

The logo consists of a white rounded square border containing the text 'ios' and 'DeCal' in white. 'ios' is in a bold, lowercase sans-serif font, and 'DeCal' is in a regular, title-case sans-serif font.

ios
DeCal

lecture 4

Multiview Applications

cs198-001 : spring 2018

announcements

- Pokedex lab this week
- Snapchat Clone Part 1 will be released this week
- office hours after every lecture in class & Thursdays from 3-4 (location TBA)
- only excusing absences for midterm conflicts - no review sessions

view controller lifecycle

view controller lifecycle

viewDidLoad()

viewWillAppear()

viewDidAppear()

viewWillDisappear()

viewDidDisappear()



someViewController.swift

view controller lifecycle

viewDidLoad()

viewWillAppear()

viewDidAppear()

viewWillDisappear()

viewDidDisappear()



someViewController.swift

automatically called when the view controller
loaded completely in the memory

view controller lifecycle

viewDidLoad()

viewWillAppear()

viewDidAppear()

viewWillDisappear()

viewDidDisappear()



someViewController.swift

called when the view controller is about to be
added to the view hierarchy

view controller lifecycle

viewDidLoad()

viewWillAppear()

viewDidAppear()

viewWillDisappear()

viewDidDisappear()



someViewController.swift

called when the view controller was added to the
view hierarchy

view controller lifecycle

viewDidLoad()

viewWillAppear()

viewDidAppear()

viewWillDisappear()

viewDidDisappear()



someViewController.swift

called when the view controller is about to be removed from the view controller hierarchy

view controller lifecycle

viewDidLoad()

viewWillAppear()

viewDidAppear()

viewWillDisappear()

viewDidDisappear()



someViewController.swift

called when the view controller was removed from
the view controller hierarchy

view controller lifecycle

viewDidLoad()

viewWillAppear()

viewDidAppear()

viewWillDisappear()

viewDidDisappear()



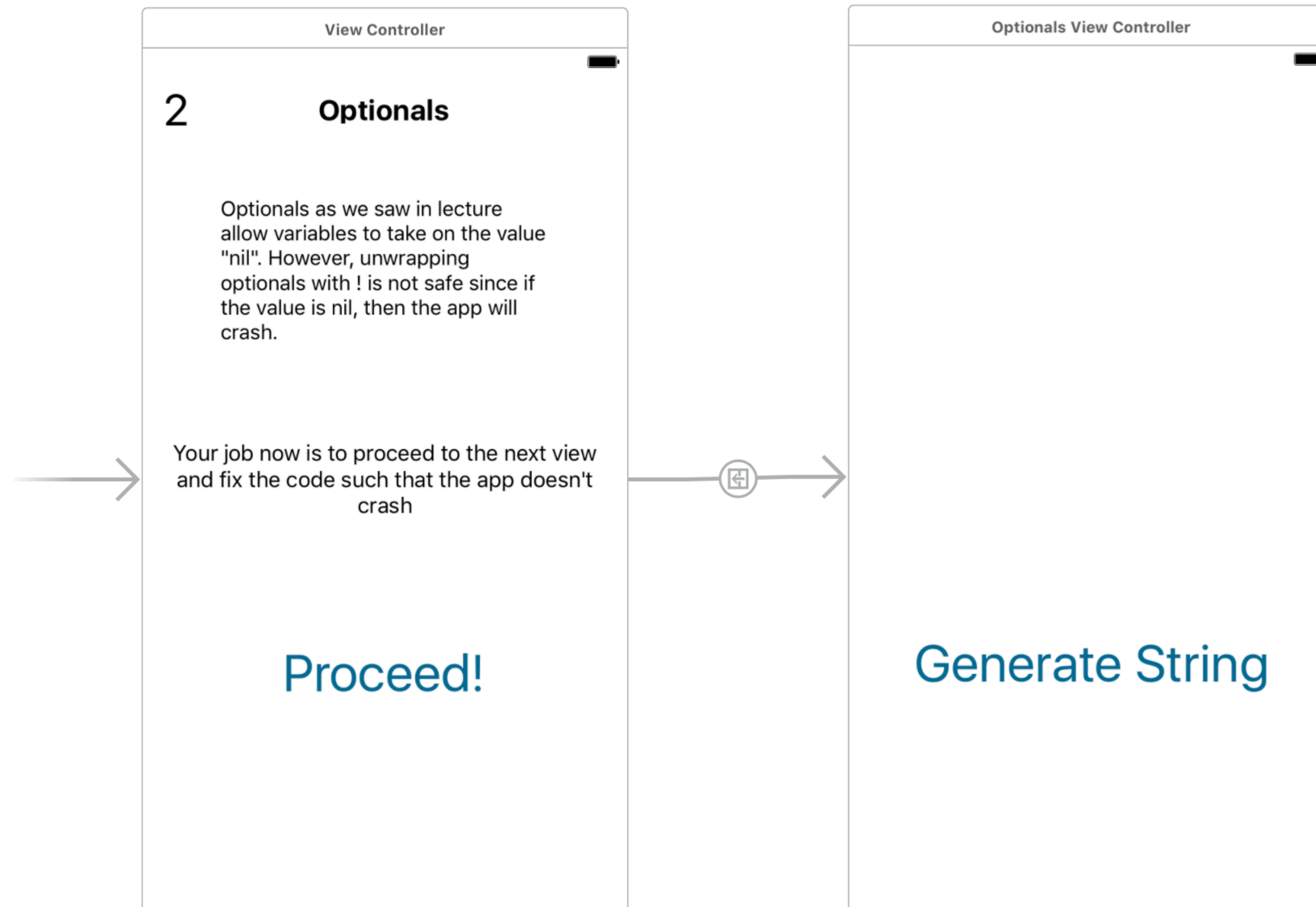
someViewController.swift

didReceiveMemoryWarning

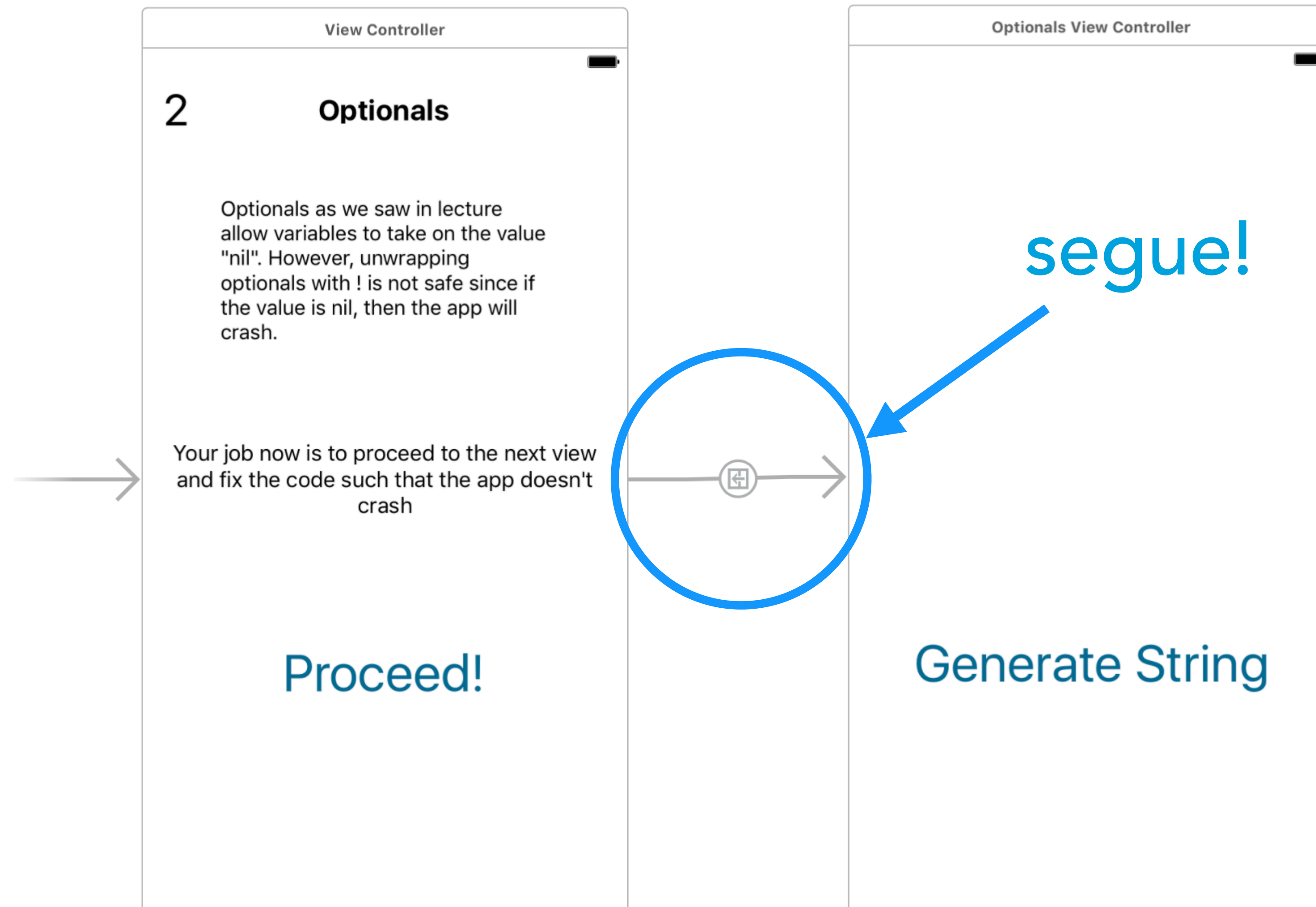
called when available memory is low

multiview applications

lab 1 (multiview example)



lab 1 (multiview example)



multiview applications

Multi-view applications are made up of multiple MVC's stringed together

To communication and transitioning between MVC's, create **segues** between View Controllers in Storyboard

Each time a user triggers a segue, a **new instance** of an MVC will be created (not an old instance).

More on this later!

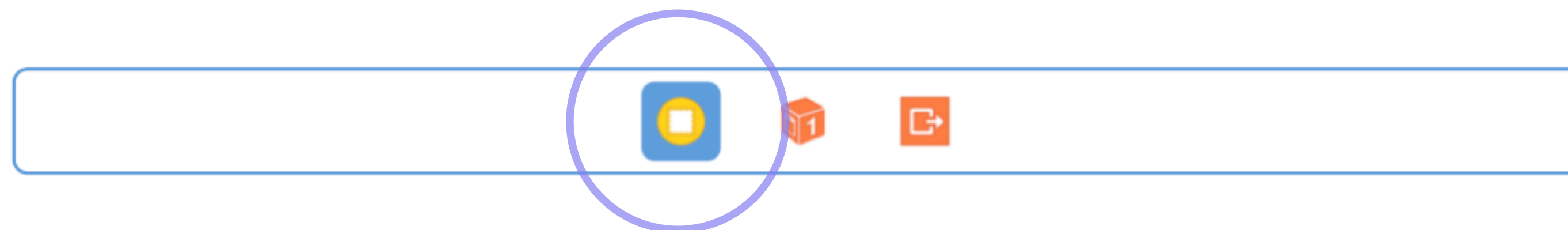
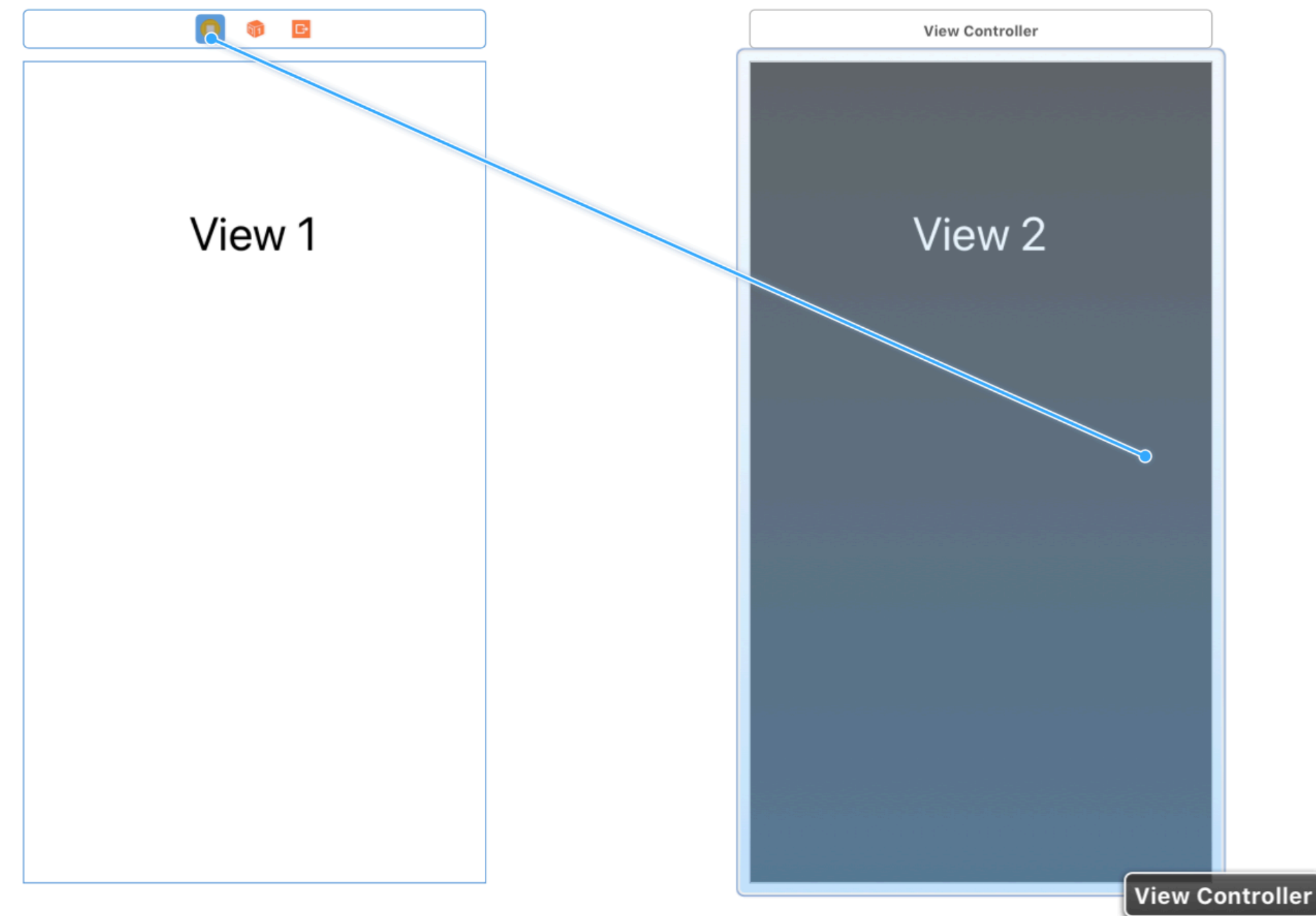
segues

To create a segue between MVC's, you need to:

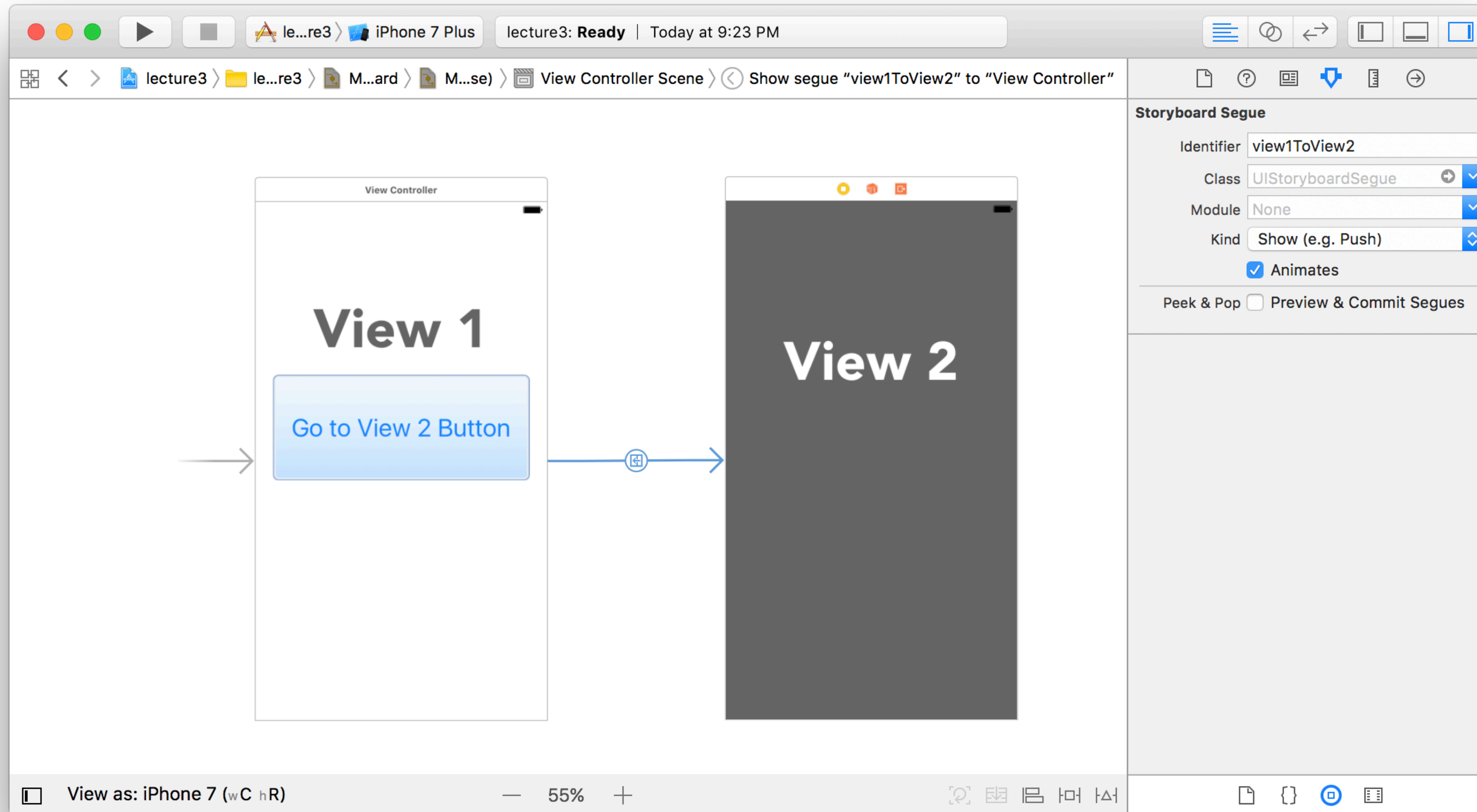
1. create the segue (control + drag)
2. create a segue identifier
3. trigger the segue using `performSegue`
 1. if you need to pass data or do any other “pre-segue” work, use the “`prepareForSegue`” method

segue creation

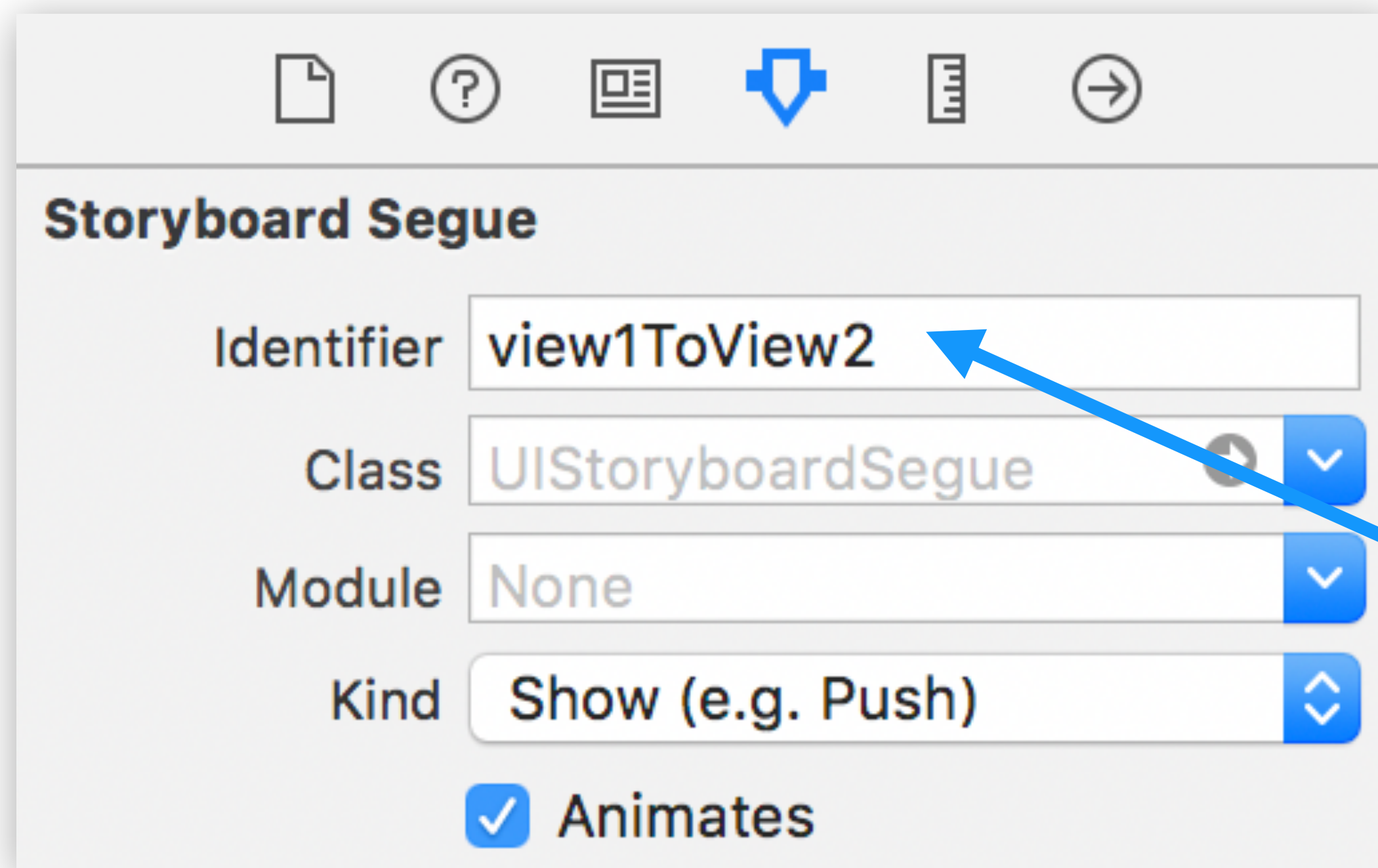
**Step 1: control
drag from your
initial view
controller to the
destination view
controller**



Segue Identification: Tap on a segue in Storyboard, and add a **Identifier** in the **Attributes Inspector**



Segue Identification: Tap on a segue in Storyboard, and add a **Identifier** in the **Attributes Inspector**

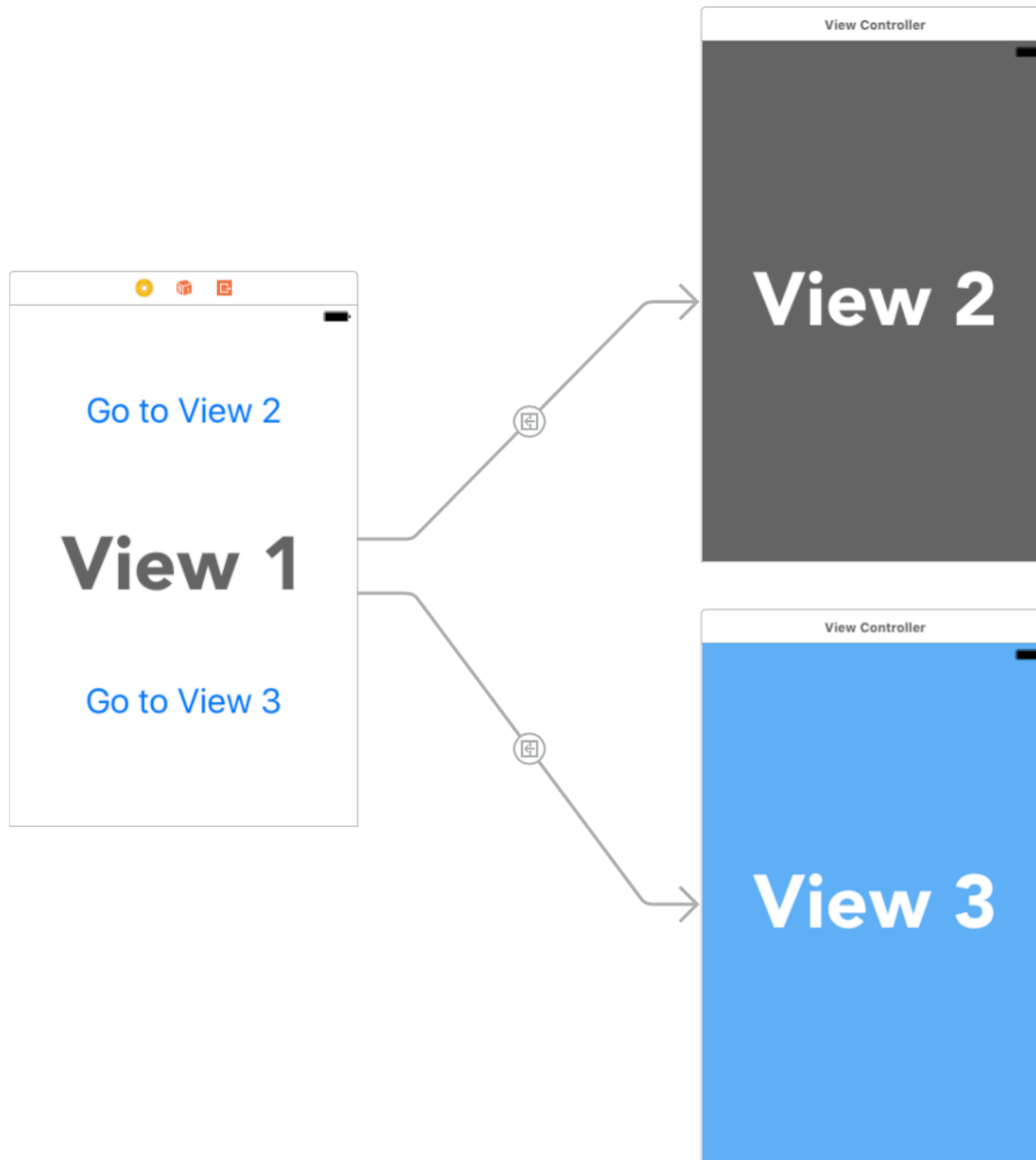


you can use this
identifier to
reference your
segue in code

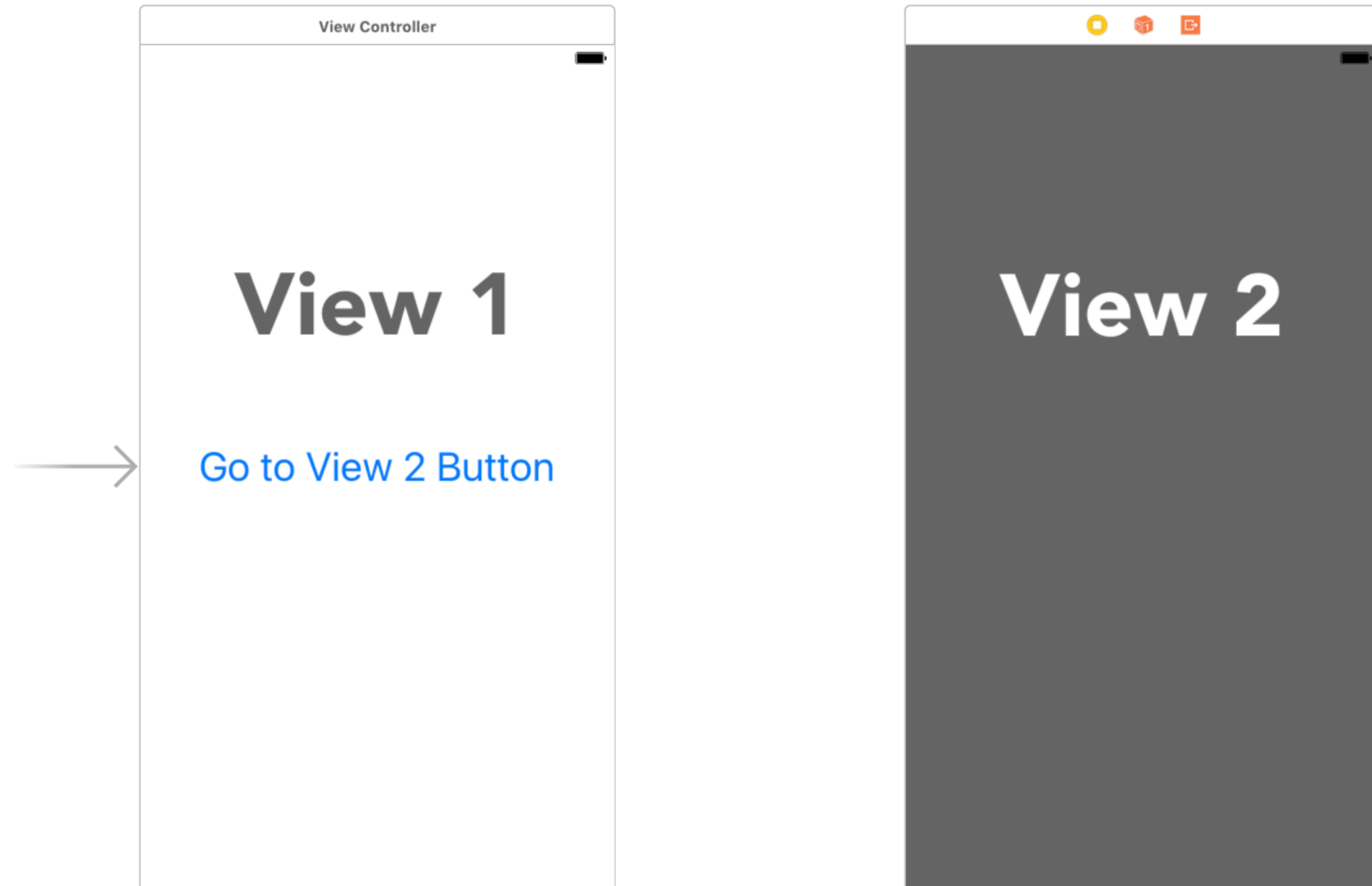
performSegue

Call `performSegue` to trigger the segue created in Storyboard using the identifier you set

```
@IBAction func buttonPressed(sender: UIButton) {  
    performSegue(withIdentifier: "view1ToView2",  
                  sender: sender)  
}
```

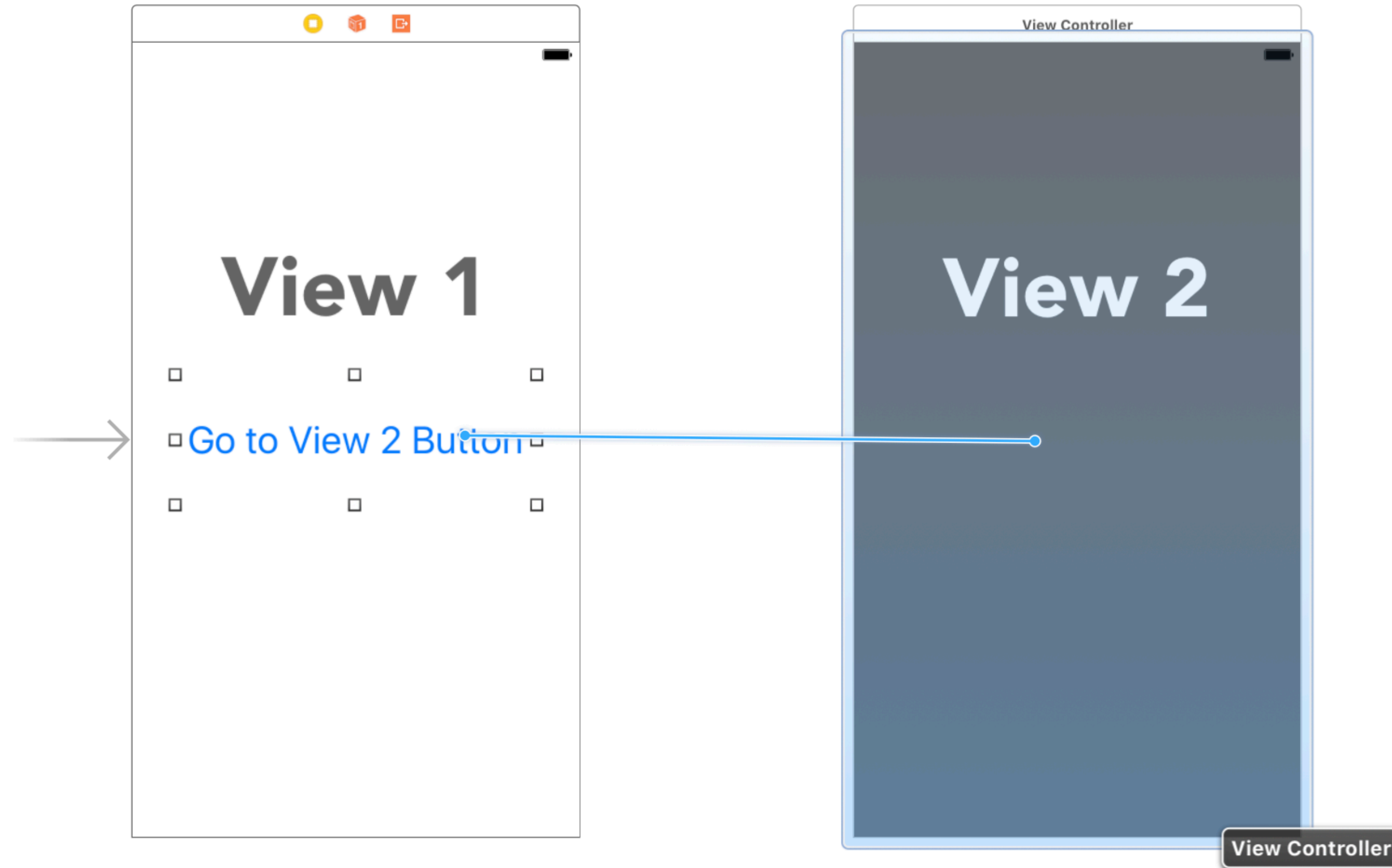


Segue Identifiers
are useful to
distinguish
between which
MVC you are going
to
(one MVC can have
a segue to multiple
others) MVC's).



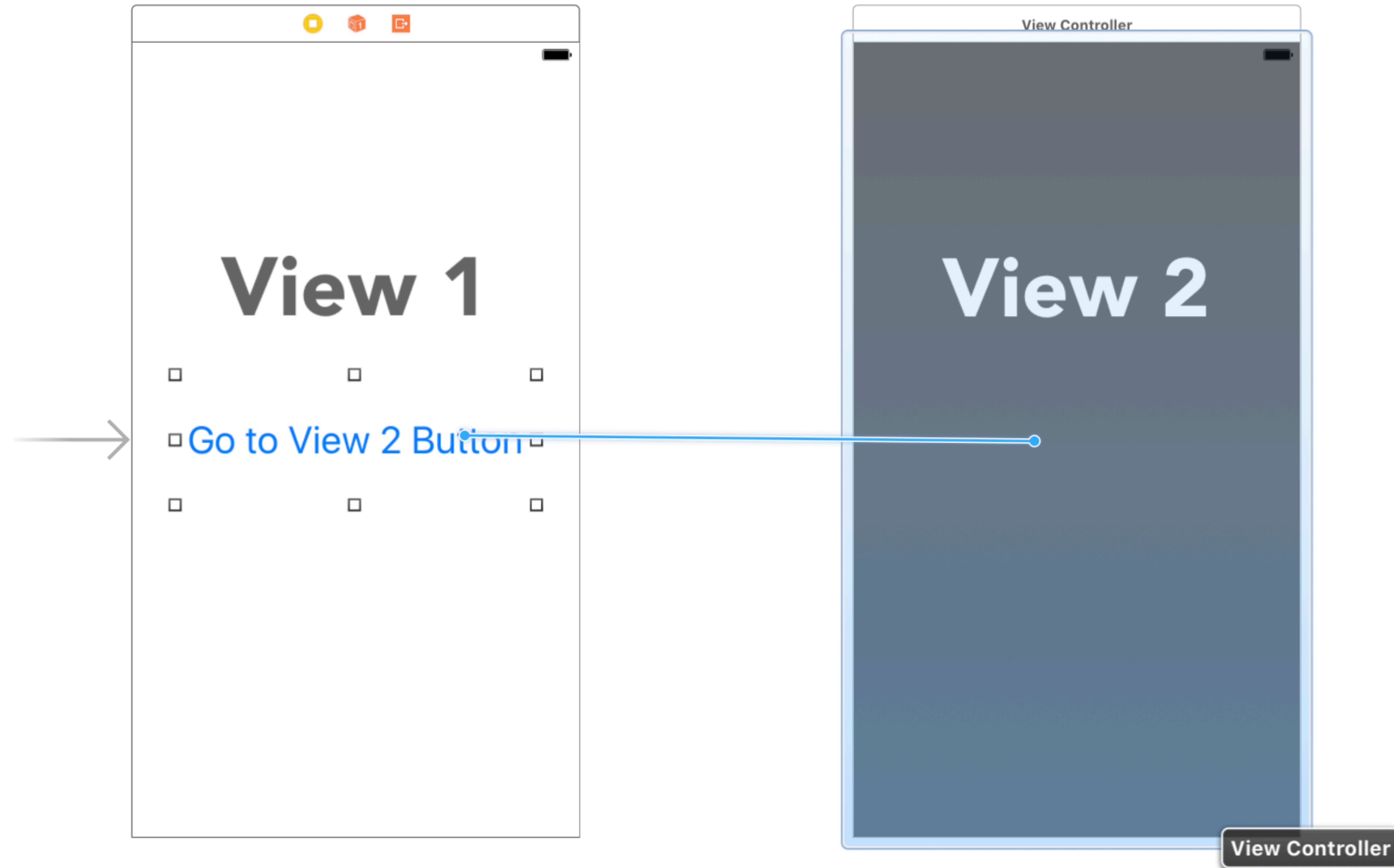
shortcut (if using button or cell)

You can also control + drag from an instigator (typically a button) from one MVC to another MVC



shortcut (if using button or cell)

You can also control + drag from an instigator (typically a button)
from one MVC to another MVC



shortcut (if using button or cell)

this eliminates the need to call "performSegue" in your code

passing data between MVC's

When the user taps on a button that triggers a segue, the MVC they will transition to will be a **new instance of that MVC**.

passing data between MVC's

When the user taps on a button that triggers a segue, the MVC they will transition to will be a **new instance of that MVC**.

Therefore, we need to pass data from the first MVC to the new MVC **before** the segue takes place.

How do we do this? —> `prepareForSegue`

prepareForSegue

```
func prepare(for segue: UIStoryboardSegue, sender: Any?) {  
    if let identifier = segue.identifier {  
        if identifier == "goToMainView" {  
            if let dest = segue.destination as? MainViewController {  
                dest.somePropertyOnMainView = "hi!"  
                dest.someMainViewSetupFunc()  
            }  
        }  
        else if identifier == "goToSettings" {  
            if let dest = segue.destination as? SettingsViewController {  
                // do stuff in the settingsVC before it loads  
            }  
        }  
    }  
}
```

prepareForSegue

```
func prepare(for segue: UIStoryboardSegue, sender: Any?) {  
    if let identifier = segue.identifier {  
        if identifier == "goToMainView" {  
            if let dest = segue.destination as? MainViewController {  
                dest.somePropertyOnMainView = "hi!"  
                dest.someMainViewSetupFunc()  
            }  
        }  
        else if identifier == "goToSettings" {  
            if let dest = segue.destination as? SettingsViewController {  
                // do stuff in the settingsVC before it loads  
            }  
        }  
    }  
}
```

segue : The segue just instigated
sender : The segue's instigator (usually a button)

prepareForSegue

```
func prepare(for segue: UIStoryboardSegue, sender: Any?) {  
    if let identifier = segue.identifier {  
        if identifier == "goToMainView" {  
            if let dest = segue.destination as? MainViewController {  
                dest.somePropertyOnMainView = "hi!"  
                dest.someMainViewSetupFunc()  
            }  
        }  
        else if identifier == "goToSettings" {  
            if let dest = segue.destination as? SettingsViewController {  
                // do stuff in the settingsVC before it loads  
            }  
        }  
    }  
}
```

First, get the identifier we created in Storyboard by
accessing segue.**identifier**

prepareForSegue

```
func prepare(for segue: UIStoryboardSegue, sender: Any?) {  
    if let identifier = segue.identifier {  
        if identifier == "goToMainView" {  
            if let dest = segue.destination as? MainViewController {  
                dest.somePropertyOnMainView = "hi!"  
                dest.someMainViewSetupFunc()  
            }  
        }  
        else if identifier == "goToSettings" {  
            if let dest = segue.destination as? SettingsViewController {  
                // do stuff in the settingsVC before it loads  
            }  
        }  
    }  
}
```

If the identifier is **"goToMainView"**, we know we are heading to the MainViewController

prepareForSegue

```
func prepare(for segue: UIStoryboardSegue, sender: Any?) {  
    if let identifier = segue.identifier {  
        if identifier == "goToMainView" {  
            if let dest = segue.destination as? MainViewController {  
                dest.somePropertyOnMainView = "hi!"  
                dest.someMainViewSetupFunc()  
            }  
        }  
        else if identifier == "goToSettings" {  
            if let dest = segue.destination as? SettingsViewController {  
                // do stuff in the settingsVC before it loads  
            }  
        }  
    }  
}
```

We can get a reference to the destination View Controller by accessing segue.destination

prepareForSegue

```
func prepare(for segue: UIStoryboardSegue, sender: Any?) {  
    if let identifier = segue.identifier {  
        if identifier == "goToMainView" {  
            if let dest = segue.destination as? MainViewController {  
                dest.somePropertyOnMainView = "hi!"  
                dest.someMainViewSetupFunc()  
            }  
        }  
        else if identifier == "goToSettings" {  
            if let dest = segue.destination as? SettingsViewController {  
                // do stuff in the settingsVC before it loads  
            }  
        }  
    }  
}
```

`type(segue.destination)` is `ViewController`,
we need to cast it as a `MainViewController`

prepareForSegue

```
func prepare(for segue: UIStoryboardSegue, sender: Any?) {  
    if let identifier = segue.identifier {  
        if identifier == "goToMainView" {  
            if let dest = segue.destination as? MainViewController {  
                dest.somePropertyOnMainView = "hi!"  
                dest.someMainViewSetupFunc()  
            }  
        }  
        else if identifier == "goToSettings" {  
            if let dest = segue.destination as? SettingsViewController {  
                // do stuff in the settingsVC before it loads  
            }  
        }  
    }  
}
```

Now we can set properties / call methods in the MainViewController before it's view loads

prepareForSegue

```
func prepare(for segue: UIStoryboardSegue, sender: Any?) {  
    if let identifier = segue.identifier {  
        if identifier == "goToMainView" {  
            if let dest = segue.destination as? MainViewController {  
                dest.somePropertyOnMainView = "hi!"  
                dest.someMainViewSetupFunc()  
            }  
        }  
        else if identifier == "goToSettings" {  
            if let dest = segue.destination as? SettingsViewController {  
                // do stuff in the settingsVC before it loads  
            }  
        }  
    }  
}
```

Since MVC's can segue to multiple other MVC's,
`segue.identifier` can take on different values

IMPORTANT

- `UIStoryboardSegue.destination` view controller has been initialized.

IMPORTANT

- `UIStoryboardSegue.destination` view controller has been initialized.
- that view controller's view has NOT been.

IMPORTANT

- **UIStoryboardSegue.destination** view controller has been initialized.
- that view controller's view has NOT been.
 - which means *viewDidLoad()* has not been called.

IMPORTANT

- `UIStoryboardSegue.destination` view controller has been initialized.
- that view controller's view has NOT been.
 - which means *viewDidLoad()* has not been called.
 - which also means storyboard connections have not been created!

Incorrect

```
func prepare(for segue: UIStoryboardSegue, sender: Any?) {  
    if let identifier = segue.identifier {  
        if identifier == "goToMainView" {  
            if let dest = segue.destination as? MainViewController {  
                dest.label.text = "hi!"  
            }  
        }  
    }  
}
```

Incorrect

```
func prepare(for segue: UIStoryboardSegue, sender: Any?) {  
    if let identifier = segue.identifier {  
        if identifier == "goToMainView" {  
            if let dest = segue.destination as? MainViewController {  
                dest.label.text = "hi!"  
            }  
        }  
    }  
}
```

Incorrect

```
func prepare(for segue: UIStoryboardSegue, sender: Any?) {  
    if let identifier = segue.identifier {  
        if identifier == "goToMainView" {  
            if let dest = segue.destination as? MainViewController {  
                dest.label.text = "hi!"  
            }  
        }  
    }  
}
```


Correct

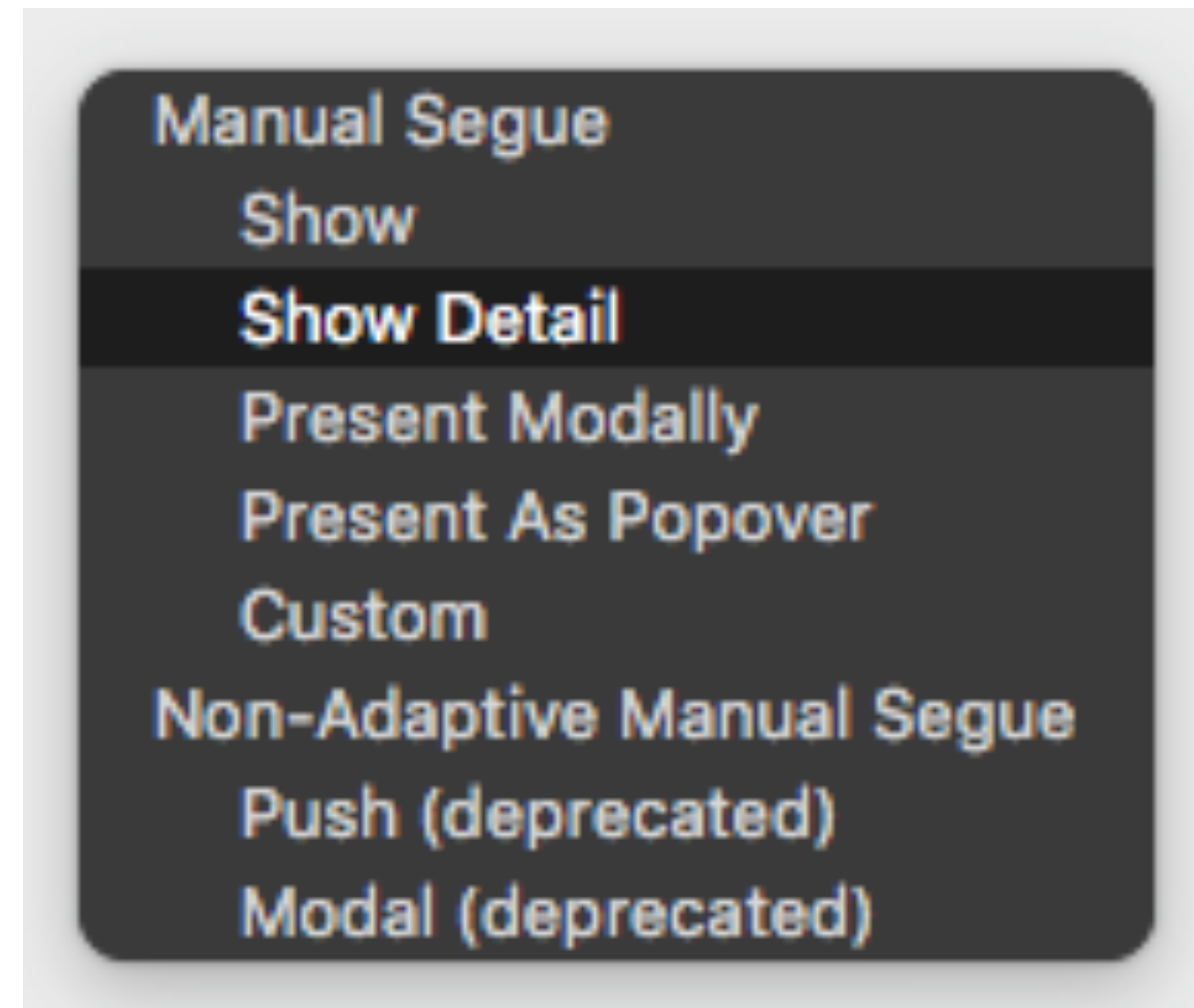
```
func prepare(for segue: UIStoryboardSegue, sender: Any?) {  
    if let identifier = segue.identifier {  
        if identifier == "goToMainView" {  
            if let dest = segue.destination as? MainViewController {  
                dest.labelText = "hi!"  
            }  
        }  
    }  
    ...  
  
    // MainViewController.swift  
    var labelText: String?  
  
    func viewDidLoad() {  
        label.text = labelText  
    }  
}
```

segue types

when control dragging to create a segue, drop down will show up, asking you to specify the segue type

available segue options

- show
- show detail
- present modally
- present as popover
- custom

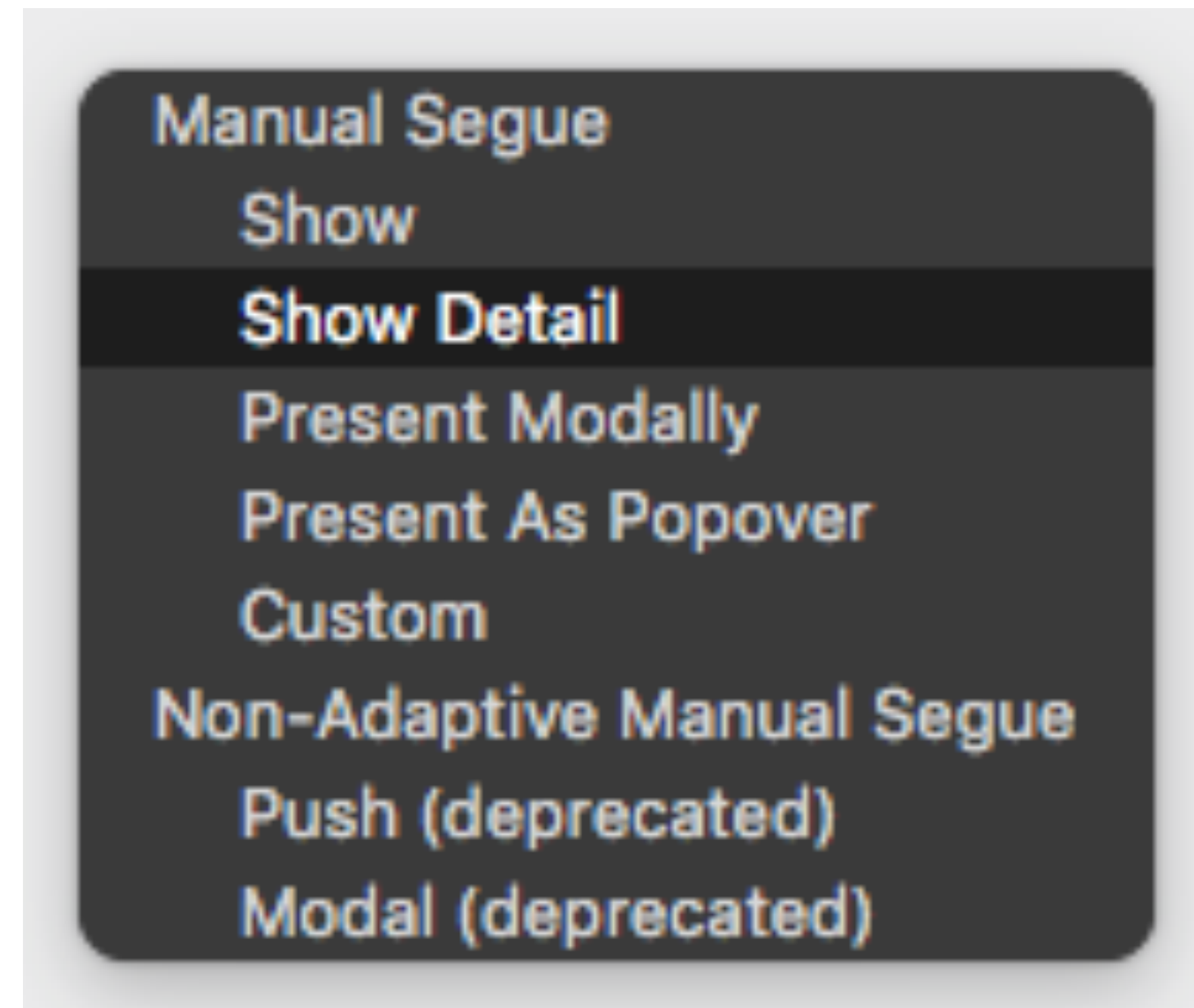


segue types

when control dragging to create a segue, drop down will show up, asking you to specify the segue type

available segue options

- show
- show detail
- present modally
- present as popover
- custom

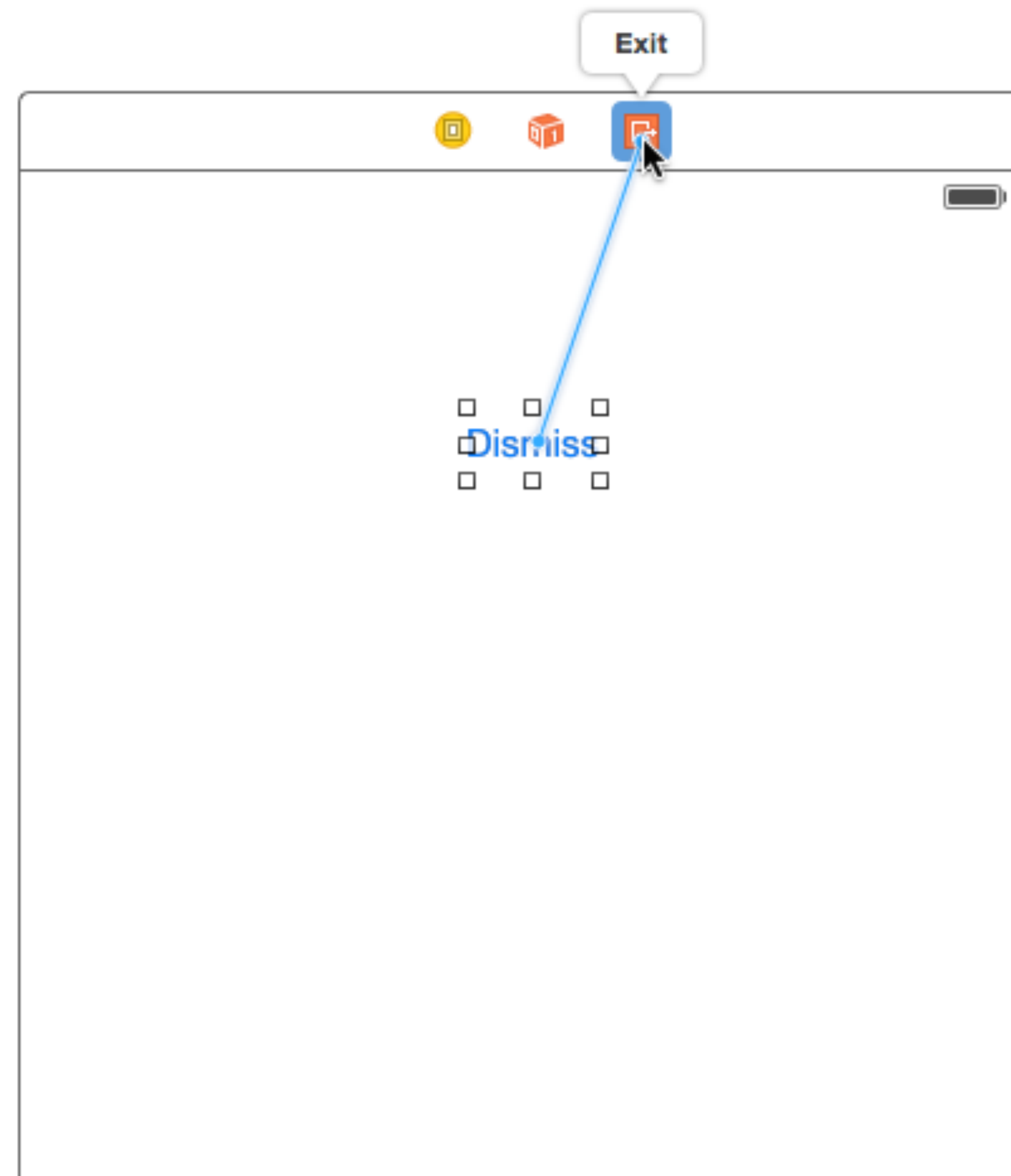


great explanation of the difference between segue types here: <https://stackoverflow.com/questions/25966215/whats-the-difference-between-all-the-selection-segues>

unwind segues

to dismiss a view controller that you've presented, create an unwind segue

1. define an unwind @IBAction in the view controller you want to return to after dismissal
2. create segue by control dragging to the "exit" in storyboard
3. select the IBAction you created

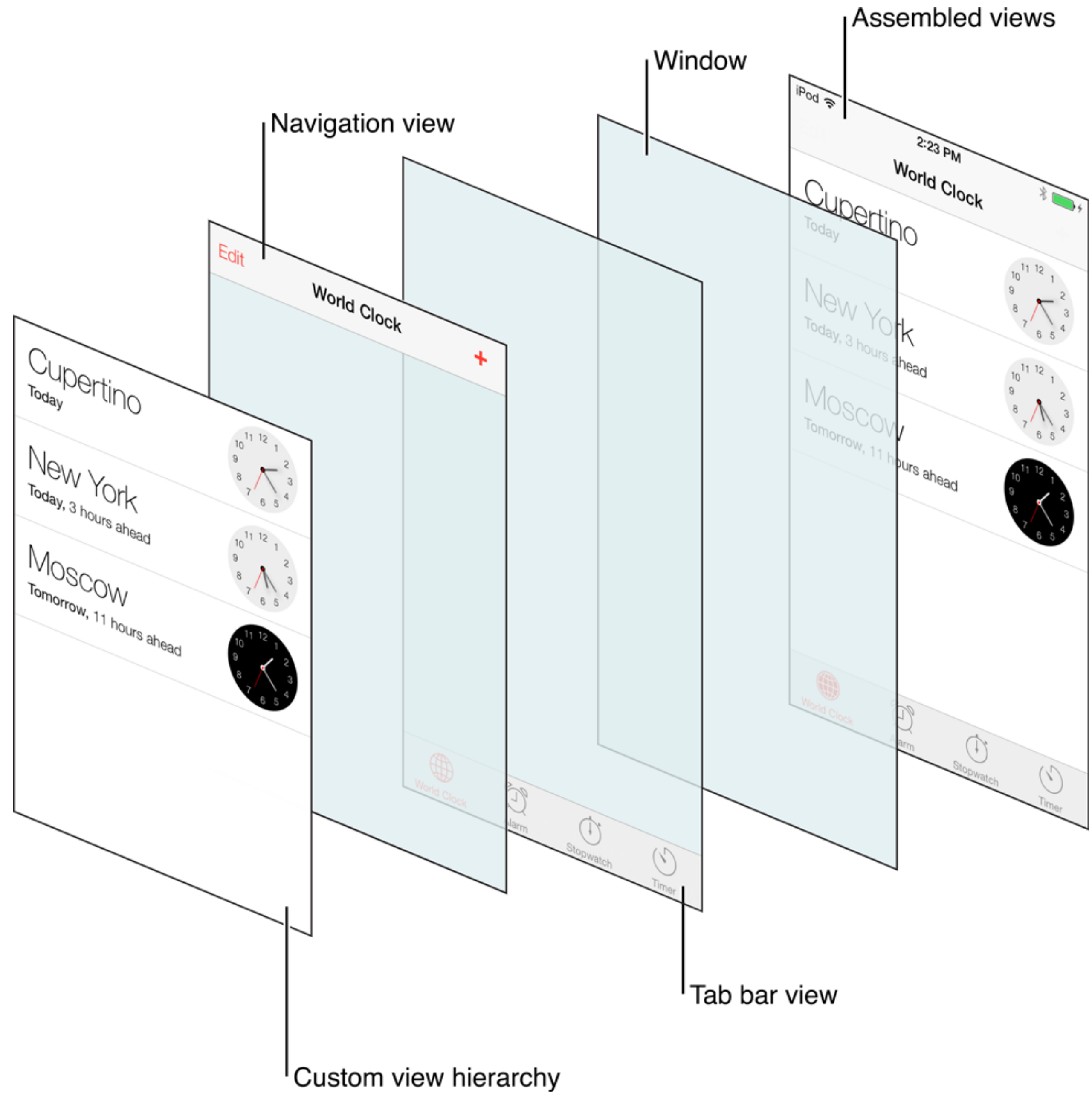


check in

**After:
segue demo**

Available on course site

Navigation + Tab bars

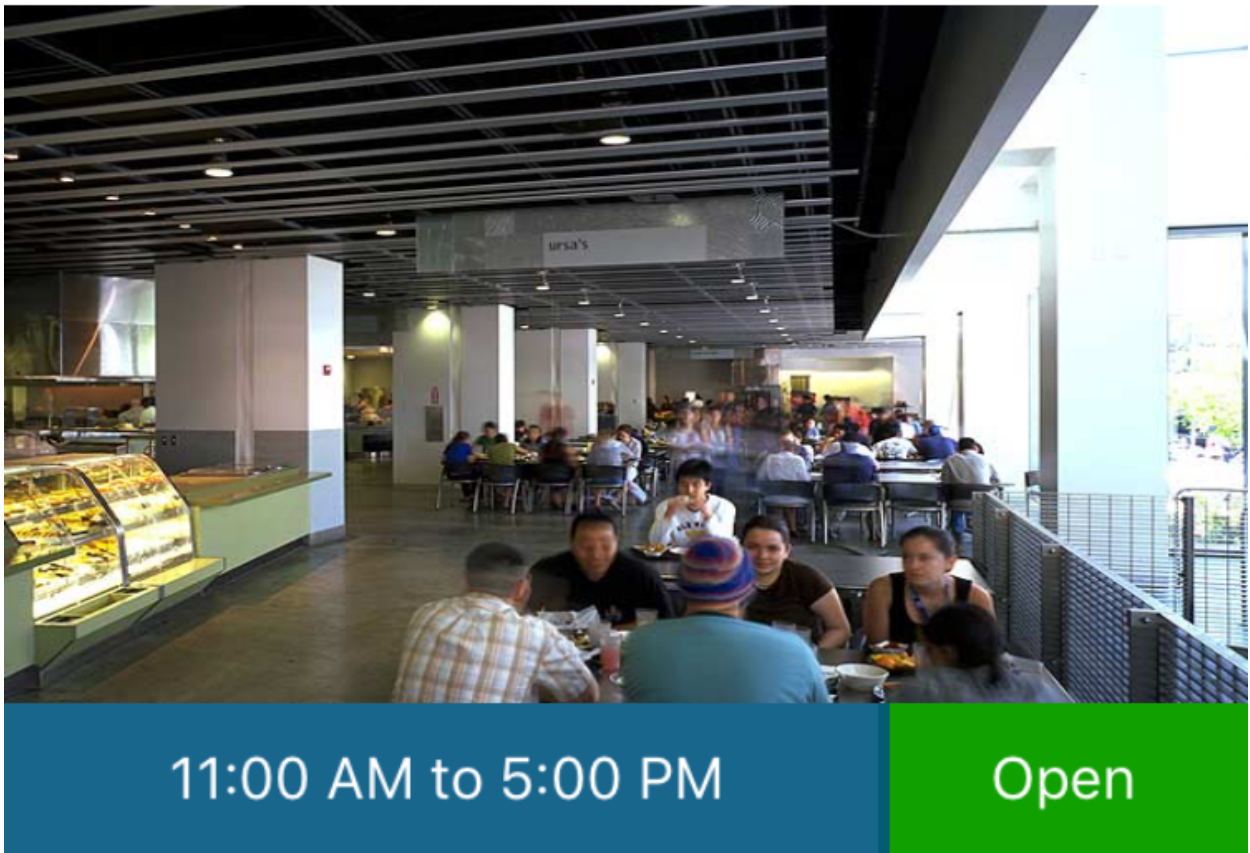


BREAKFAST

LUNCH

DINNER

NIGHT



Baked Potato Bar

♥

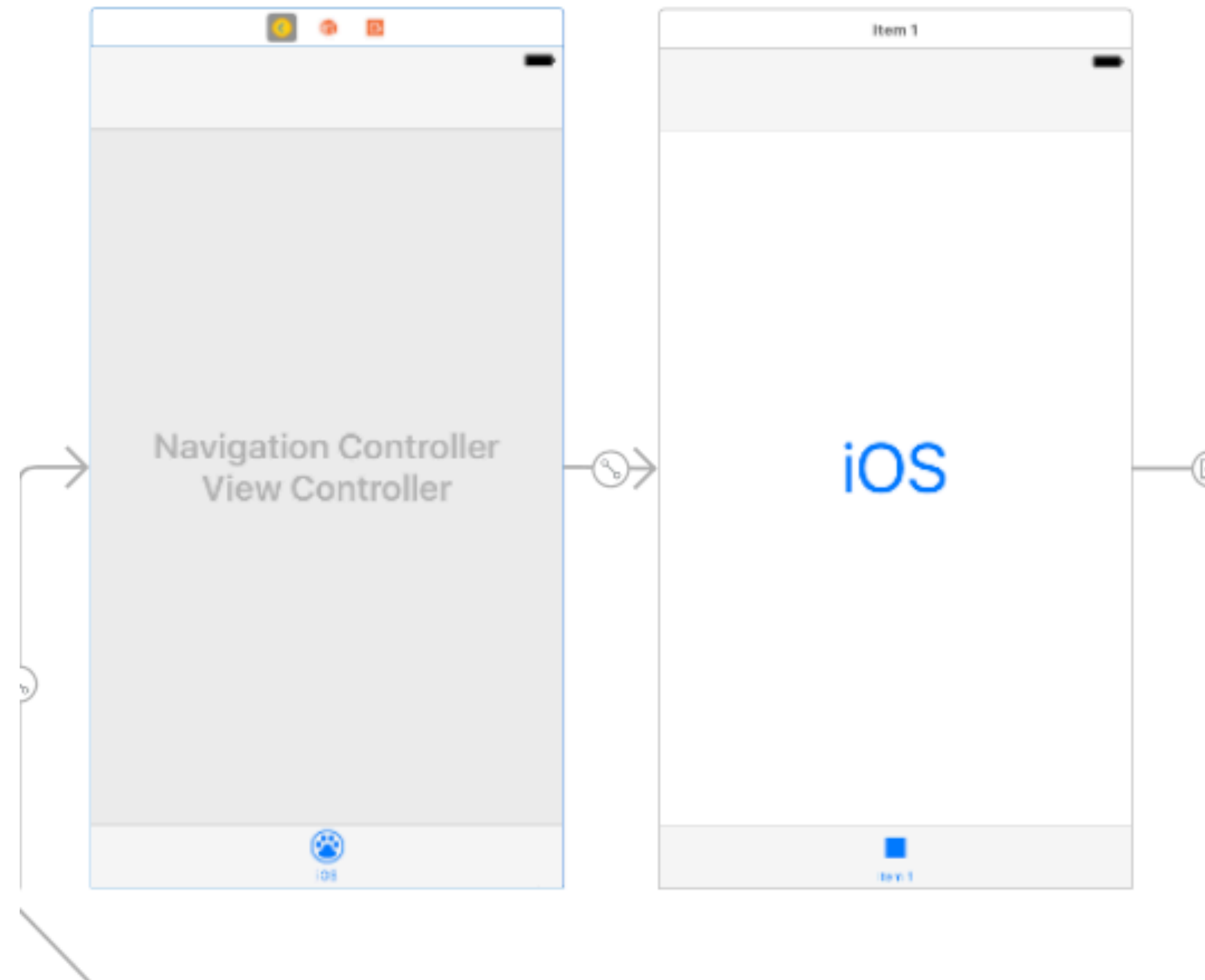
Aurora Sauce

♥

navigation controllers

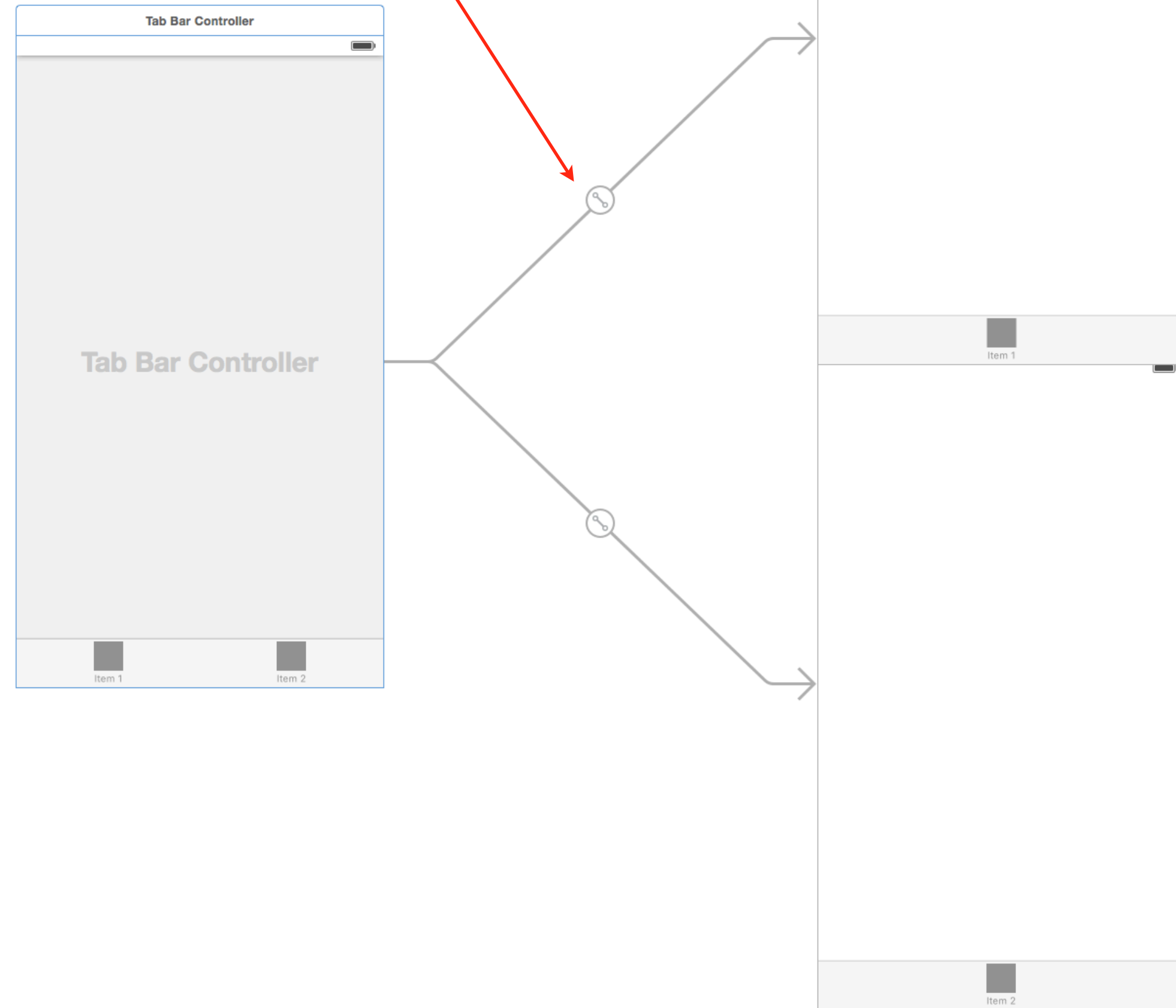
1. pull out a navigation controller from the object library
2. set a “root view controller”
3. add view controllers to your navigation stack using “show” segues
4. if needed, create “unwind segues” to dismiss view controllers

Navigation Bars

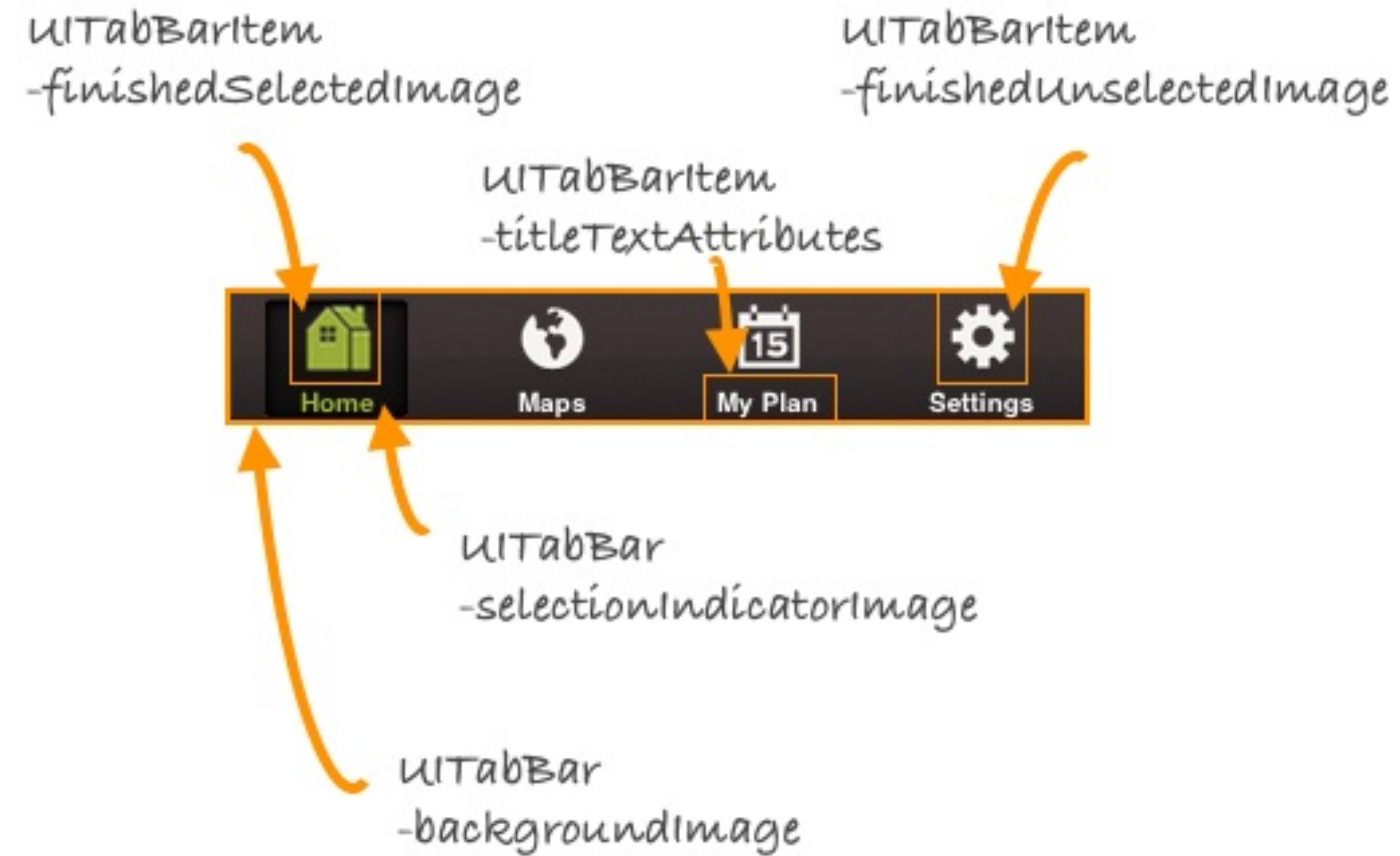


Tab Bars

Relationship Segue



Tab Bars : Customization



Tab Bars : Customization

```
class AppDelegate: UIResponder, UIApplicationDelegate {  
  
    var window: UIWindow?  
  
    func application(_ application: UIApplication,  
        didFinishLaunchingWithOptions launchOptions:  
        [UIApplicationLaunchOptionsKey: Any]?) -> Bool {  
  
        UITabBar.appearance().tintColor =  
        UIColor.lightGray//selected tab color  
        UITabBar.appearance().backgroundImage =  
        UIImage(named:"tabbarbg.png")  
        UITabBar.appearance().barTintColor =  
        UIColor.white  
        return true  
    }  
}
```

Navigation Bars : Customization

```
class ViewController: UIViewController

func viewDidLoad() {

    navigationBar.barTintColor =
    UIColor(colorLiteralRed: 51/255, green:
    90/255, blue: 149/255, alpha: 1)

    navigationBar.titleTextAttributes =
    [NSForegroundColorAttributeName:
    UIColor.white]

    navigationBar.tintColor = UIColor.white
}
```