

# 1

## 实验



# 创建和管理 View 用于构建用户界面

## 实验结构

- ▶▶ 在应用中创建和管理 View

## 实验目标

本实验结束后，你将能够：

- ▶▶ 在 Storyboard 中创建 View，用于构建应用用户界面
- ▶▶ 管理 Storyboard 中创建的 View

## 导论

Xcode 被用于在 Xcode terminal 上执行 **Objective-C** 程序。你可以使用 Xcode 中的 **View Controller** 来确定应用需要哪些 View。Xcode 为应用提供有很多 UI View 选项。本节中，你将学会这些 View 的创建和管理。

## 实验：创建和管理 VIEW，用于构建用户界面

用户在使用应用时首先注意到的就是用户界面（UI）。前端 UI 需要简单直观，它是通过 **View** 来创建的。所有 **View** 和 **Controller** 都在实用工具面板的对象库中。你需要通过在 **Storyboard** 中创建和管理 **View** 来构建用户界面。你还有可能需要用到一些音频、图像、背景、按钮、字体及其它支持文件，来增强用户界面。

## 背景

假设你要创建一个注册表应用，用户在其中填入信息并点 **Submit** 按钮提交。点提交按钮会打开一个细节页面，用户可以看到要提交的所有信息。该应用还需要包含一个用于访问社交网站的按钮，该按钮会另外打开页面。你可以使用 **View** 和 **Controller** 来添加这些页面。

你需要执行下面这些任务：

1. 在 **Xcode** 中创建一个项目，用于注册表单应用
2. 通过实现 **View** 来为注册表单应用创建用户界面

## 实验准备

要执行这些任务，你需要有：

- iOS 8 和 Xcode
- 个人苹果帐户

## 实验推荐解决方案

### 任务 1 解决方案：在 Xcode 中创建项目

1. 通过点击图 1 所示的 **Xcode** 图标来打开 Xcode：



图 1: 开启 Xcode

2. 在 Xcode 欢迎页面，选择 **Create a new Xcode project** 选项（参见图 2）。

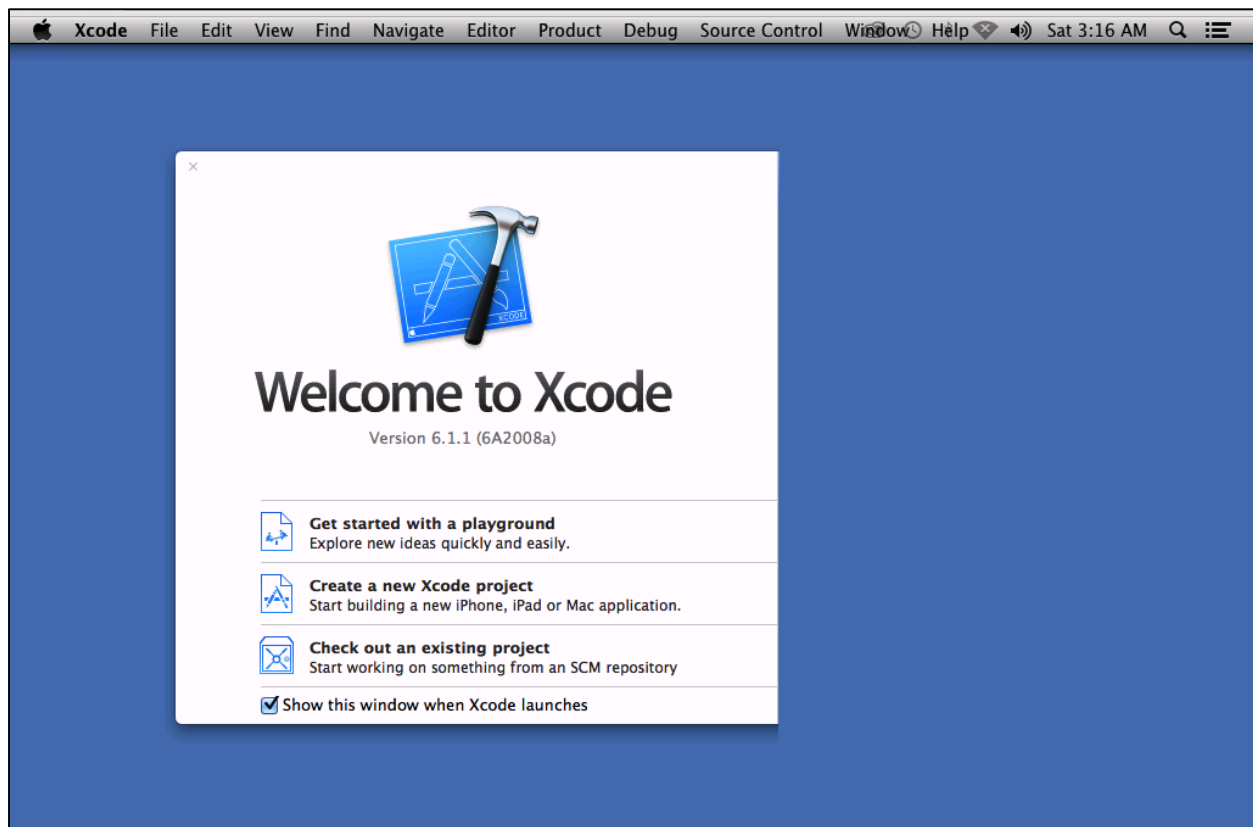


图 2: 选择创建新 Xcode 项目

3. 要开发应用，选择 **Single View Application** 模板，并点 **Next**，如图 3 所示：

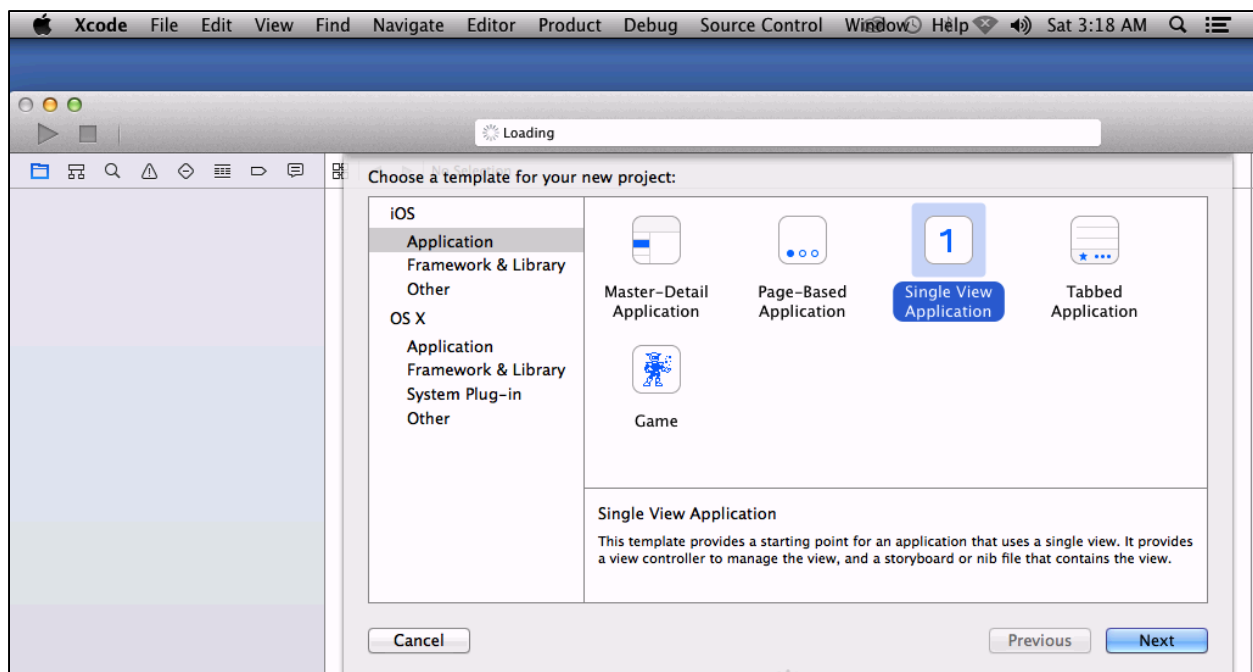


图 3: 选择 Single View Application 模板

4. 在 **Product Name** 字段中输入恰当的应用名（参见图 4）。应用名需要匹配应用用途，且不能超过 12 个字符。为应用填入其它选项，点 **Next**。

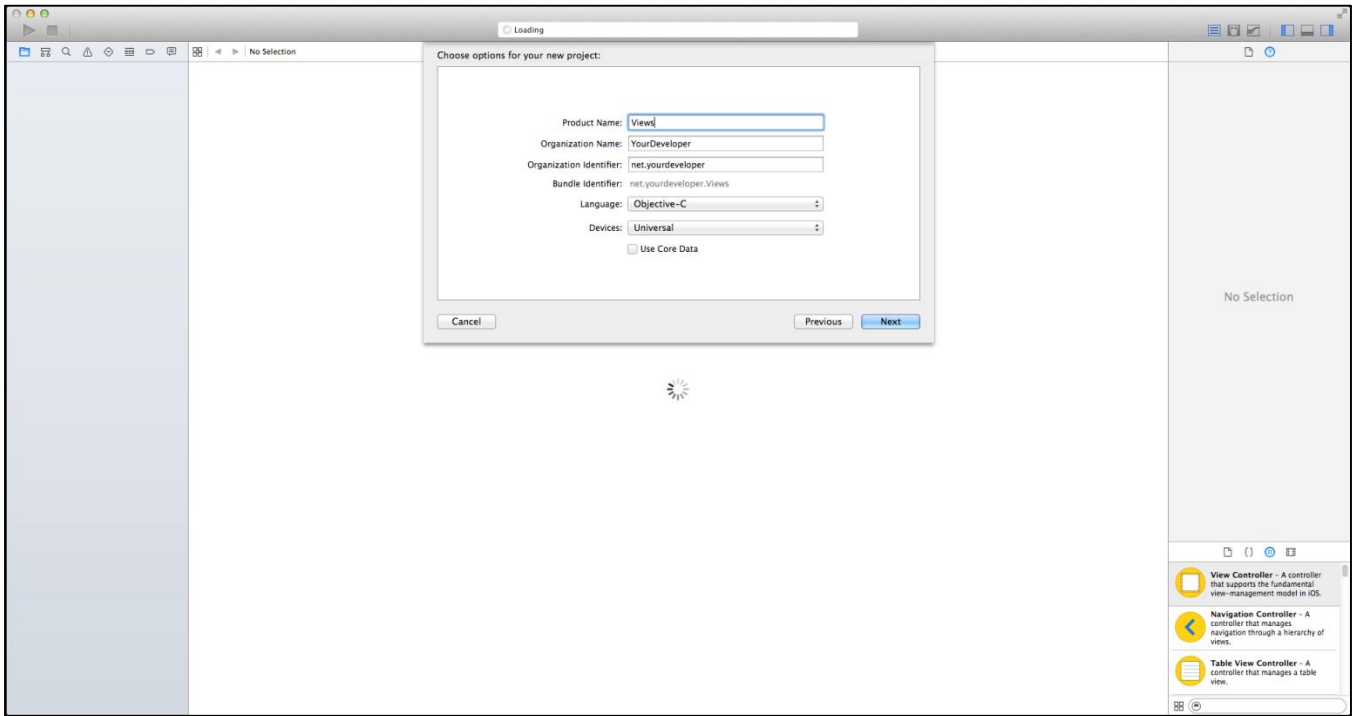


图 4: 命名新项目

5. 选择保存项目的位置，然后点 **Create** 来创建，如图 5 所示：

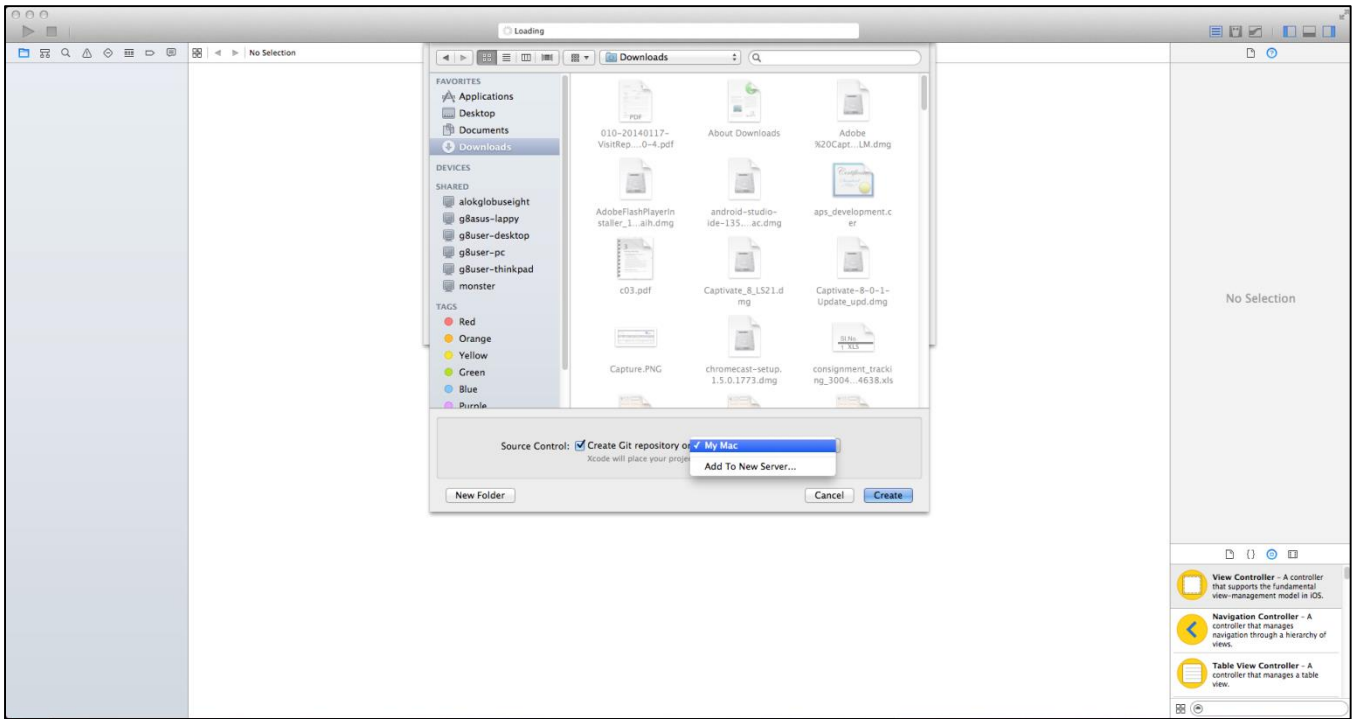


图 5: 保存和创建项目

6. 点 **Create** 按钮会在 Xcode 中显示出 Views 表单，如图 6 所示。表单中，你需要填入项目细节，例如构建标识符、版本号 and 团队。

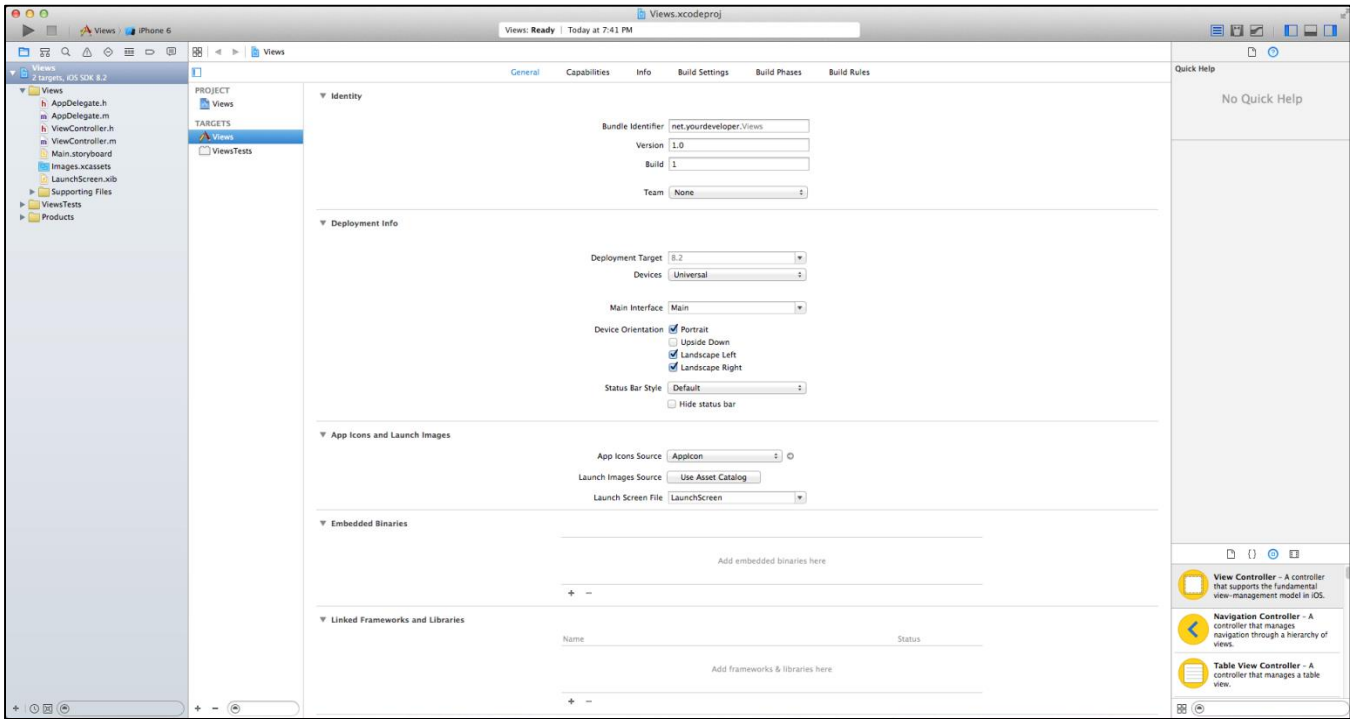


图 6: 应用开发的项目编辑器表单

7. 从导航面板选择 Main.storyboard 文件，如图 7 所示。这会在编辑器中添加 View Controller UI，如图 8 所示。

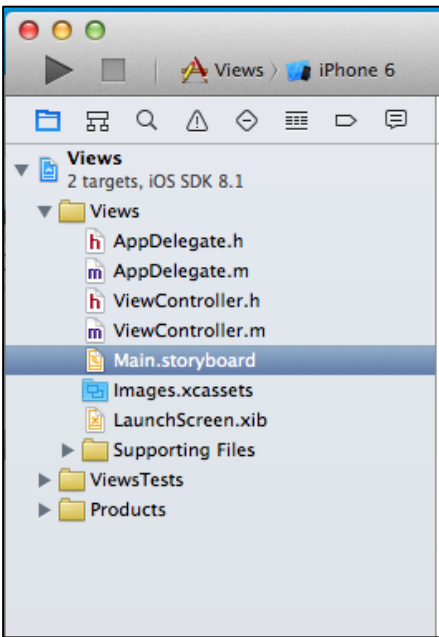


图 7: Main.storyboard 文件

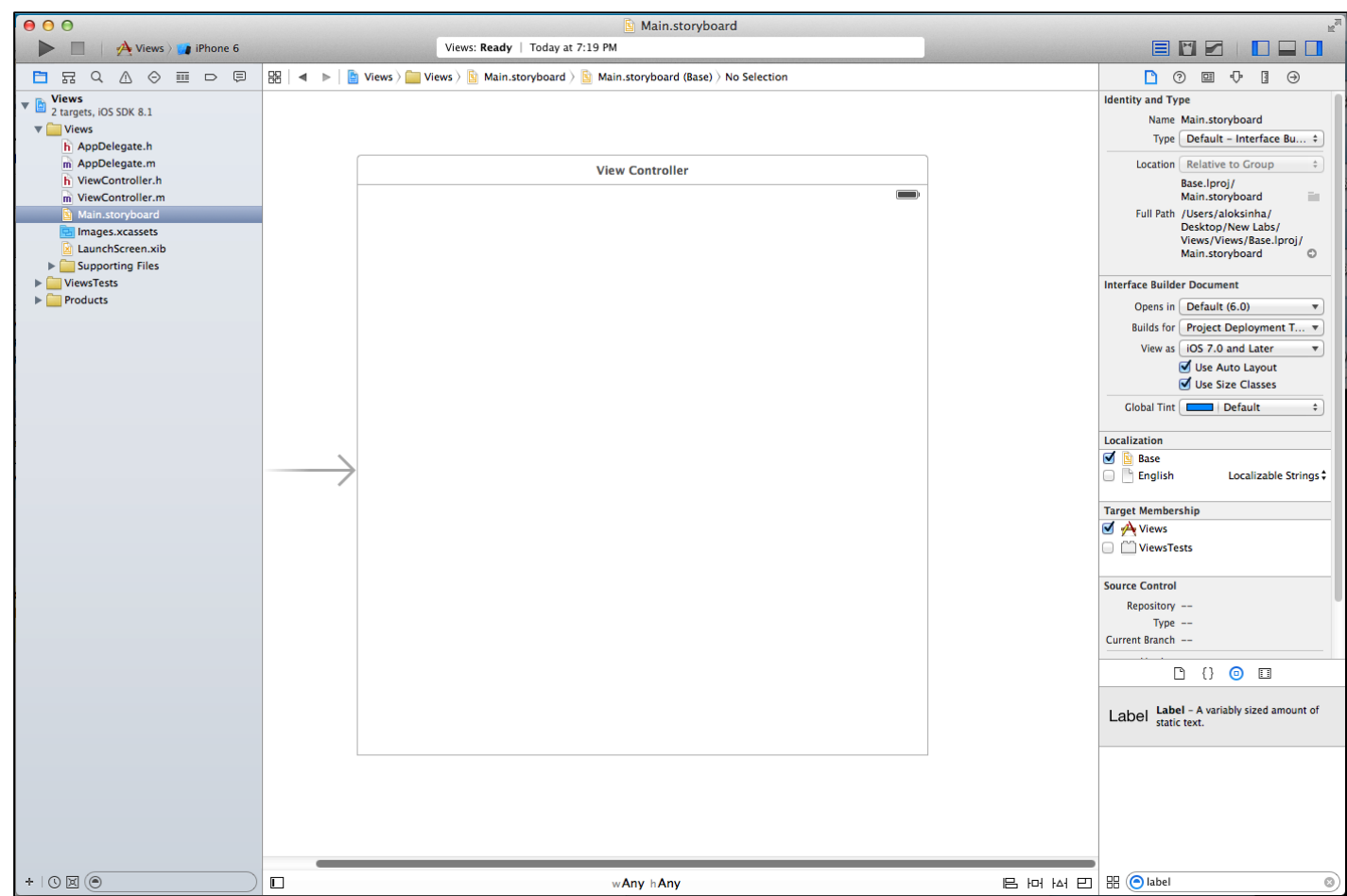


图 8: 编辑器中的 View Controller

## 任务 2 解决方案：管理 View Controller 尺寸

1. 管理 View Controller 尺寸的方法有两种：
  - a. 手动调节高度和宽度，如图 9 所示。
  - b. 在文件检查器中勾选和取消勾选 **Use Size Classes** 复选框，如图 10 所示。

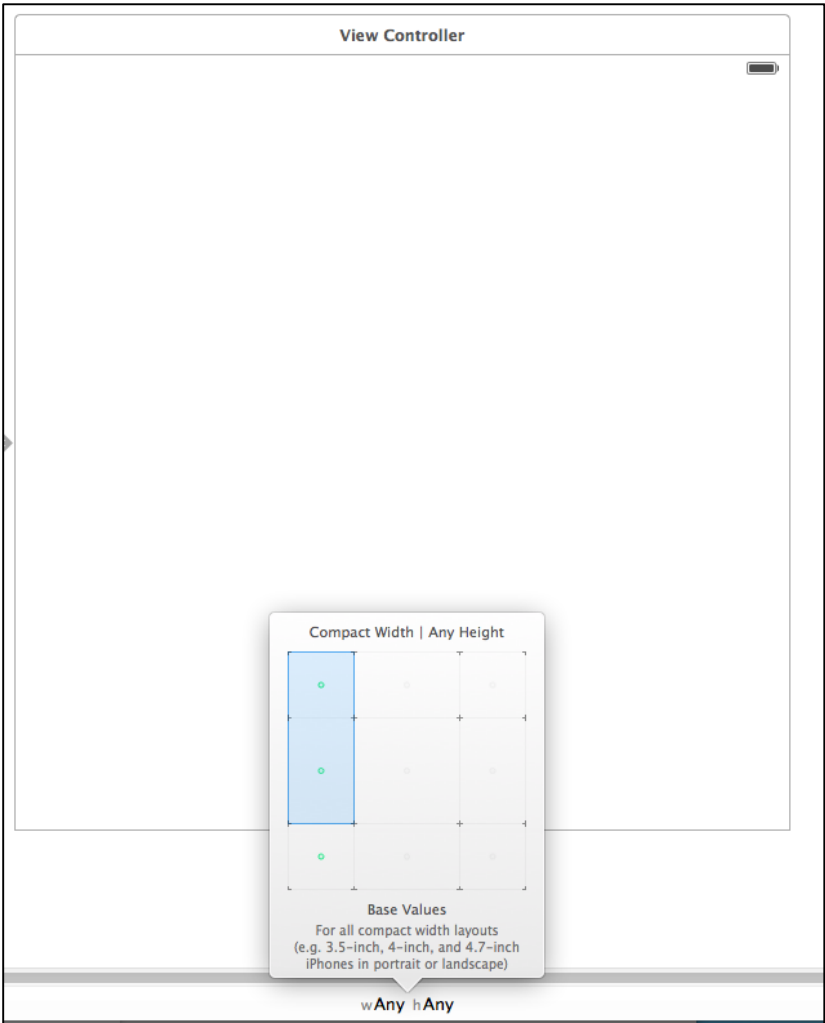


图 9: 手动调节 View Controller 的高度和宽度

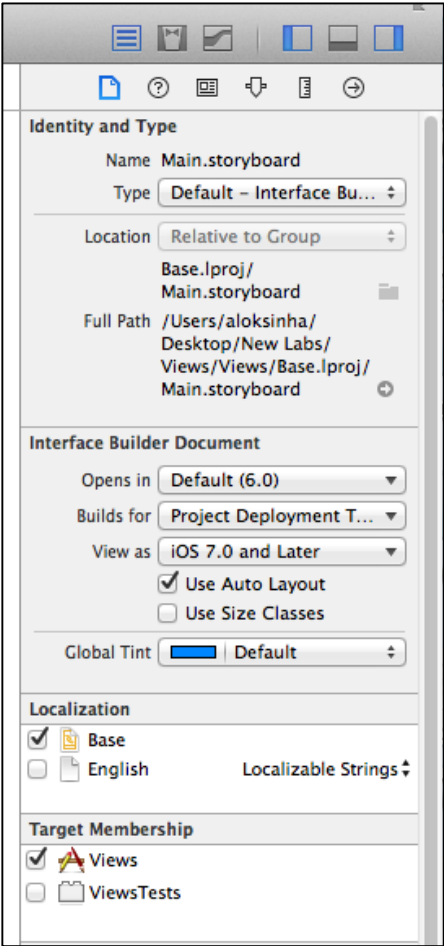


图 10: 取消对文件检查器中 Use Size Classes 复选框的勾选

2. 如果你在保持 **Use Size Classes** 复选框勾选的情况下管理 View，会有一个弹窗消息出现，让你禁用 size classes，如图 11 所示：

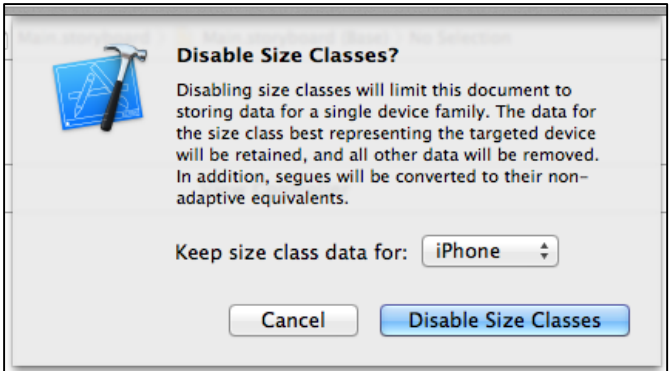


图 11: 要求禁用 Size Classes 的弹窗

3. 这样你就在应用中有了一个 View Controller。下面从实用工具面板对象库中选择并拖动 Table View Controller 到 Storyboard，如图 12 所示：

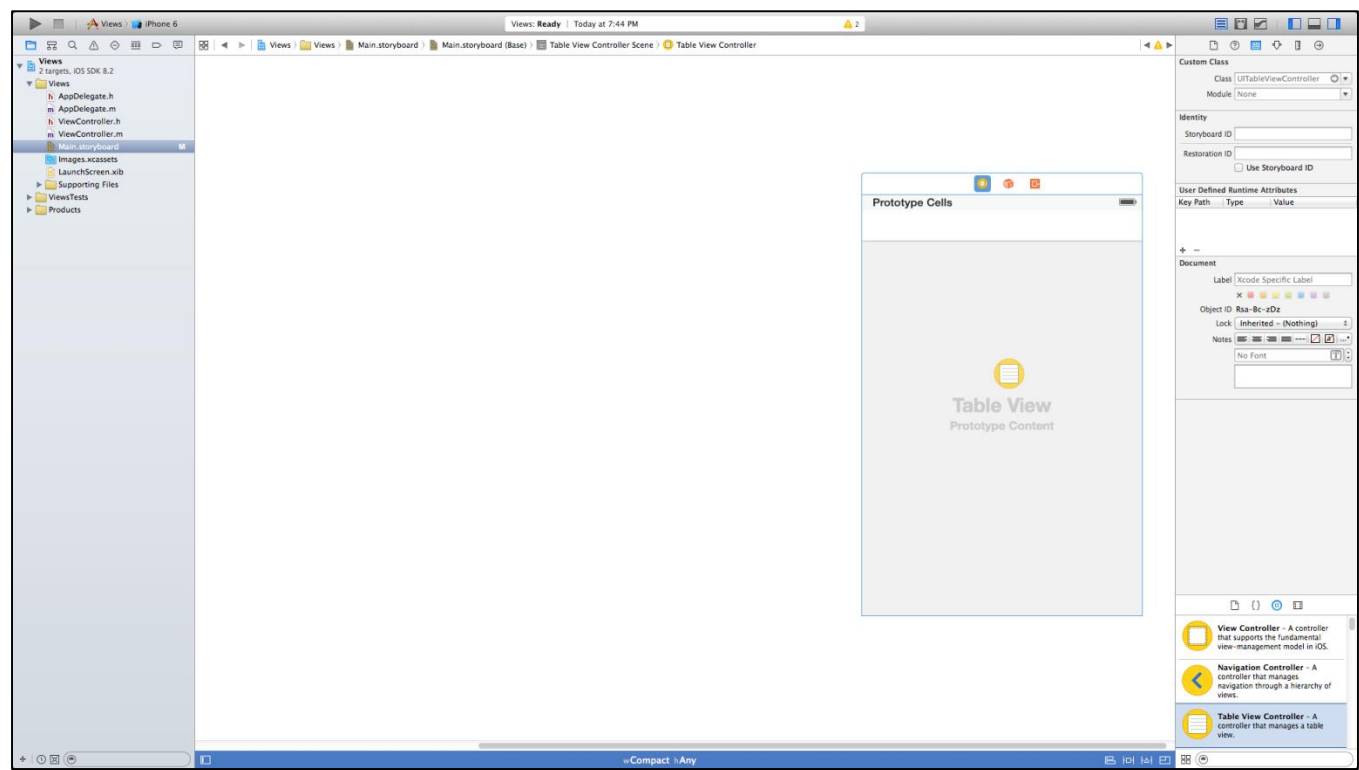


图 12: 从对象库中拖动 Table View Controller 到 Storyboard

4. 类似添加其它 View 到应用，如图 13 和 14 所示：

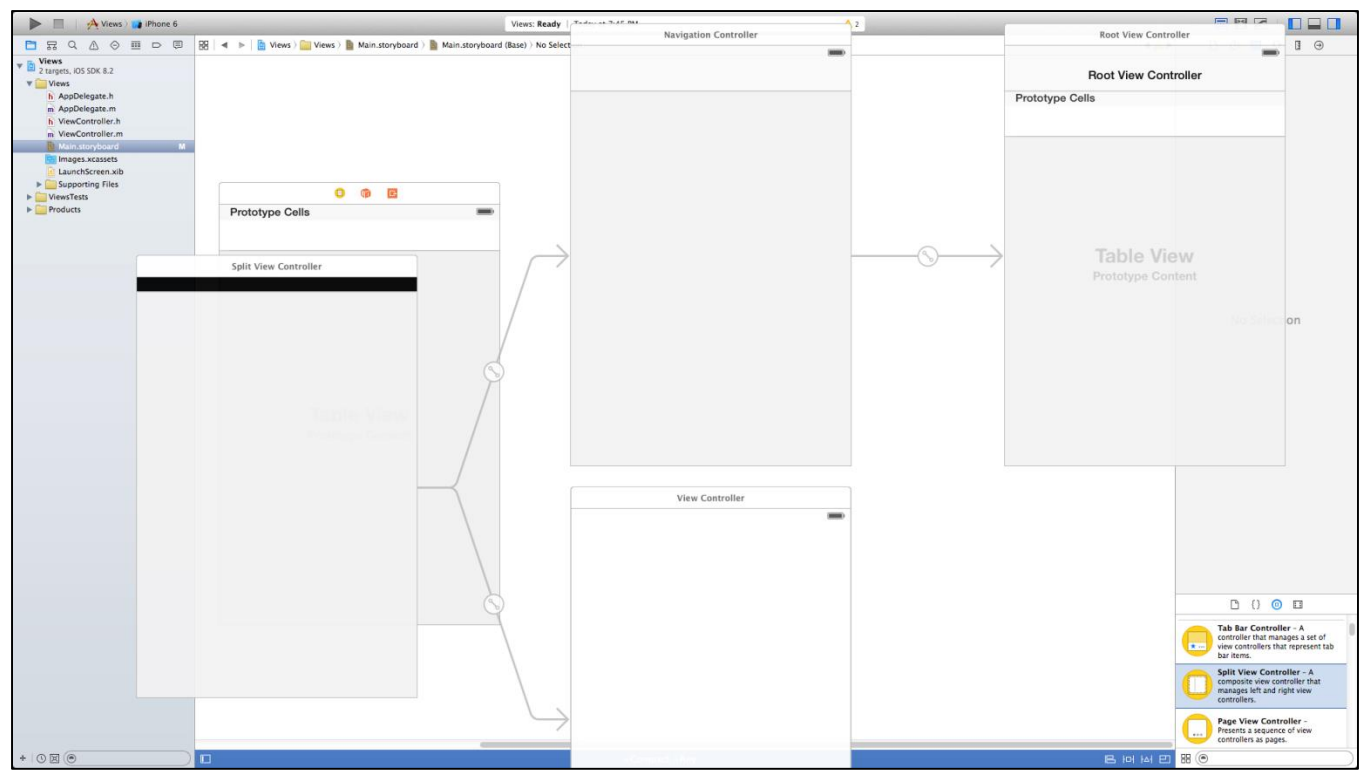


图 13: 从对象库拖放 Split View



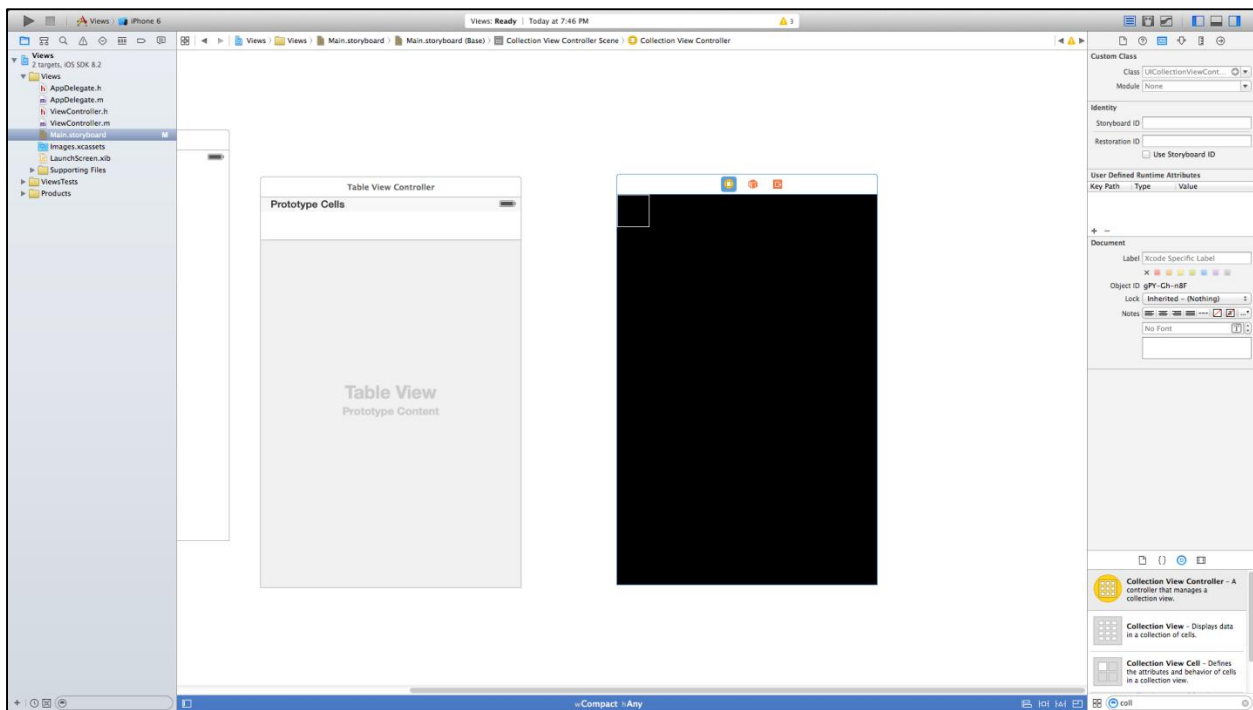


图 14: 拖动添加 Collection View Controller

5. 下面需要连接各 View。应用中需要在页面之间切换，因此各 View 需要连接起来，如图 15 所示。这里考虑 Table View 和 View Controller。要连接这两个 View，你需要执行下面这些步骤：
  - i. 右键点击 Table View。
  - ii. 按住并拖动 Table View 到 View Controller。

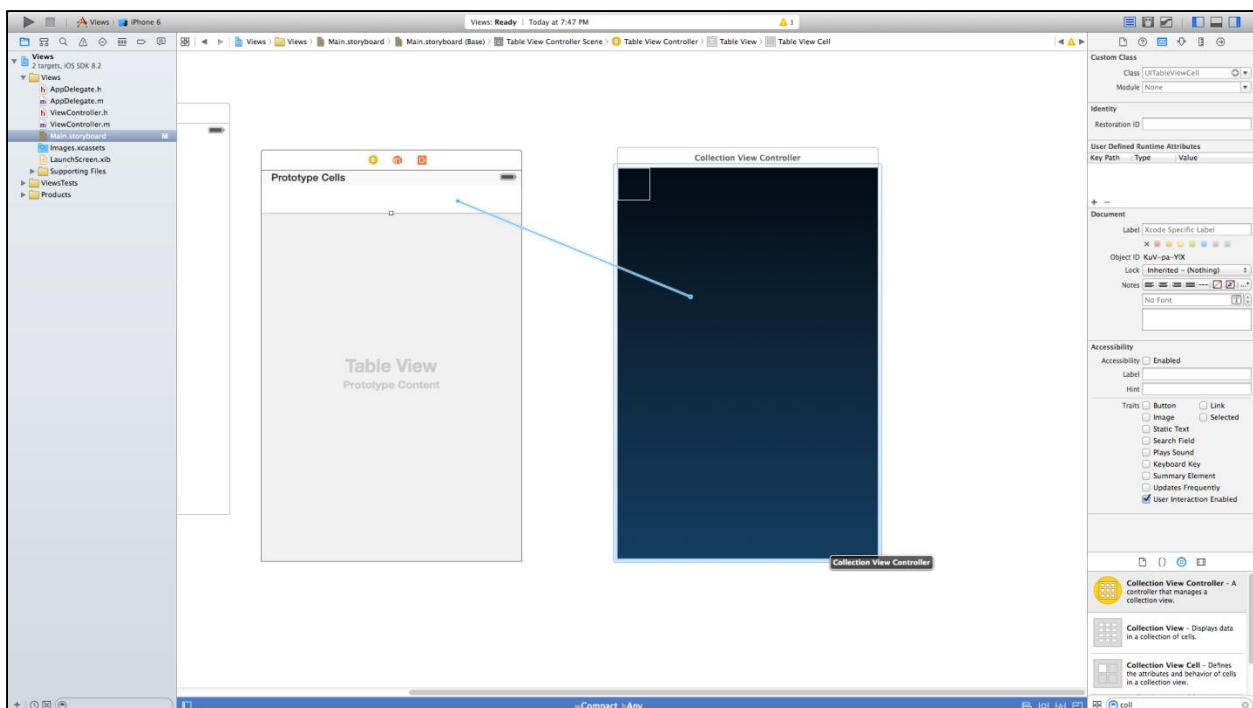


图 15: 连接 Table View 和 Collection View Controller

- iii. 拖动 Table View 到 View Controller 并释放鼠标后，你会得到一个弹出菜单，如图 16 所示。菜单中，你可以选择在 View 之间创建怎样的连接。

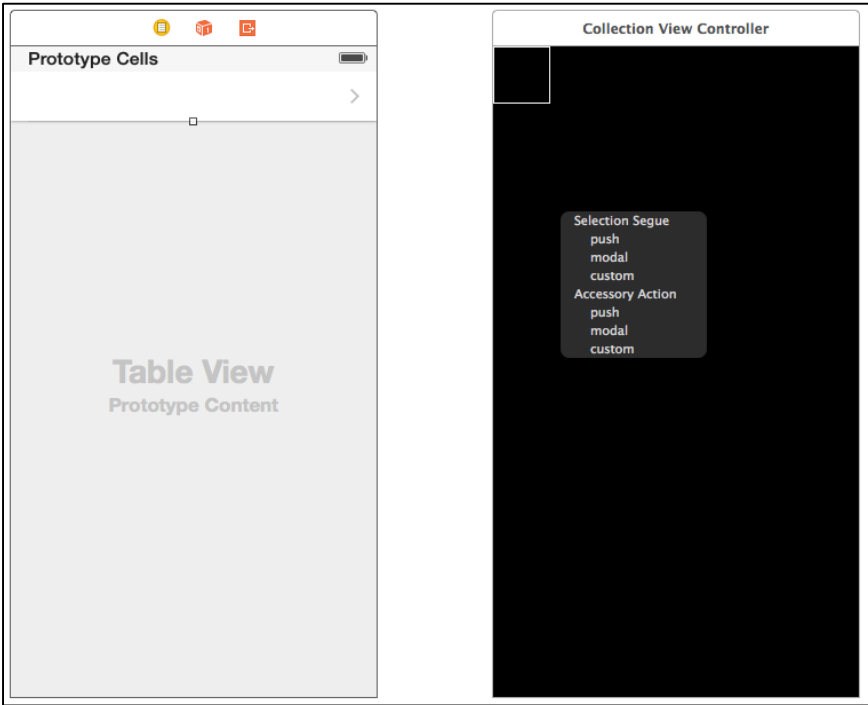


图 16: 从弹出菜单中选择连接类型

- iv. 连接 View 之后，你会得到图 17 所示的画面：

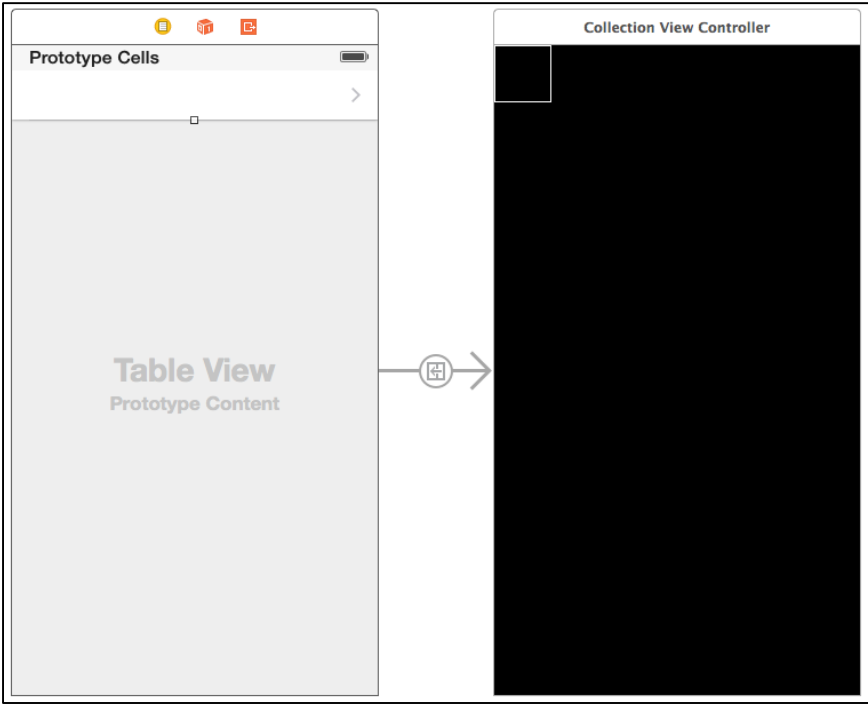


图 17: View 之间的连接完成

6. 之前你向 Storyboard 添加了 View Controller、Table View 和 Connection View Controller。你也可以像这样添加对象库中的其它 Controller 及实用工具，如图 18 所示：

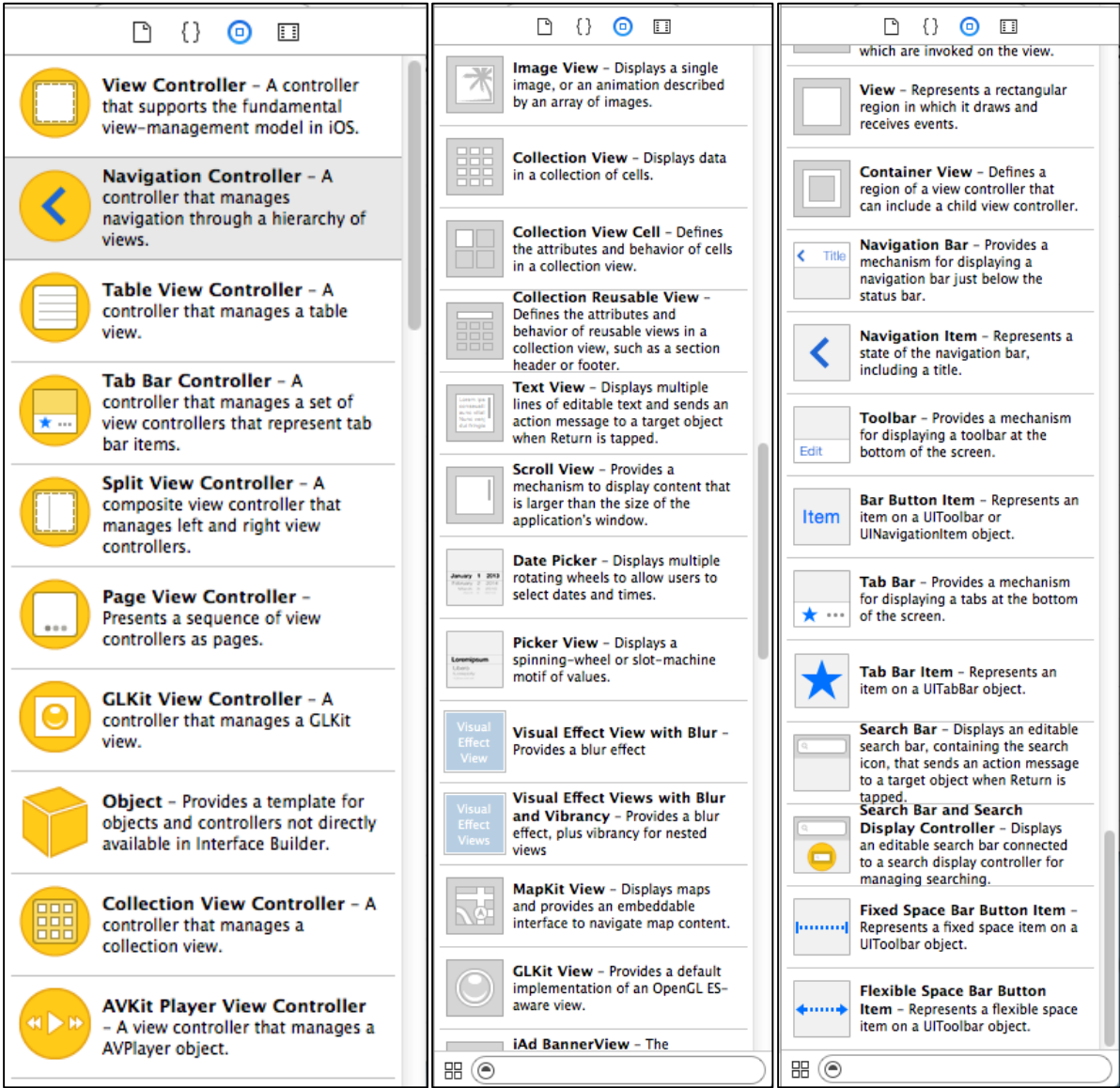


图 18: Xcode 右侧实用工具面板中的对象库

7. 连接各 View 之后，从对象库中添加对象，如图 18 所示。添加对象后，你可以从对象库中选择并添加 Label，如图 19 所示。

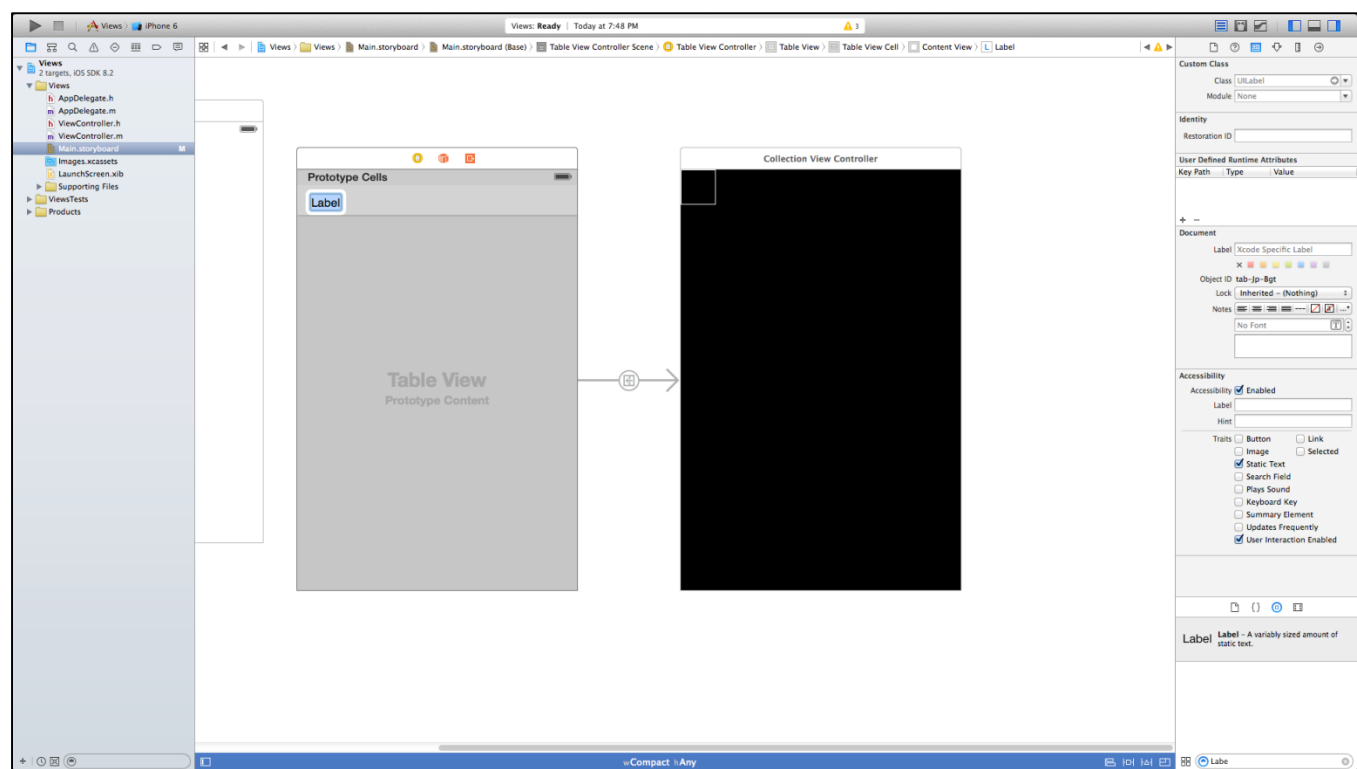


图 19: 为 Table View 添加标签

8. 设计好 Storyboard 后，切换到 ViewController.m 文件，进入应用代码编写部分，如图 20 所示。Xcode 在 **Single View Application** 中提供 ViewController 作为默认。

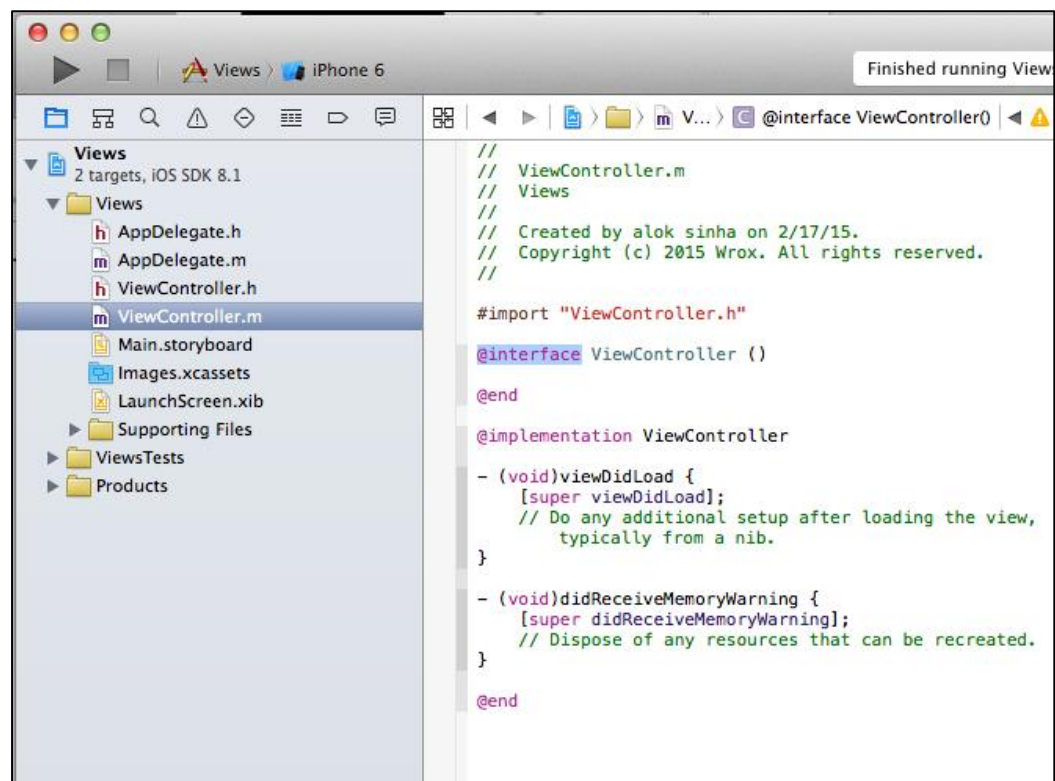


图 20: Single View Application 中默认的 ViewController.m 文件

9. 如果创建新项目（参见图 3）时，你选择的是 **Master-Detail Application** 模板，默认控制器文件就将是 MasterViewController 和 DetailViewController，如图 21 所示：

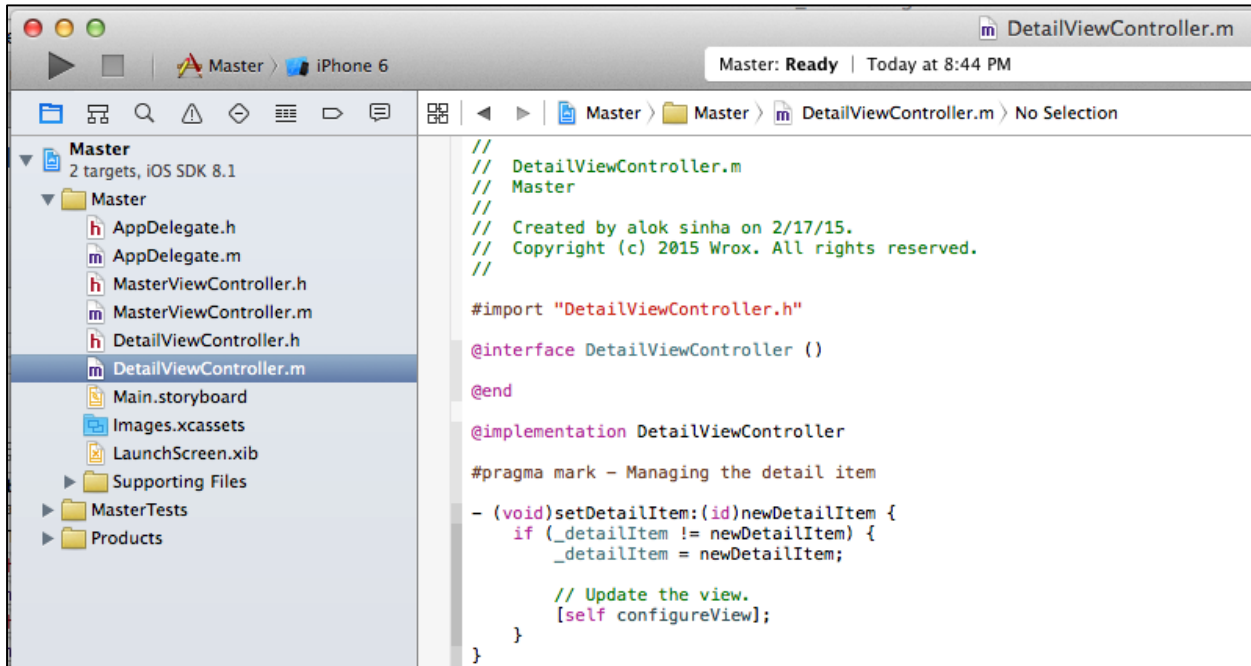


图 21: Master-Detail Application 中默认的 MasterViewController 和 DetailViewController 文件

10. 类似地，**Tabbed Application** 模板的默认控制器如图 22 所示：

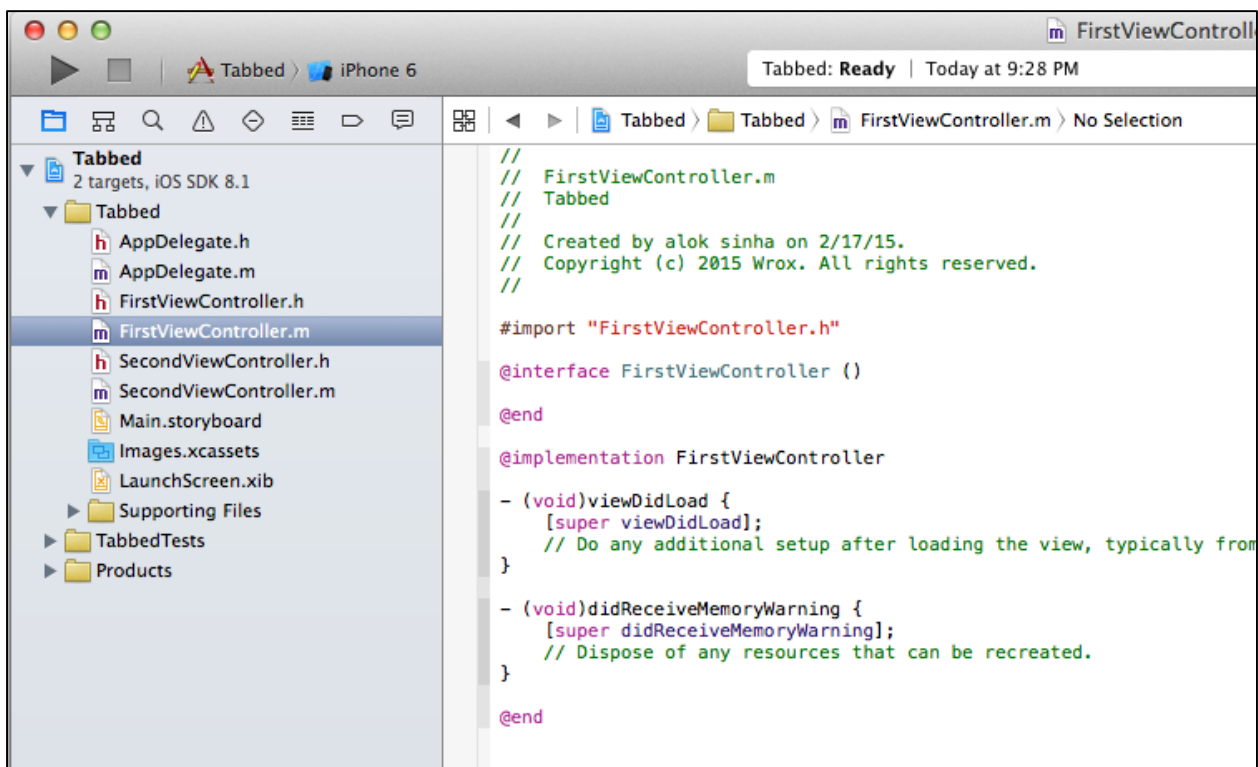


图 22: Tabbed Application 默认的 FirstViewController 和 SecondViewController

11.要在应用中使用这些 Controller，你可以使用 MasterViewController 和 DetailViewController 中的如下样码。

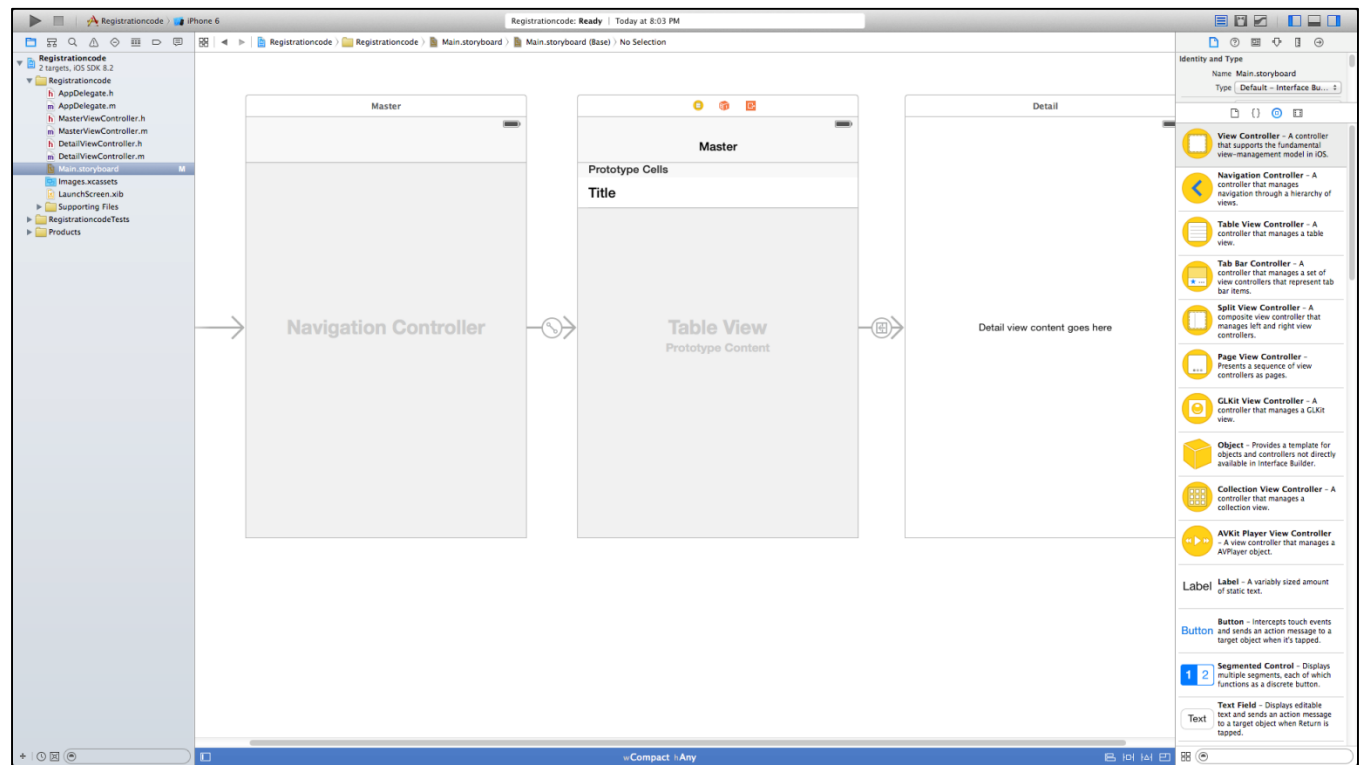


图 23: 样本应用的 View

### 源码

AppDelegate.m

```
#import "AppDelegate.h"

@implementation AppDelegate

- (BOOL)application:(UIApplication *)application
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    // Override point for customization after application launch.
    if ([[UIDevicecurrentDevice] userInterfaceIdiom] == UIUserInterfaceIdiomPad) {
        UISplitViewController *splitViewController = (UISplitViewController
*)self.window.rootViewController;
        UINavigationController *navigationController =
[splitViewController.viewControllerslastObject];
        splitViewController.delegate = (id)navigationController.topViewController;
    }
    return YES;
}

MasterViewController.h
```

```

#import <UIKit/UIKit.h>

@class DetailViewController;

@interface MasterViewController : UITableViewController

@property (strong, nonatomic) DetailViewController *detailViewController;

@end

```

MasterViewController.m

```

#import "MasterViewController.h"

#import "DetailViewController.h"

@interface MasterViewController () {
    NSMutableArray *_objects;
}
@end

@implementation MasterViewController
- (void)awakeFromNib
{
    if ([[UIDevicecurrentDevice] userInterfaceIdiom] == UIUserInterfaceIdiomPad) {
        self.clearsSelectionOnViewWillAppear = NO;
        self.preferredContentSize = CGSizeMake(320.0, 600.0);
    }
    [superawakeFromNib];
}
- (void)viewDidLoad
{
    [super viewDidLoad];
    // Do any additional setup after loading the view, typically from a nib.
    self.navigationItem.leftBarButtonItem = self.editButtonItem;
    UIBarButtonItem *addButton = [[UIBarButtonItemalloc]
initWithBarButtonSystemItem:UIBarButtonSystemItemAdd target:self
action:@selector(insertNewObject:)];
    self.navigationItem.rightBarButtonItem = addButton;
    self.detailViewController = (DetailViewController
*)[[self.splitViewController.viewControllerslastObject] topViewController];
}

- (void)didReceiveMemoryWarning
{
    [superdidReceiveMemoryWarning];
    // Dispose of any resources that can be recreated.
}

```

```

- (void)insertNewObject:(id)sender
{
    if (!_objects) {
        _objects = [[NSMutableArray alloc] init];
    }
    [_objects insertObject:[NSDate date] atIndex:0];
    NSIndexPath *indexPath = [NSIndexPath indexPathForRow:0 inSection:0];
    [self.tableView insertRowsAtIndexPaths:@[indexPath]
    withRowAnimation:UITableViewRowAnimationAutomatic];
}

#pragma mark - Table View

- (NSInteger)numberOfSectionsInTableView:(UITableView *)tableView
{
    return 1;
}

- (NSInteger)tableView:(UITableView *)tableView numberOfRowsInSection:(NSInteger)section
{
    return _objects.count;
}

- (UITableViewCell *)tableView:(UITableView *)tableView cellForRowAtIndexPath:(NSIndexPath *)indexPath
{
    UITableViewCell *cell = [tableView dequeueReusableCellWithIdentifier:@"Cell"
    forIndexPath:indexPath];

    NSDate *object = _objects[indexPath.row];
    cell.textLabel.text = [object description];
    return cell;
}

- (BOOL)tableView:(UITableView *)tableView canEditRowAtIndexPath:(NSIndexPath *)indexPath
{
    // Return NO if you do not want the specified item to be editable.
    return YES;
}

- (void)tableView:(UITableView *)tableView commitEditingStyle:(UITableViewCellEditingStyle)editingStyleforRowAtIndexPath:(NSIndexPath *)indexPath
{
    if (editingStyle == UITableViewCellEditingStyleDelete) {
        [_objects removeObjectAtIndex:indexPath.row];
        [tableView deleteRowsAtIndexPaths:@[indexPath]
        withRowAnimation:UITableViewRowAnimationFade];
    } else if (editingStyle == UITableViewCellEditingStyleInsert) {
        // Create a new instance of the appropriate class, insert it into the array, and
        add a new row to the table view.
    }
}

```



```

    }
}

- (void)tableView:(UITableView *)tableViewdidSelectRowAtIndexPath:(NSIndexPath *)indexPath
{
    if ([[UIDevicecurrentDevice] userInterfaceIdiom] == UIUserInterfaceIdiomPad) {
        NSDate *object = _objects[indexPath.row];
        self.detailViewController.detailItem = object;
    }
}

- (void)prepareForSegue:(UIStoryboardSegue *)segue sender:(id)sender
{
    if ([segue identifier] isEqualToString:@"showDetail"]) {
        NSIndexPath *indexPath = [self.tableViewindexPathForSelectedRow];
        NSDate *object = _objects[indexPath.row];
        [[segue destinationViewController] setDetailItem:object];
    }
}

@end

```

DetailViewController.h

```

@interface DetailViewController :UIViewController<UISplitViewControllerDelegate>

@property (strong, nonatomic) id detailItem;

@property (weak, nonatomic) IBOutletUILabel *detailDescriptionLabel;

@end

```

DetailViewController.m

```

#import "DetailViewController.h"

@interface DetailViewController ()
@property (strong, nonatomic) UIPopoverController *masterPopoverController;
- (void)configureView;
@end

@implementation DetailViewController

#pragma mark - Managing the detail item

- (void)setDetailItem:(id)newDetailItem
{
    if (_detailItem != newDetailItem) {
        _detailItem = newDetailItem;
    }
}

```

```

        // Update the view.
        [selfconfigureView];
    }

    if (self.masterPopoverController != nil) {
        [self.masterPopoverControllerdismissPopoverAnimated:YES];
    }
}

- (void)configureView
{
    // Update the user interface for the detail item.

    if (self.detailItem) {
        self.detailDescriptionLabel.text = [self.detailItem description];
    }
}

- (void)viewDidLoad
{
    [superviewDidLoad];
    // Do any additional setup after loading the view, typically from a nib.
    [selfconfigureView];
}

- (void)didReceiveMemoryWarning
{
    [superdidReceiveMemoryWarning];
    // Dispose of any resources that can be recreated.
}

#pragma mark - Split view

- (void)splitViewController:(UISplitViewController
*)splitControllerwillHideViewController:(UIViewController
*)viewControllerwithBarButtonItem:(UIBarButtonItem
*)barButtonItemforPopoverController:(UIPopoverController *)popoverController
{
    barButtonItem.title = NSLocalizedString(@"Master", @"Master");
    [self.navigationItemsetLeftBarButtonItem:barButtonItemanimated:YES];
    self.masterPopoverController = popoverController;
}

- (void)splitViewController:(UISplitViewController
*)splitControllerwillShowViewController:(UIViewController
*)viewControllerinvalidatingBarButtonItem:(UIBarButtonItem *)barButtonItem
{
    // Called when the view is shown again in the split view, invalidating the button and
    popover controller.
    [self.navigationItemsetLeftBarButtonItem:nilanimated:YES];
    self.masterPopoverController = nil;
}

```

@end

---