



# Lập trình iOS

## Bài 3. *Storyboard* và *Navigation Controller*

Ngành Mạng & Thiết bị di động





# Nội dung

---

## 1. Storyboard

- Tổng quan về storyboard
- Khảo sát lớp UIStoryboard
- Các segue trong storyboard
- Ưu điểm và hạn chế của storyboard

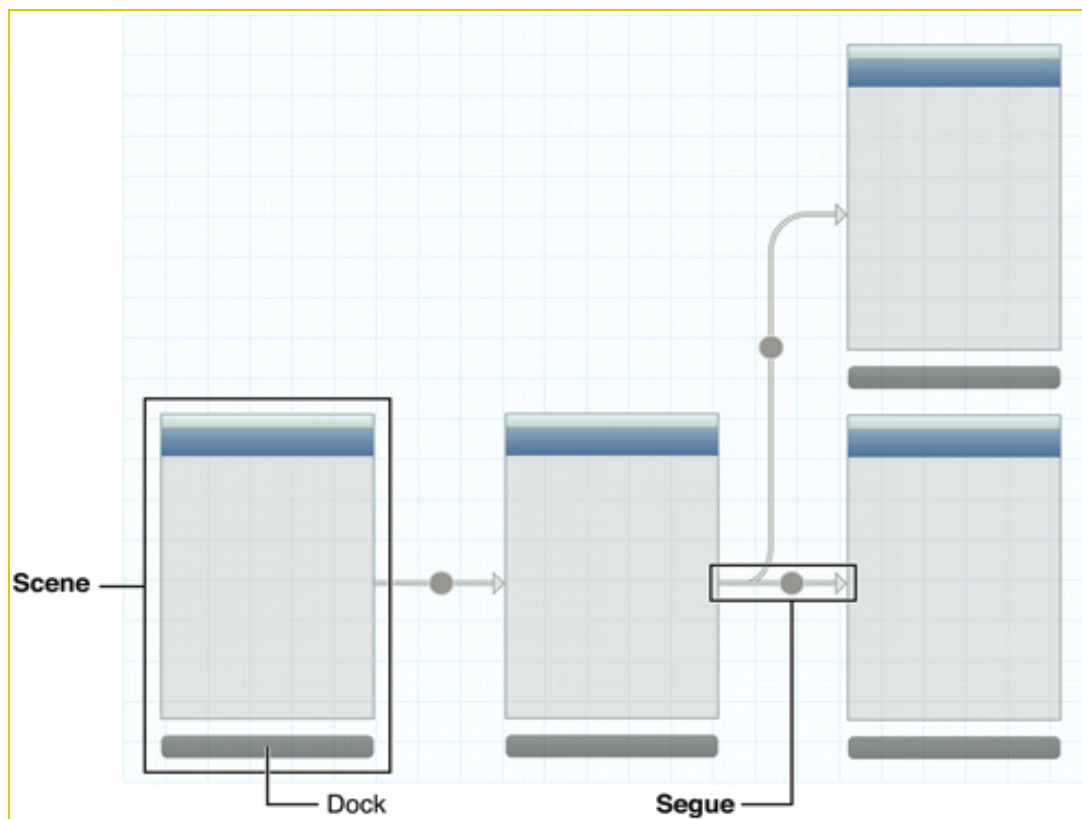
## 2. Navigation controller

- Mục đích sử dụng trong ứng dụng
- Khảo sát lớp UINavigationController
- Xây dựng Navigation Controller



# 1.1 Tổng quan về Storyboard

- ❑ Storyboard là một cách thức tổ chức, quản lý các tập tin giao diện của ứng dụng ở những version trước của iOS thường gọi chung là User Interface Builder





# 1.1 Tổng quan về Storyboard

---

- ❑ Storyboard được apple giới thiệu ra mắt khi IOS 5 ra đời.
- ❑ Storyboard bao gồm một chuỗi trình tự các view. Mỗi view có một controller điều khiển hoặc chung đại diện cho mình được kết nối với nhau thông qua đối tượng segue là đối tượng dùng để chuyển tiếp giữa hai view trên một storyboard



## 1.2 Khảo sát lớp UIStoryboard

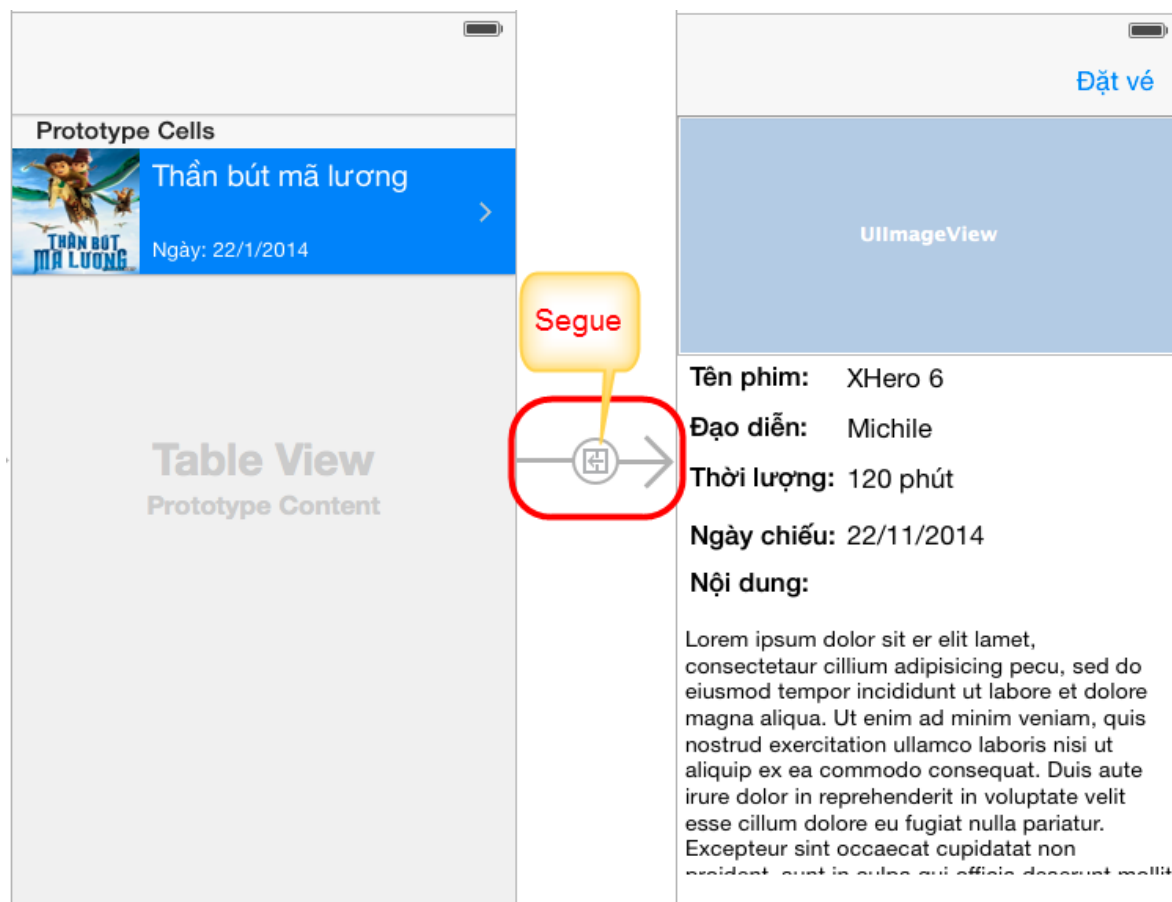
---

- ❑ **+ storyboardWithName: bundle:** Tạo ra một đối tượng storyboard theo một resource storyboard đã thiết kế..
- ❑ **- (id)instantiateInitialViewController:** Khởi tạo và trả về một View Controller ban đầu của một storyboard.
- ❑ **- (id)instantiateViewControllerWithIdentifier:** Khởi tạo và trả về một View Controller với một tên định danh



## 1.3 Các segue trong storyboard

- ❑ Segue là một đối tượng thuộc lớp UIStoryboardSegue, nó dùng để liên kết các View Controller trong storyboard với nhau





## 1.3 Các segue trong storyboard

---

- ❑ Một View Controller có thể liên kết tới một hoặc nhiều View Controller khác. Do đó ta có thể tạo nhiều segue cho một View Controller
  
- ❑ Segue bao gồm các loại như sau:
  - **Push** : Được sử dụng trong trường hợp có Navigation Controller.
  - **Modal** : Gọi giữa các View Controller với nhau.
  - **Custom** : Các loại segue tùy chỉnh.



## 1.4 Ưu điểm và hạn chế của storyboard

---

### ❑ Ưu điểm:

- Giúp cho việc hình dung, quan sát luồng UI một cách dễ dàng.
- Kiểm soát cũng như thiết kế các chuyển động của các màn hình dễ dàng hơn.
- Giúp giảm số lượng mã lệnh để xử lý việc chuyển đổi các View Controller, thay vào đó ta có thể dùng segue trong storyboard sẽ đơn giản hơn rất nhiều.

### ❑ Nhược điểm:

- Với project phức tạp có nhiều số lượng màn hình, khi đó sơ đồ trên Storyboard sẽ rối mắt.
- Mang lại sự bất tiện khi làm việc với nhóm. Vì mọi thành viên đều có thể sẽ cần sửa đổi, thêm xóa các thành phần trong storyboard nên sẽ dẫn đến bị đụng (conflict).





# Nội dung

---

## 1. Storyboard

- Tổng quan về storyboard
- Khảo sát lớp UIStoryboard
- Các segue trong storyboard
- Ưu điểm và hạn chế của storyboard

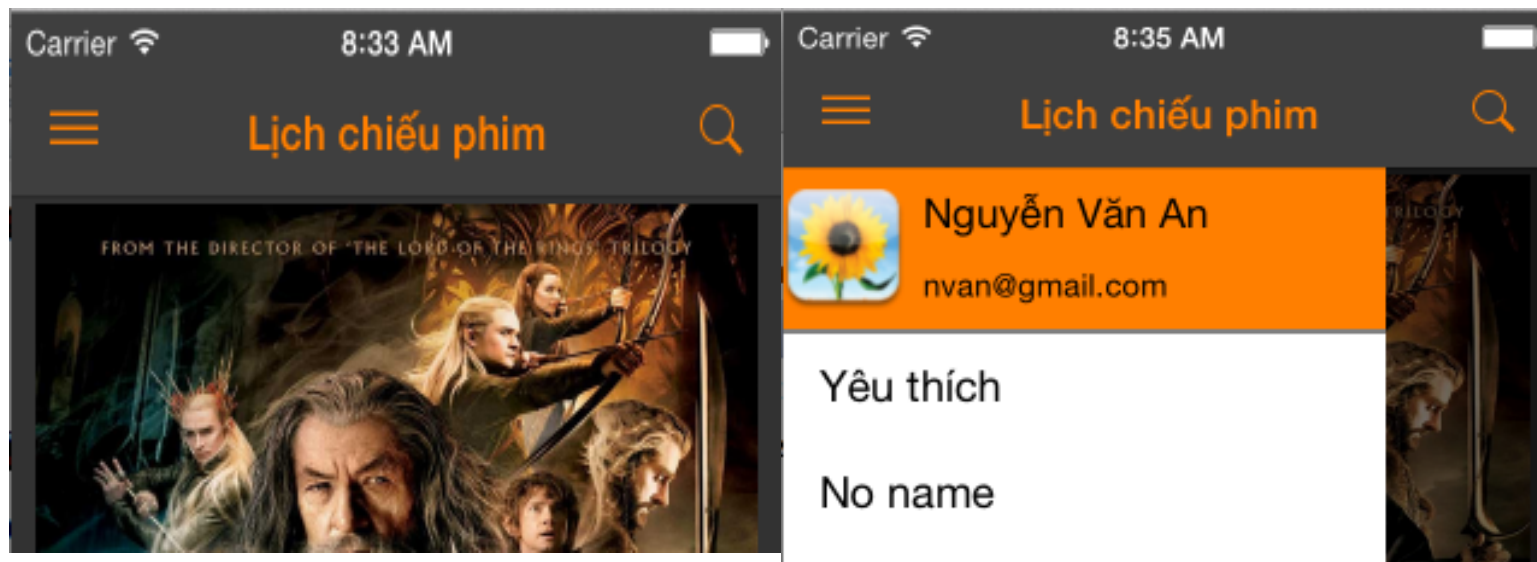
## 2. Navigation controller

- Mục đích sử dụng trong ứng dụng
- Khảo sát lớp UINavigationController
- Xây dựng Navigation Controller



# 1.1 Mục đích sử dụng trong ứng dụng

- ❑ Navigation controller được sử dụng nhằm hiển thị tiêu đề của màn hình, hỗ trợ chuyển hướng các màn hình khác.





## 1.2 Khảo sát lớp UINavigationController

---

### ❑ Phương thức khởi tạo

- - **initWithRootViewController**: phương thức khởi tạo navigation controller.
- - **initWithNavigationBarClass:toolbarClass**: phương thức khởi tạo navigation controller sử dụng lớp toolbar tùy chỉnh.



## 1.2 Khảo sát lớp UINavigationController

---

### ❑ Truy cập vào item trên Navigation Stack

- **topViewController**: thuộc tính cho biết view controller nằm ở trên cùng của stack, nghĩa là view controller được push vào gần nhất.
- **visibleViewController**: thuộc tính cho biết view controller đang được hiển thị. View controller này có thể là **topViewController** trong stack hoặc là một view controller được **topViewController** gọi hiển thị theo kiểu modal.
- **viewControllers**: danh sách các view controller hiện có trong stack
- - **setViewControllers:animated**: phương thức hỗ trợ thay thế các view controller có trong stack hiện tại



## 1.2 Khảo sát lớp UINavigationController

---

### ❑ Push và pop Stack item

- - **pushViewController:animated:** phương thức push view controller vào trong stack đồng thời hiển thị view controller lên màn hình
- - **popViewControllerAnimated:** lấy topViewController ra khỏi stack. Đồng nghĩa với việc hiển thị màn hình trước đó. Lúc này màn hình trước đó sẽ trở thành topViewController.
- - **popToRootViewControllerAnimated:** lấy ra tất cả các view controller có trong stack ngoại trừ view controller gốc và hiển thị view controller này lên màn hình
- - **popToViewController:animated:** là phương thức lấy ra các view controller cho đến view controller truyền vào.



# 1.2 Khảo sát lớp UINavigationController

---

## ❑ Tùy chỉnh thanh Navigation

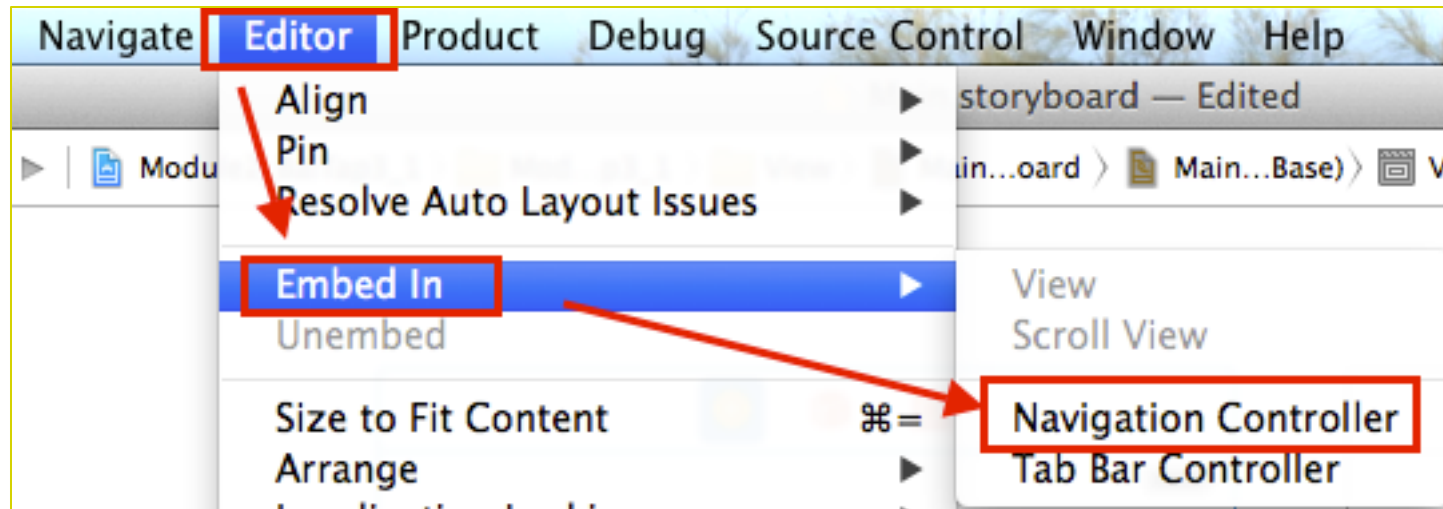
- **navigationBar**: là thuộc tính lưu trữ đối tượng navigationBar.
- - **setNavigationBarHidden:animated**: phương thức cấu hình việc ẩn hiển của navigation bar
- **hidesBarsOnTap**: thuộc tính quy định ẩn navigation bar khi có sự kiện **tap** trên màn hình.
- **hidesBarsOnSwipe**: thuộc tính quy định ẩn navigation bar khi có sự kiện **swipe** trên màn hình.
- **hidesBarsWhenVerticallyCompact**: thuộc tính quy định ẩn navigation bar khi người dùng xoay ngang màn hình.
- **hidesBarsWhenKeyboardAppears**: thuộc tính quy định ẩn navigation bar khi người dùng bàn phím hiển thị.
- **navigationBarHidden**: thuộc tính quy định ẩn navigation bar.



## 1.3 Xây dựng Navigation Controller

### ❑ Xây dựng Navigation Controller từ Storyboard:

- Bấm chọn View Controller → Trên thanh công cụ, chọn "**Editor**" → chọn "**Embed In**" → chọn "**Navigation Controller**"





## 1.3 Xây dựng Navigation Controller

### ❑ Xây dựng Navigation Controller từ mã nguồn:

```
- (BOOL)application:(UIApplication *)application
didFinishLaunchingWithOptions: (NSDictionary *)launchOptions {
    UIStoryboard *storyboard = [UINavigationController storyboardWithName:
@"Main" bundle: nil];
    ViewController *myViewController = [storyboard
instantiateViewControllerWithIdentifier: @"MH1"];
    UINavigationController *navigationController =
[[UINavigationController alloc] initWithRootViewController:
myViewController];
    self.window.rootViewController = navigationController;
    return YES;
}
```





## 1.3 Xây dựng Navigation Controller

- ❑ Để chuyển đổi sang một View Controller khác ta sử dụng phương thức push và pop

- Dùng phương thức push để hiển thị một View Controller.

```
SecondViewController *view = [[SecondViewController alloc] init];  
[self.navigationController pushViewController:view animated:YES];
```

- Dùng phương thức pop để quay trở lại View Controller trước.

```
[self.navigationController popToRootViewControllerAnimated:YES];  
[self.navigationController popViewControllerAnimated:YES];
```



## 1.3 Xây dựng Navigation Controller

- ❑ Sau khi ta thiết lập Navigation làm root view cho ứng dụng khi chạy ứng dụng thì phía trên cùng của màn hình sẽ xuất hiện thanh navigation.



