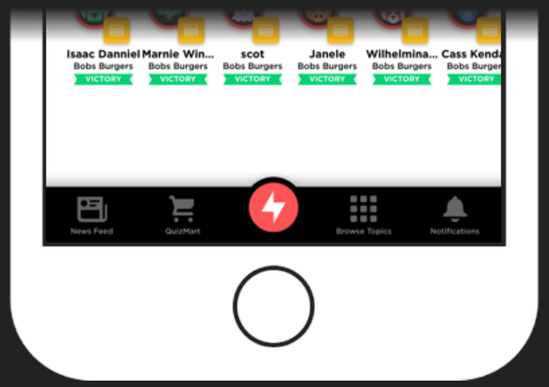
CUSTOM TAB BARS

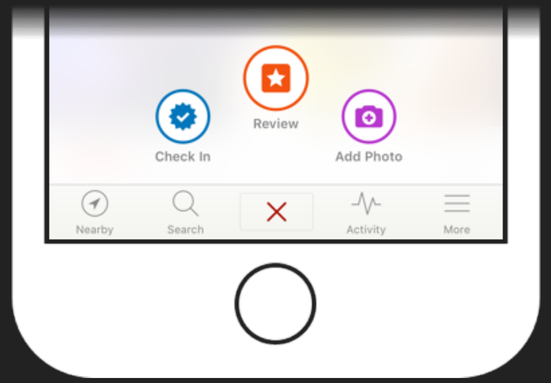
WHAT WE'LL COVER

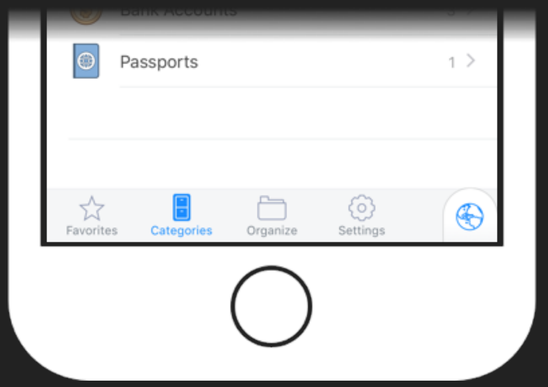
- An example implementation of a custom tab bar controller
  - We'll use a subclassed UITabBarController: `CustomTabBarController`
    - Allow the underlying (Apple's) tab bar controller to manage the view switching
  - We'll use a subclassed UIView for the user interface: `CustomTabBarView`
  - We'll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

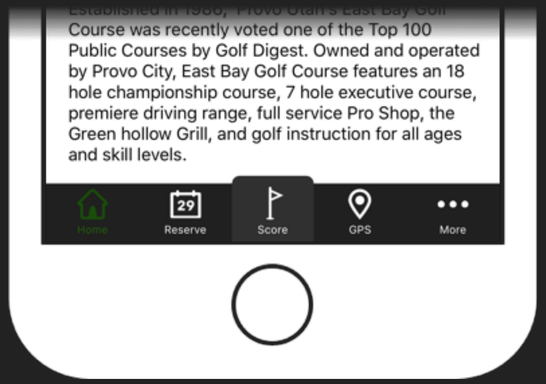
WHY?

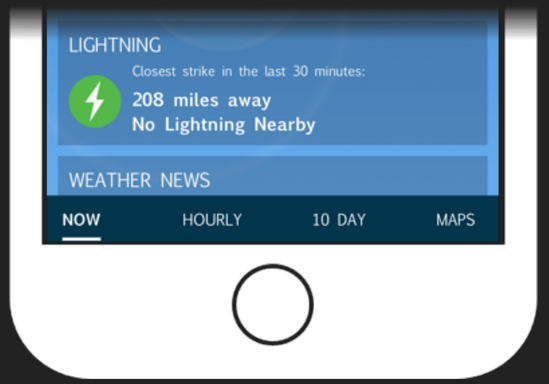
- DESIGNERS LOVE THEM!



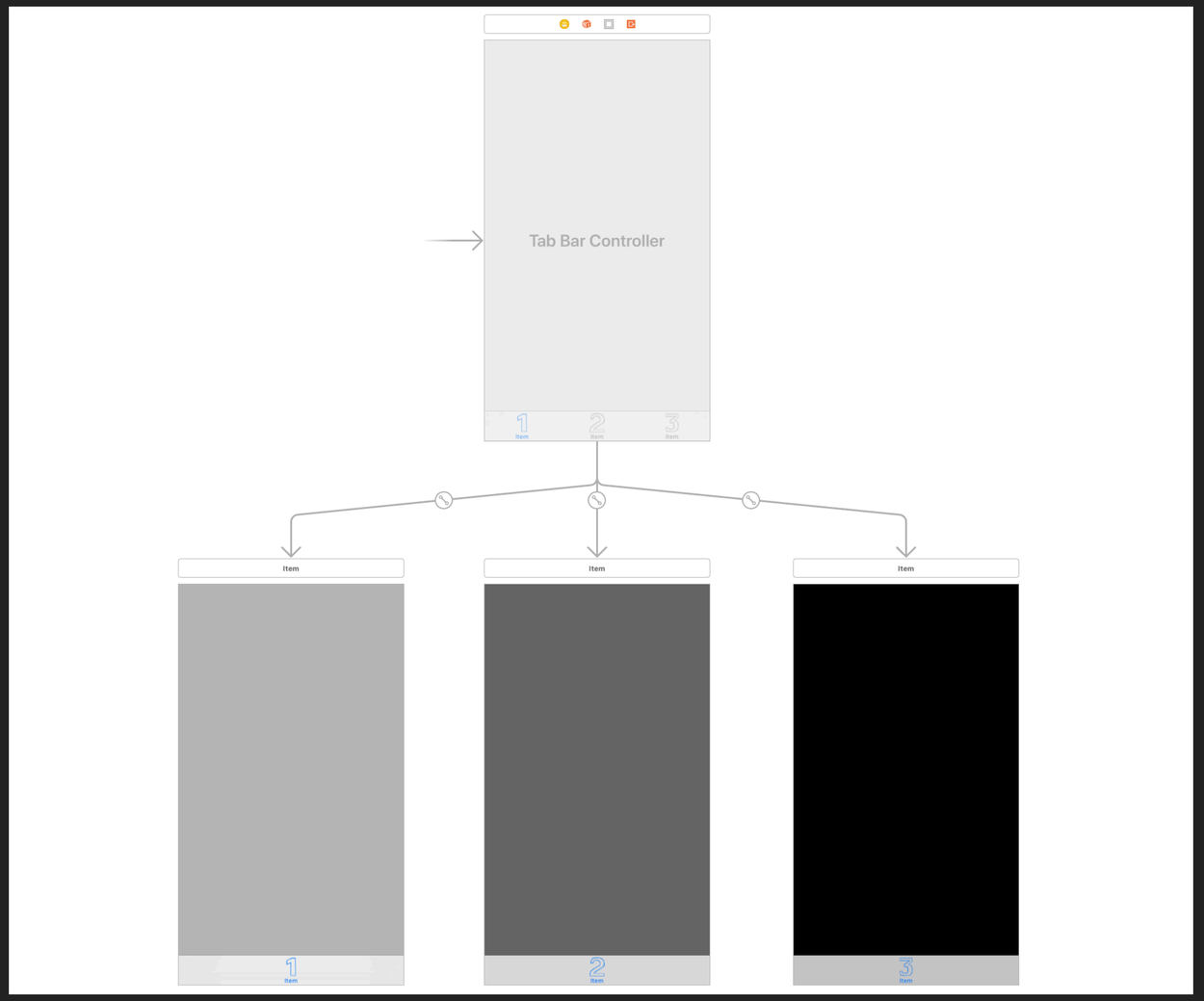




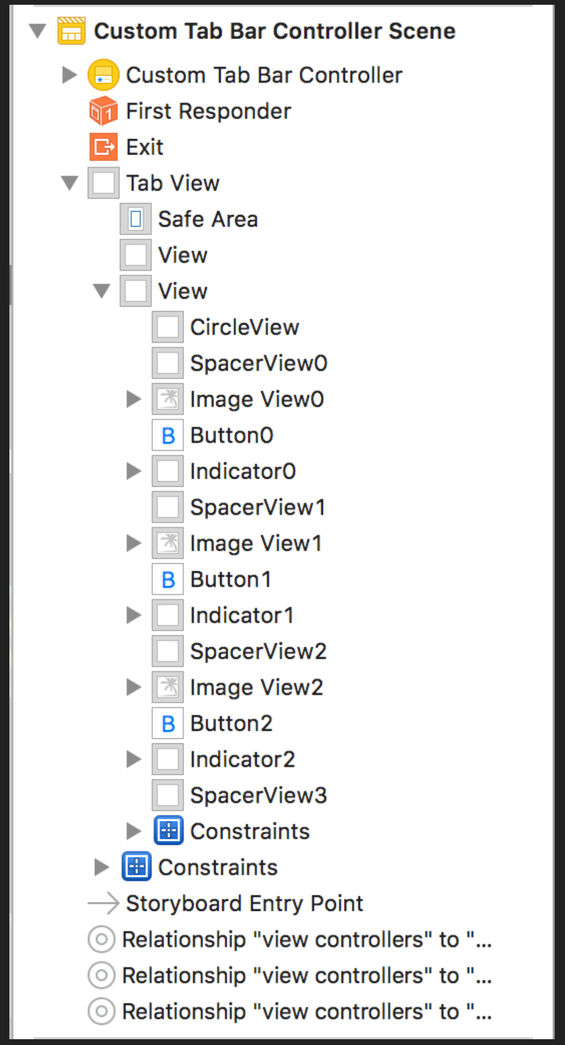


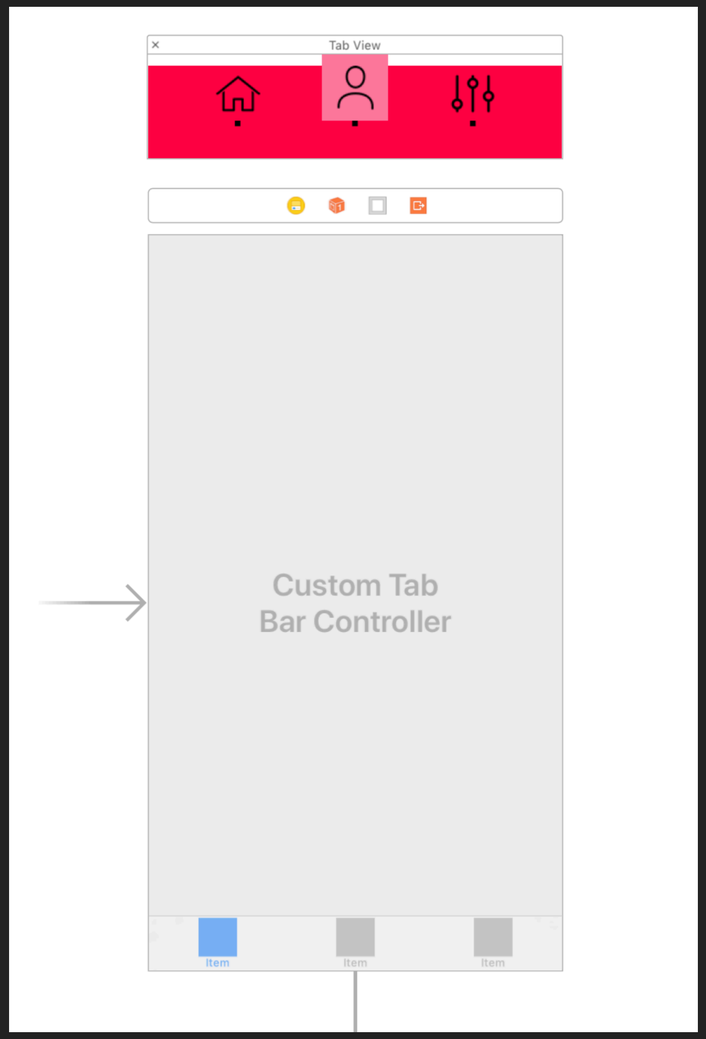


SETUP STORYBOARD STRUCTURE

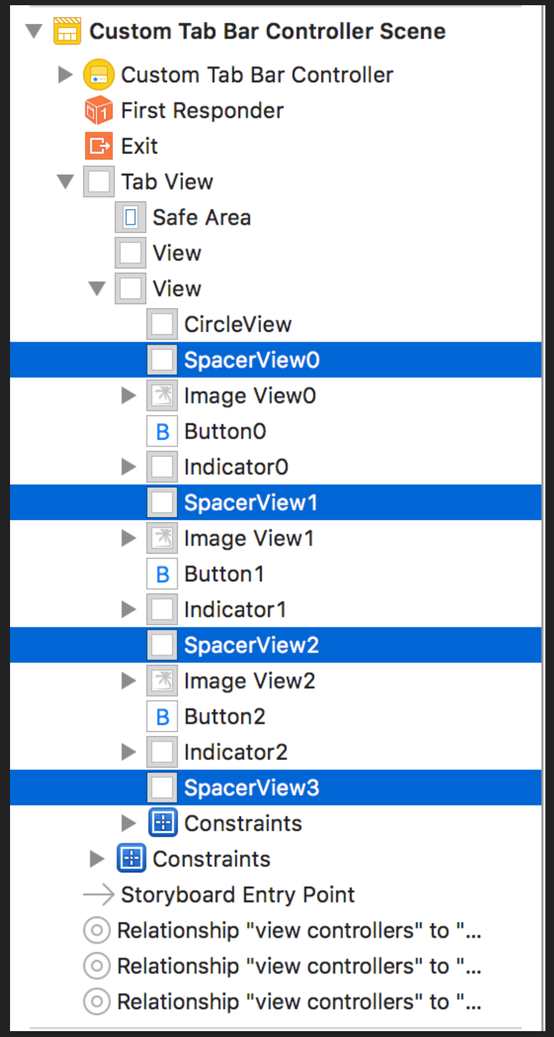


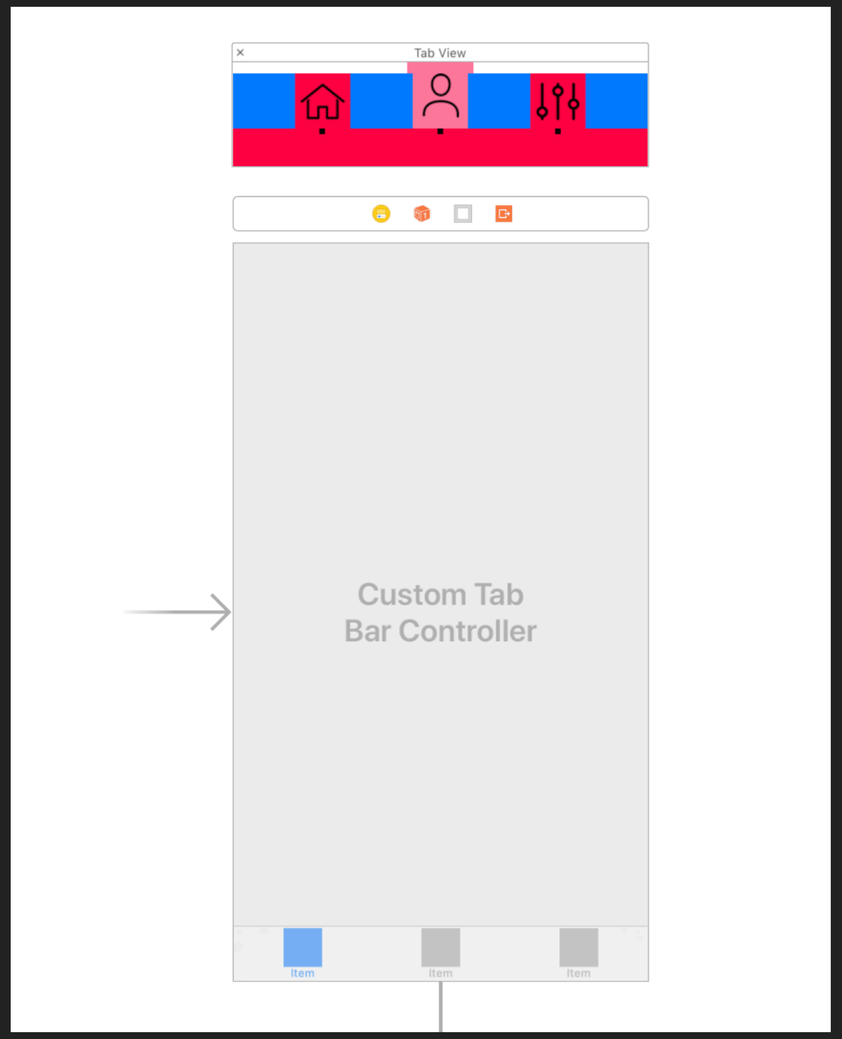
BUILD TAB BAR UI





BUILD TAB BAR UI





CUSTOM TAB BAR VIEW CLASS

```
class CustomTabBarView: UIView {
   //INTERFACE BUILDER OUTLETS
   @IBOutlet weak var imageView0: UIImageView!
   ...
   @IBOutlet weak var button0: UIButton!
   ...
   @IBOutlet weak var indicator0: UIView!
   ...


   func setAppearance(forIndex index: Int) {
      let imageViews: [UIImageView] = [imageView0, imageView1, imageView2]
      let indicators: [UIView] = [indicator0, indicator1, indicator2]
      for i in 0..<imageViews.count {
      if i == index {
         //selected tab
         indicators[i].isHidden = false
      } else {
         //unselected tab
         indicators[i].isHidden = true
      }
   }


   @IBAction func tabBarButtonTapped(_ sender: UIButton) { }
}
```

TAB BAR VIEW DELEGATE

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

class CustomTabBarView: UIView {

   @IBOutlet weak var imageView0: UIImageView!
   ...
   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()
      //Next Slide
   }

   // MARK: - Custom Tab Bar Delegate

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

}
```

SUBCLASS TAB BAR CONTROLLER (CONT'D)

```
import UIKit

class CustomTabBarController: UITabBarController, CustomTabBarViewDelegate {

   @IBOutlet weak var tabView: CustomTabBarView!

   override var selectedIndex: Int {...}

   override func viewDidLoad() {
      super.viewDidLoad()

      tabView.translatesAutoresizingMaskIntoConstraints = false
      tabView.delegate = self
      selectedIndex = 0
      view.addSubview(tabView)
      let leadingConstraint = tabView.leadingAnchor.constraint(equalTo: view.leadingAnchor)        //↓ to extend behind
      let trailingConstraint = tabView.trailingAnchor.constraint(equalTo: view.trailingAnchor)     //↓ iPhone X home indicator
      let bottomConstraint = tabView.bottomAnchor.constraint(equalTo: view.safeAreaLayoutGuide.bottomAnchor, constant: 34.0)
      let heightConstraint = tabView.heightAnchor.constraint(equalToConstant: 104.0) //← includes 34.0 points for home indicator

      NSLayoutConstraint.activate([leadingConstraint, trailingConstraint, bottomConstraint, heightConstraint])
   }

   func tabBarViewChangedSelectedIndex(at index: Int) {...}
}
```

ADDITIONAL SAFE AREAS

```
import UIKit

class HomeViewController: UIViewController {
override func viewDidLoad() {
   super.viewDidLoad()

   additionalSafeAreaInsets = UIEdgeInsets(top: 0.0,
                                           left: 0.0,
                                         bottom: 60.0,
                                          right: 0.0)
  }
}
```

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
  - Add 34.0 points to the height of your tab bar to accommodate for the iPhone X home indicator
  - Set `additionalSafeAreaInsets`

DEMO

QUESTIONS?