

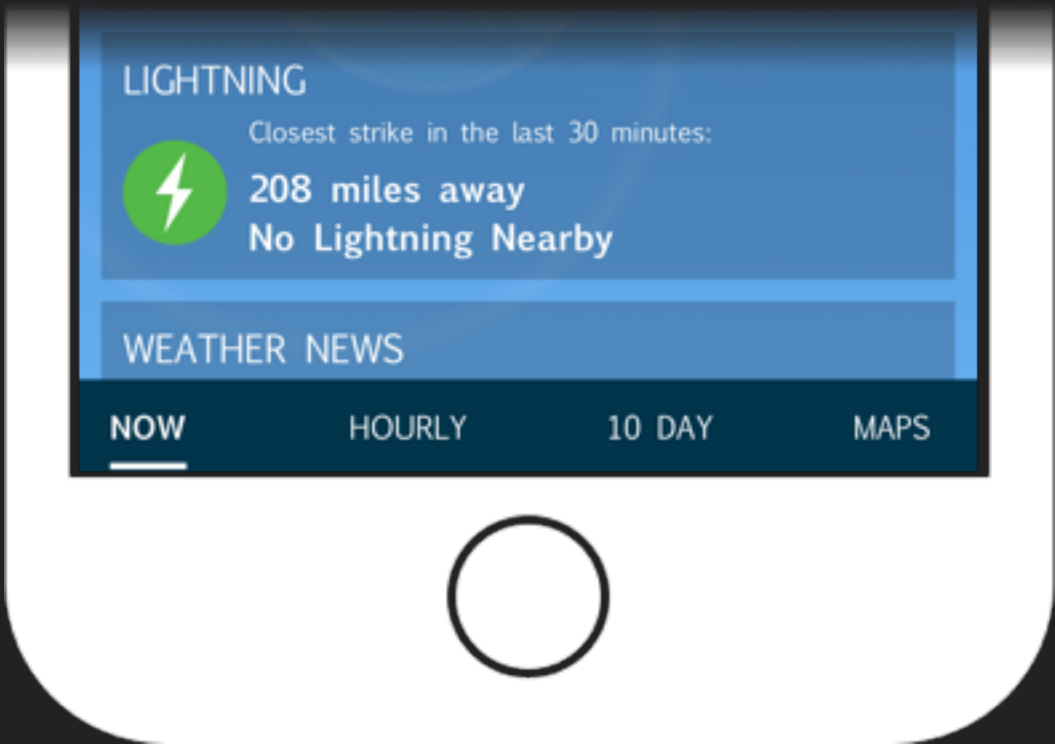
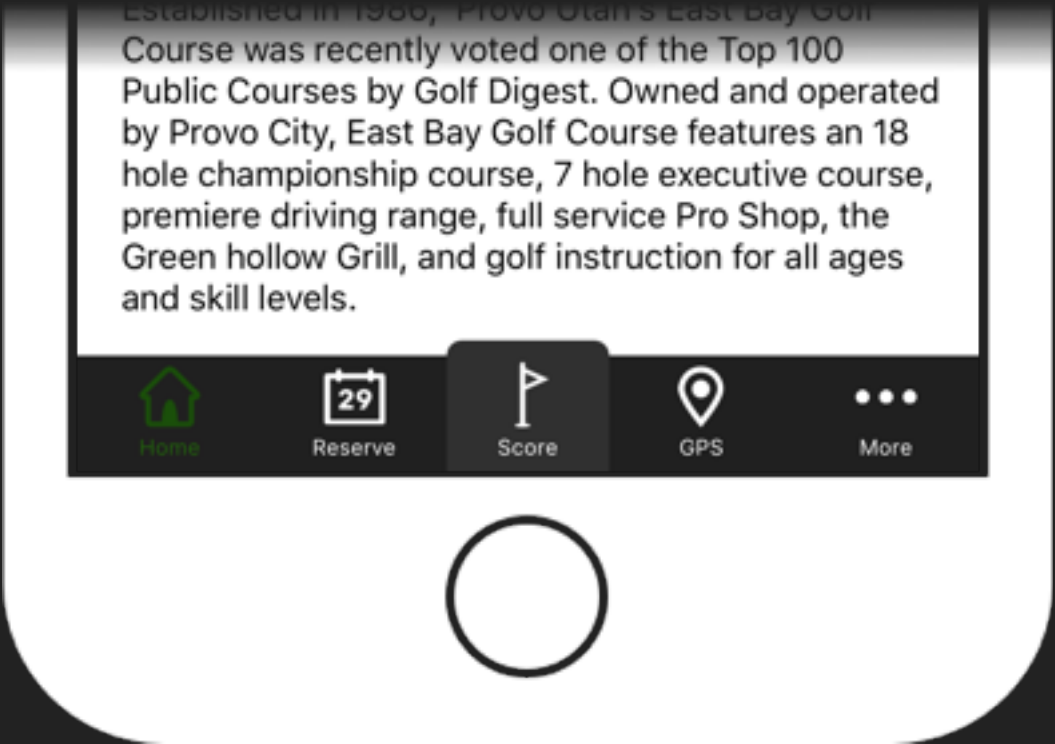
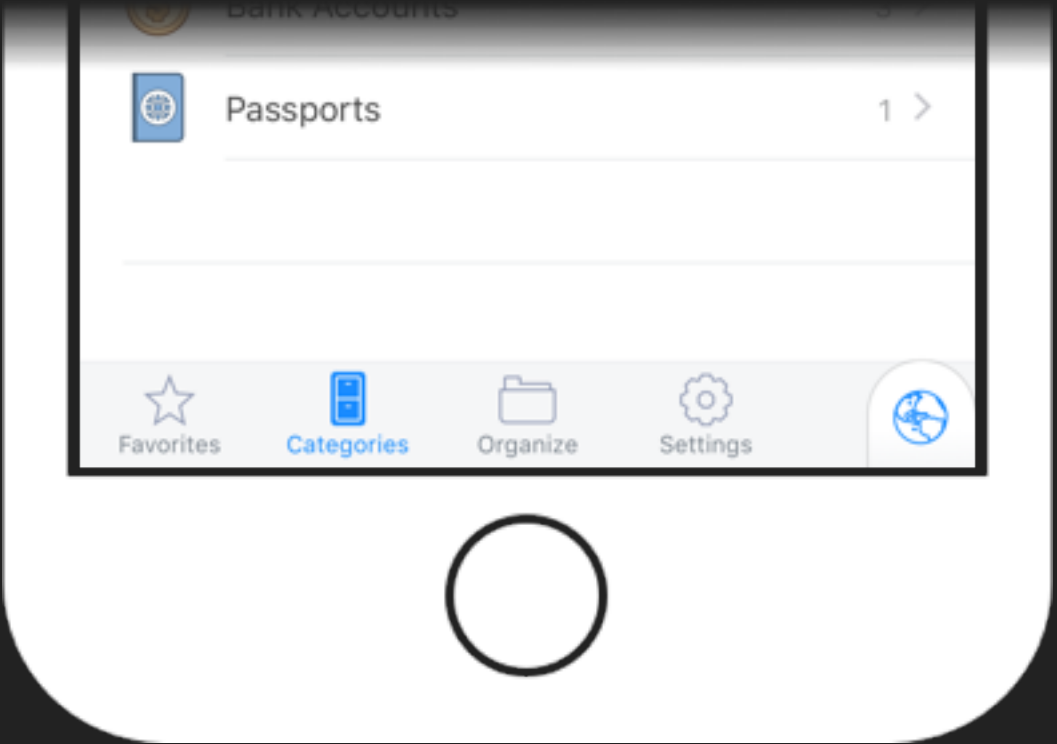
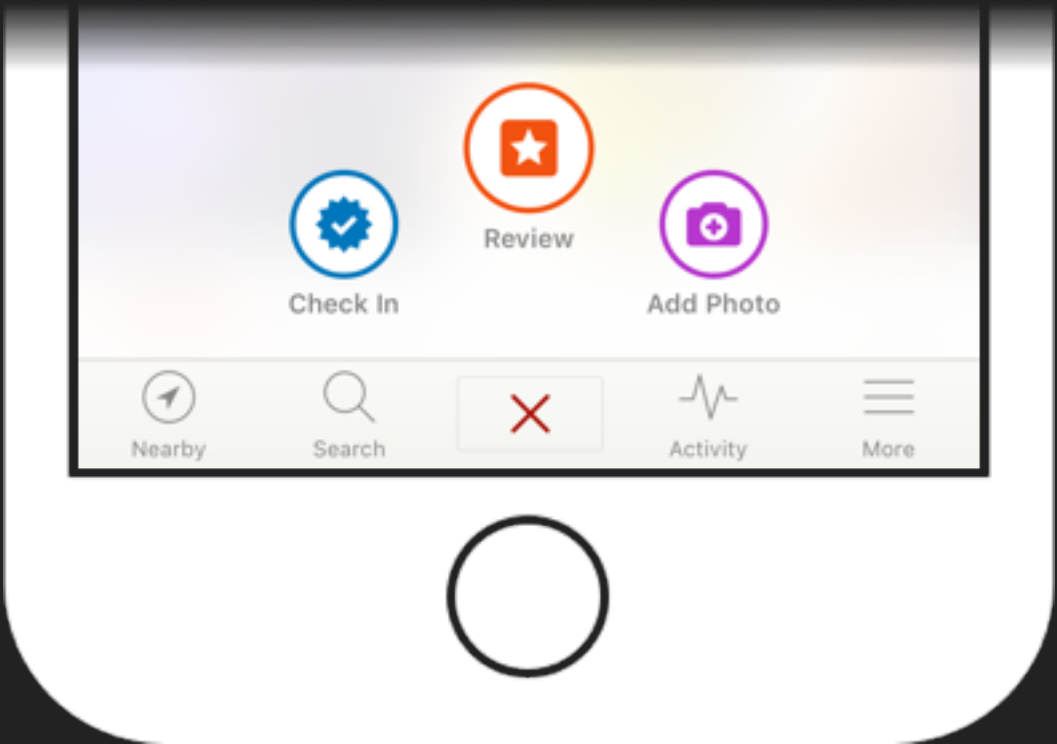
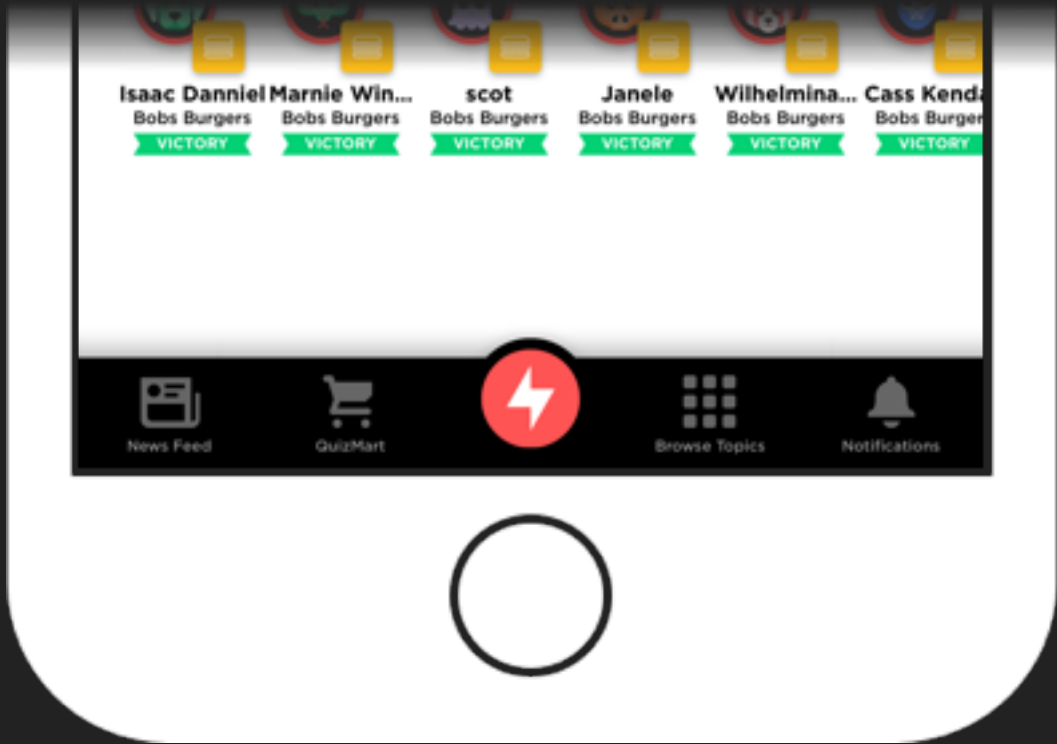
CUSTOM TAB BARS

WHAT WE'LL COVER

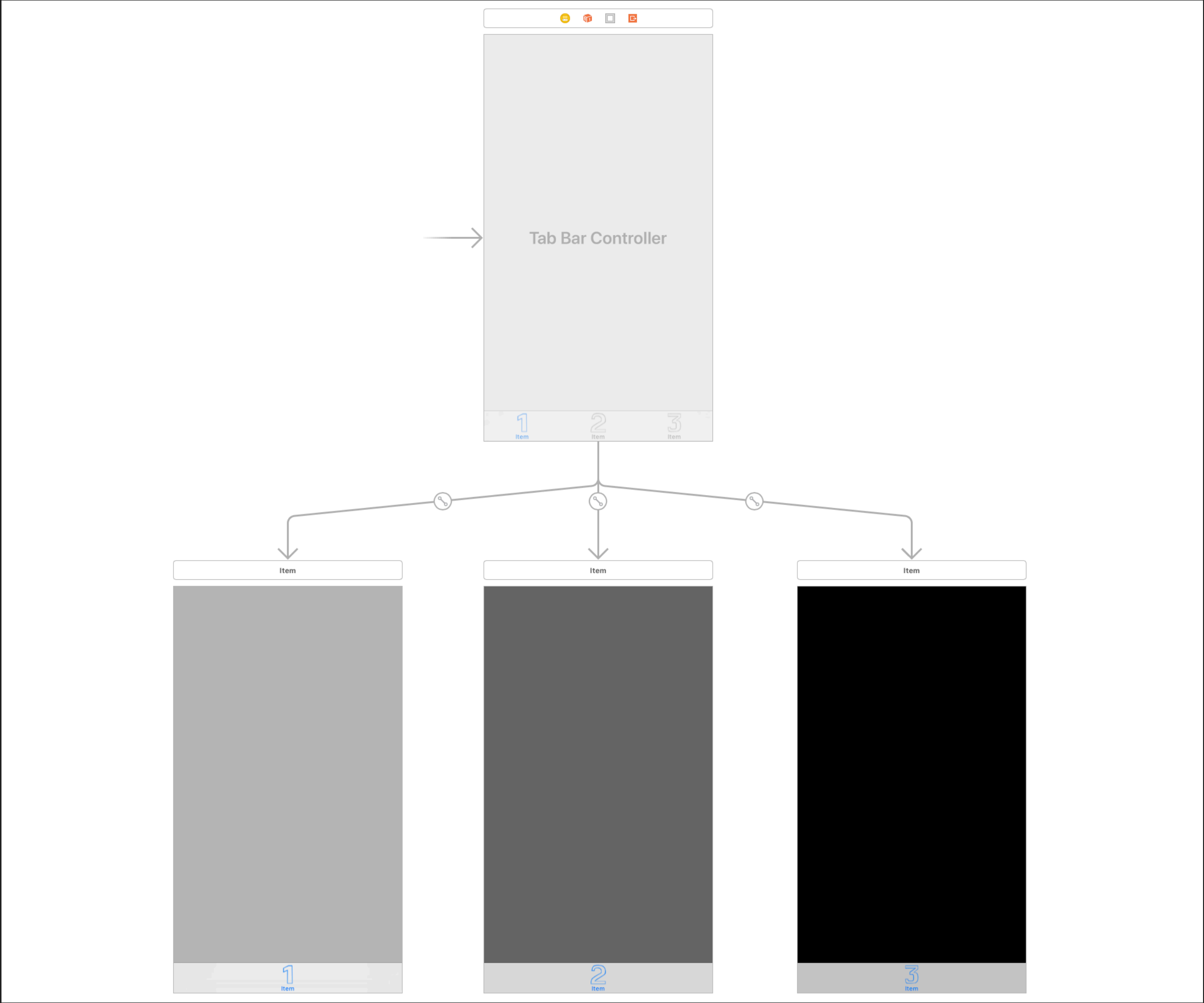
- ▶ An example implementation of a custom tab bar controller
 - ▶ We'll use a subclassed UITabBarController: `CustomTabBarController`
 - ▶ Allow the underlying (Apple's) tab bar controller to manage the view switching
 - ▶ We'll use a subclassed UIView for the user interface: `CustomTabBarView`

WHY?

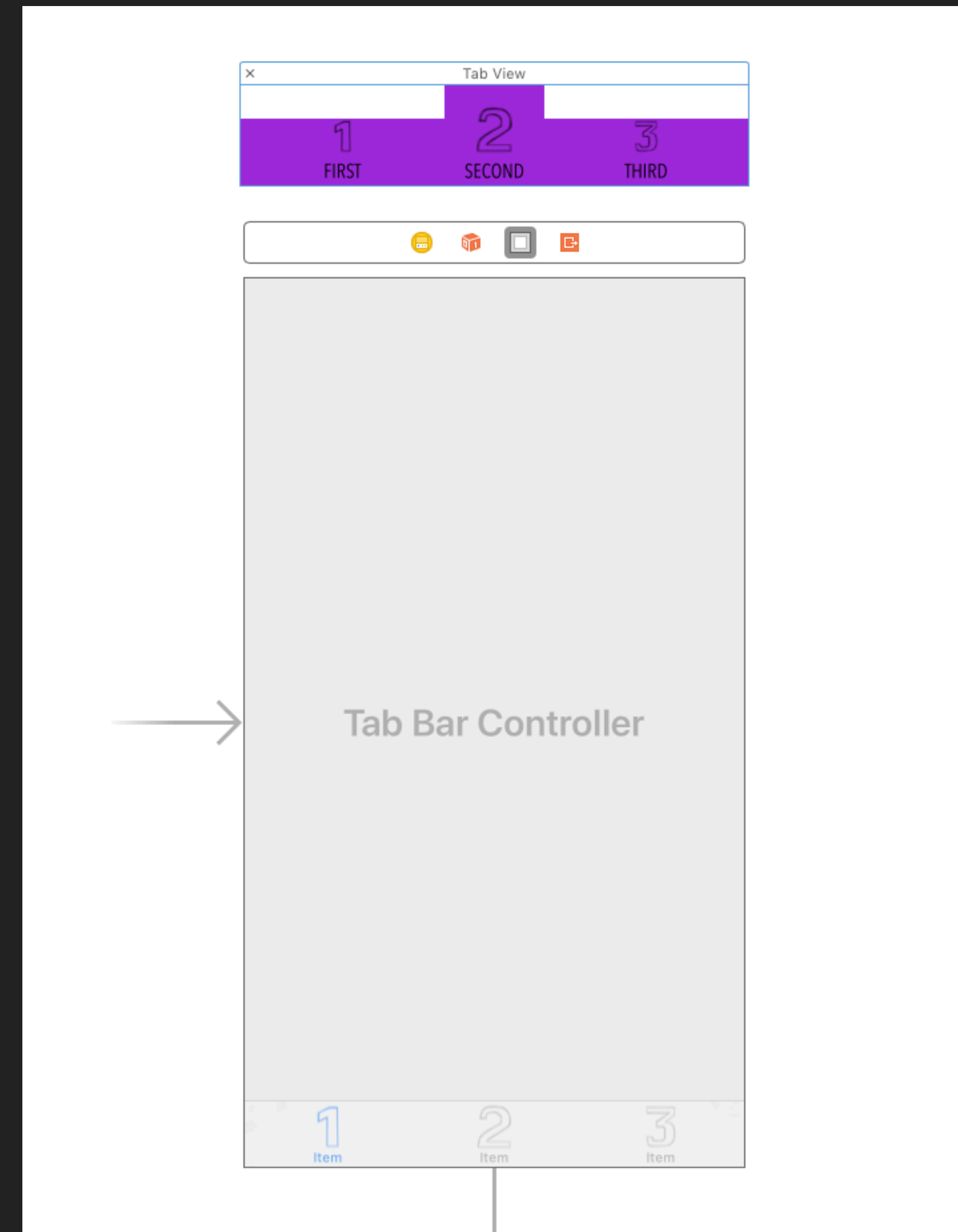
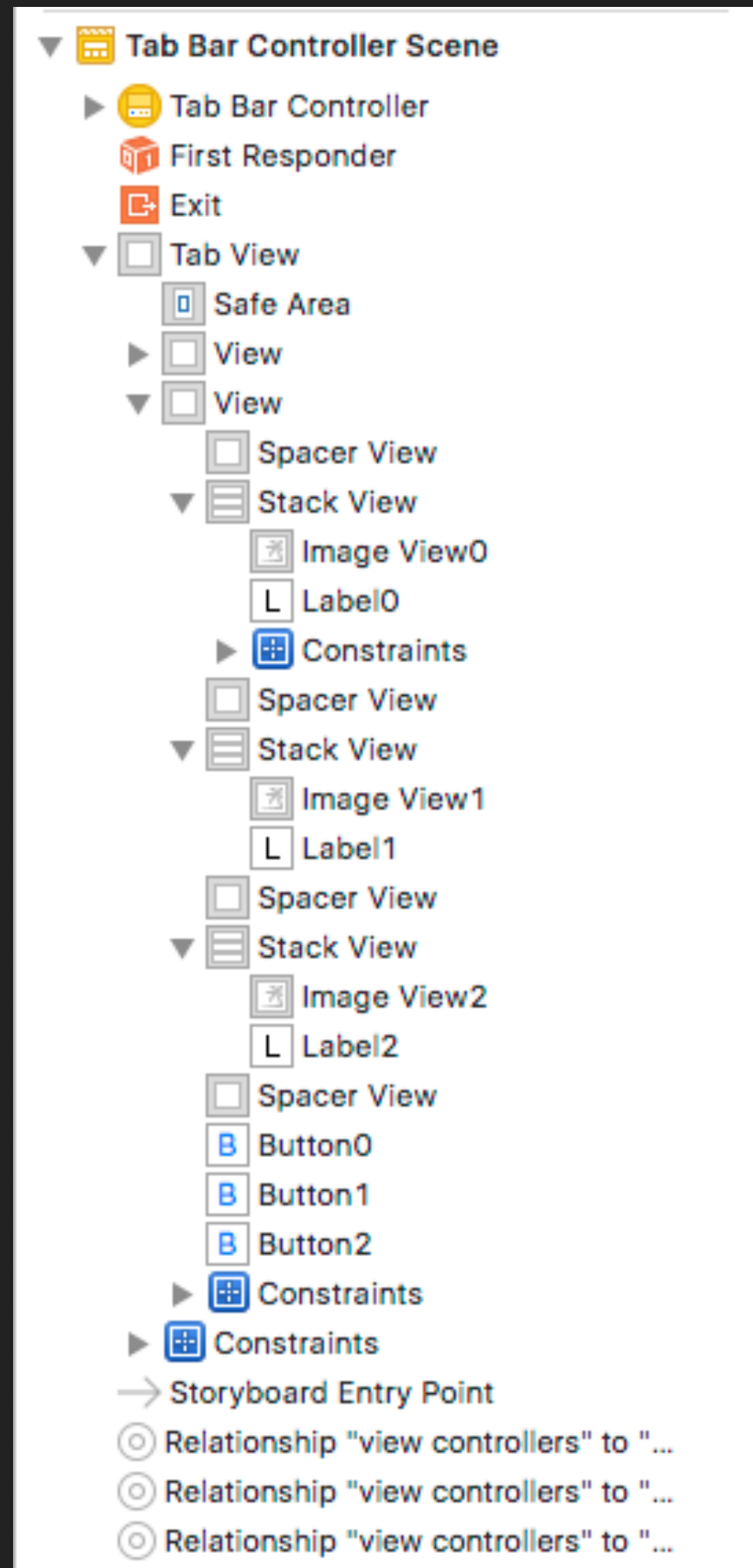
► DESIGNERS LOVE THEM!



SETUP STORYBOARD STRUCTURE



BUILD TAB BAR UI



BUILD TAB BAR UI

Tab Bar Controller Scene

Tab Bar Controller

First Responder

Exit

Tab View

Safe Area

View

View

Spacer View

Stack View

Image View0

Label0

Constraints

Spacer View

Stack View

Image View1

Label1

Spacer View

Stack View

Image View2

Label2

Spacer View

Button0

Button1

Button2

Constraints

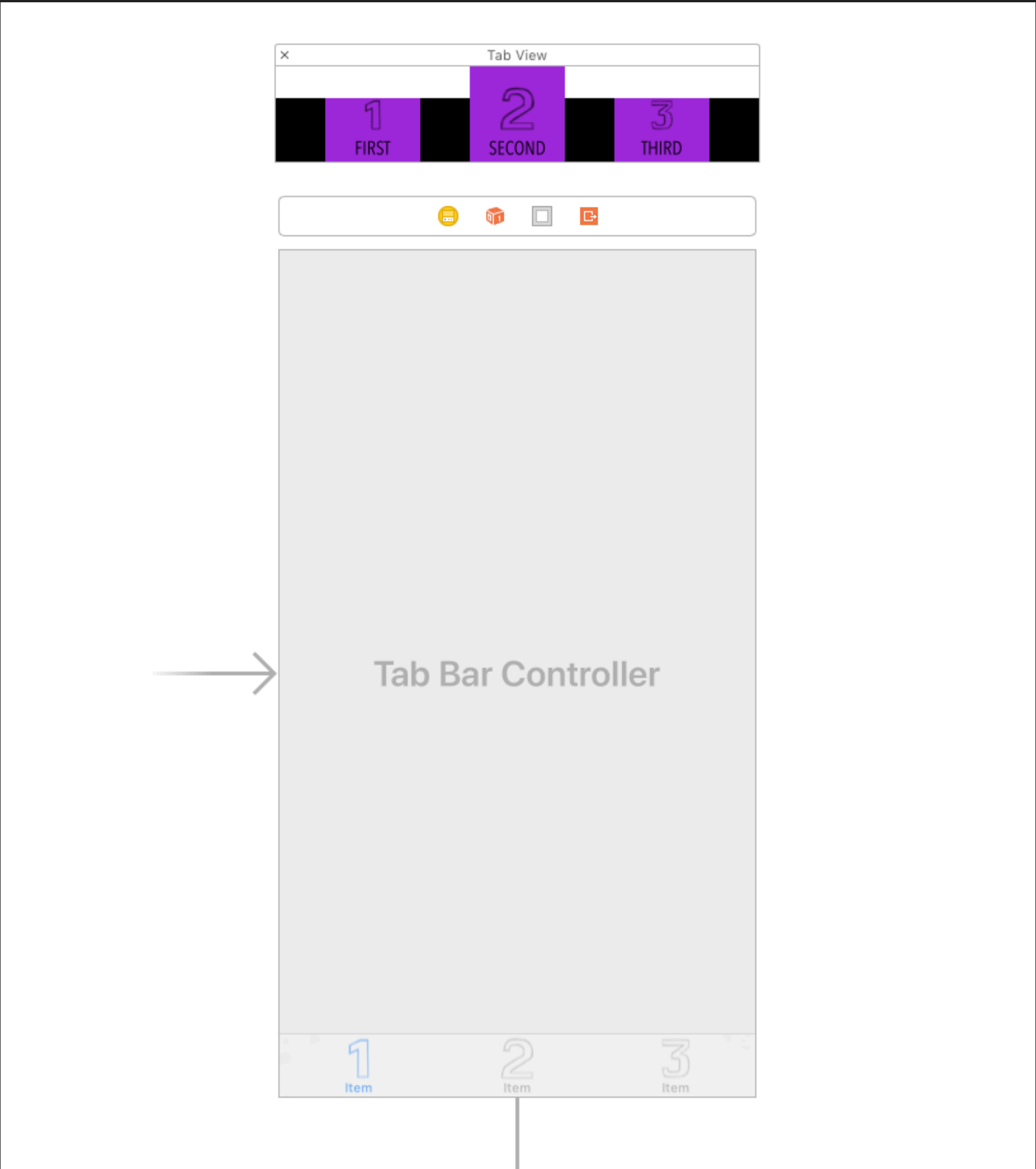
Constraints

Storyboard Entry Point

Relationship "view controllers" to "...

Relationship "view controllers" to "...

Relationship "view controllers" to "...



CUSTOM TAB BAR VIEW CLASS

```
class CustomTabBarView: UIView {
    //INTERFACE BUILDER OUTLETS
    @IBOutlet weak var imageView0: UIImageView!
    ...
    @IBOutlet weak var label0: UILabel!
    ...
    @IBOutlet weak var button0: UIButton!
    ...

    var icons: [(unselected: UIImage, selected: UIImage)] = [(UIImage(named: "one")!, UIImage(named: "oneS")!),
                                                             (UIImage(named: "two")!, UIImage(named: "twoS")!),
                                                             (UIImage(named: "three")!, UIImage(named: "threeS")!)]

    func setAppearance(forIndex index: Int) {
        let imageViews: [UIImageView] = [imageView0, imageView1, imageView2]
        let labels: [UILabel] = [label0, label1, label2]

        for i in 0..
```

TAB BAR VIEW DELEGATE

```
protocol CustomTabBarViewDelegate: class {
    func tabBarViewChangedSelectedIndex(at index: Int)
}

class CustomTabBarView: UIView {

    @IBOutlet weak var imageView0: UIImageView!
    ...
    var icons: [(unselected: UIImage, selected: UIImage)] = ...
    func setAppearance(forIndex index: Int) {...}

    weak var delegate: CustomTabBarViewDelegate?

    @IBAction func tabBarButtonTapped(_ sender: UIButton) {
        delegate?.tabBarViewChangedSelectedIndex(at: sender.tag)
    }
}
```


SUBCLASS TAB BAR CONTROLLER

```
import UIKit

class CustomTabBarController: UITabBarController, CustomTabBarViewDelegate {

    @IBOutlet weak var tabView: CustomTabBarView!

    override var selectedIndex: Int {
        didSet {
            tabView.setAppearance(forIndex: selectedIndex)
        }
    }

    override func viewDidLoad() {
        super.viewDidLoad()

        let frame = CGRect(x: 0, y: view.frame.height - 75.0, width: view.frame.width, height: 75)
        tabView.frame = frame
        selectedIndex = 0
        tabView.delegate = self
        view.addSubview(tabView)
    }

    // MARK: - Custom Tab Bar Delegate

    func tabBarViewChangedSelectedIndex(at index: Int) {
        selectedIndex = index
    }
}
```

GOTCHAS

- ▶ Don't forget to:
 - ▶ Set up the identify of your storyboard objects
 - ▶ Set up the `tag` property on your buttons
 - ▶ Hide the `UITabBarController`'s tab bar
 - ▶ Set the `delegate` property of your `CustomTabBarView` instance

DEMO

QUESTIONS?