
}
```

ii. Remove VC

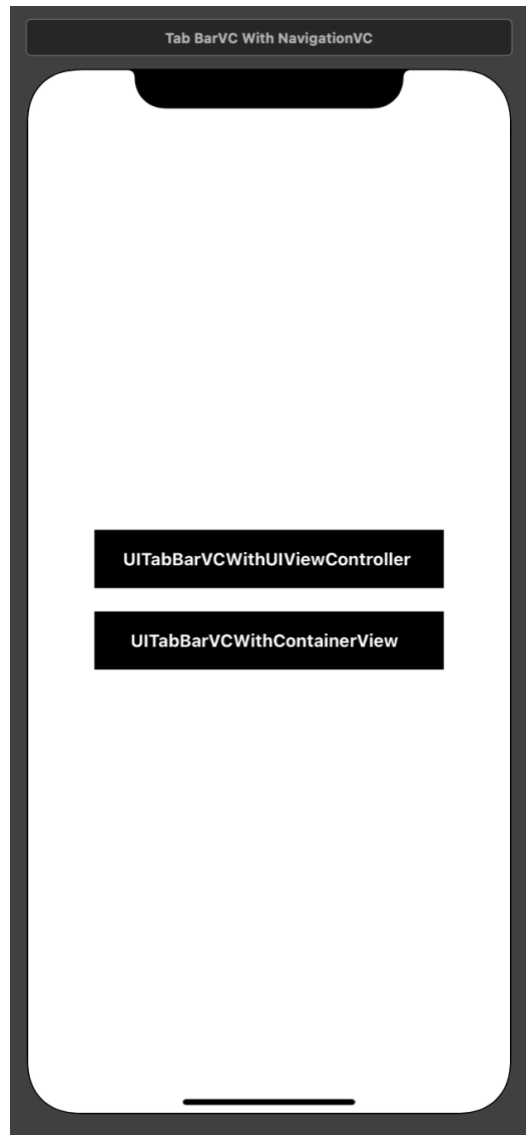
- In SetVC4 when click **btnRemoveVC**, it will remove VC:
 - `navVCArray?.removeLast()` //Remove the last view controller
 - `navVCArray?.removeFirst()` // Remove the first view controller
 - `navVCArray?.removeAll()` //Remove all view controllers
 - `navVCArray?.remove(at: 0)` //Remove at specific position

iii. Insert VC

- In SetVC4 when click **btnInsertVC**, it will insert VC2

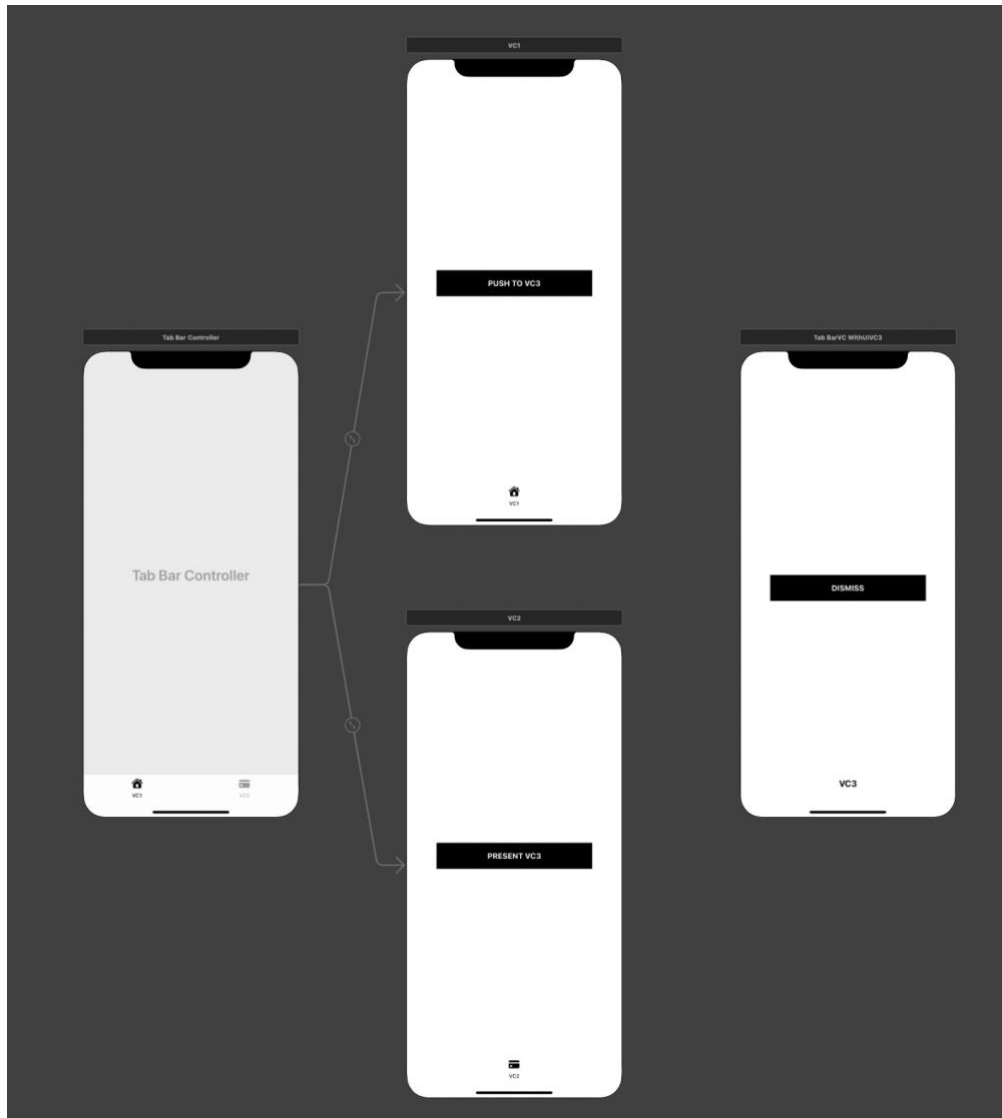
```
@IBAction func btnInsertVC(_ sender: Any) {  
  
    let vc = storyboard?.instantiateViewController(identifier: "SetVC2")  
    navigationController?.viewControllers.insert(vc!, at: 0)  
  
}
```

f. Using UITabBarController with UINavigationController



i. UINavigationController with UINavigationController

- In TabBarVCWithUIVC, add Tab Bar Controller that has VC1 & VC2



- In TabBarVCWithUINavigationController, on **btnPushToVC3Click** we push to TabBarVCWithUINavigationController

```
@IBAction func btnPushToVC3Click(_ sender: Any) {  
    pushNavigation(storyboard: "TabBarVCWithUINavigationController", identifier: "TabBarVCWithUINavigationController")  
}
```

- In TabBarVCWithUINavigationController2, on **btnPresentVC3Click** we present to TabBarVCWithUINavigationController

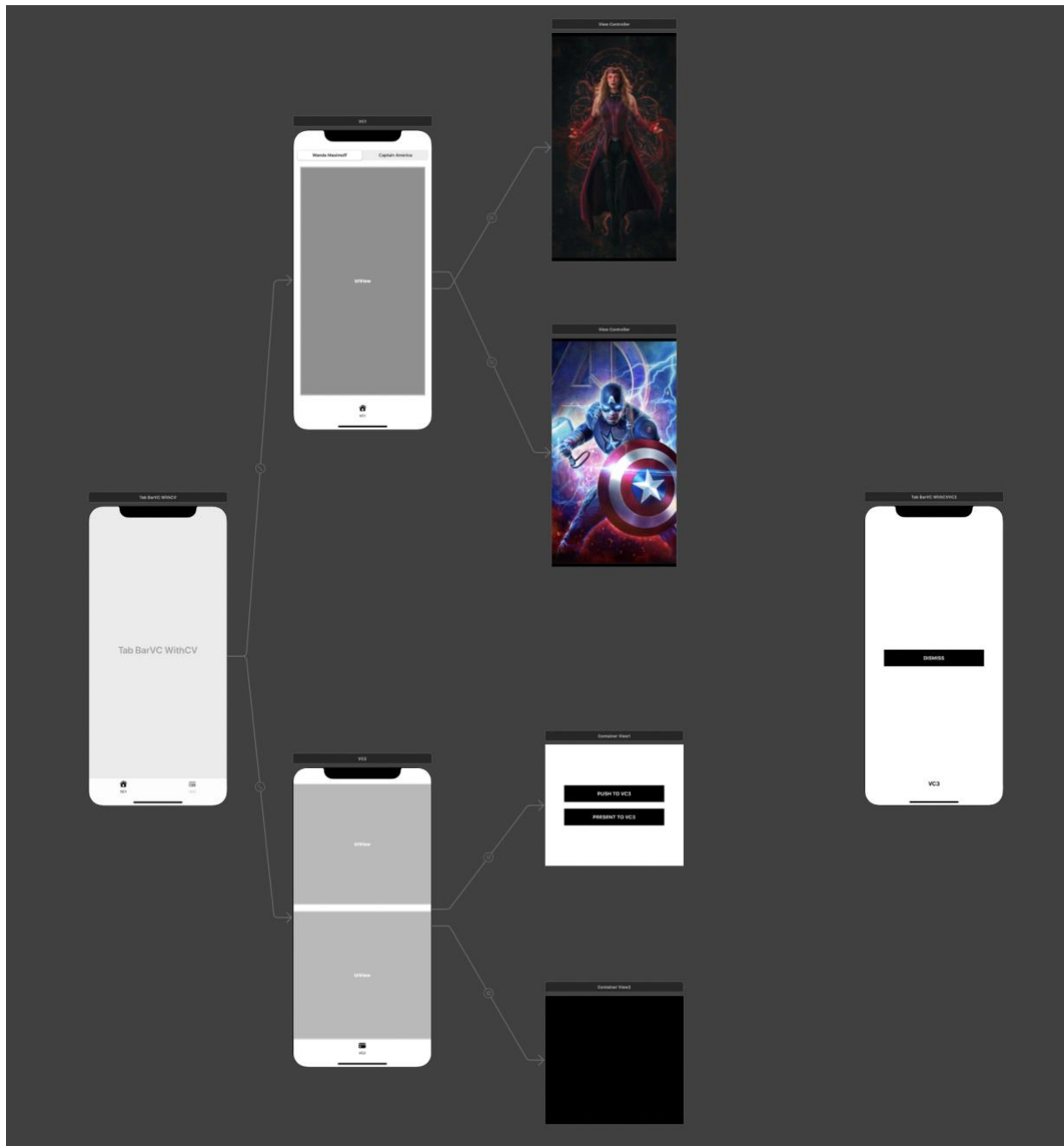
```
@IBAction func btnPresentVC3Click(_ sender: Any) {  
    presentVC(storyboard: "TabBarVCWithUIVCSB", identifier: "TabBarVCWithUIVC3")  
}
```

- In TabBarVCWithUIVC3, on **btnDismissClick** we dismiss to TabBarVCWithUIVC2

```
@IBAction func btnDismissClick(_ sender: Any) {  
    self.presentingViewController?.dismiss(animated: true)  
}
```

ii. UITabBarController with Container View

- In TabBarVCWithCVSB, add Tab Bar Controller that has VC1 & VC2 that contain Container View



- In TabBarVCWithCVVC, on **segmentedControlClick** we hide show firstView & secondView

```

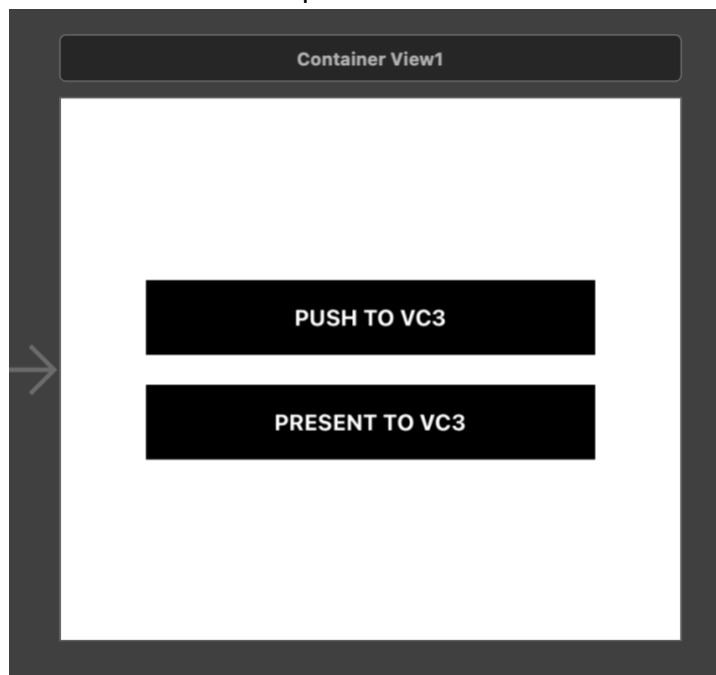
@IBAction func segmentedControlClick(_ sender: UISegmentedControl) {

    if sender.selectedSegmentIndex == 0 {
        firstView.alpha = 1
        secondView.alpha = 0
    } else {
        firstView.alpha = 0
        secondView.alpha = 1
    }

}
}

```

- In ContainerView1, on **btnPushToVC3** we push to TabBarVCWithCVVC3, on **btnPresentToVC3** we present to TabBarVCWithCVVC3



```

@IBAction func btnPushToVC3(_ sender: Any) {
    pushNavigation(storyboard: "TabBarVCWithCVSB", identifier: "TabBarVCWithCVVC3")
}

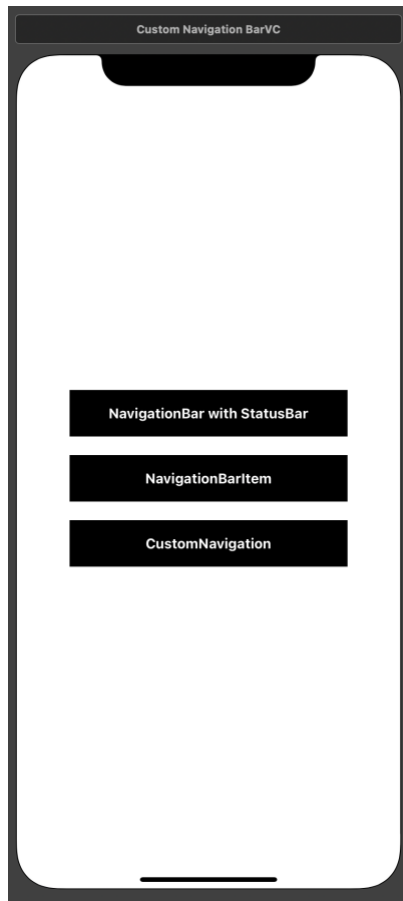
@IBAction func btnPresentToVC3(_ sender: Any) {
    presentVC(storyboard: "TabBarVCWithCVSB", identifier: "TabBarVCWithCVVC3")
}

```

- In TabBarVCWithCVVC3, on **btnDismissClick** we dismiss to TabBarVCWithCVVC2

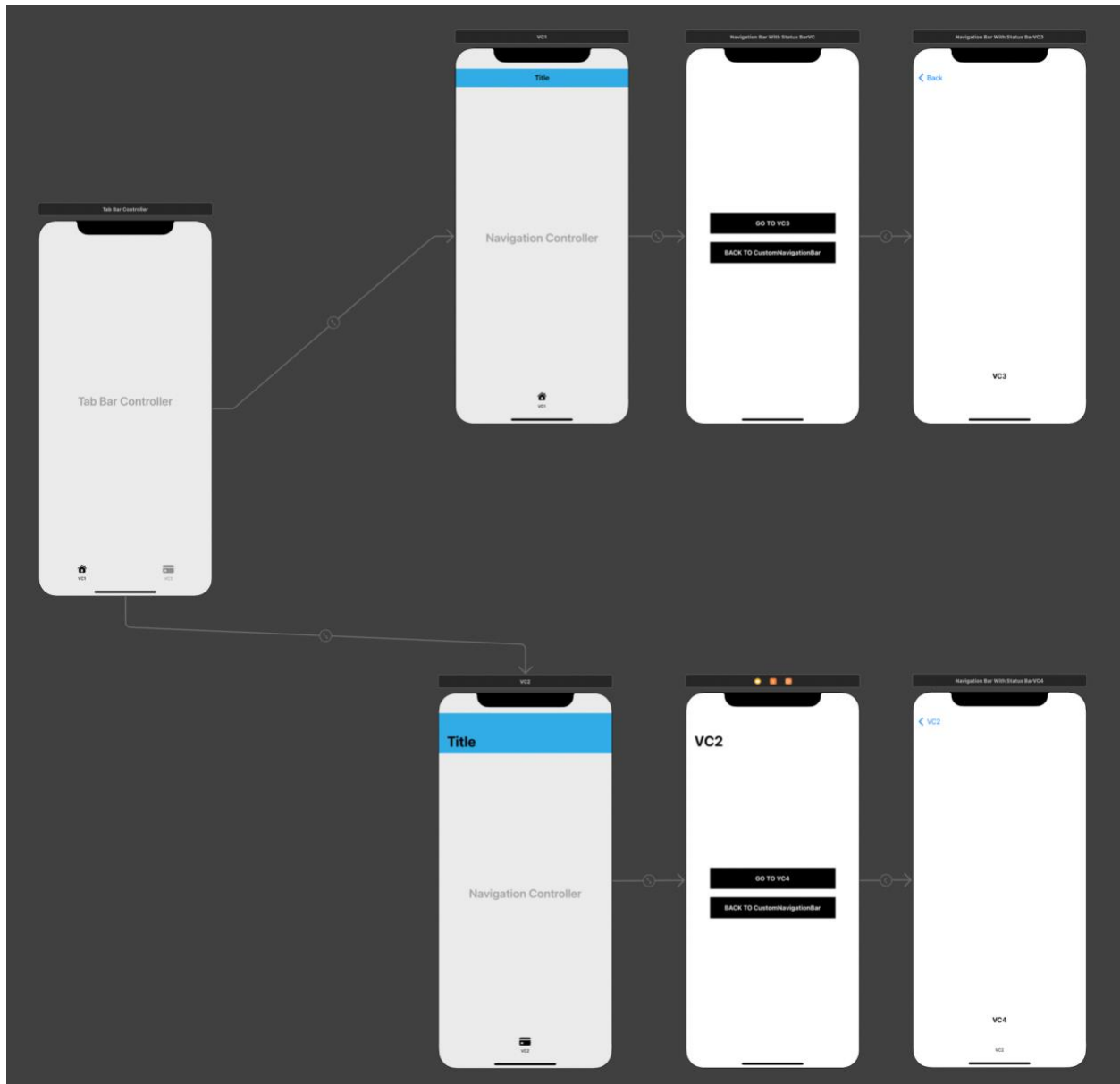
```
@IBAction func btnDismissClick(_ sender: Any) {  
    self.presentingViewController?.dismiss(animated: true)  
}
```

g. Custom NavigationBar



i. NavigationBar with StatusBar: Normal Navigation & Large Title Navigation

- In NavigationBarWithStatusBarSB, add Tab Bar Controller that has VC1 & VC2 that contain VC1 is normal titles & VC2 is large titles



- In **NavigationBarWithStatusBarVC**, on **viewDidLoad** we call function **setUpNavAndStatusBarColor** (status bar has color the same as navigation)

```
func setUpNavAndStatusBarColor() {
    let appearance = UINavigationBarAppearance()
    appearance.configureWithOpaqueBackground()
    appearance.backgroundColor = UIColor.black
    appearance.titleTextAttributes = [.foregroundColor: UIColor.white]

    navigationController?.navigationBar.standardAppearance = appearance;
    navigationController?.navigationBar.scrollEdgeAppearance = navigationController?.navigationBar.standardAppearance

    self.navigationItem.title = "NAV VC1"
    self.tabBarController?.viewControllers?[0].tabBarItem.title = NSLocalizedString("TAB VC1", comment: "")
    self.tabBarController?.viewControllers?[1].tabBarItem.title = NSLocalizedString("TAB VC2", comment: "")
}
```

- In UINavigationControllerWithStatusBarVC2, on **viewDidLoad** we call function **setUpNavAndStatusBarColor** (status bar has color the same as navigation)

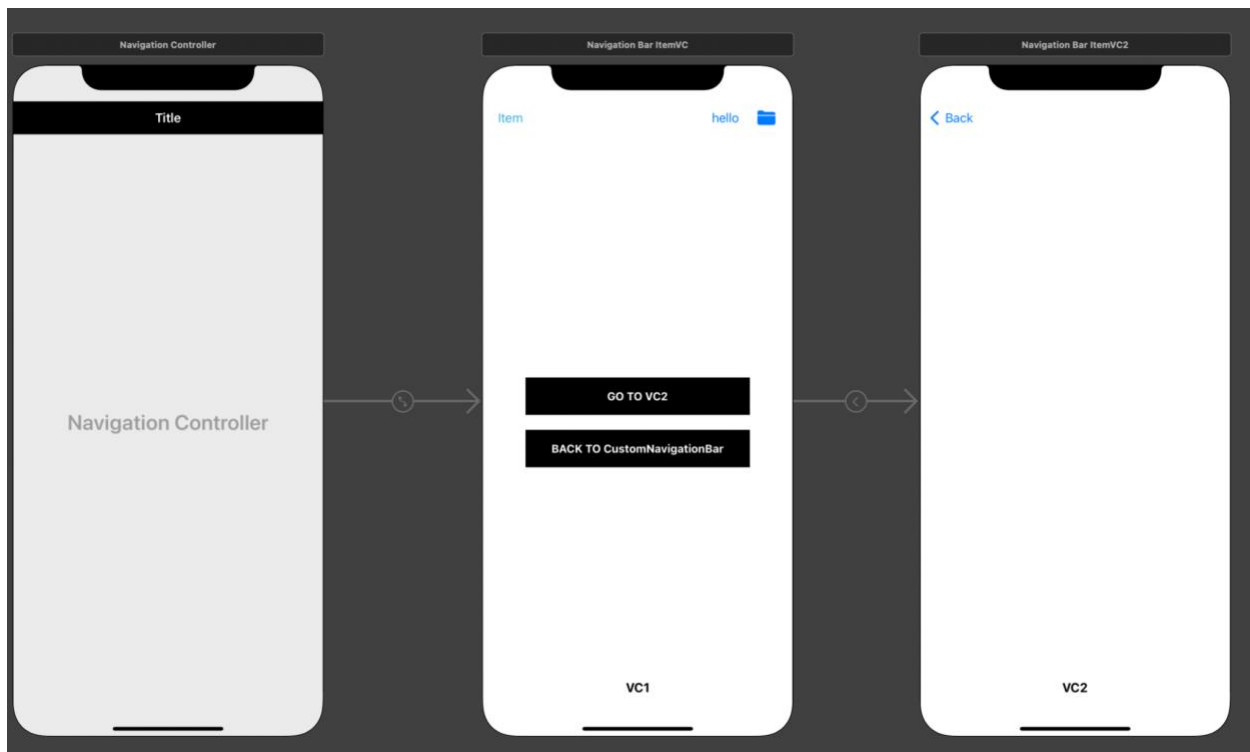
```
func setUpNavAndStatusBarColor() {
    let appearance = UINavigationControllerAppearance()
    appearance.configureWithOpaqueBackground()
    appearance.backgroundColor = UIColor.black
    appearance.largeTitleTextAttributes = [.foregroundColor: UIColor.white]

    navigationController?.navigationBar.standardAppearance = appearance;
    navigationController?.navigationBar.scrollEdgeAppearance = navigationController?.navigationBar.standardAppearance

    self.navigationItem.title = "NAV VC2"
}
```

ii. UINavigationControllerItem

- In UINavigationControllerItemSB, we have VC1 that contain Bar Button Item such as Item, hello and folder.fill



- In UINavigationController, we connect @IBAction

```
//MARK: - @IBAction
@IBAction func barItemHelloClick(_ sender: Any) {
    print("==> barItemHelloClick")
}

@IBAction func barItemClick(_ sender: Any) {
    print("==> barItemClick")
}
```

- In UINavigationController2, on **viewDidLoad** we call function
setUpRightBarButtonItems (add rightBarButtonItems)

```
//MARK: - Function
func setUpRightBarButtonItems() {

    let add = UIBarButtonItem(barButtonSystemItem: .add, target: self, action: #selector(addClick))
    let hello = UIBarButtonItem(title: "hello", style: .plain, target: self, action: #selector(helloClick))

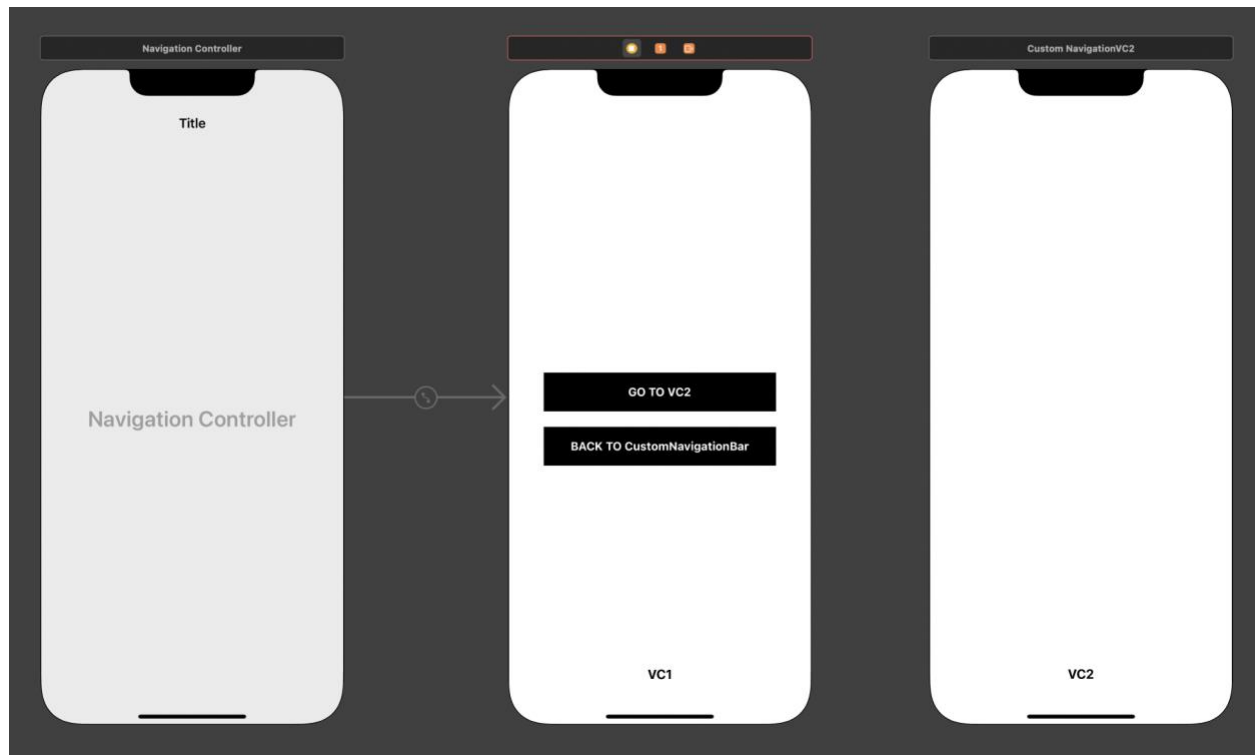
    navigationItem.rightBarButtonItem = [add, hello]
}

//MARK: - @objc
@objc func addClick() {
    print("==> addClick")
}

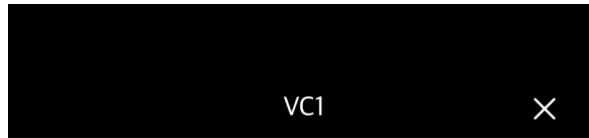
@objc func helloClick() {
    print("==> helloClick")
}
```

iii. CustomNavigation

- In CustomNavigationSB, we have VC1 & VC2



- In CustomNavigationVC, on **viewWillAppear** we call function `setNavigationBarStyle` for set title, titleColor, barTintColor, rightBarButtonImage & action of rightBarButtonImage



- In CustomNavigationVC, when click **goToCustomNavigationVC2** it will go to CustomNavigationVC2 & get completionData when **backward**

```
@IBAction func goToCustomNavigationVC2(_ sender: Any) {  
    let vc = UIStoryboard(name: "CustomNavigationSB", bundle: nil).instantiateViewController(withIdentifier: "CustomNavigationVC2") as! CustomNavigationVC2  
  
    vc.completionData = { data in  
        print("==> data", data)  
    }  
  
    self.navigationController?.pushViewController(vc, animated: true)  
}
```

- In CustomNavigationVC2, on **viewWillAppear** we call function **setNavigationBarStyle** for set title, titleColor, barTintColor, array of rightBarButtonImage & action of rightBarButtonImage

```
override func viewWillAppear(_ animated: Bool) {
    var rightBarImages = [UIImage]()
    rightBarImages.append(UIImage(named: "main_close_icon")!)
    rightBarImages.append(UIImage(systemName: "arrowshape.turn.up.backward.2")!)
    rightBarImages.append(UIImage(systemName: "arrowshape.turn.up.backward.fill")!)

    self.setNavigationBarStyle(isHidden: false, animated: animated, title: "VC2", titleColor: .white, barTintColor: .systemPink, leftBarButtonImage:
    UIImage(named: "main_back_icon"), leftSelector: #selector(backClicked), rightBarButtonImage: rightBarImages, selectors :
    [#selector(rightCloseClicked), #selector(backward), #selector(backward)])
}
```

- In CustomNavigationVC2, when backward it will pop & pass data to CustomNavigationVC

```
@objc private func backward() {
    self.popOrDismissVC(animated: true) {
        self.completionData("jisoo")
    }
}
```

GitHub

- <https://github.com/lymanny/UINavigationController-Sample-Project.git>