http://github.com/GroundSpeed/ParseSwift

Locking Out the Abominable Snowman in iOS

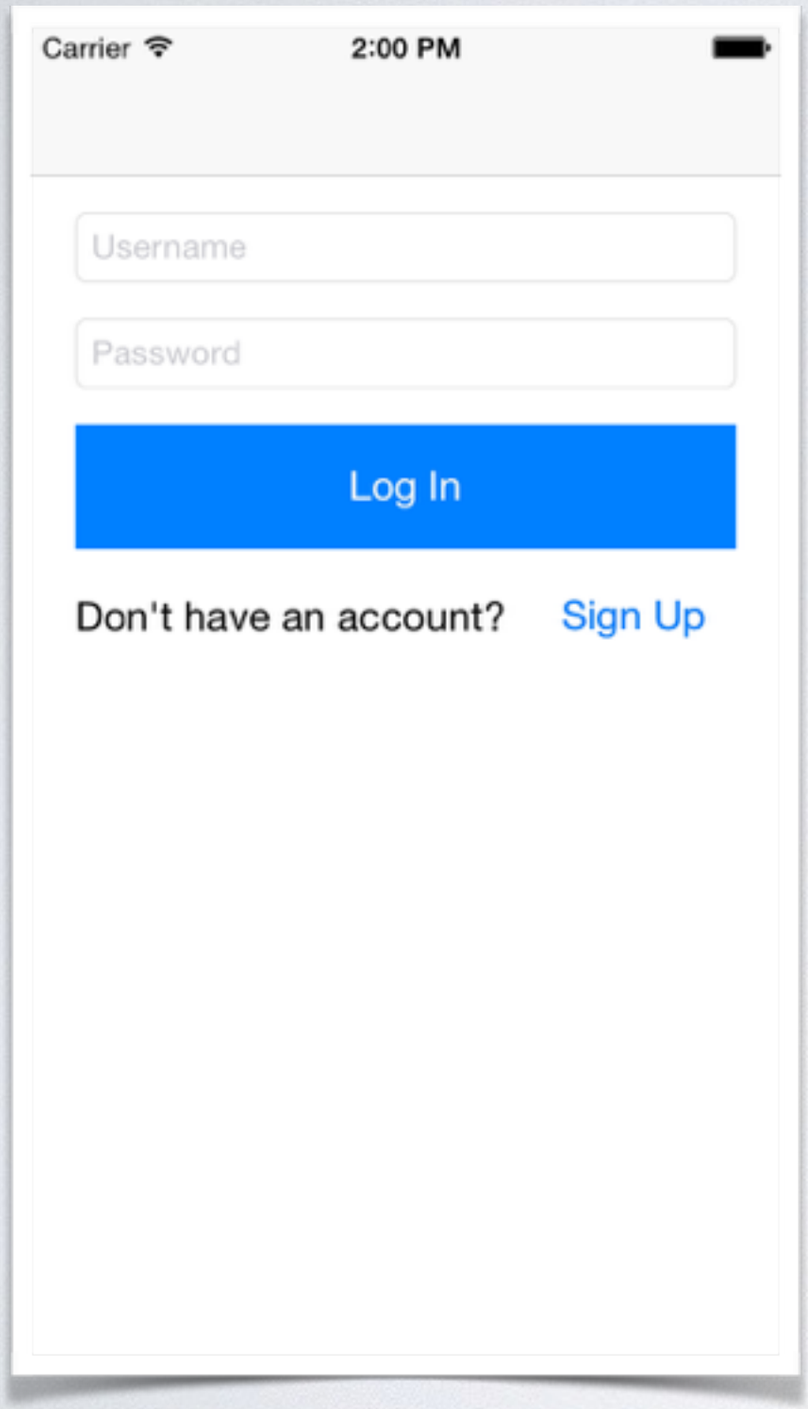
A CodeMash 2015 Introduction to parse.com using Swift

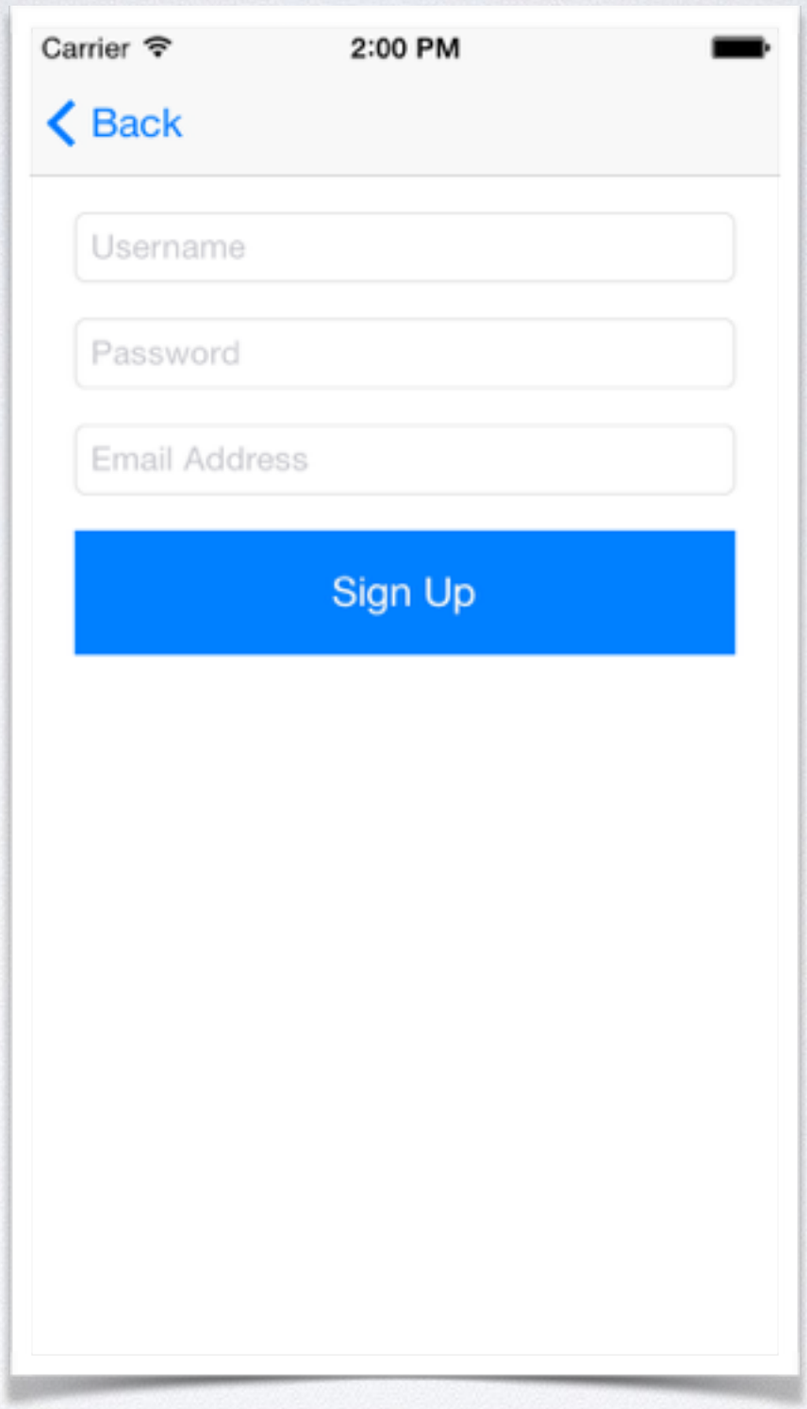
Don Miller | GroundSpeed™
rapid web + mobile software

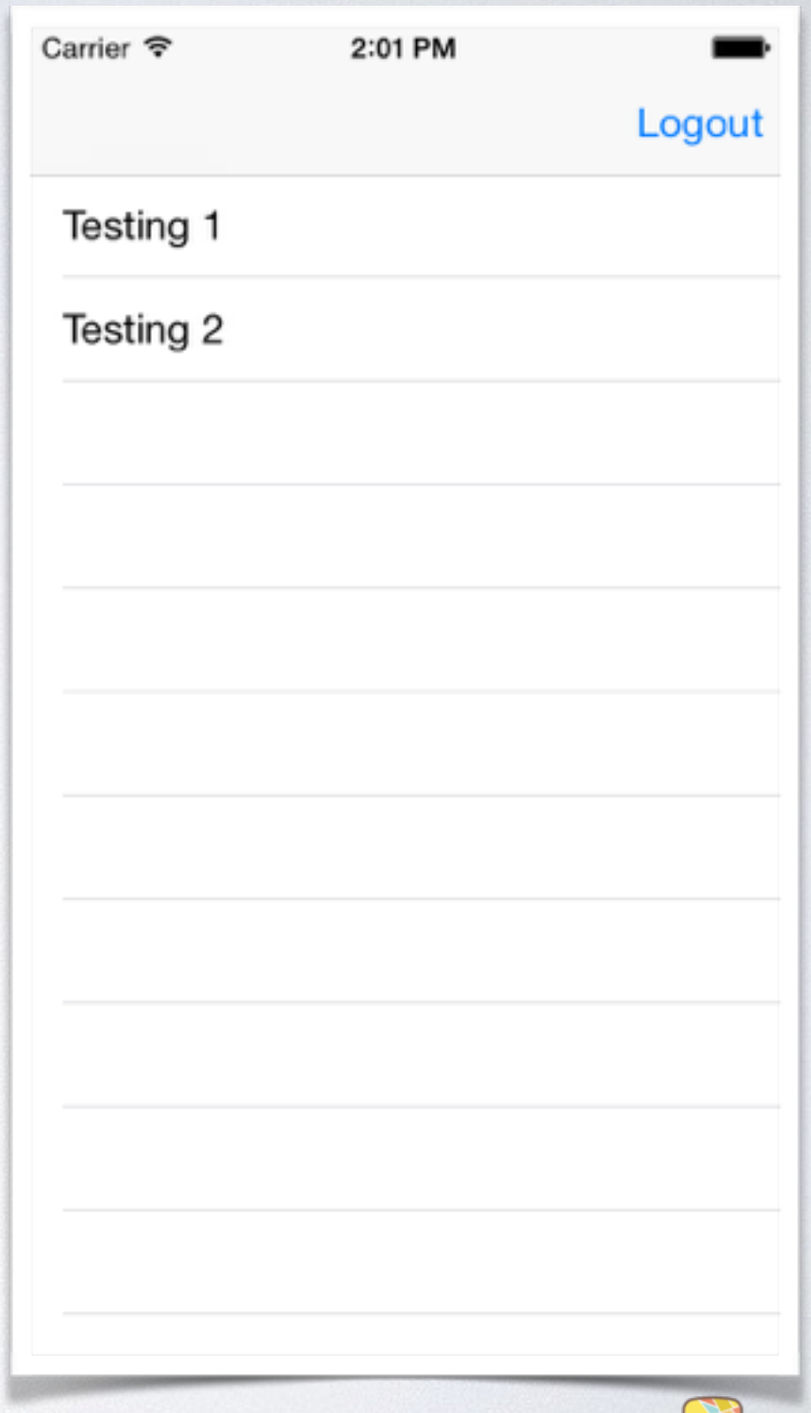
A bit about Parse

- “Cloud” service to store various pieces of data and notifications
  - Users for Security
  - Classes for data tables (Core)
    - We can pull, push, update, or delete records
  - Analytics
  - Push Notifications

Let's Create a Simple App to Authenticate to Parse

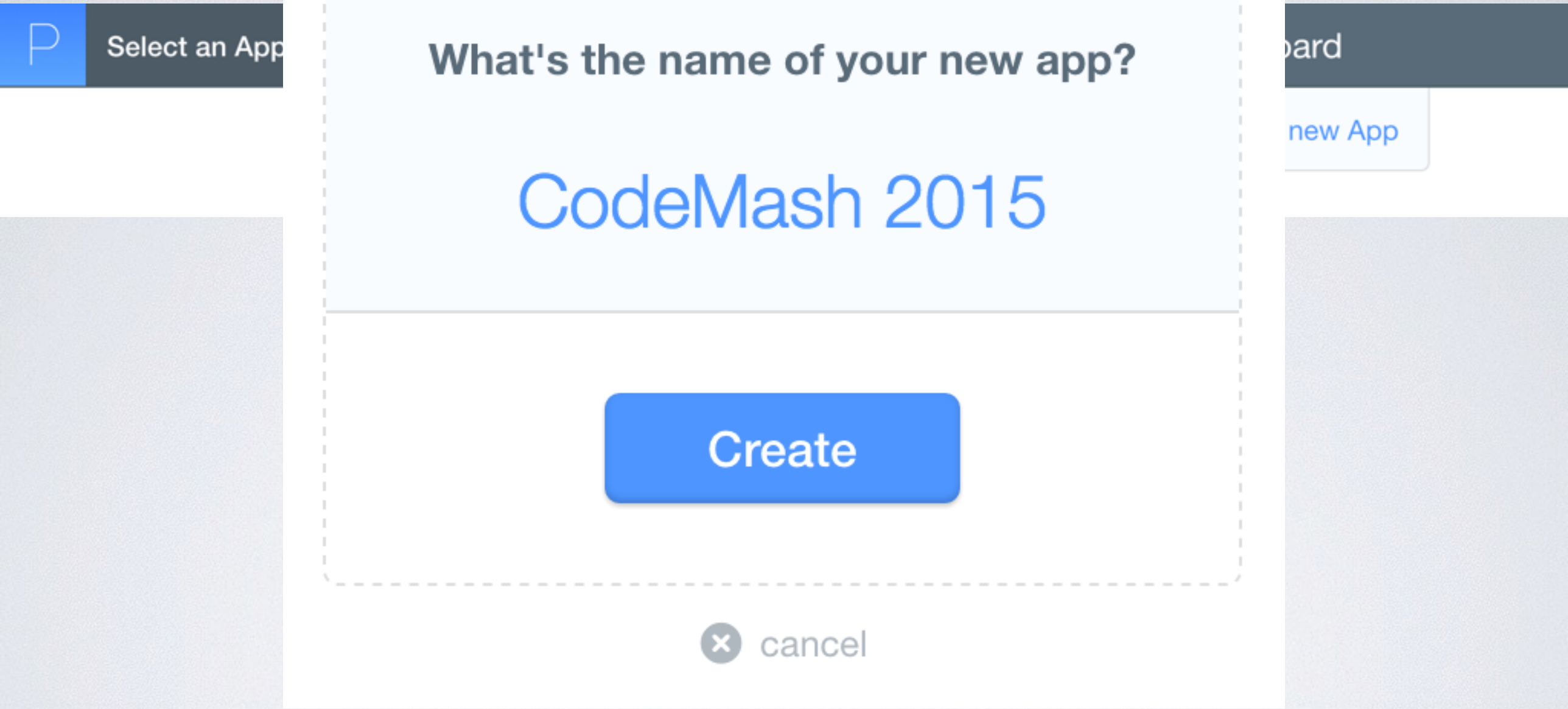




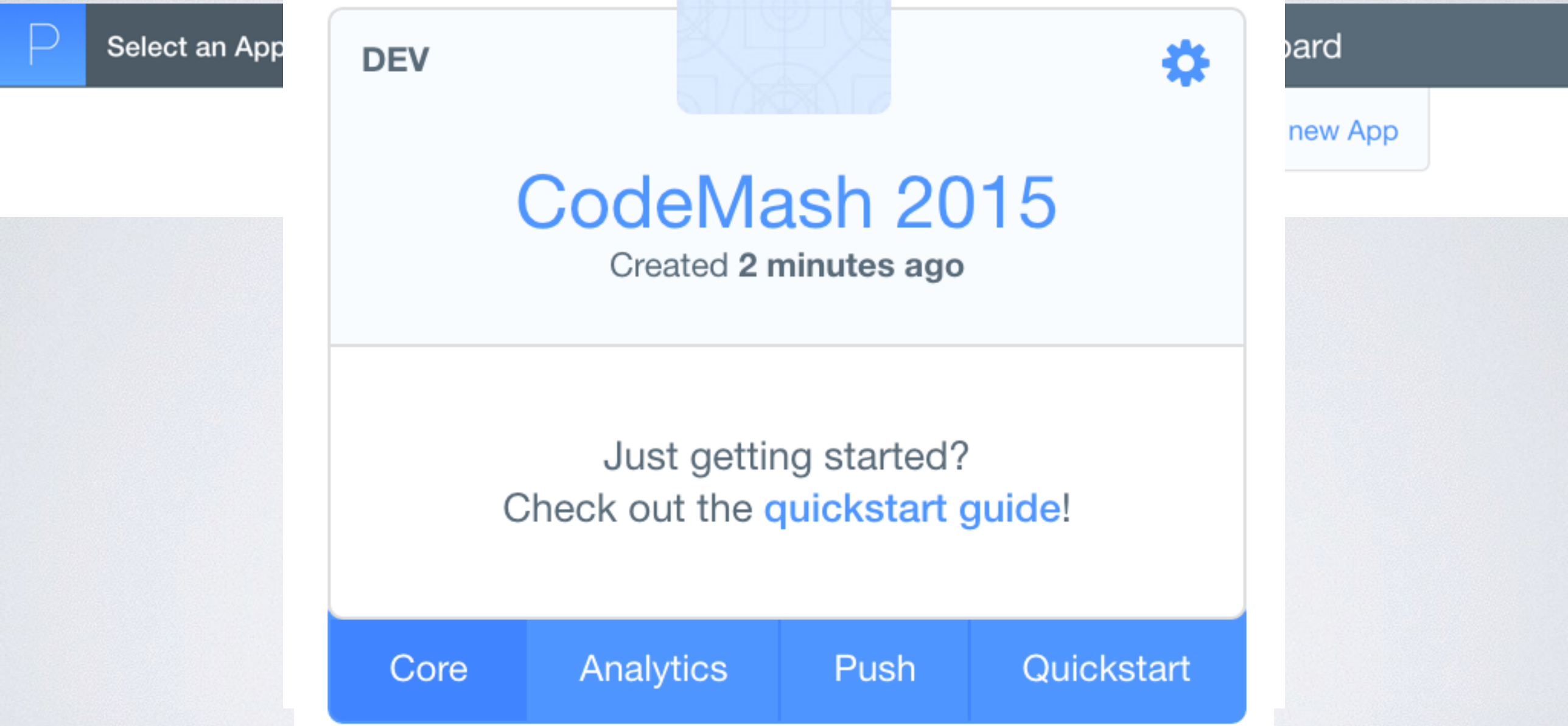


Create Parse App

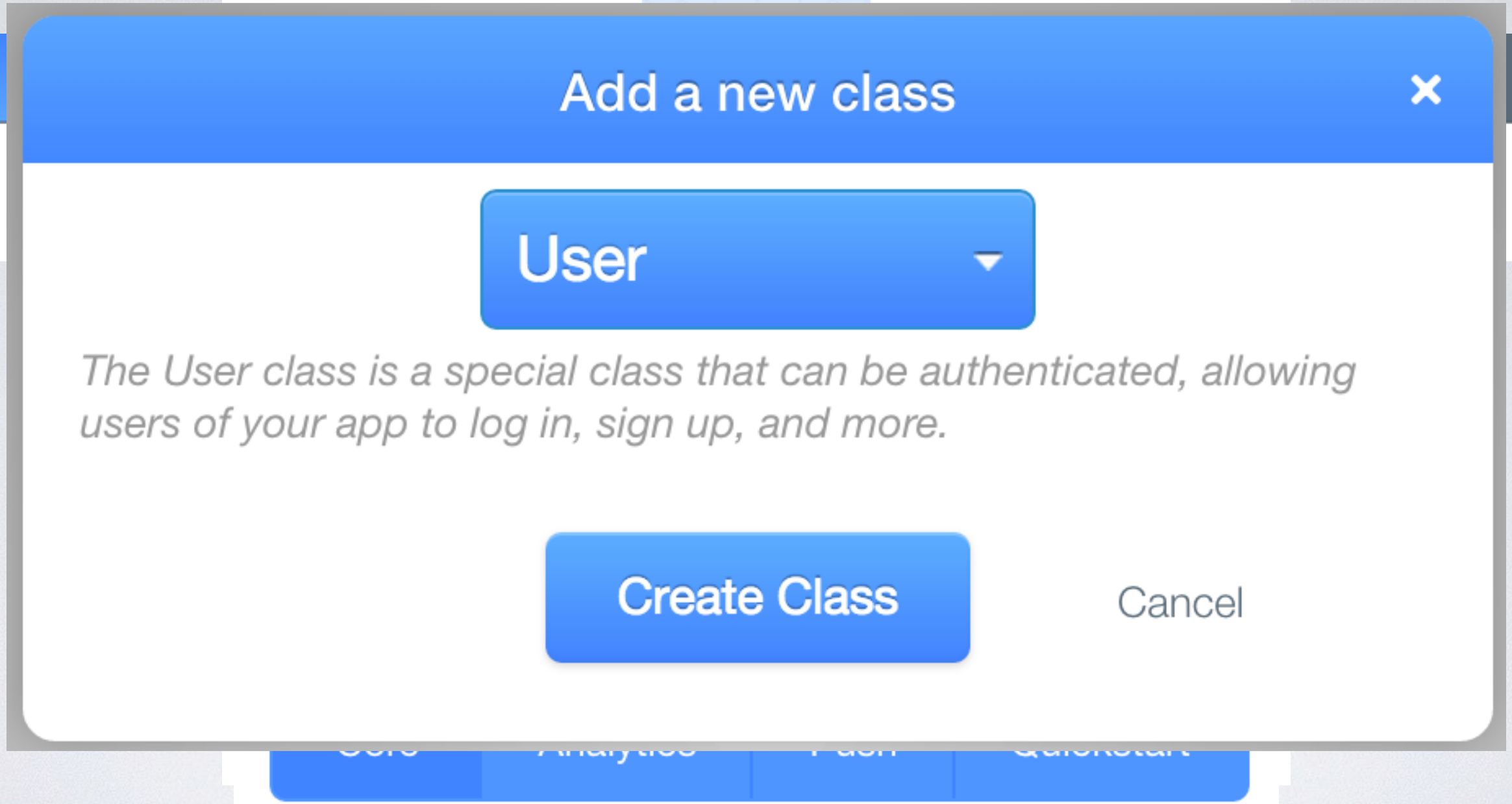
Create Parse App



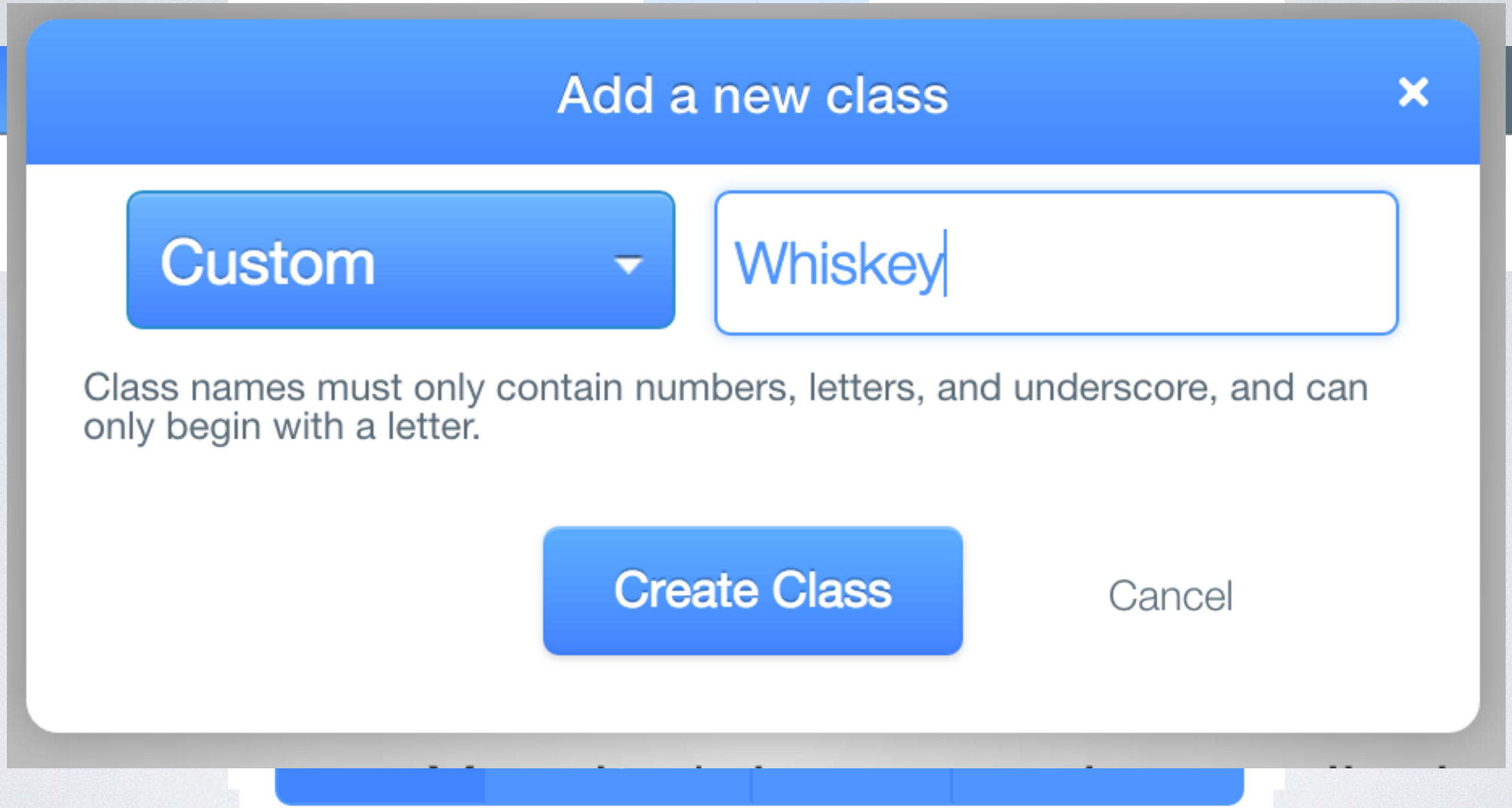
Create Parse App



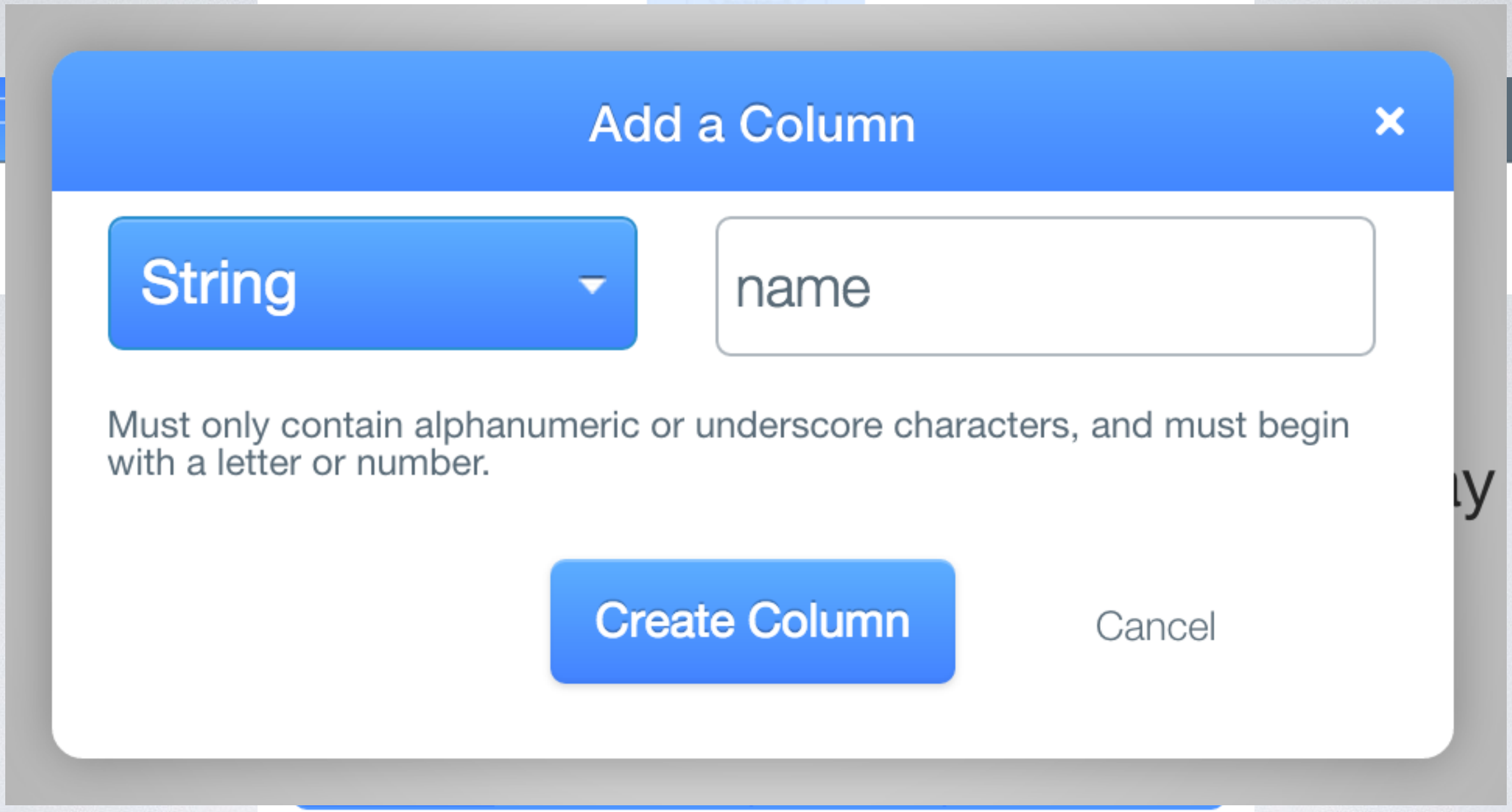
Create Parse App



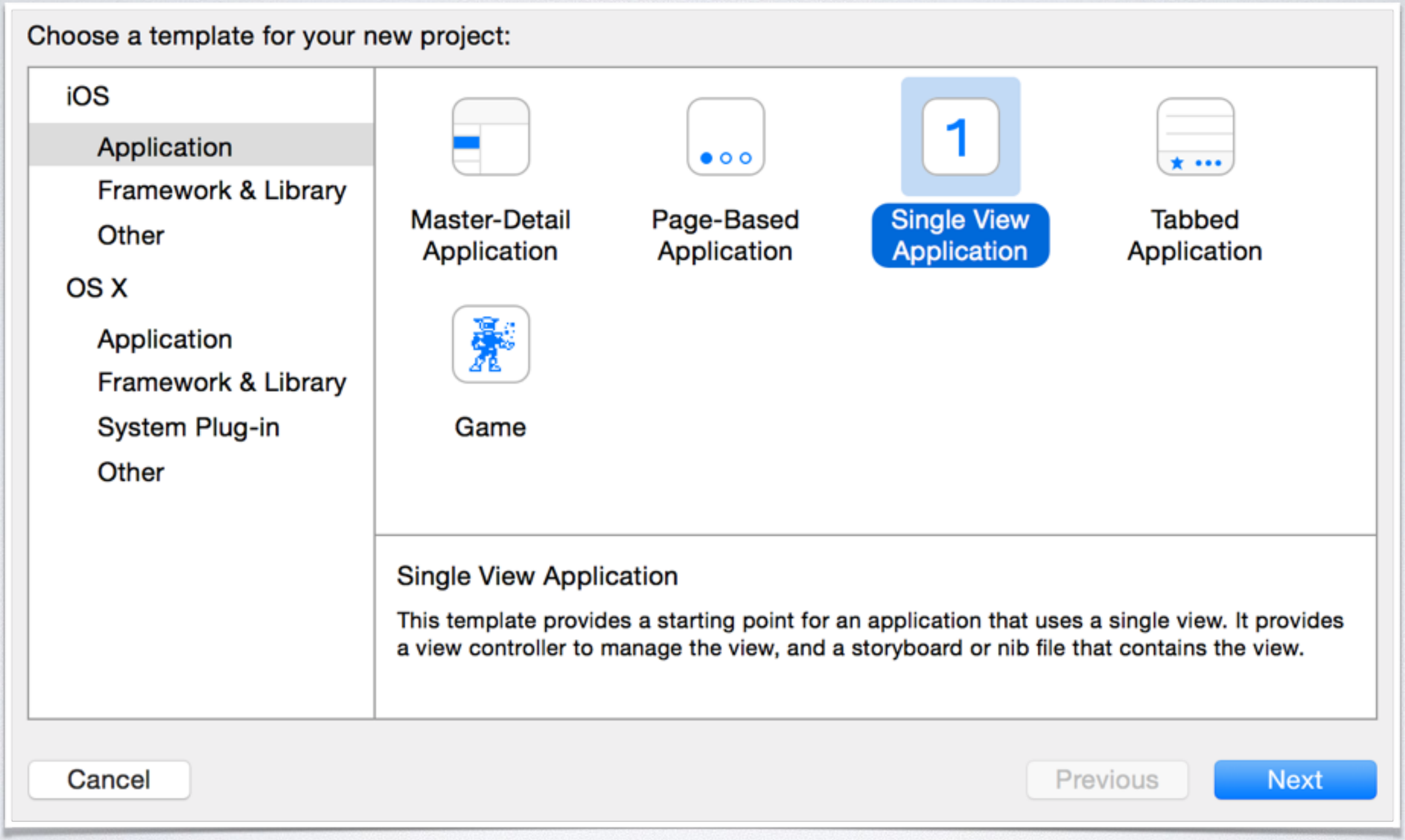
Create Parse App



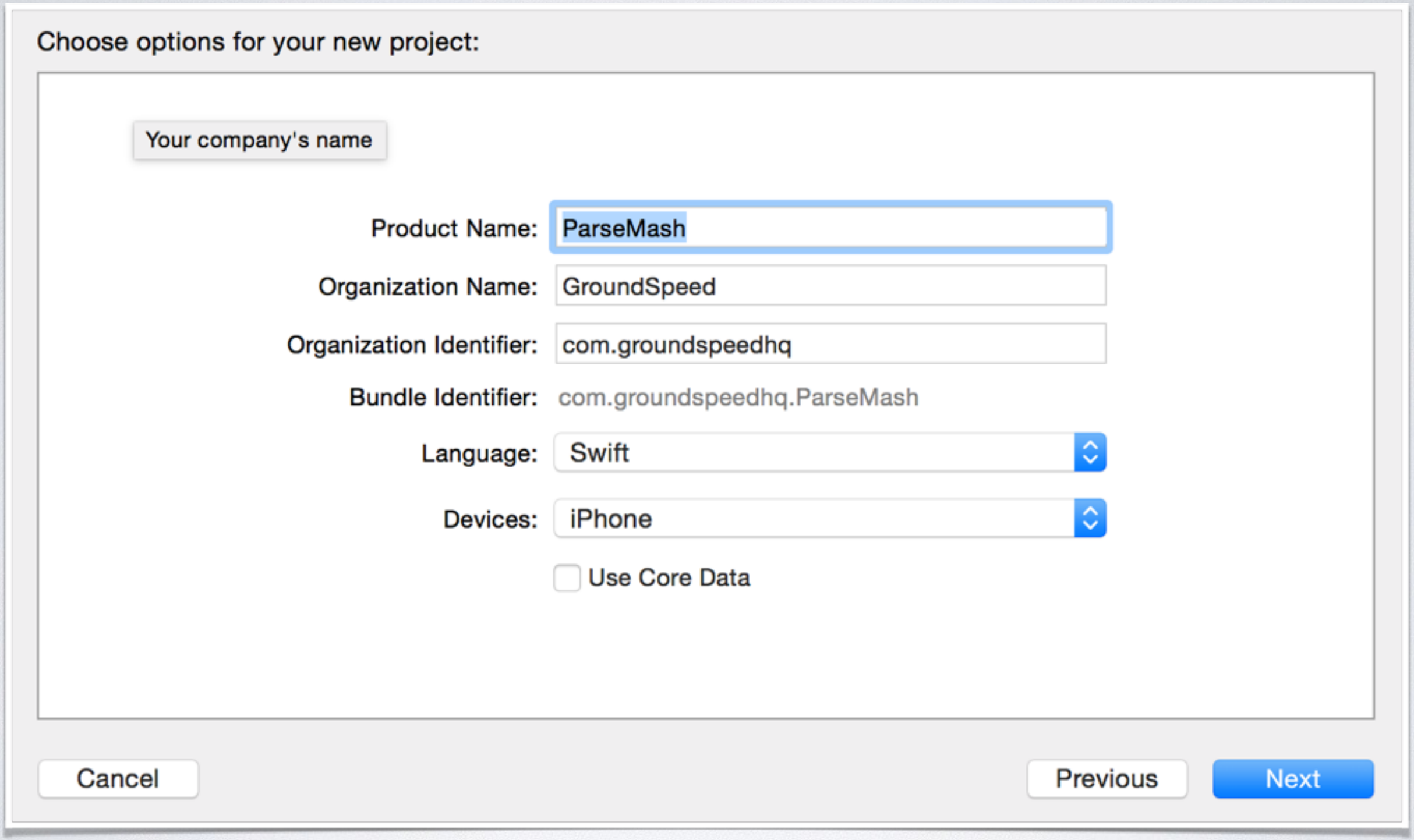
Create Parse App



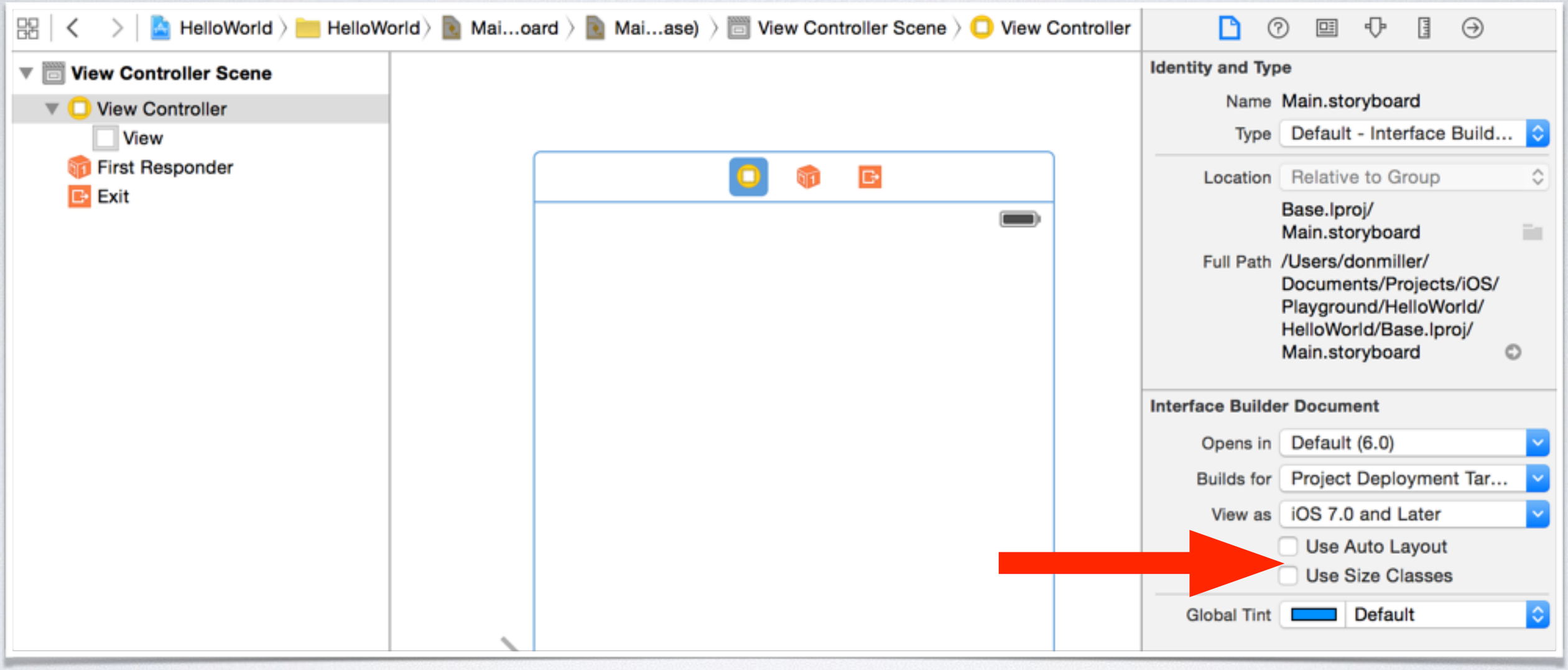
Create A New Single View Application



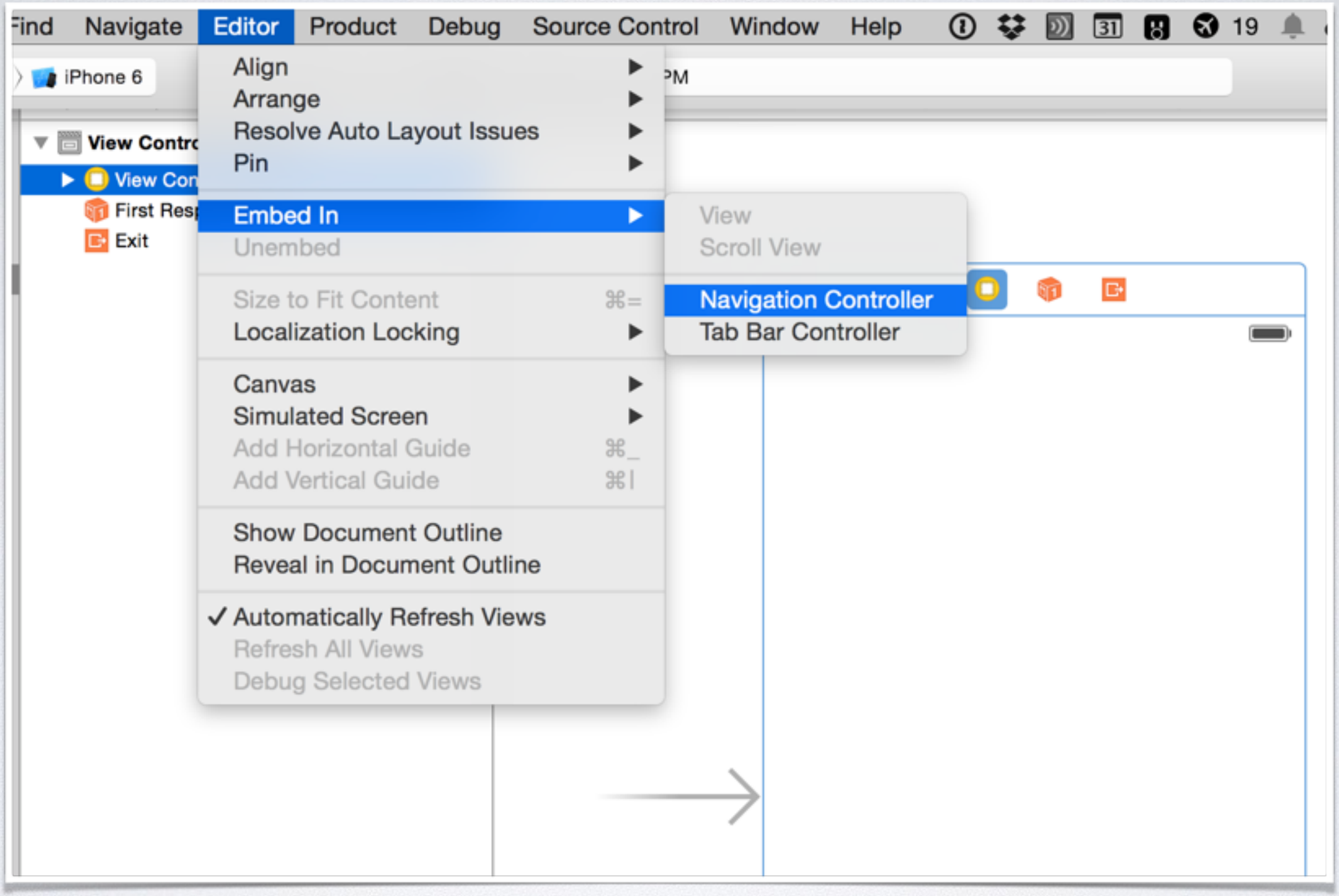
Enter the Product Name



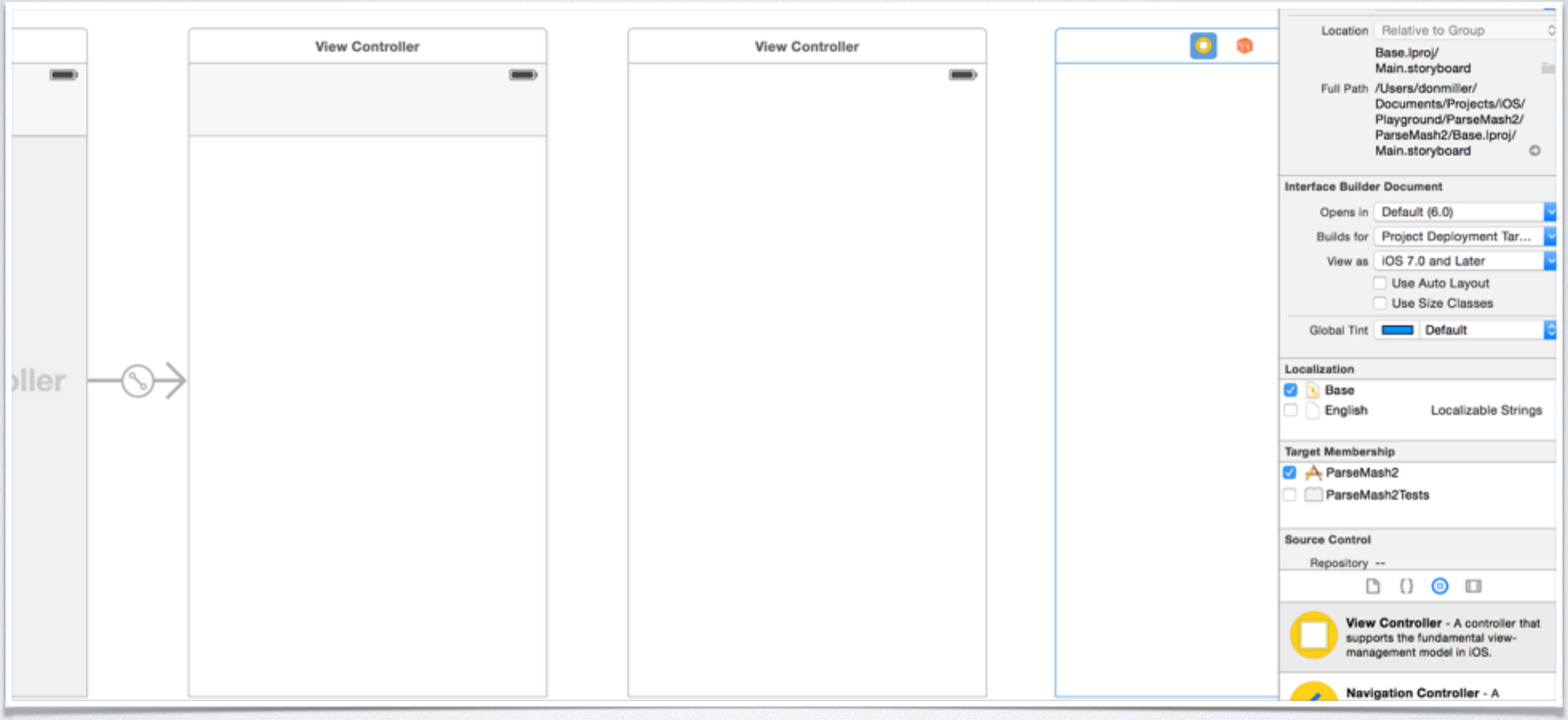
Turn Off Size Classes and Auto Layout



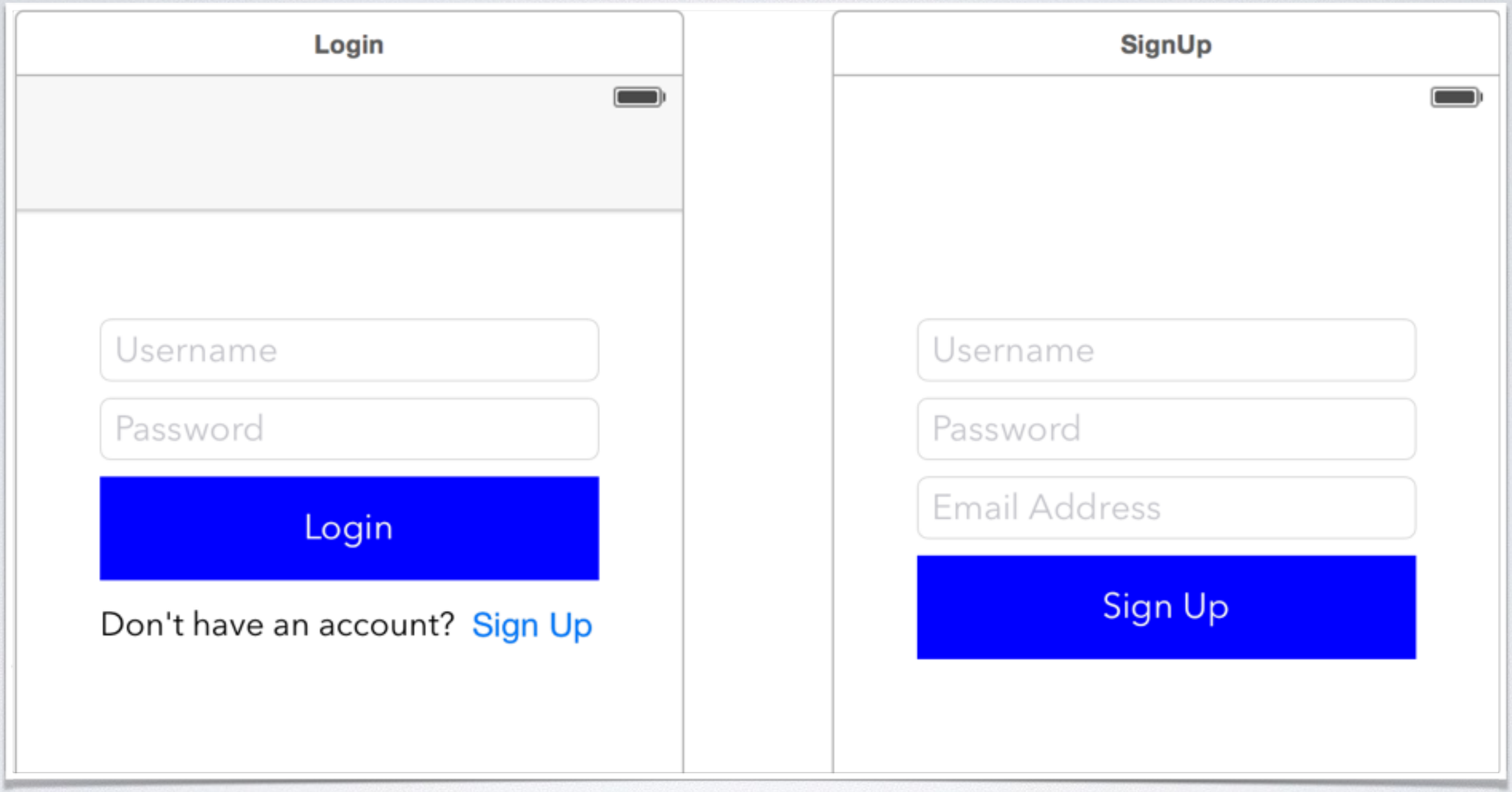
Embed Main View Inside Navigation



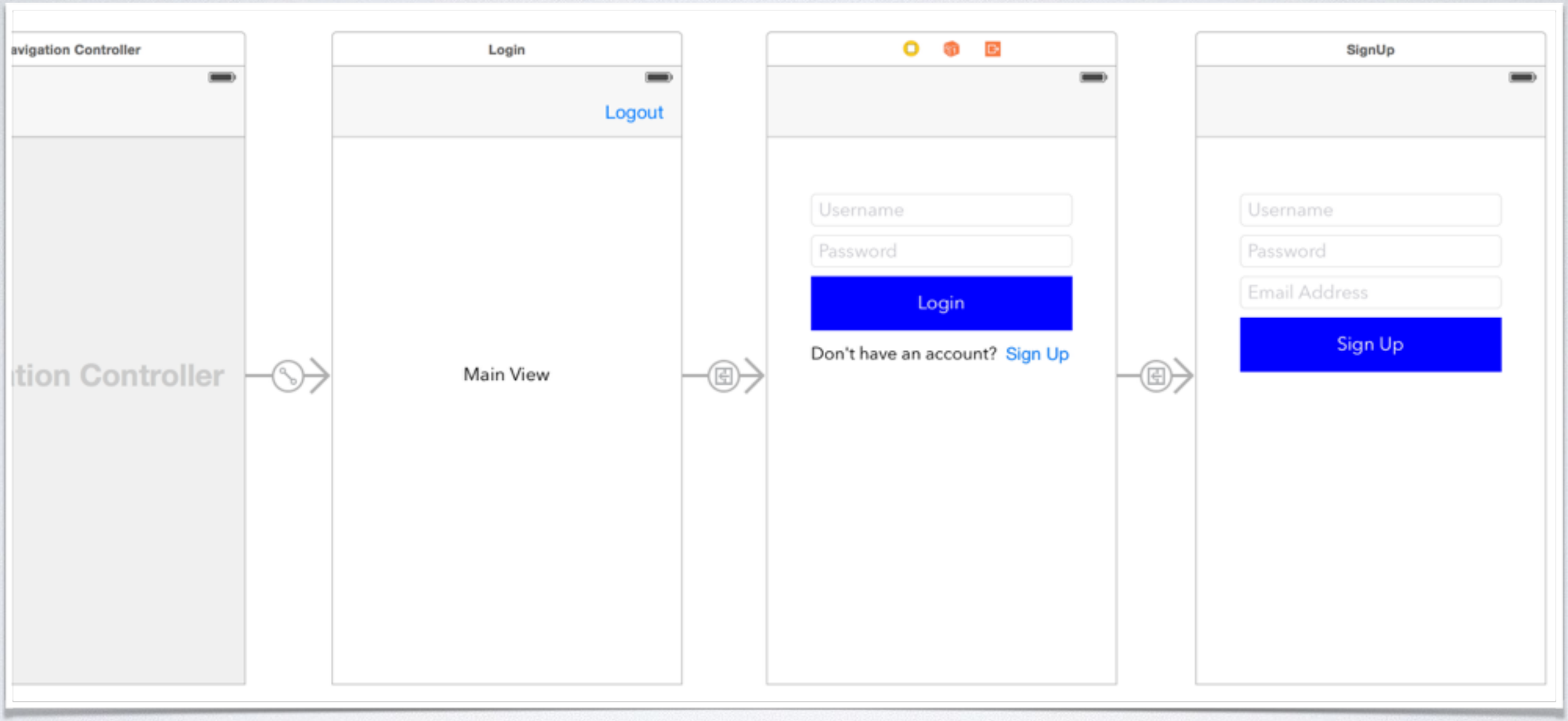
Create 2 Additional View Controllers



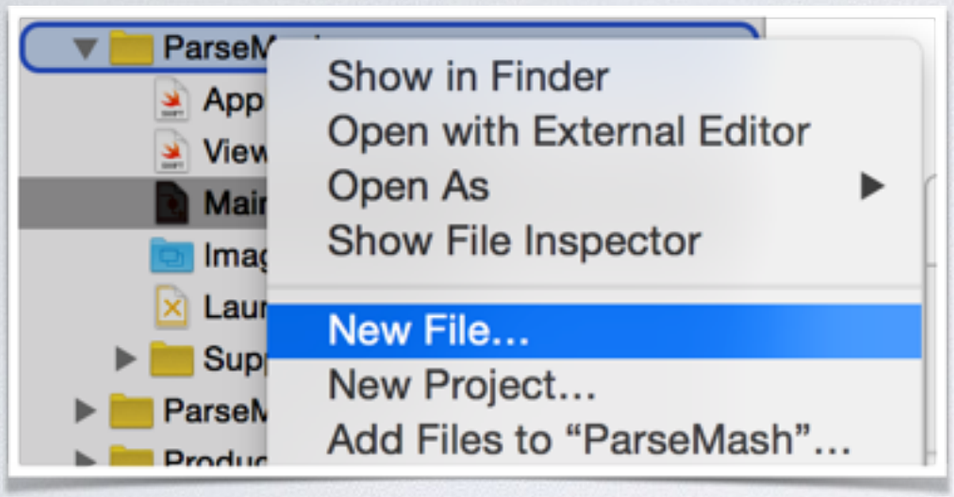
Add TextFields and Buttons to the Login and Signup



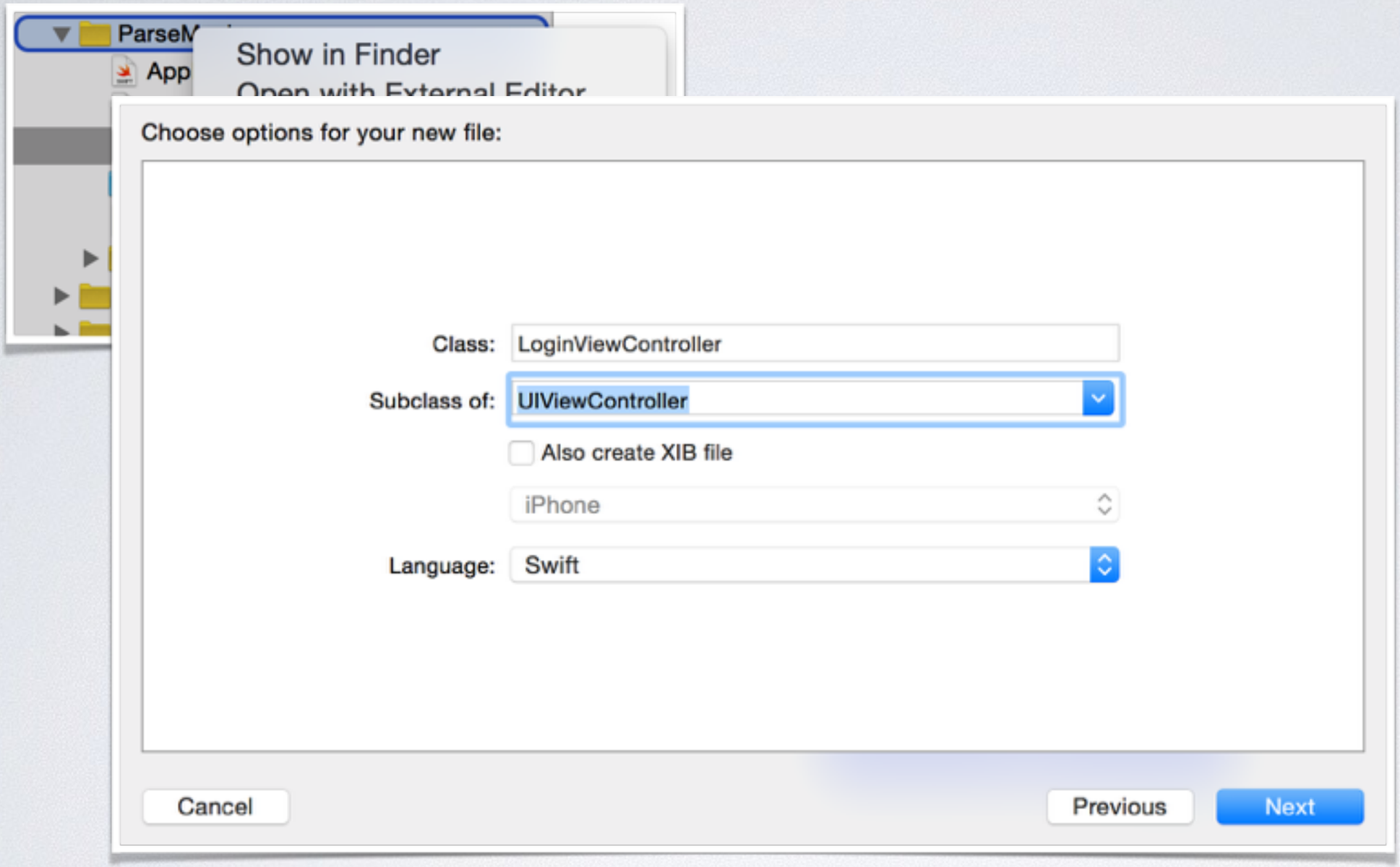
Add Logout Bar Button and Push Segues



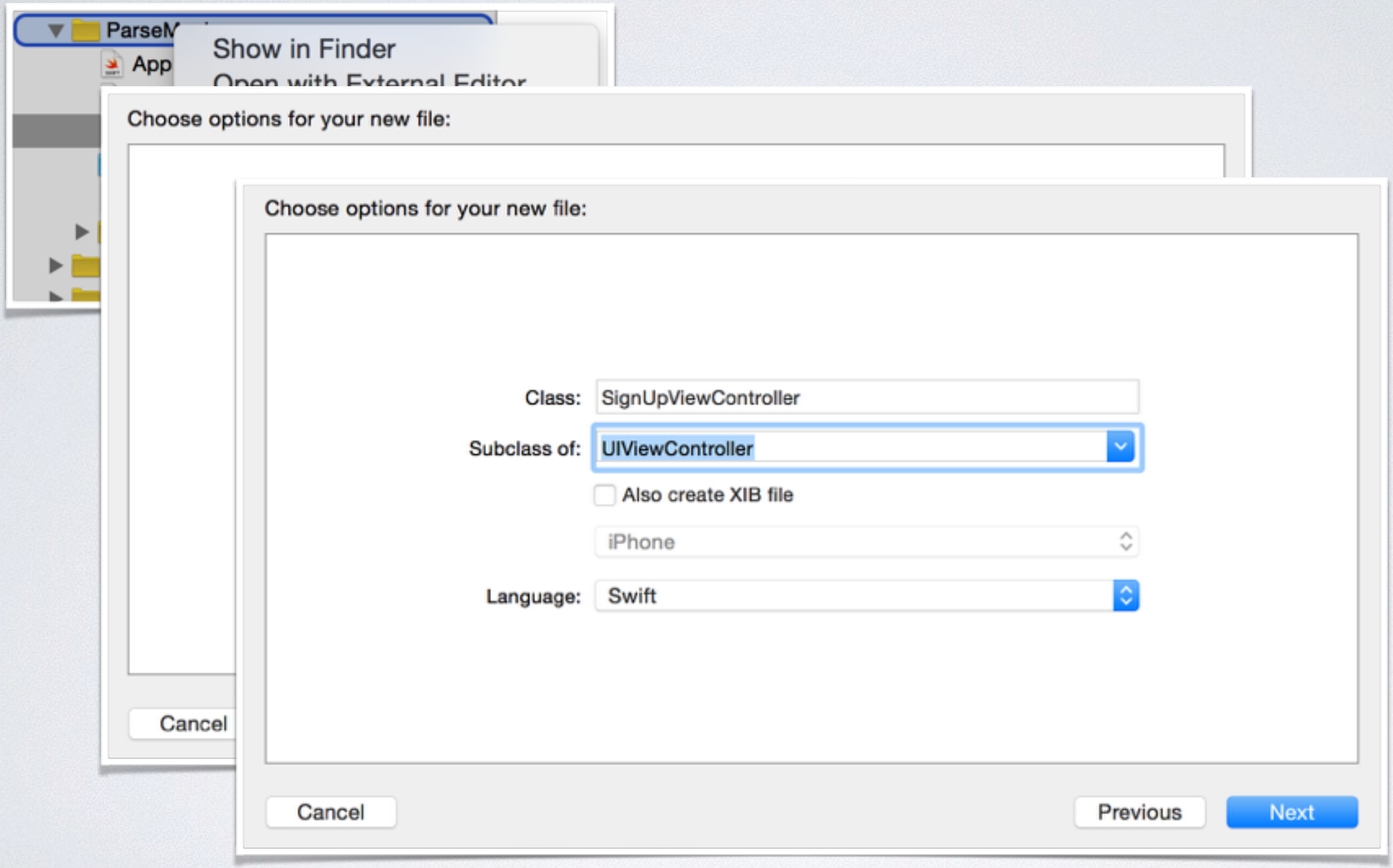
Create Login and Signup View Controllers



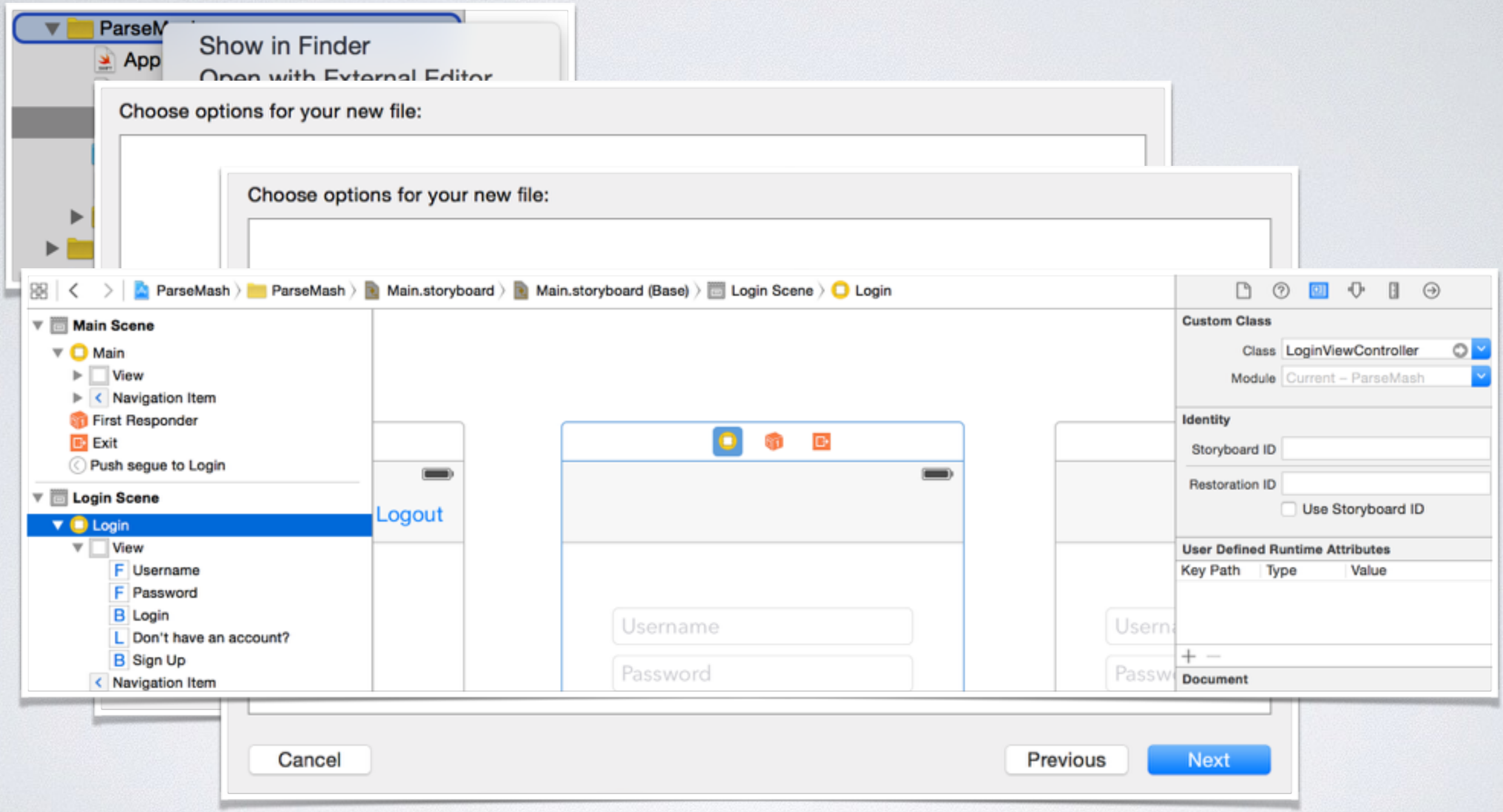
Create Login and Signup View Controllers



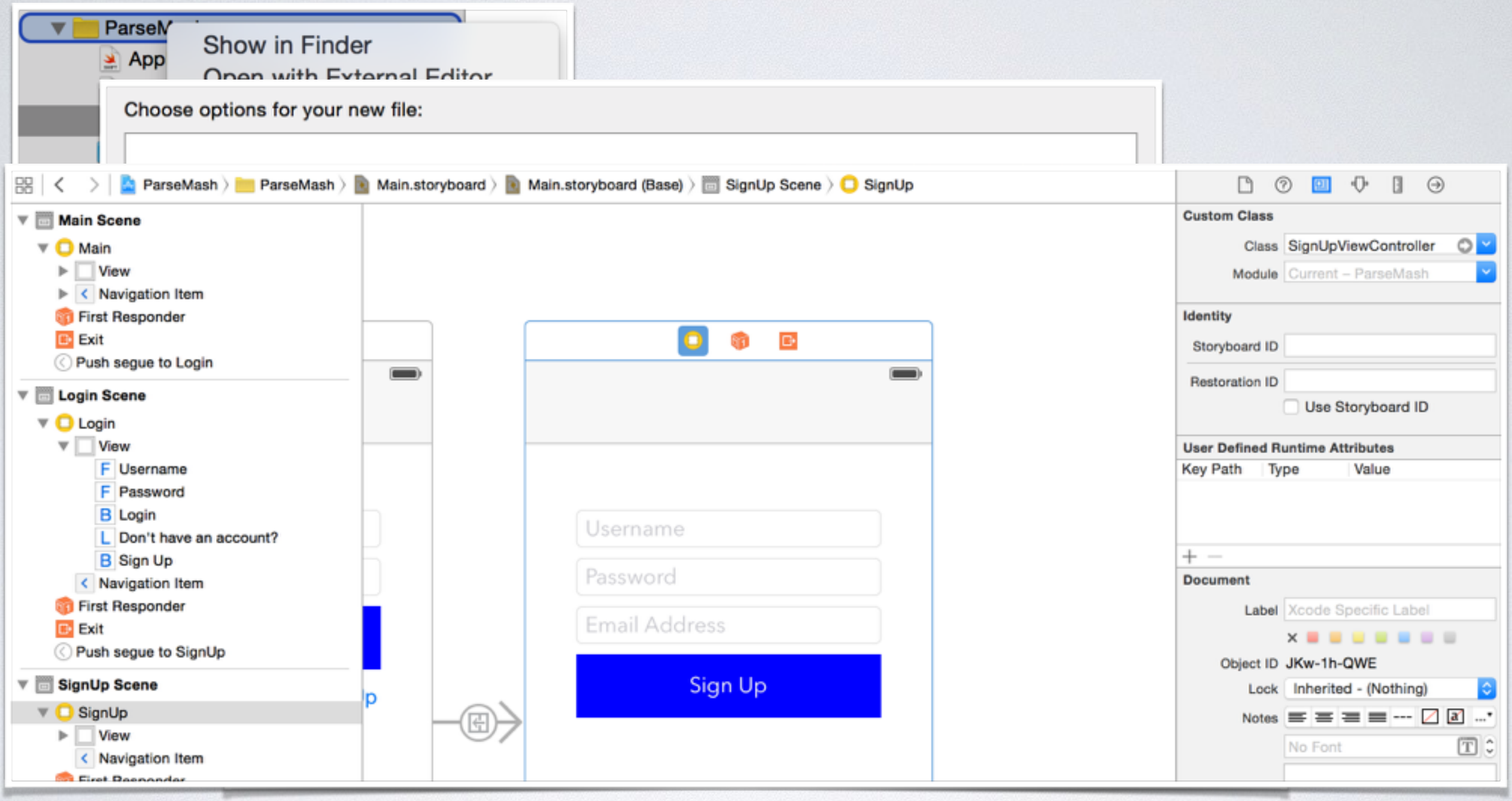
Create Login and Signup View Controllers



Create Login and Signup View Controllers



Create Login and Signup View Controllers

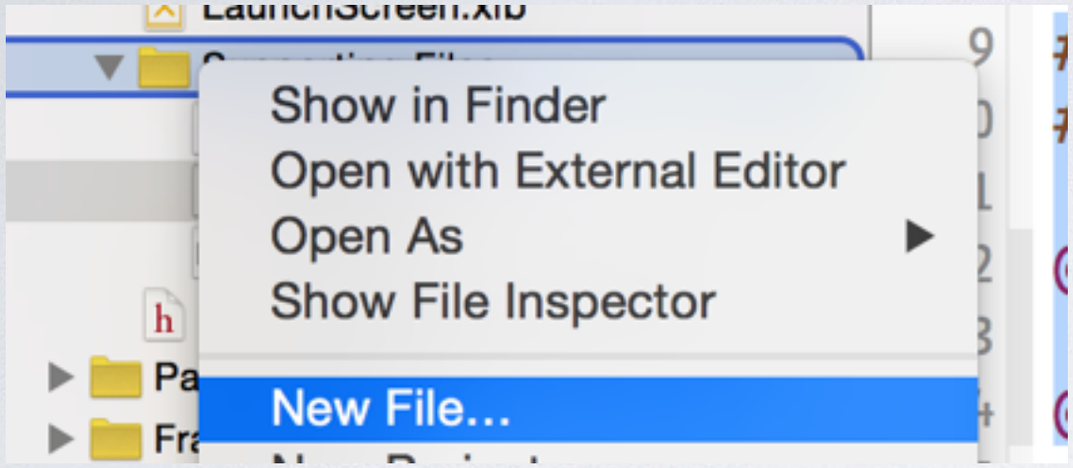


GroundSpeed™
rapid web + mobile software

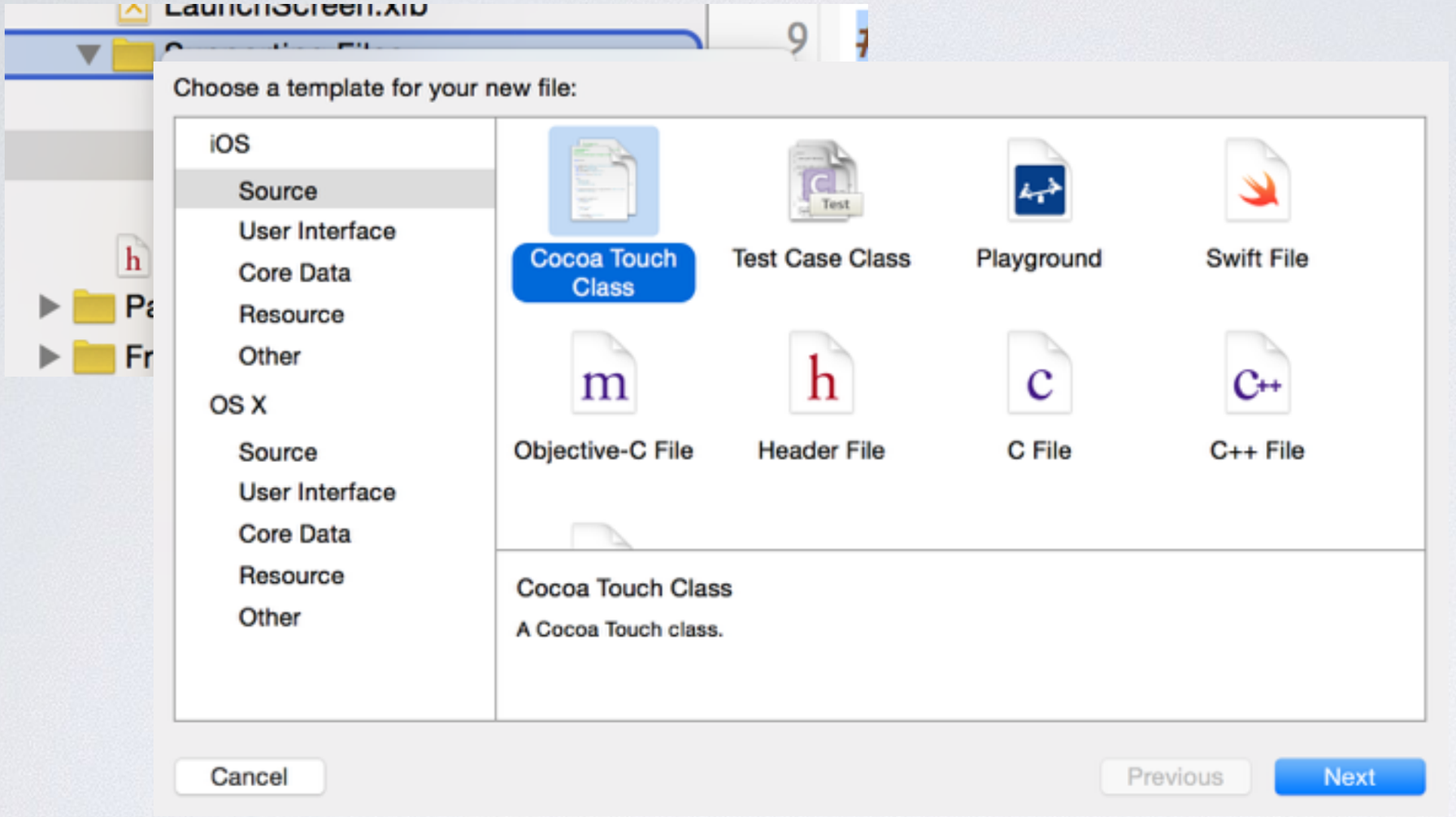
Add the Needed Parse Framework and Prerequisites
The red are required by Parse, but the screenshot is required for our needs

- AudioToolbox.framework
- CFNetwork.framework
- CoreGraphics.framework
- CoreLocation.framework
- libz.dylib
- libsqlite3.dylib
- MobileCoreServices.framework
- QuartzCore.framework
- Security.framework
- StoreKit.framework
- SystemConfiguration.framework
- Bolts.framework
- Parse.framework (obviously)

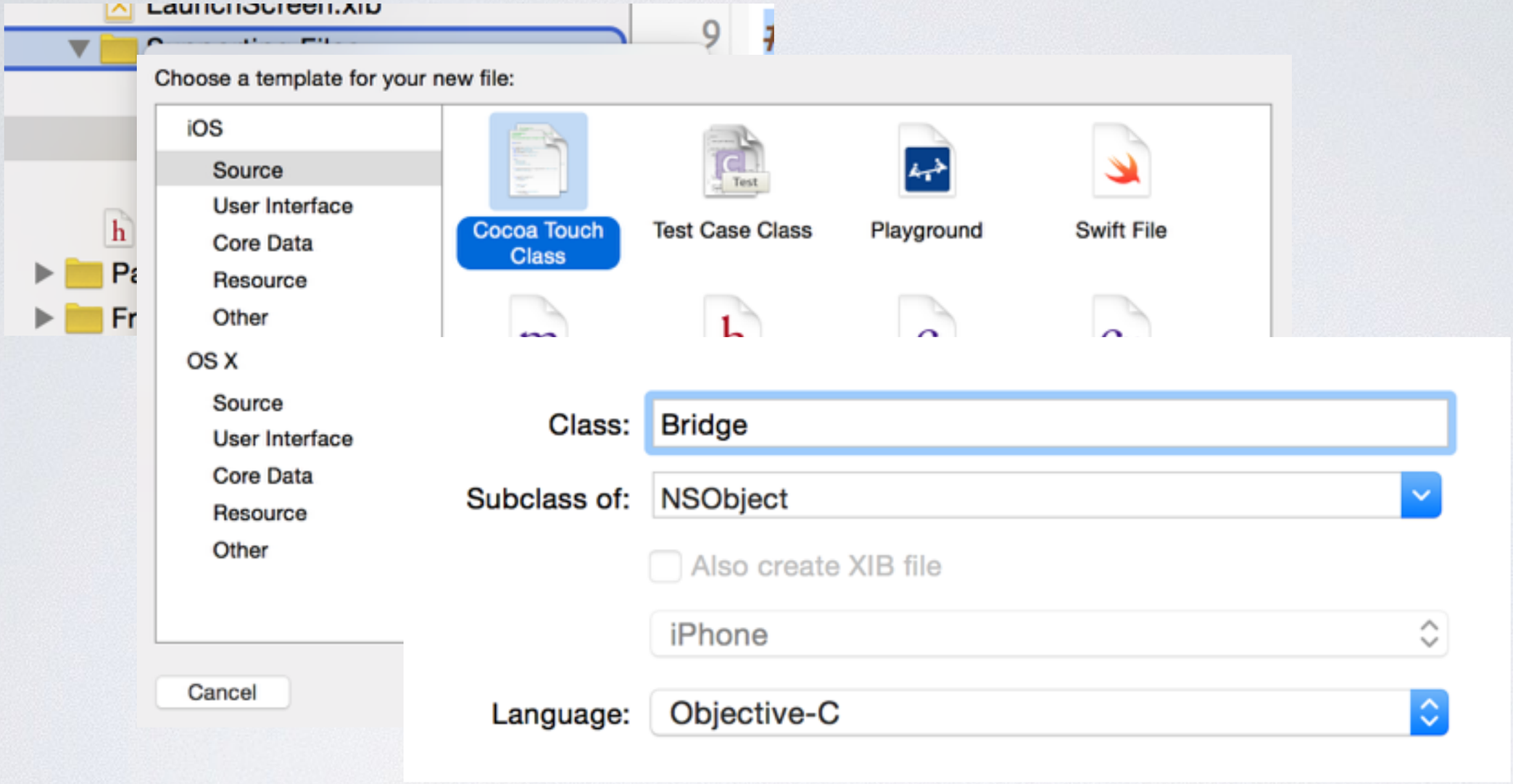
Create Objective-C Bridge



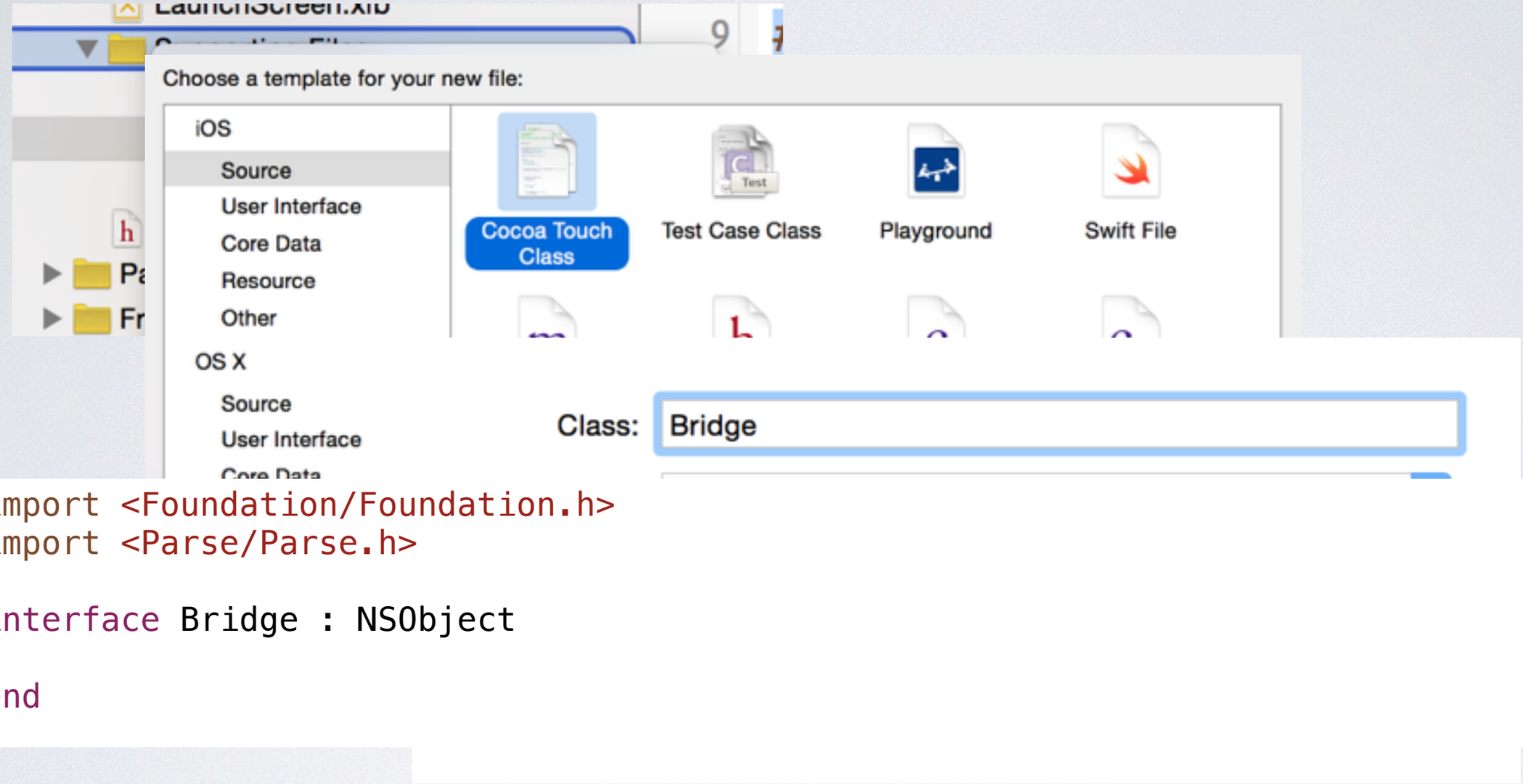
Create Objective-C Bridge



Create Objective-C Bridge

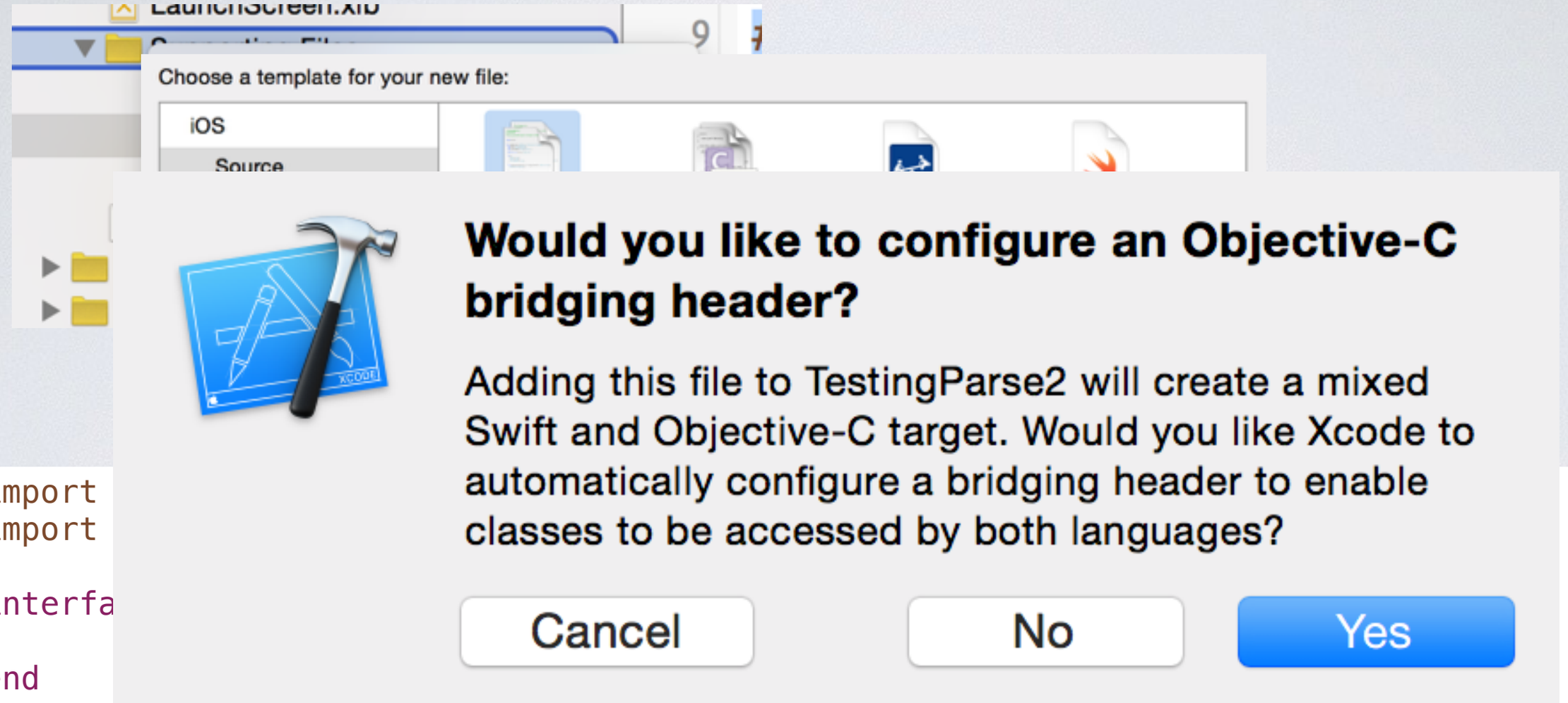


Create Objective-C Bridge



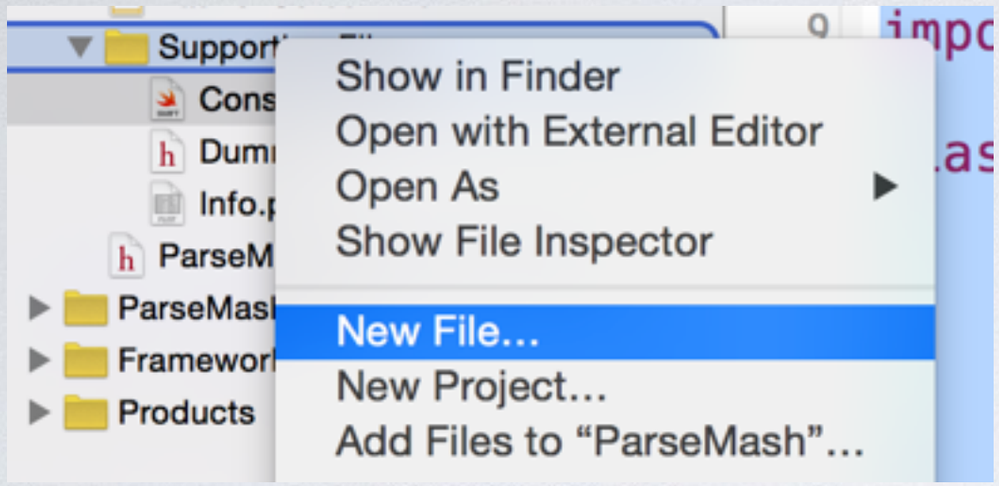

```
#import <Foundation/Foundation.h>
#import <Parse/Parse.h>

@interface Bridge : NSObject

@end
```

Create Objective-C Bridge

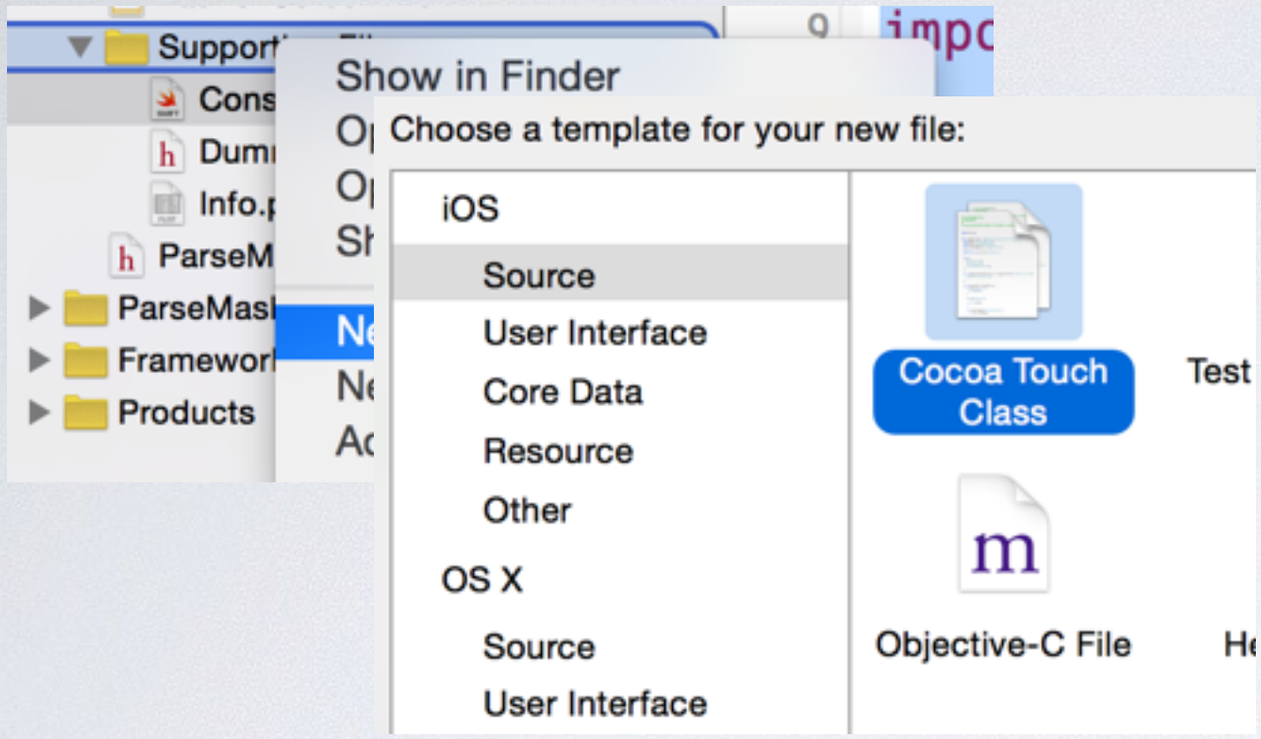


GroundSpeed™
rapid web + mobile software

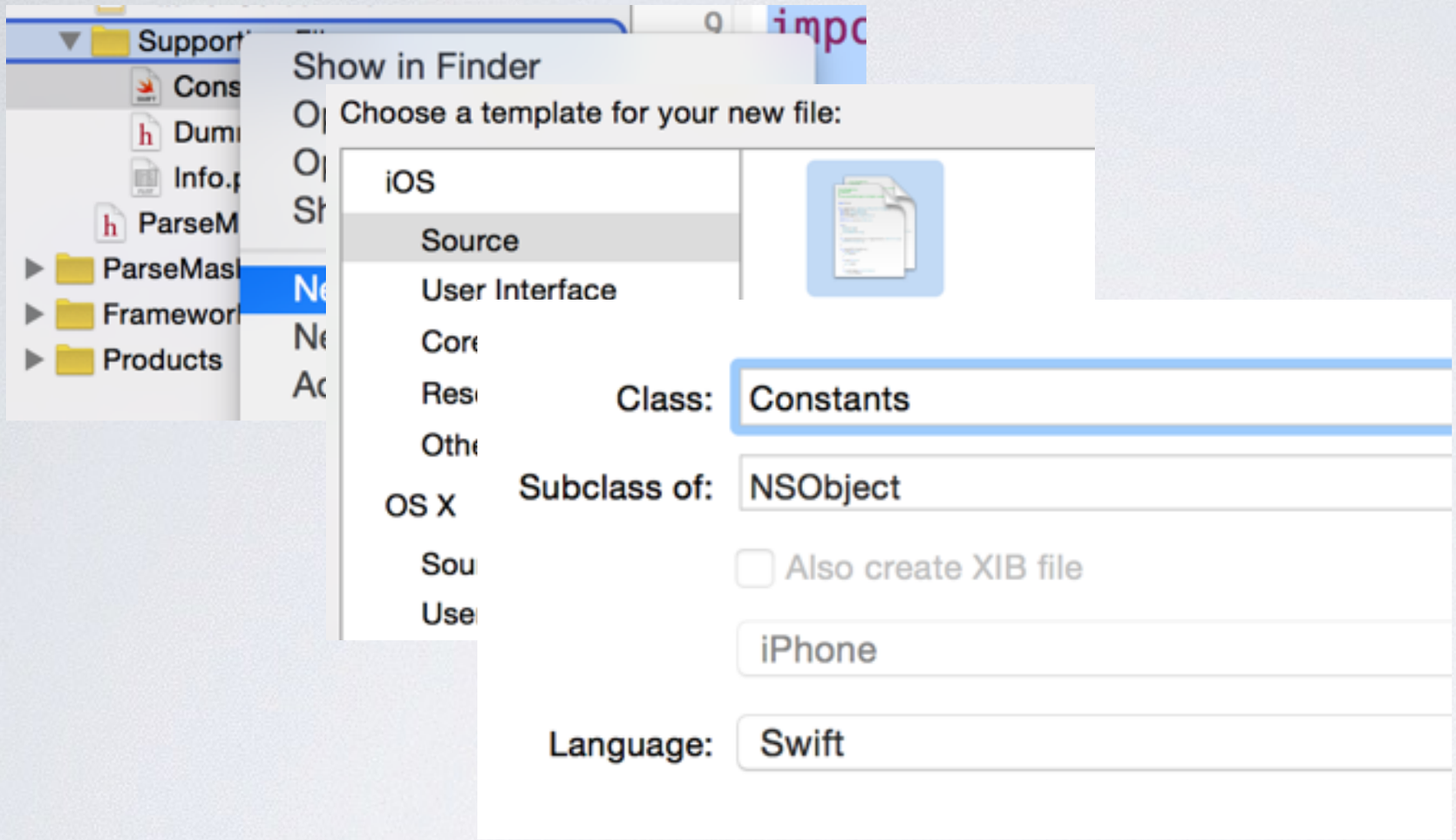
Create a Constants Class



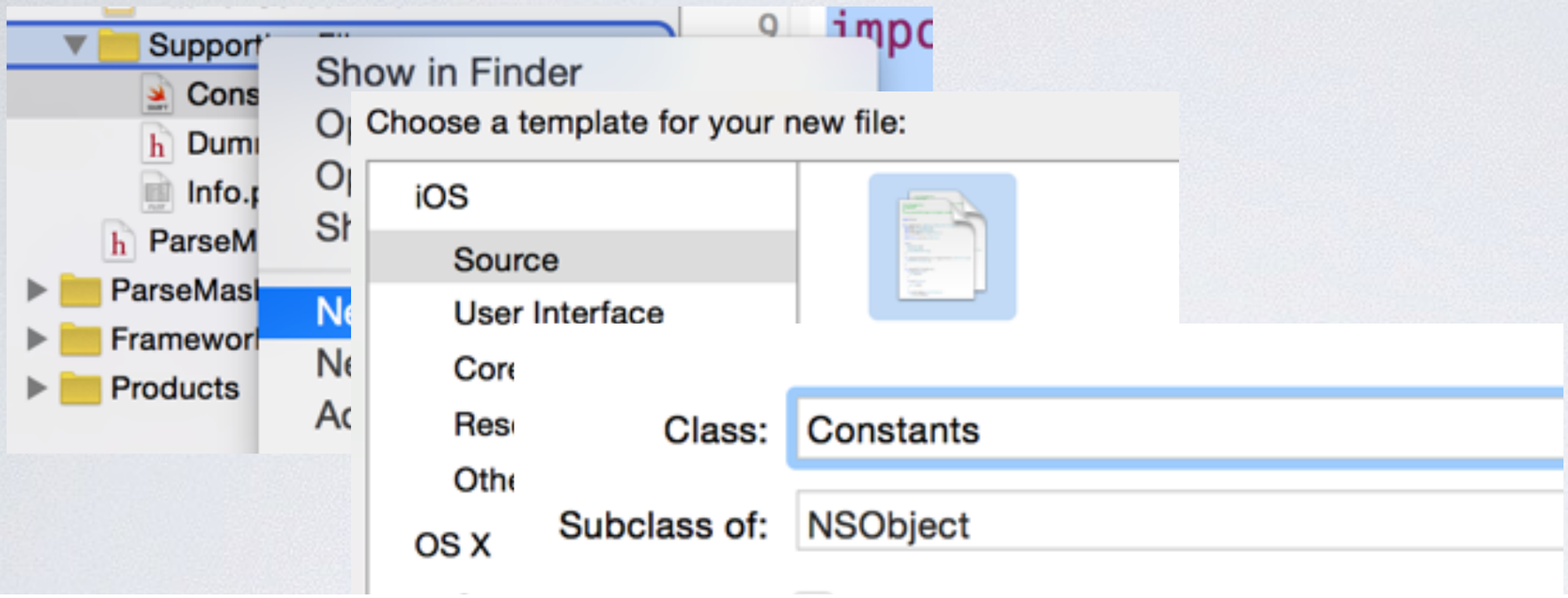
Create a Constants Class



Create a Constants Class

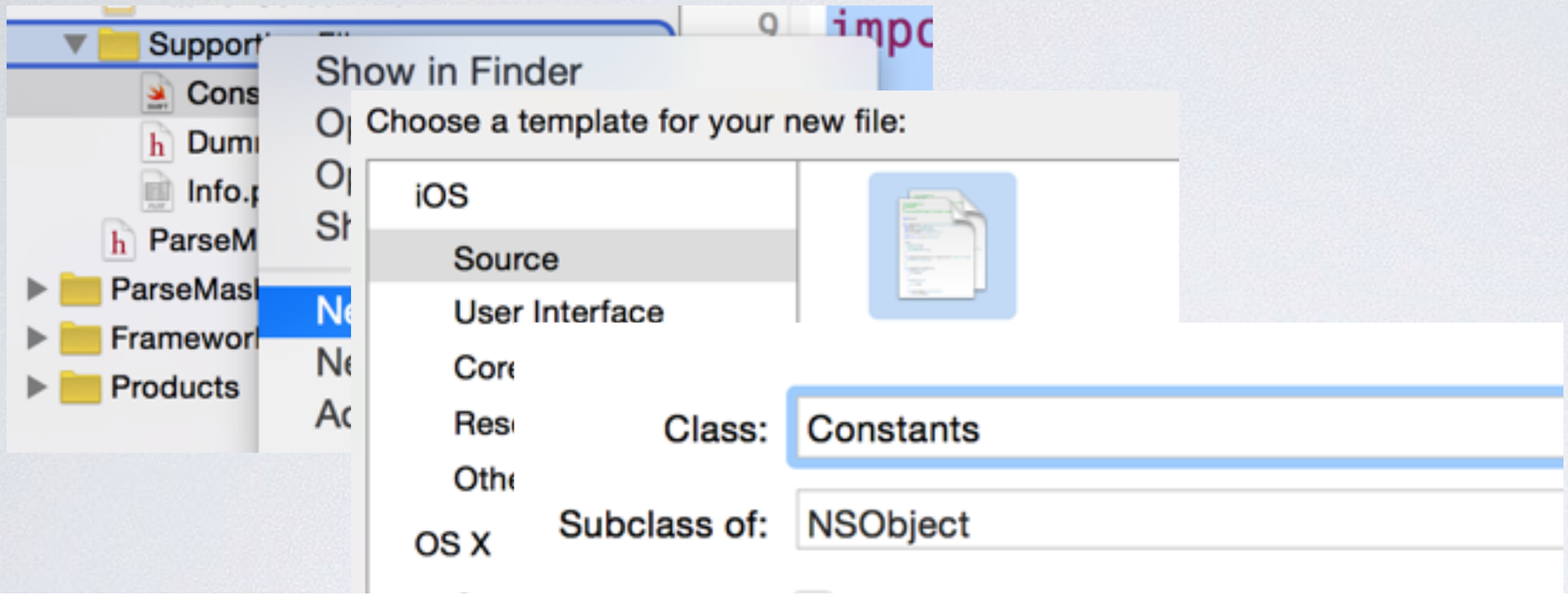


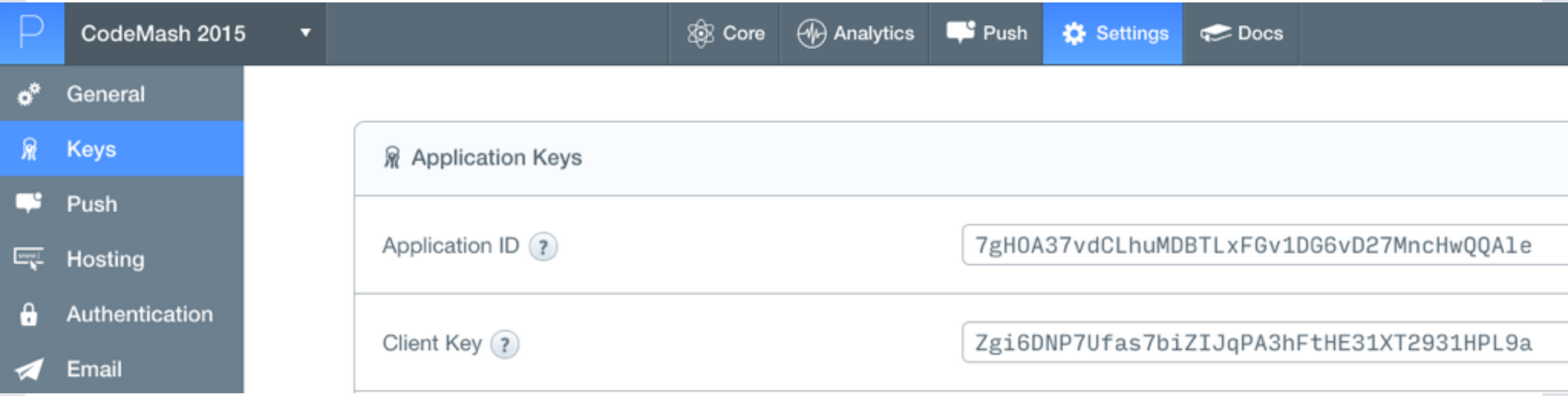
Create a Constants Class




```
import UIKit

class Constants: NSObject {
    let kParseClassName = "Whiskey"
    let kApplicationId  = “Copy and paste from settings"
    let kClientKey      = “Copy and paste from settings"

}
```

Create a Constants Class





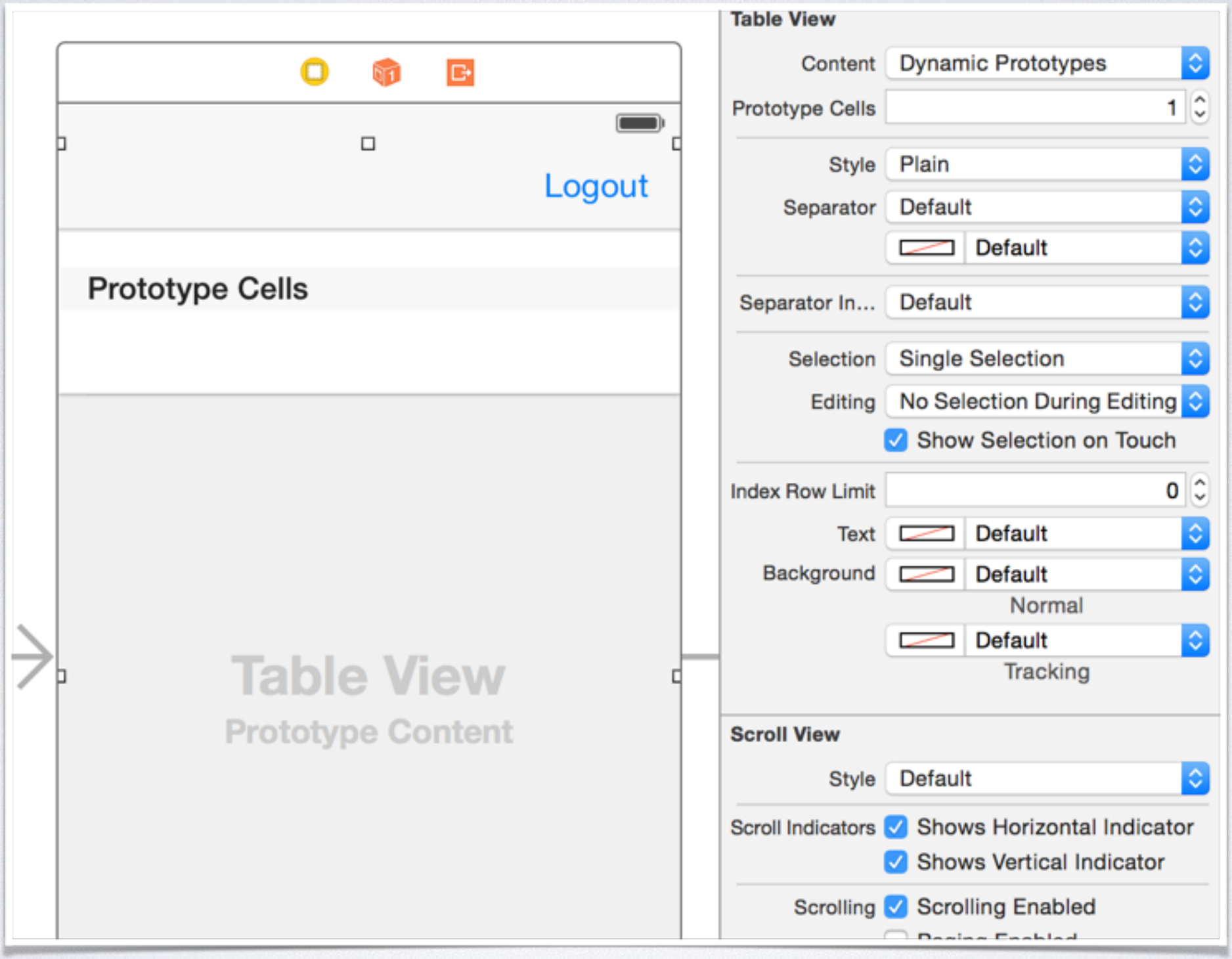
GroundSpeed™
rapid web + mobile software

Add Parse to the App Delegate

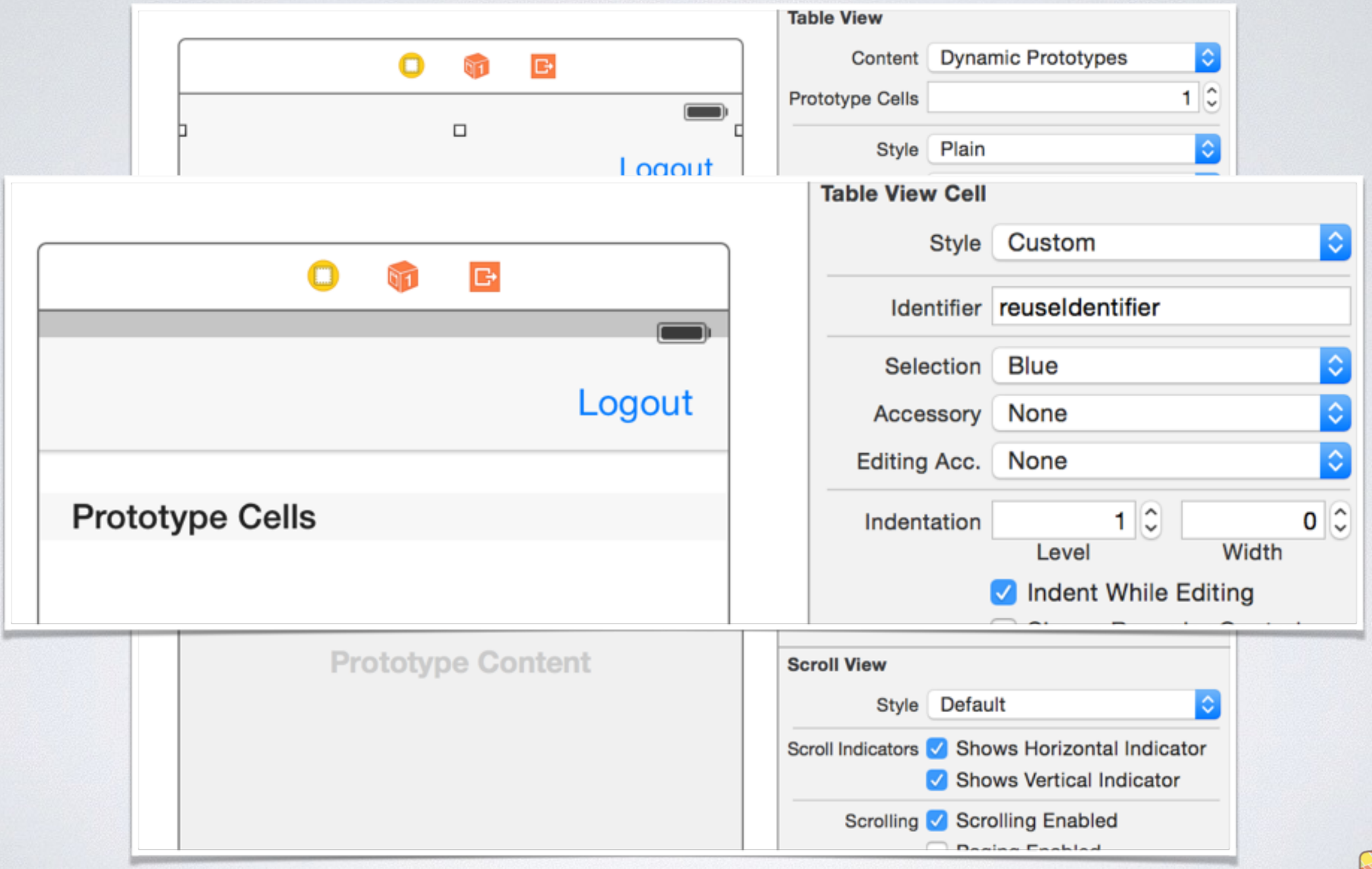
Add Parse to the App Delegate

```
import Parse

@UIApplicationMain
class AppDelegate: UIResponder, UIApplicationDelegate {

    var window: UIWindow?
    let c = Constants()

    func application(application: UIApplication, didFinishLaunchingWithOptions
launchOptions: [NSObject: AnyObject]?) -> Bool {
        // Override point for customization after application launch.
        Parse.setApplicationId(c.kApplicationId, clientKey: c.kClientKey)
        return true
    }
```

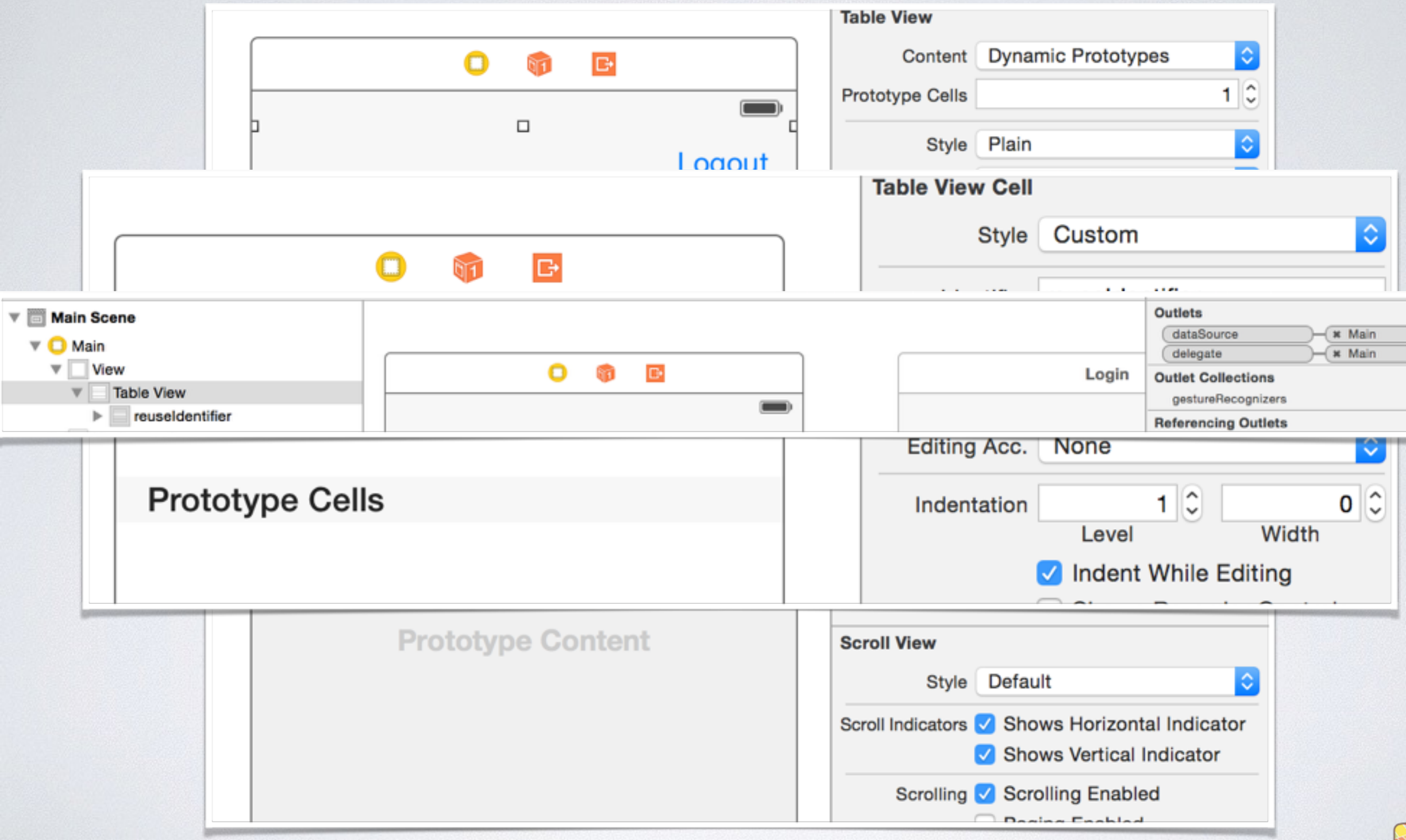
Add a TableView to Your Main Controller



Add a TableView to Your Main Controller



Add a TableView to Your Main Controller



Program Your Main ViewController

```
import UIKit
import Parse

class MainViewController: UIViewController, UITableViewDelegate, UITableViewDataSource {
……
}
```

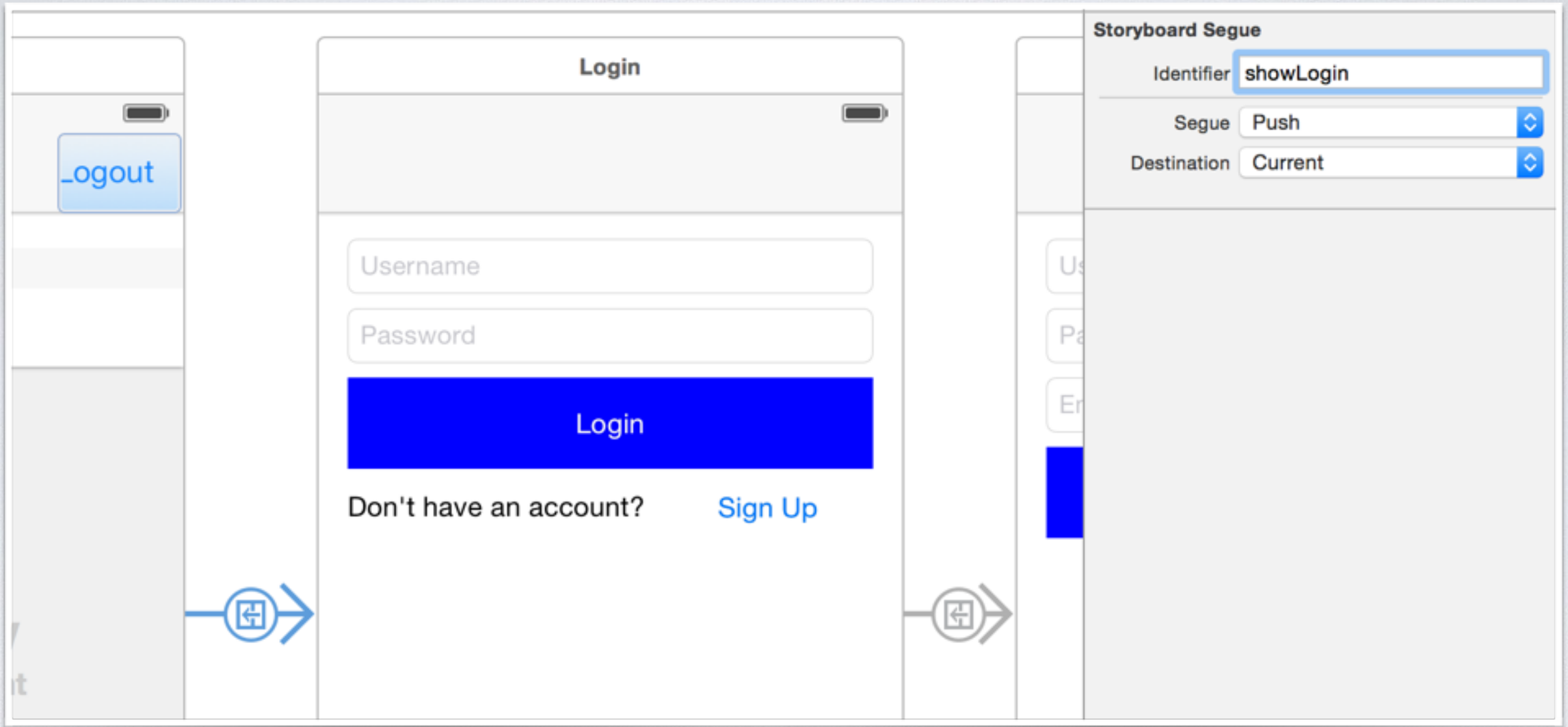
Program Your Main ViewController

```
@IBOutlet var tblWhiskey: UITableView!
let c = Constants()
var whiskeys = [String]()

override func viewDidLoad() {
    super.viewDidLoad()
    // Do any additional setup after loading the view.
    var currentUser = PFUser.currentUser()
    if currentUser != nil {
        var query = PFQuery(className:c.kParseClassName)
        query.findObjectsInBackgroundWithBlock({(NSArray objects, NSError error) in
            if (error != nil) {
                NSLog("error " + error.localizedDescription)
            }
            else {
                NSLog("objects %@", objects as NSArray)
                var whiskeyNames = NSArray(array: objects)

                for whiskey in whiskeyNames {
                    self.whiskeys.append(whiskey.objectForKey("Name") as String);
                }
                self.tblWhiskey.reloadData()
            }
        })
    } else {
        self.performSegueWithIdentifier("showLogin", sender: self)
    }

}
```

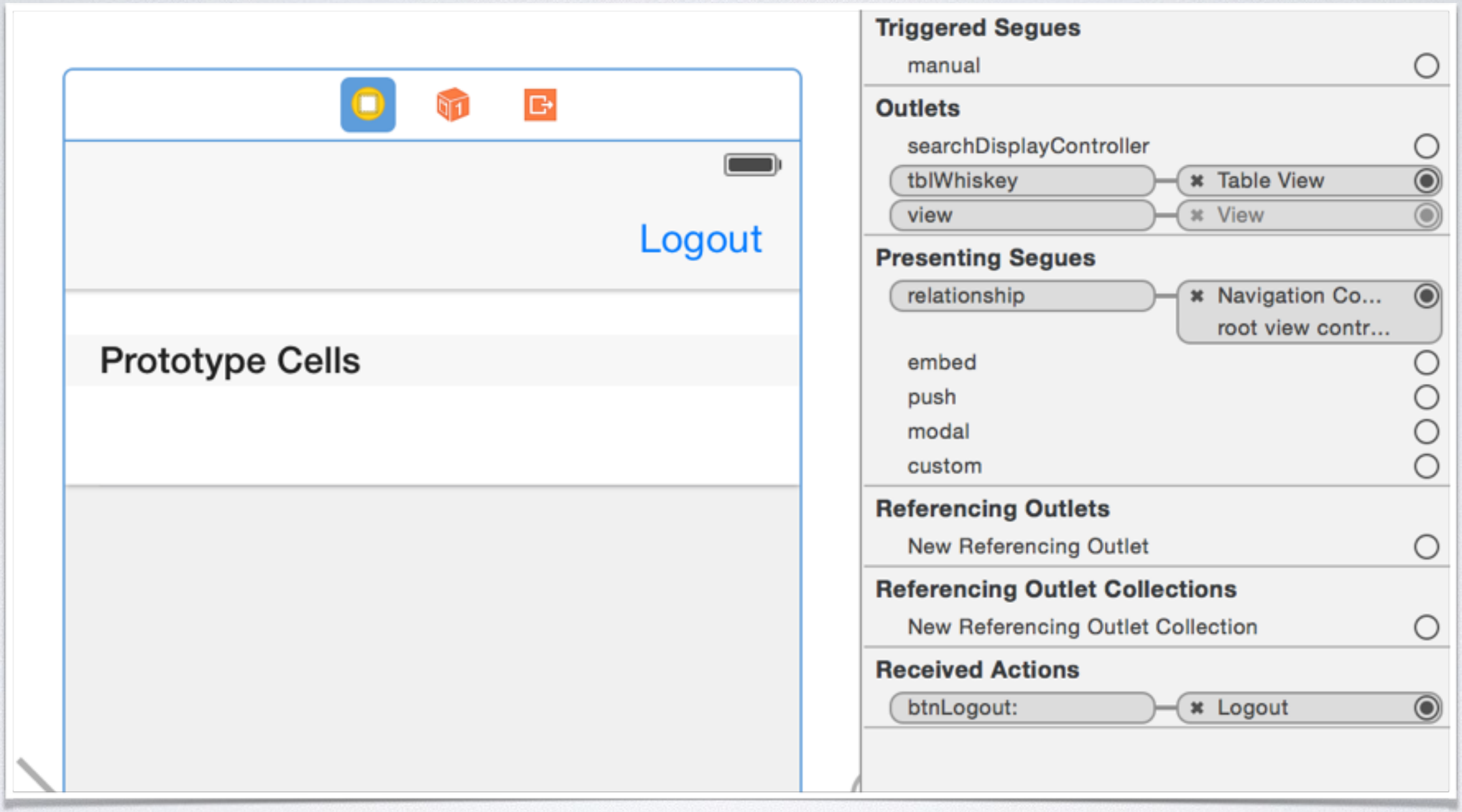
Program Your Main ViewController

```
    override func viewDidAppear(animated: Bool) {
        tblWhiskey.reloadData()
    }

    func tableView(tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
        return self.whiskeys.count
    }

    func tableView(tableView: UITableView, cellForRowAtIndexPath indexPath: NSIndexPath) -
> UITableViewCell {
        let cell = tableView.dequeueReusableCellWithIdentifier("reuseIdentifier",
forIndexPath: indexPath) as UITableViewCell

        // Configure the cell...
        cell.textLabel?.text = self.whiskeys[indexPath.row]

        return cell
    }

    @IBAction func btnLogout(sender: AnyObject) {
        PFUser.logOut()
        PFUser.currentUser().save()
        self.performSegueWithIdentifier("showLogin", sender: self)
    }
```

Update Identifier in Login Segue



Add your IBAction and IBOutlet Connections to the Main View



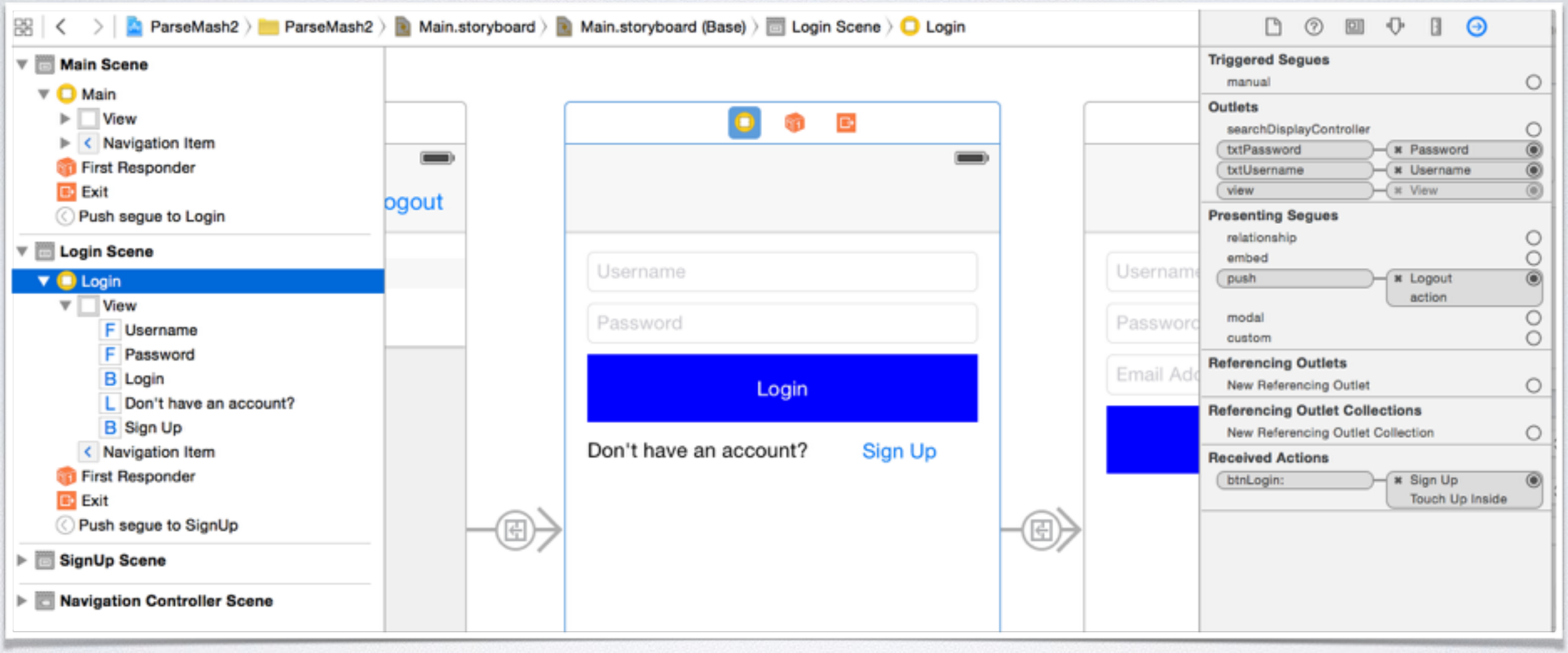
Program Your LoginViewController

Program Your LoginViewController

```
@IBOutlet var txtUsername: UITextField!
@IBOutlet var txtPassword: UITextField!

override func viewDidLoad() {
    super.viewDidLoad()
    self.navigationItem.hidesBackButton = true
}

override func didReceiveMemoryWarning() {
    super.didReceiveMemoryWarning()
    // Dispose of any resources that can be recreated.
}
```

```
@IBAction func btnLogin(sender: AnyObject) {
    PFUser.logInWithUsernameInBackground(txtUsername.text, password:txtPassword.text)
{
        (user: PFUser!, error: NSError!) -> Void in
        if user != nil {
            self.navigationController?.popToRootViewControllerAnimated(true)
        } else {
            let alert = UIAlertView()
            alert.title = "Sorry!"
            alert.message = error.userInfo!["error"] as NSString
            alert.addButtonWithTitle("OK")
            alert.show()
        }
    }
}
```

Add your IBAction and IBOutlet Connections to the Login View



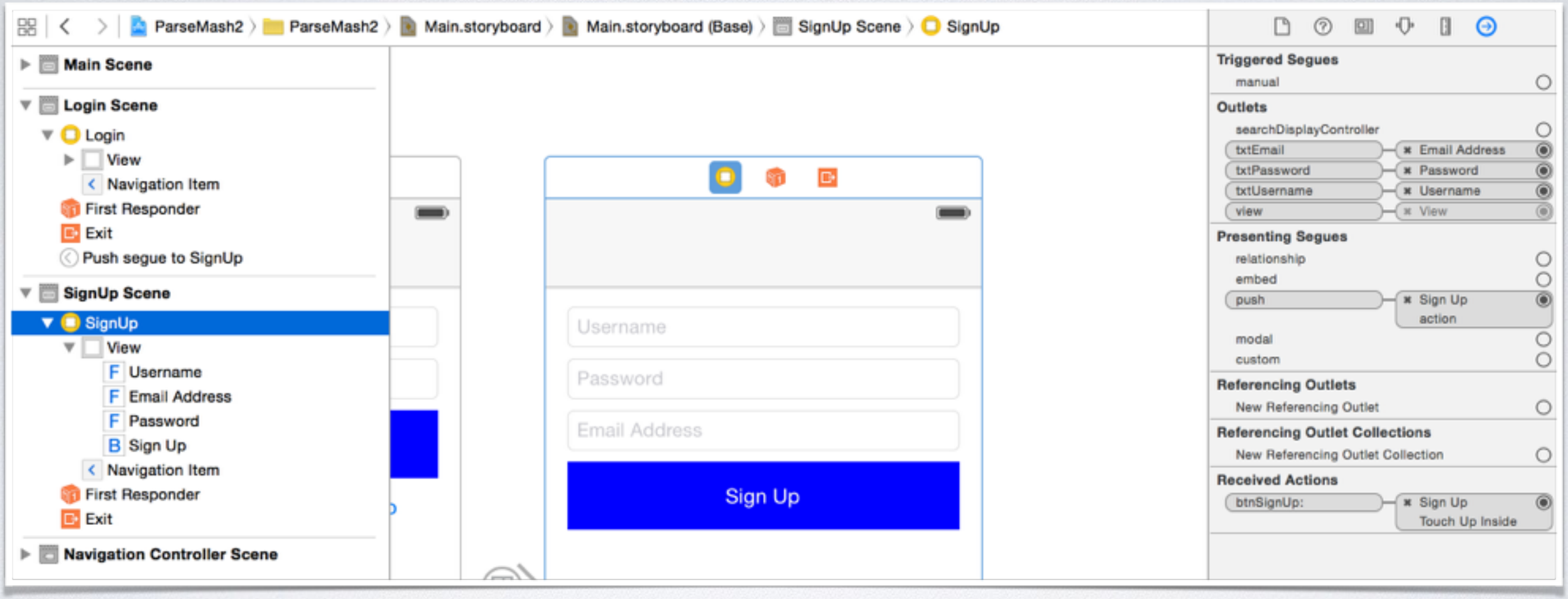
Program Your SignUpViewController

```
    @IBOutlet var txtUsername: UITextField!
    @IBOutlet var txtPassword: UITextField!
    @IBOutlet var txtEmail: UITextField!

    @IBAction func btnSignUp(sender: AnyObject) {
        let username: String =
self.txtUsername.text!.stringByTrimmingCharactersInSet(NSCharacterSet.whitespaceAndNewline
CharacterSet())
        let password: String =
self.txtPassword.text!.stringByTrimmingCharactersInSet(NSCharacterSet.whitespaceAndNewline
CharacterSet())
        let email: String =
self.txtEmail.text!.stringByTrimmingCharactersInSet(NSCharacterSet.whitespaceAndNewlineCha
racterSet())
```

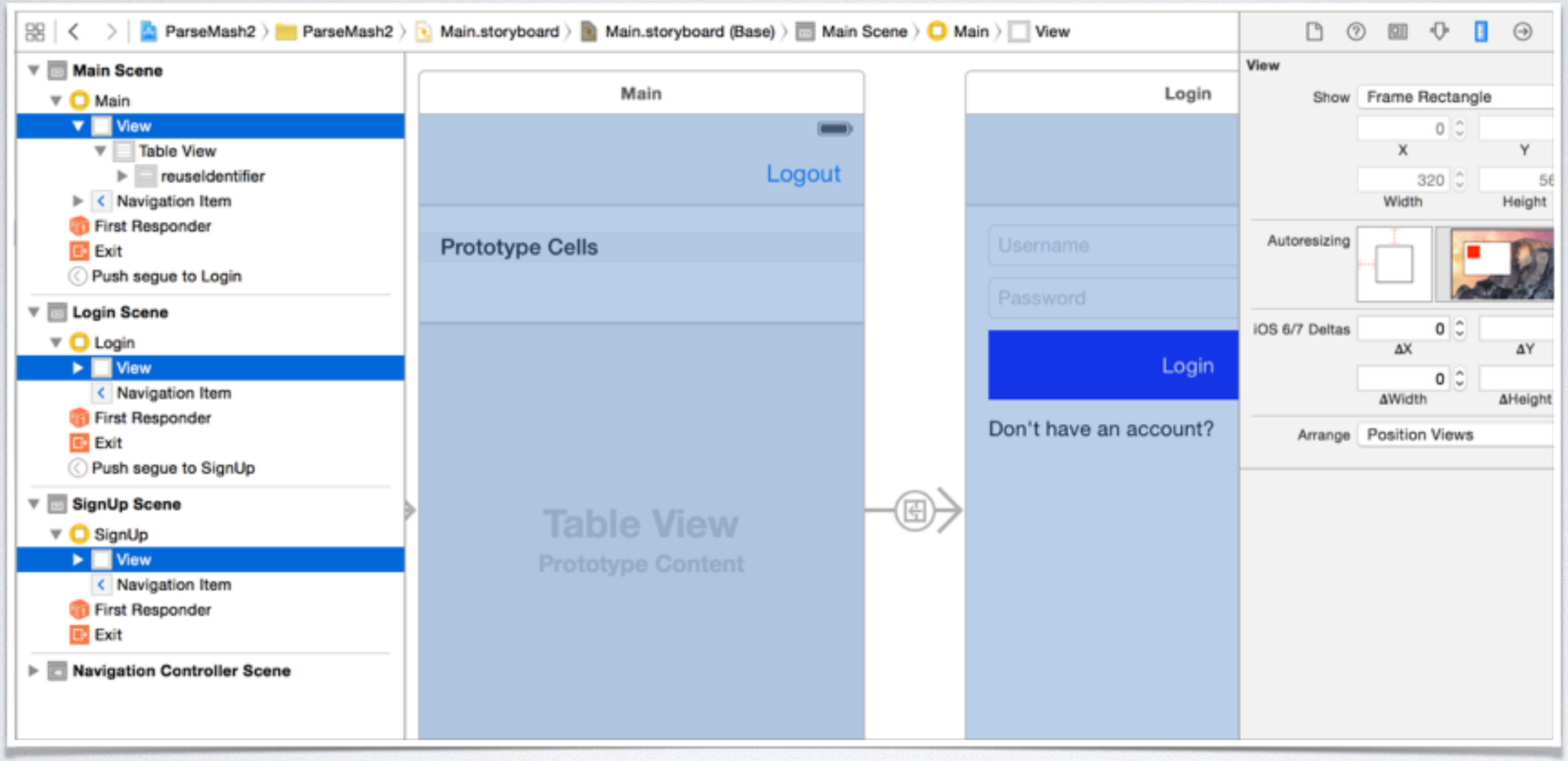
Program Your SignUpViewController

```
    if (countElements(username) == 0 || countElements(password) == 0 || countElements(email) == 0)
    {
        let alert = UIAlertView()
        alert.title = "Oops!"
        alert.message = "Make sure you enter an username, password, and email address!"
        alert.addButtonWithTitle("OK")
        alert.show()
     }
     else
     {
        var newUser = PFUser()
        newUser.username = username
        newUser.password = password
        newUser.email = email

            newUser.signUpInBackgroundWithBlock {
                (succeeded: Bool!, error: NSError!) -> Void in
                if error == nil {
                    self.navigationController?.popToRootViewControllerAnimated(true)
                } else {
                    let alert = UIAlertView()
                    alert.title = "Sorry!"
                    alert.message = error.userInfo!["error"] as NSString
                    alert.addButtonWithTitle("OK")
                    alert.show()
                }
            }
        }
    }
```

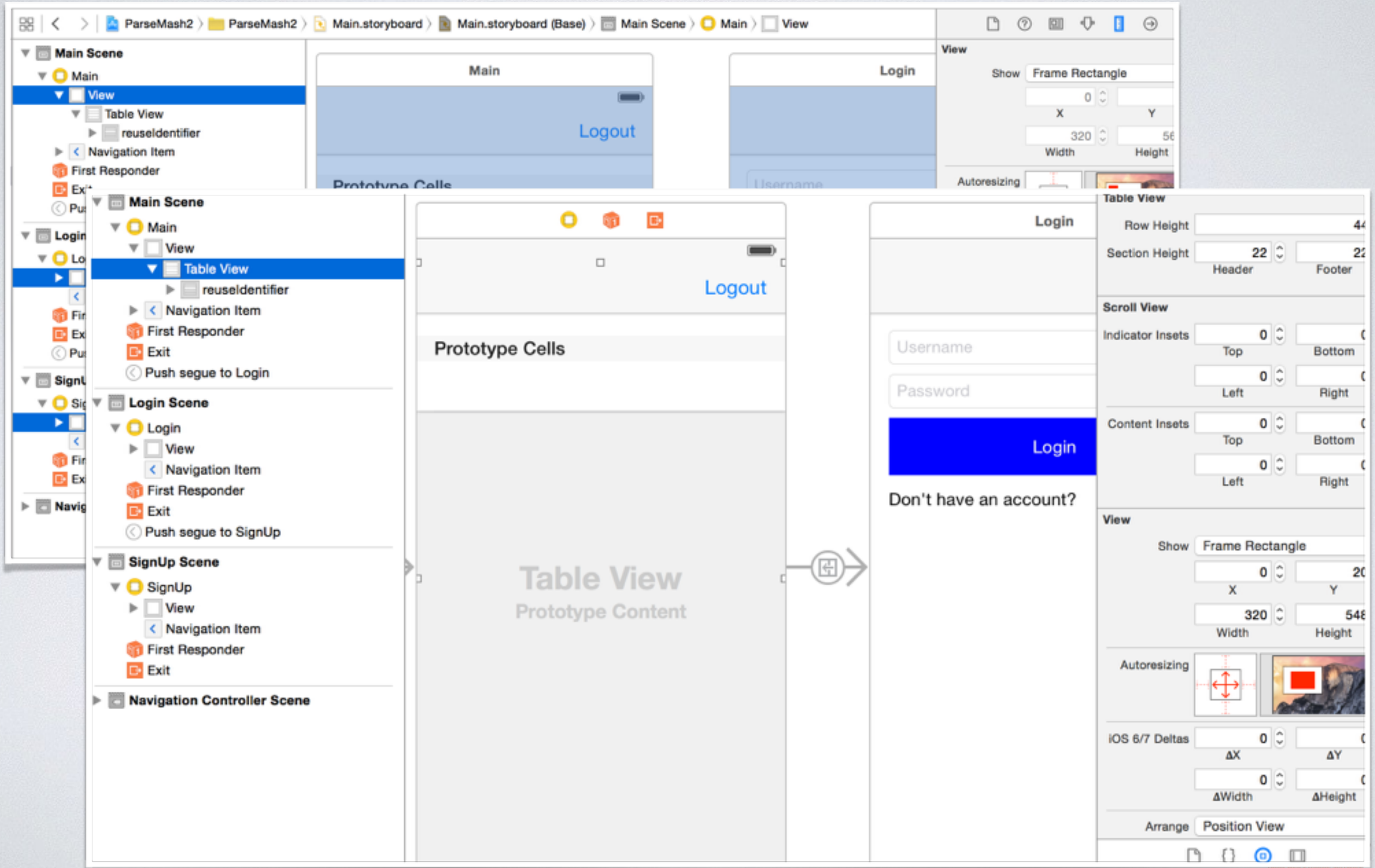
Add your IBAction and IBOutlet Connections to the SignUp View



Clean Up - Because We Aren't Using Auto Layout



Clean Up - Because We Aren't Using Auto Layout



Don Miller

don@groundspeedhq.com

@donmiller

Or follow @groundspeedhq

groundspeedhq.com