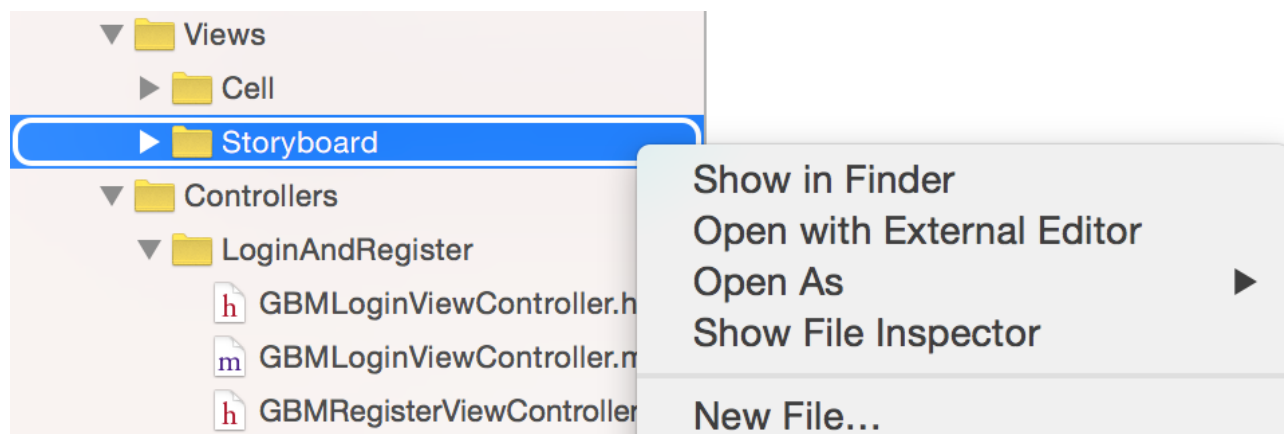


蓦然项目个人信息模块——“我的”

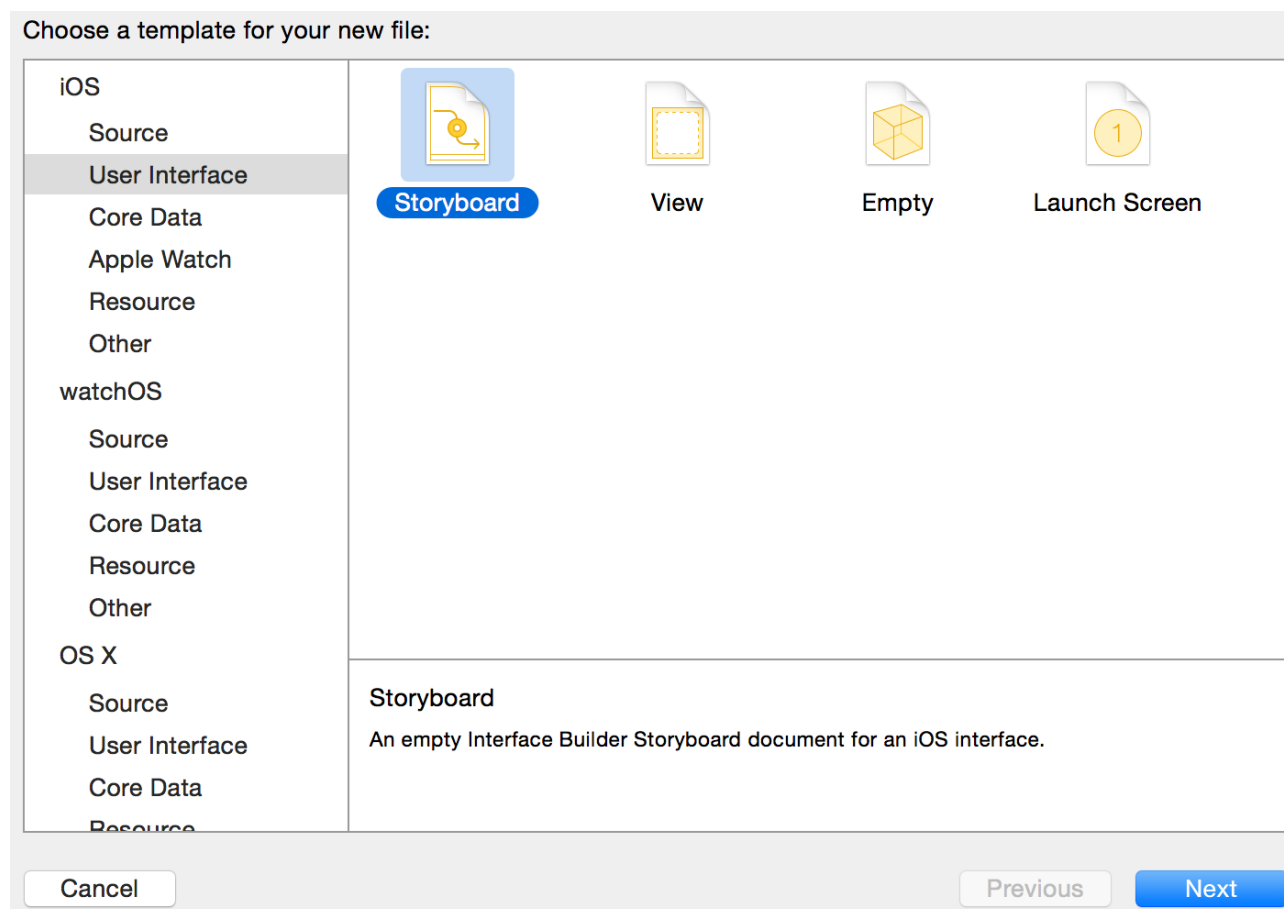
1. 创建TabBar框架

1.1 首先需要创建一个UITabBarController，有两个页面组成，一个是“广场”页面，一个是“我的”个人信息页面。

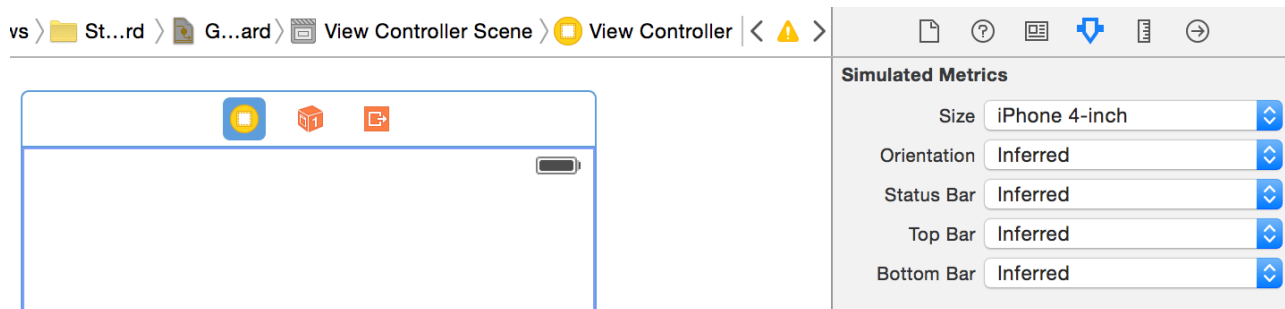
首先在Xcode左侧的导航栏中，选中storyboard文件夹，右键选择“New File”；



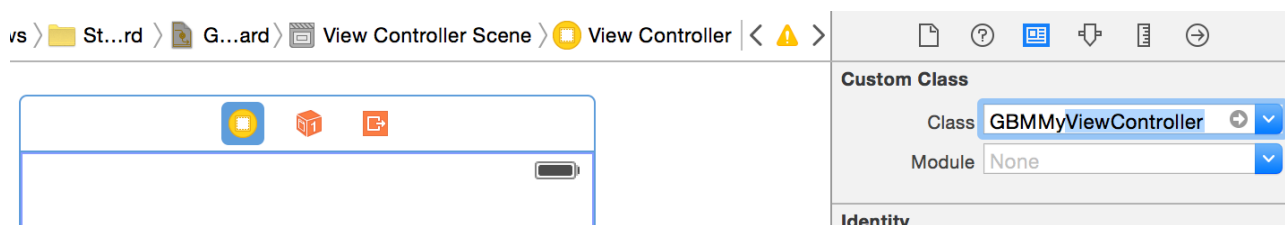
1.2 然后在弹出窗口的“User Interface”中选择“Storyboard”，创建一个名为“GBMMy”的storyboard文件（GBM为项目的前缀，在类和新建的文件名称前面加上）；



1.3 在GBMMy.storyboard中，拖入一个UITableViewController，为了少占屏幕空间，可以在右侧的属性选项中，将模拟器的尺寸设置为4寸



1.4 接下来新建一个名叫“GBMMyViewController”的类，并用来管理刚刚创建的storyboard，在Xcode右侧菜单的第三个“Identity inspector”中，将class设置为“GBMMyViewController”；



1.5 打开“AppDelegate.m”文件，添加loadMainViewWithController: 方法，创建一个包含了UITabBarController，包含两个页面——广场页面和“我的”个人信息页面；

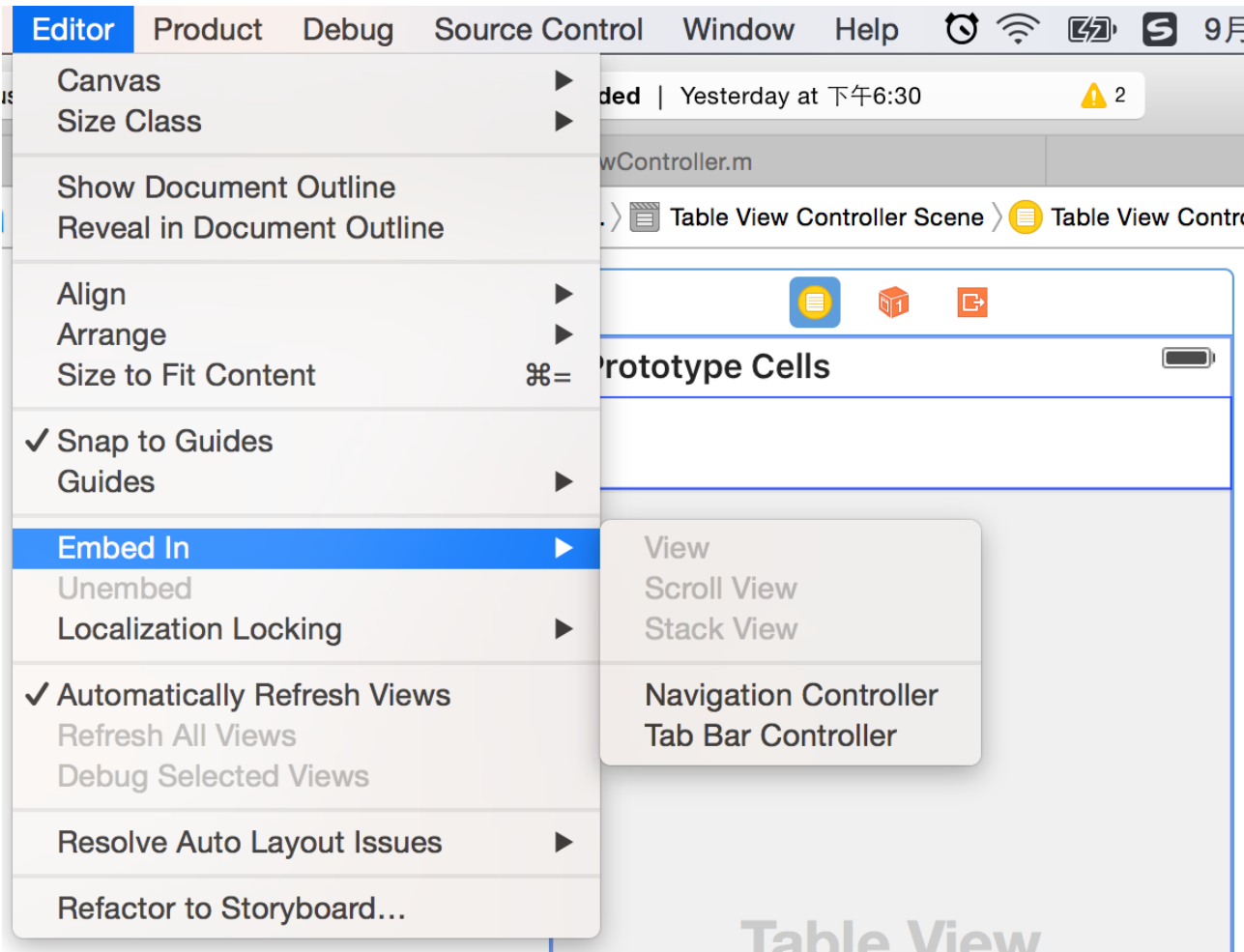
```
- (void)loadMainViewWithController:(UIViewController *)controller
{
    UIStoryboard *squareStoryboard = [UINavigationController storyboardWithName:@"GBMSquare" bundle:[NSBundle mainBundle]];
    GBMSquareViewController *squareVC = [squareStoryboard
    instantiateViewControllerWithIdentifier:@"SquareStoryboard"];
    squareVC.tabBarItem.title = @"广场";
    squareVC.tabBarItem.image = [UIImage imageNamed:@"square"];

    UIStoryboard *myStoryboard
    = [UINavigationController storyboardWithName:@"GBMMy" bundle:[NSBundle mainBundle]];
    GBMMyViewController *myVC = [myStoryboard instantiateViewControllerWithIdentifier:@"MyStoryboard"];
    myVC.tabBarItem.title = @"我的";
    myVC.tabBarItem.image = [UIImage imageNamed:@"my"];

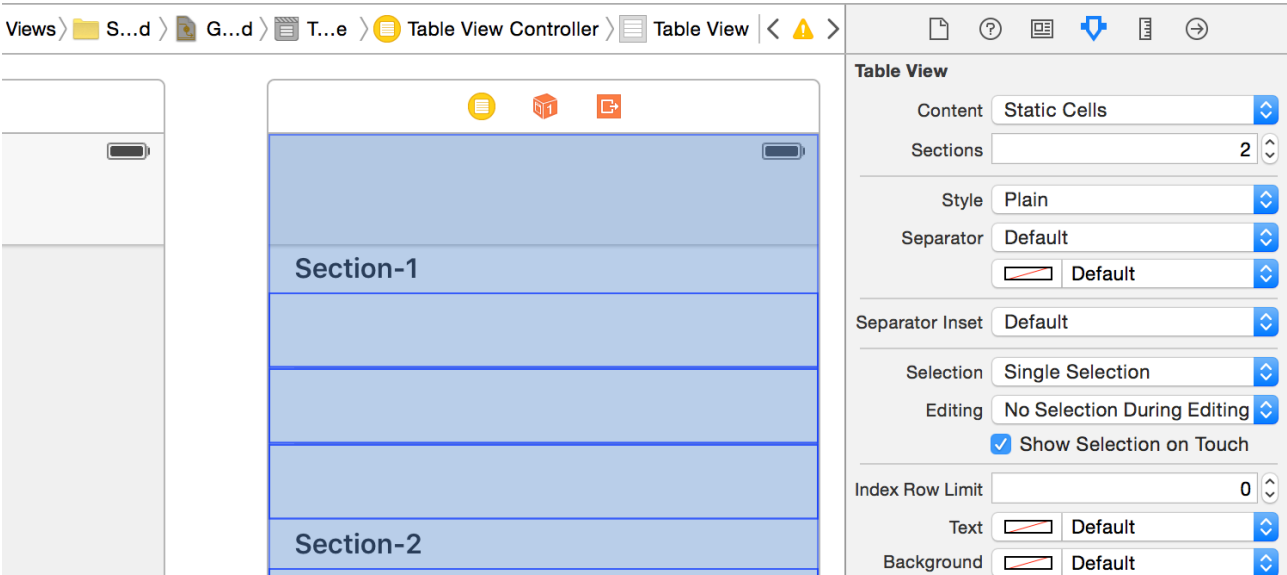
    self.tabBarController = [[UITabBarController alloc] init];
    self.tabBarController.viewControllers = @[squareVC, myVC];
}
```

2. 创建个人信息页面UI

2.1 选中视图中的Table View Controller，点击上方菜单栏的Editor，并选择Embed in菜单中的Navigation Controller，来给Table View加入一个导航控制器；



2.2 选中Table View（注意不是Controller），在右侧的属性选择器中，将之前默认的Dynamic Prototypes改为Static Cells，并将Sections设置为2；因为像这种row是固定行数的table view，可以用Static Cells比较方便；



2.3 参照原型图（左图）的要求，将各个UI组件拖到storyboard上面，最终的效果如右图所示，有几点需要注意：

1) 如何将“我的”下面显示用户头像、昵称的那栏放到table view上？

可以拖一个UIView到“更改昵称”那个row的上方，并按原型图将高度设为128（原型图为256px，4寸的iPhone转换成点就是128个点）；

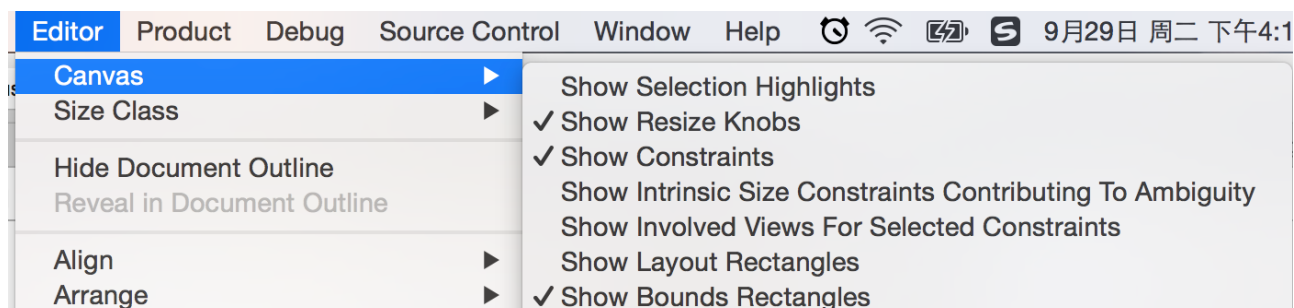
2) 将section的header去掉，因为根据原型图的要求，我们可以在代码中自定义header的高度（第一个section的header高度为13，第二个section为10）；

3) 自动布局，别忘了适配各个屏幕尺寸，我们的App是适配4寸到4.7寸及5.5寸的iPhone；



4) 如何为控件显示框线?

建议在调整、摆放UI控件的时候，可以显示控件的边框，这样可以方便直观地调整位置，打开方法为：点击Xcode菜单栏的Editor，如何勾选Canvas菜单中的“Show Bounds Rectangles”选项，这些框线只在Xcode里面显示，应用运行时是不显示的；



5) 如何把头像设置成圆形显示?

在GBMMyViewController.m中的viewDidLoad方法中，加上以下代码即可

```
@implementation GBMMyViewController

- (void)viewDidLoad {
    [super viewDidLoad];

    // 给navigation bar 设置背景色，颜色选择和蓦然登录注册界面的button颜色一致
    self.navigationController.navigationBar.barTintColor = [UIColor colorWithRed:237 / 255.0f green:127 / 255.0f blue:74 / 255.0f alpha:1.0f];

    // 把头像显示成圆形
    self.headImageView.layer.cornerRadius = self.headImageView.frame.size.width / 2.0f;
    self.headImageView.clipsToBounds = YES;
}
```

6) 设置每个section自定义的header和footer，footer都为0，如下图所示，还是在GBMMyViewController.m中，添加table view的Data Source代理方法；

```
#pragma mark - TableView Data Source methods

- (CGFloat)tableView:(UITableView *)tableView heightForHeaderInSection:(NSInteger)section
{
    CGFloat header;
    if (section == 0) {
        header = 13.0;
    } else if (section == 1) {
        header = 10;
    }
    return header;
}

- (CGFloat)tableView:(UITableView *)tableView heightForFooterInSection:(NSInteger)section
{
    return 0;
}
```

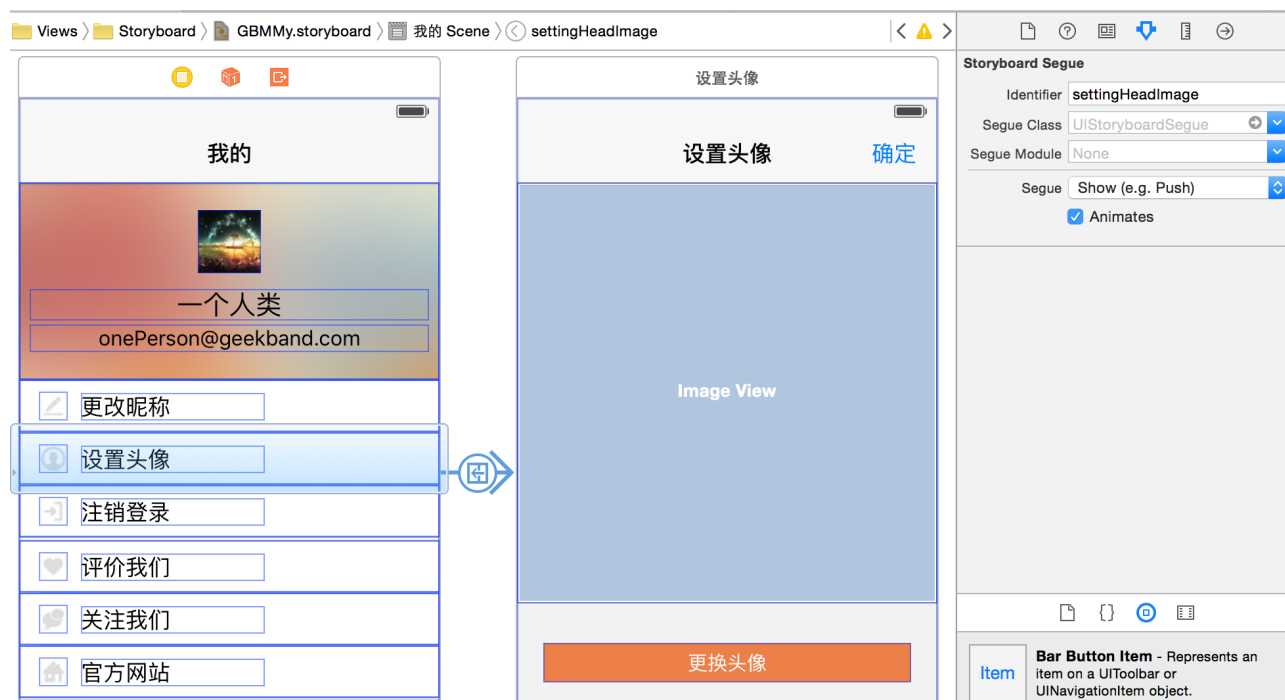
2.4 新增“更改昵称”页面

- 1) 首先在“我的”界面旁边，添加一个ViewController；
- 2) 选中“更改昵称”这一个cell，拖动一根线到新增的ViewController，在弹出菜单中选中Selection segue中的“show”选项，并在右侧的属性栏中设置segue的identifier为“settingNickName”；
- 3) 两个视图连接完成后，新增的ViewController上面会出现一个Navigation Item，点击可以把title改成“更改昵称”，并在右侧拖动添加一个Bar Button Item，名称为“确定”；
- 4) 新增一个名称为“GBMNickNameViewController”的类，用来作为“更改昵称”视图的控制器；



2.5 新增“设置头像”页面

- 1) 如2.4所示的步骤一样，在storyboard上新增一个新的ViewController，并加入需要的各个UI控件；
- 2) 新增一个“GBMHeadImageViewController”的类来作为控制器来管理；

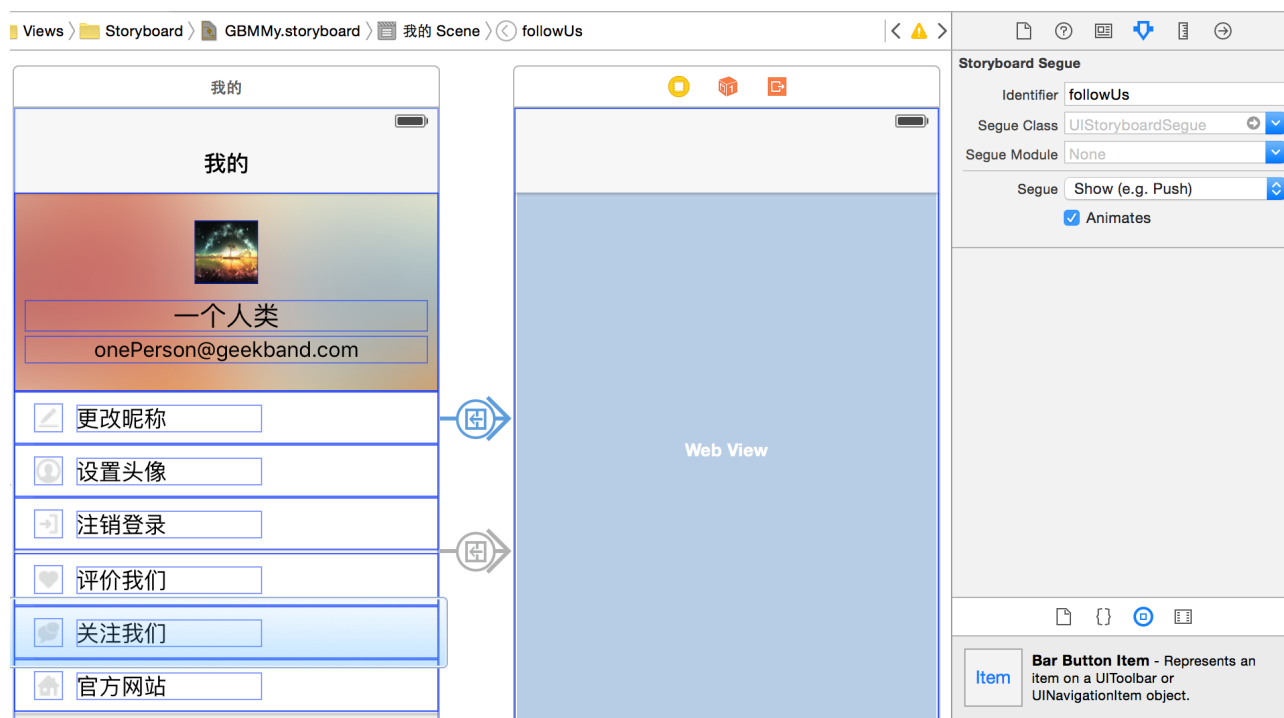


2.6 注销登录

注销登录这个功能，我们无需创建视图，稍后我们用代码来实现注销登录的功能（参见3.3节）；

2.7 关注我们、官方网站

- 1) 这两个功能，目前我们均为打开“GeekBand”的官网，所以可以共用一个视图、一个控制器；
- 2) 从“关注我们”和“官方网站”两个cell，分别拖动一根线到新增的ViewController上，并设置相应的 segue identifier，“关注我们”是“followUs”，“官方网站”是“homePage”；



3. 实现功能

3.1 更改昵称

- 1) 希望实现的效果：点击“更改昵称”这个cell，在打开的界面的textField中，会显示目前的用户昵称，修改过后，点击右上角的“确定”，即可更新昵称在“我的”界面；
- 2) 通过选中用户昵称的label（这里是“一个人人类”），连线到GBMMyViewController.h中，设置一个IBOutlet的属性，名称为“nickNameLabel”；

```
@interface GBMMyViewController : UITableViewController

@property (weak, nonatomic) IBOutlet UILabel *nickNameLabel;
```


3) 在GBMNickNameViewController.h中，添加以下代码：创建了一个名为“GBMNickNameViewControllerDelegate”的协议，并添加代理的属性；

```
@class GBMNickNameViewController;

@protocol GBMNickNameViewControllerDelegate <NSObject>

- (void)updateNickName:(NSString *)newName;

@end

@interface GBMNickNameViewController : UIViewController

@property (weak, nonatomic) IBOutlet UITextField *nickNameTextField;
@property (nonatomic, weak) NSString *nickName;
@property (nonatomic, weak) id <GBMNickNameViewControllerDelegate> delegate;

- (IBAction)doneBarButtonClicked:(id)sender;
```

4) 在GBMNickNameViewController.m中调用协议中的代理方法，这样完成了添加代理的前三步；

```
- (IBAction)doneBarButtonClicked:(id)sender
{
    [self.delegate updateNickName:self.nickNameTextField.text];
    [self.navigationController popViewControllerAnimated:YES];
}
```

5) 实现代理的后三步：首先在GBMMyViewController.h中，声明实现代理，剩下两步按如图的代码所示：在prepareForSegue: sender: 方法中，设置代理的值，并把nickNameLabel的当前昵称传入下一个视图的nickNameTextField中；最后实现代理方法updateNickName: 方法；

```
#pragma mark - Storyboard segue methods

- (void)prepareForSegue:(UIStoryboardSegue *)segue sender:(id)sender
{
    if ([segue.identifier isEqualToString:@"settingNickName"]) {
        GBMNickNameViewController *nickNameVC = segue.destinationViewController;
        nickNameVC.nickName = self.nickNameLabel.text;
        nickNameVC.delegate = self;
    } else if ([segue.identifier isEqualToString:@"settingHeadImage"]) {
        GBMHeadImageViewController *headImageVC = segue.destinationViewController;
        headImageVC.headImage = self.headImageView.image;
        headImageVC.delegate = self;
    }
}

#pragma mark - GBMNickNameViewController delegate method

- (void)updateNickName:(NSString *)newName
{
    self.nickNameLabel.text = newName;
}
```


3.2 设置头像

- 1) 创建需要交互的UI控件的IBOutlet，方法和3.1描述的基本一致，也需要让GBMMyViewController成为GBHeadImageViewController的代理，来完成相应的操作，来更新头像；
- 2) 点击页面中的“更换头像”按钮时，需要弹出一个选项（如下图所示），让用户选择从使用哪种途径获得图片，这个效果是用什么实现呢？



这个可以用一个UIActionSheet，在点击“更换头像”按钮时弹出，并根据用户的点击来执行不同的操作，具体代码如下：

```
- (IBAction)doneBarButtonClicked:(id)sender
{
    [self.delegate updateHeadImage:self.headImageView.image];
    [self.navigationController popViewControllerAnimated:YES];
}

// 添加ActionSheet
- (void)addActionSheet
{
    UIActionSheet *actionSheet = [[UIActionSheet alloc] initWithTitle:nil
                                                                    delegate:self
                                                                    cancelButtonTitle:@"取消"
                                                                    destructiveButtonTitle:nil
                                                                    otherButtonTitles:@"拍照", @"从手机相册选择", nil];

    [actionSheet showInView:self.view];
}

- (void)actionSheet:(UIActionSheet *)actionSheet clickedButtonAtIndex:(NSInteger)buttonIndex
{
    UIImagePickerController *imagePicker = [[UIImagePickerController alloc] init];
    if (buttonIndex == 0) {
        if ([UIImagePickerController isSourceTypeAvailable:UIImagePickerControllerSourceTypeCamera]) {
            imagePicker.sourceType = UIImagePickerControllerSourceTypeCamera;
        } else {
            UIAlertView *alert = [[UIAlertView alloc] initWithTitle:nil
                                                                message:@"无法获取相机"
                                                                delegate:nil
                                                                cancelButtonTitle:@"确定"
                                                                otherButtonTitles:nil];
        }
    }

    // 获取用户选择的图片
    - (void)imagePickerController:(UIImagePickerController *)picker didFinishPickingMediaWithInfo:
        (NSDictionary<NSString *, id> *)info
    {
        self.headImage = info[UIImagePickerControllerOriginalImage];
        self.headImageView.image = self.headImage;
        [self dismissViewControllerAnimated:YES completion:nil];
    }
}
```

3.3 注销登录

- 1) 功能设计: 点击“注销登录”, 弹出alert view, 提示用户是否确定注销, 点击“确定”则注销并返回登录界面;
- 2) 代码如下, 注意需要响应用户点击alert view上的按钮的话, 需要让GBMMyViewController成为UIAlertView的代理;

```
@interface GBMMyViewController () <UIAlertViewDelegate>

#pragma mark - TableView delegate methods

- (void)tableView:(UITableView *)tableView didSelectRowAtIndexPath:(NSIndexPath *)indexPath
{
    if (indexPath.section == 0) {
        if (indexPath.row == 2) {
            UIAlertView *alert = [[UIAlertView alloc] initWithTitle:nil
                                                                message:@"确定注销吗? "
                                                                delegate:self
                                                                cancelButtonTitle:nil
                                                                otherButtonTitles:@"取消", @"确定", nil];

            [alert show];
        }
    }
    [tableView deselectRowAtIndexPath:indexPath animated:YES];
}

#pragma mark - UIAlertView delegate method

- (void)alertView:(UIAlertView *)alertView clickedButtonAtIndex:(NSInteger)buttonIndex
{
    if (buttonIndex == 1) {
        [self dismissViewControllerAnimated:YES completion:nil];
        AppDelegate *appDelegate = (AppDelegate *)[UIApplication sharedApplication].delegate;
        [appDelegate loadLoginView];
    }
}
}
```

3.4 关注我们及官方网站

- 1) 在打开后的详情页上, 添加UIWebView, 并连线生成IBOutlet属性;
- 2) 具体代码如下;

```
- (void)viewDidLoad {
    [super viewDidLoad];

    [self showHomePage];
}

- (void)didReceiveMemoryWarning {
    [super didReceiveMemoryWarning];
}

- (void)showHomePage
{
    NSString *urlString = @"http://geekband.com";
    NSURL *url = [NSURL URLWithString:urlString];
    NSURLRequest *request = [NSURLRequest requestWithURL:url];
    [self.webView loadRequest:request];
}
```