



Lập trình iOS

Bài 5. Ứng dụng iOS và cách thức hoạt động

Ngành Mạng & Thiết bị di động





Nội dung

1. Kiến trúc ứng dụng iOS

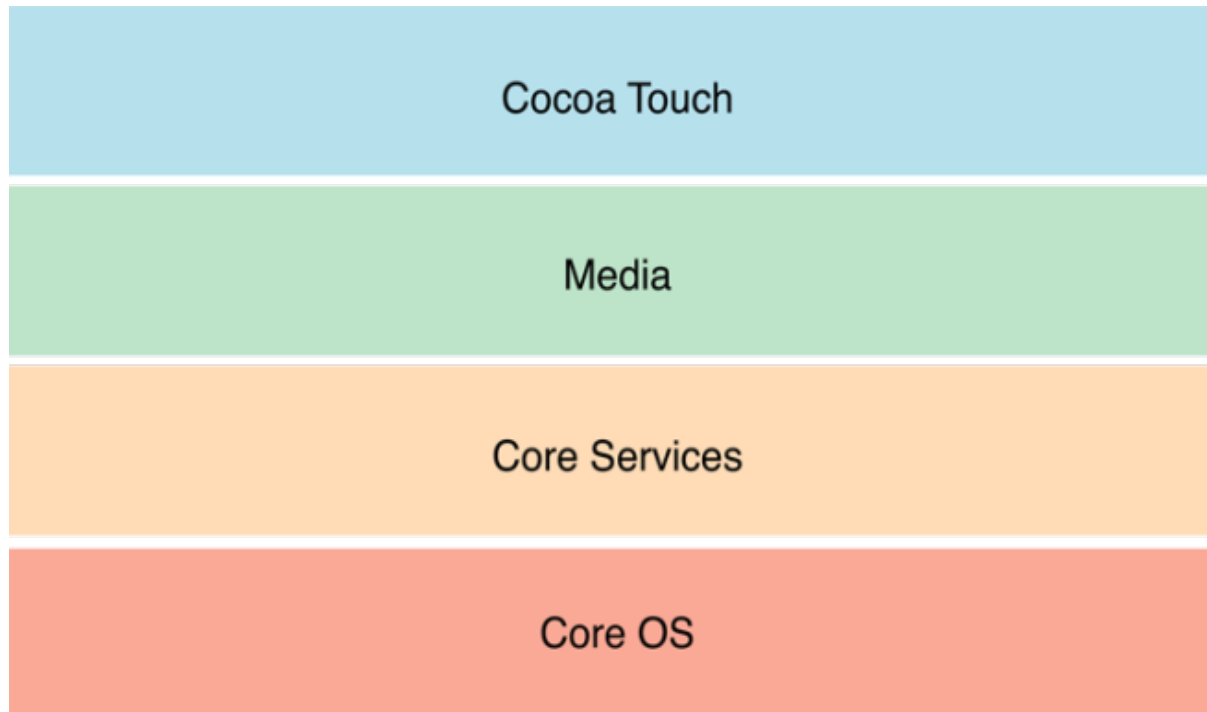
- Cocoa Touch
- Media
- Core Service
- Core OS

2. Quy trình xây dựng ứng dụng



1. Kiến trúc ứng dụng iOS

- ❑ Ứng dụng thực thi trên môi trường iOS giao tiếp với thiết bị phần cứng thông qua tập các lớp giao diện từ ngôn ngữ cấp thấp đến ngôn ngữ cấp cao.
- ❑ Kiến trúc ứng dụng iOS bao gồm:





1.1 Cocoa Touch

- ❑ Cocoa Touch là bộ phát triển ứng dụng iOS dựa trên nền tảng Cocoa tập trung vào các xử lý chạm màn hình.
- ❑ Các bộ nền tảng trong Cocoa Touch cung cấp bao gồm:
 - Address Book UI
 - Event Kit UI
 - Game Kit
 - iAd
 - Map Kit
 - Message UI
 - Twitter
 - UIKit



1.2 Media

- ❑ Media là lớp cơ sở cho phép lập trình viên có thể thực hiện các xây dựng các ứng dụng đáp ứng hầu hết các tính năng đa truyền thông về âm thanh, hình ảnh, chuyển hoạt...
- ❑ Các bộ nền tảng được Media cung cấp bao gồm:
 - Asset Library
 - AV Foundation
 - Core Audio
 - Core Graphic
 - Core Image
 - Core Text
 - Core Video
 - Game Controller
 - GLKit
 - Image I/O
 - Media Accessibility
 - MediaPlayer
 - OpenAL
 - OpenGL ES
 - Quartz Core
 - Sprite Kit



1.3 Core Service

- ❑ Core Service cung cấp hầu hết các dịch vụ cơ bản được sử dụng trong ứng dụng như định vị, mạng kết nối, điện toán đám mây...
- ❑ Các bộ nền tảng được Core Service cung cấp bao gồm:
 - Account
 - Address Book
 - Ad support
 - CF Network
 - Core Data
 - Core Foundation
 - Core Location
 - Core Media
 - Core Motion
 - Core Telephony
 - Event Kit
 - Foundation
 - JavaScript Core
 - Mobile Core Service
 - Mulipeer
 - Newsstand Kit
 - Pass Kit
 - Quick Look
 - Safari
 - Store
 - System Configuration



1.4 Core OS

- ❑ Core OS là lớp truy xuất cấp thấp đến phần cứng thiết bị, thường được sử dụng gián tiếp thông qua các lớp khác. Khi đó các khai báo về trao đổi thông tin và bảo mật là điều bắt buộc.
- ❑ Các bộ nền tảng Core OS cung cấp bao gồm:
 - Accelerate
 - Core Bluetooth
 - External Accessory
 - Generic Security Services
 - Security
 - System
 - 64-Bit Support



Nội dung

1. Kiến trúc iOS

2. Quy trình xây dựng ứng dụng

- Phát thảo ý tưởng và định hướng phát triển ứng dụng
- Thiết kế giao diện
- Tổ chức dữ liệu
- Xây dựng tương tác
- Kiểm định ứng dụng và đưa ứng dụng lên Apple Store



2.1 Phát thảo ý tưởng - định hướng phát triển

Lịch Chiếu Phim

Đang chiếu | Sắp chiếu

Prototype Cells

- Hobbit 2 - Desolation of Smaug
- Đạo diễn:

Màn hình xem lịch phim:

- Thanh chứa biểu tượng và nút tìm kiếm
- Phim đang chiếu và sắp chiếu
- Danh sách lịch xem phim
- Thông tin phim gồm:
 - Poster
 - Tên phim
 - Đạo diễn
 - Đánh giá...

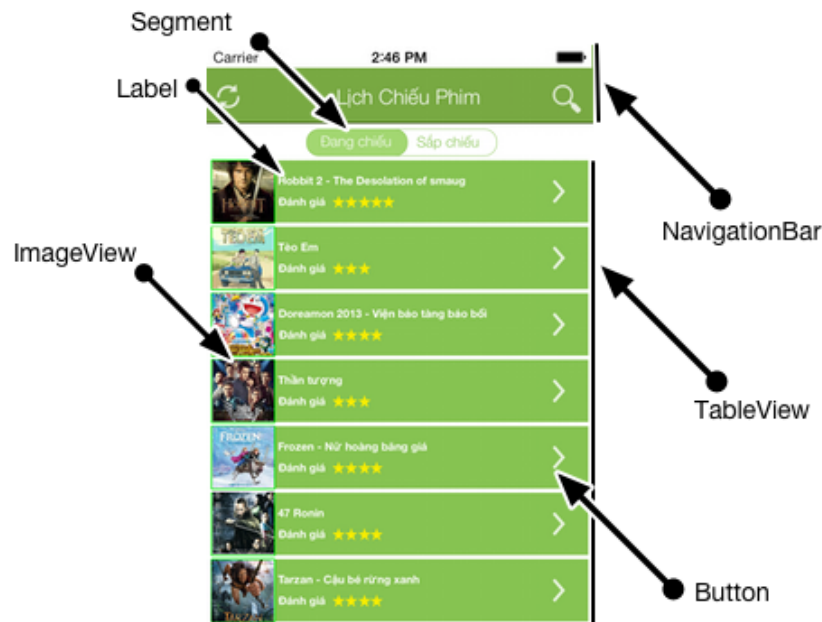
```
#import "HTSAppDelegate.h"
@implementation HTSAppDelegate
- (BOOL)application:(UIApplication *)application
    launchOptions:(NSDictionary *)launchOptions
{
    // Override point for customization after application launch.
    return YES;
}
- (void)applicationWillResignActive:(UIApplication *)application
{
    // Sent when the application is about to move into the background.
    // Use this method to save user data to disk and stop long-running
    // processes and other power-intensive activities that are not
    // suitable for execution while the application is in the background.
```

Poster	Tên phim	Đánh giá
	Hobbit 2 - The Desolation of Smaug	★★★★★
	Tèo Em	★★★
	Doreamon 2013 - Viện bảo tàng bảo bối	★★★★★
	Thần tượng	★★★
	Frozen - Nữ hoàng băng giá	★★★★★
	47 Ronin	★★★★★
	Tarzan - Cậu bé rừng xanh	★★★★★



2.2 Thiết kế giao diện

- ❑ Trong thiết kế giao diện ứng dụng iOS, View được xem là thành phần quan trọng nhất. View cho phép cấu thành nên tất cả các thuộc tính một màn hình giao diện cần có.
- ❑ Khi thiết kế ứng dụng, cần lên trước kế hoạch các đối tượng View sẽ sử dụng, vì nó ảnh hưởng toàn bộ quá trình phát triển về sau.





2.2 Thiết kế giao diện

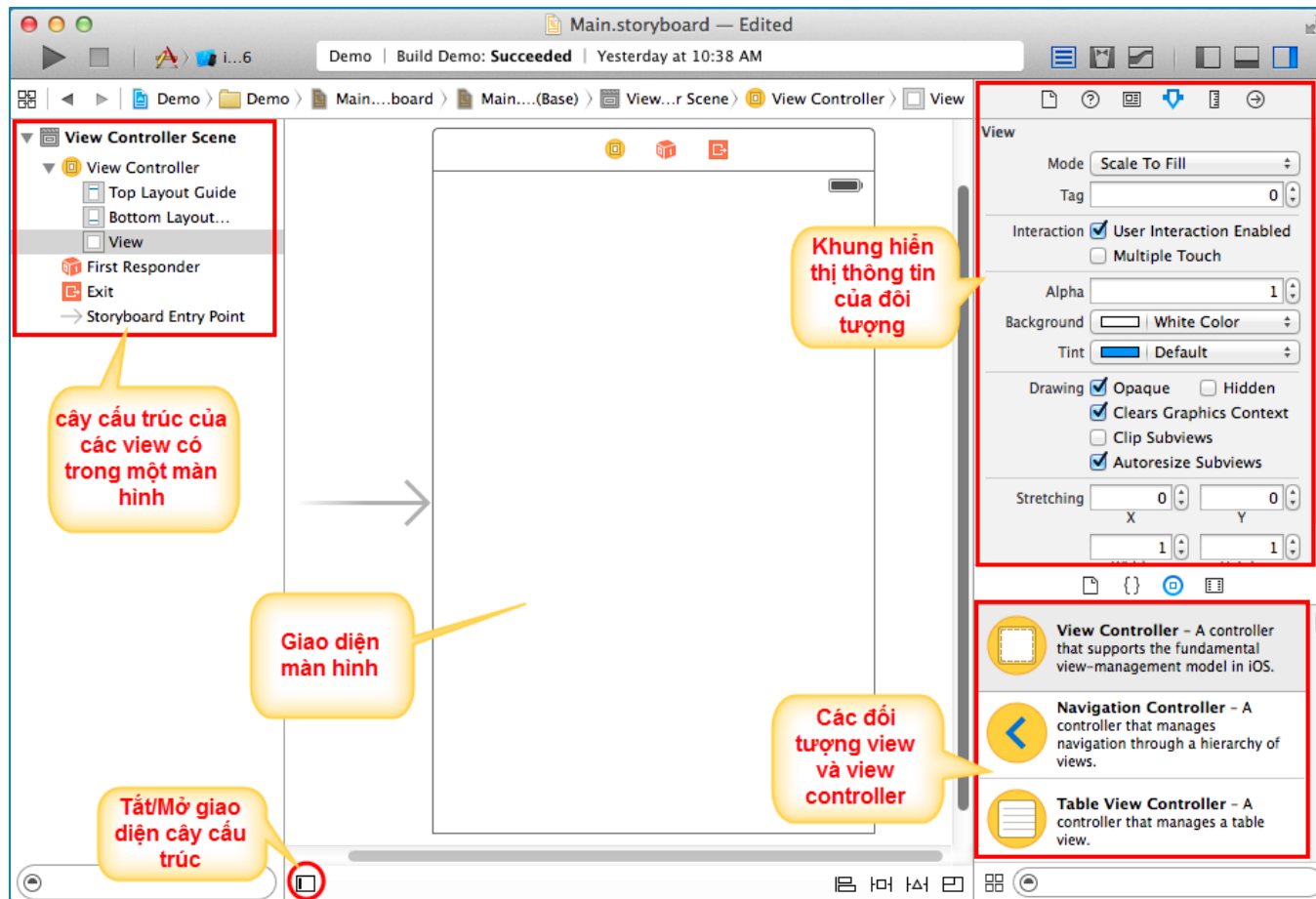
- ❑ Tùy loại đối tượng View có những thuộc tính khác nhau, tuy nhiên có thể gom thành 7 nhóm sau:

Nhóm đối tượng	Chức năng	Ví dụ
Nội dung	Hiển thị một nội dung bất kì dựa trên thuộc tính đối tượng.	ImageView, Label
Tập hợp	Hiển thị tập hợp hoặc nhóm các đối tượng.	CollectionView, TableView
Điều khiển	Cho phép thực thi tương tác hoặc hiển thị thông tin.	Button, Slider, Switch
Thanh điều khiển	Gom nhóm các tương tác hoặc điều hướng.	Toolbar, Tabbar, Navigationbar
Nhập liệu	Nhận dữ liệu được nhập từ người dùng	Searchbar, TextView
Nhóm điều khiển	Chứa các điều khiển khác nhau	View, ScrollView
Thông báo	Bao gồm các điều khiển cho phép hiển thị thông báo trong ứng dụng	ActionSheet, UIAlertView

2.2 Thiết kế giao diện



- ❑ Xcode hỗ trợ Interface Builder cho phép lập trình viên xây dựng giao diện một cách tường minh.





2.3 Tổ chức dữ liệu

❑ Tổ chức dữ liệu cho phép ta miêu tả các đối tượng ở thế giới thực vào mã nguồn.

- Ví dụ: ứng dụng xem lịch chiếu phim cần mô tả đối tượng Phim bao gồm các thuộc tính:

- Tên phim
- Thể loại
- Đạo diễn
- Ngày chiếu
- Thời lượng
- Diễn viên
- Nội dung tóm tắt
- ...



Đặt vé

Đánh giá

Đạo diễn: Peter Jackson

Thời lượng: 2 giờ và 41 phút

Thể loại: Phiêu Lưu, Tâm lý, KH Viễn Tưởng/
Thần Thoại

Khởi chiếu: 03/01/2014

Phần 2 tiếp tục kể về cuộc phiêu lưu của nhân vật chính Bilbo Baggins và cuộc hành trình với phù thủy Gandalf cùng mười ba người lùn dẫn đầu bởi Thorin Oakenshield, họ mang trên mình một nhiệm vụ sử thi để đòi lại Lonely Mountain và Vương quốc người lùn đã mất của Erebor.

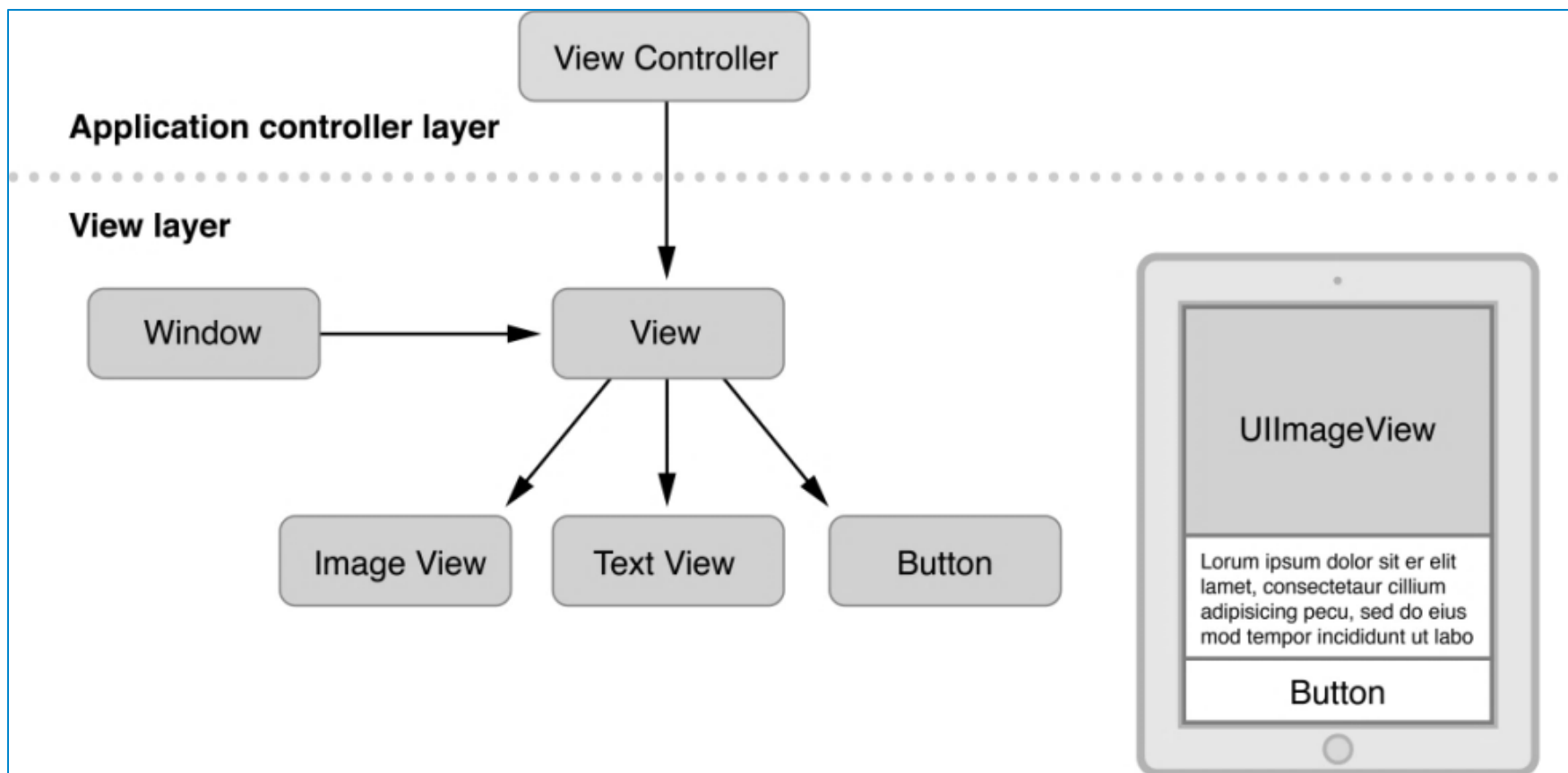


2.4 Xây dựng tương tác

- ❑ Mỗi giao diện màn hình đều có một View Controller (UIViewController) được dùng để quản lý view và các subview (các view được sắp xếp nằm bên trong view) nằm trên màn hình đó.
- ❑ Mỗi View Controller được liên kết với một lớp kế thừa từ UIViewController. Lớp này sẽ quản lý mọi hoạt động của View Controller, là nơi cho phép chúng ta viết mã lệnh để can thiệp vào các phương thức có sẵn của View Controller hoặc tạo mới các phương thức theo ý đồ xây dựng ứng dụng của mình.



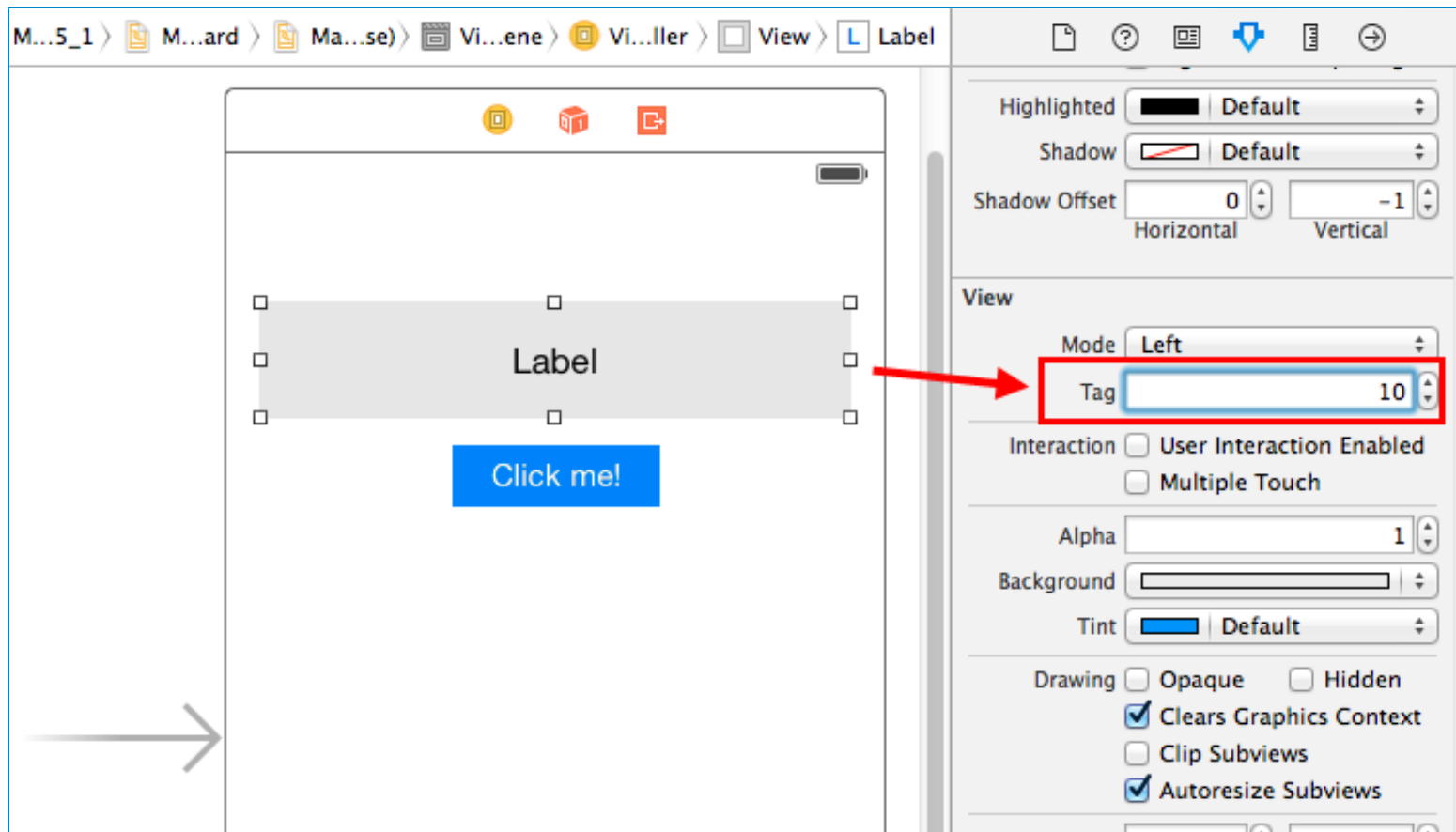
2.4 Xây dựng tương tác





2.4 Xây dựng tương tác

- ❑ Để thao tác tới các view chỉ định ta có thể thông qua thuộc tính tag của view.

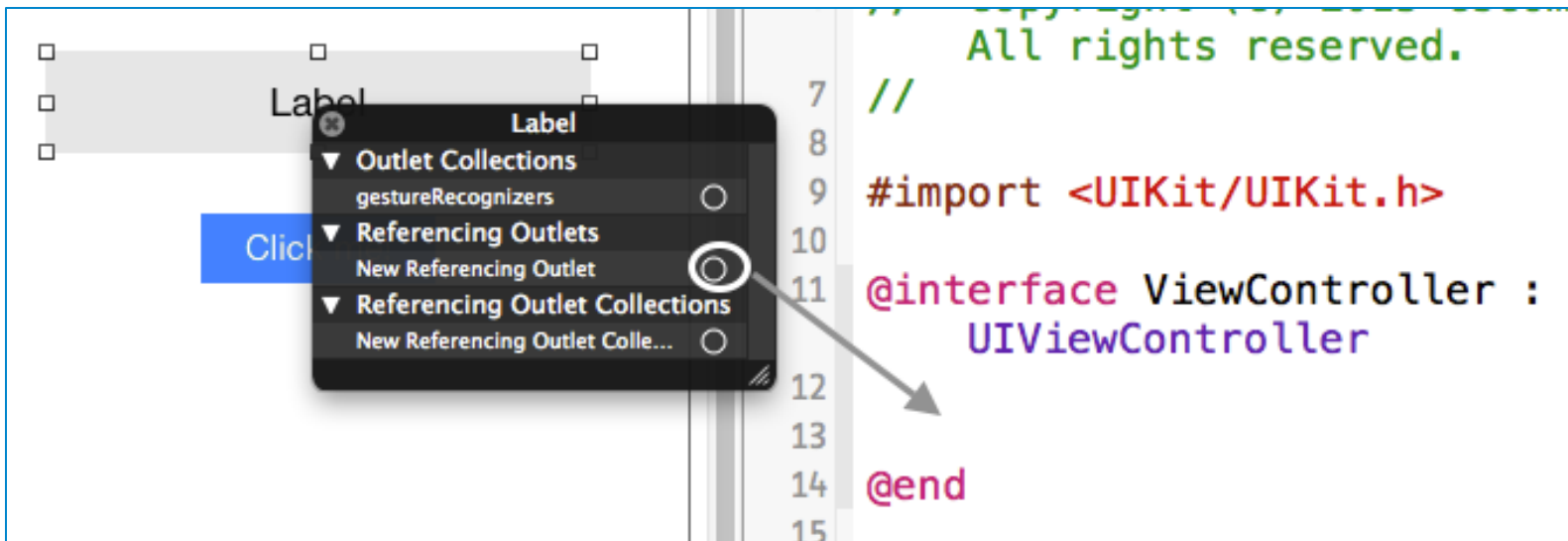




2.4 Xây dựng tương tác

❑ Kết nối IBOutlet:

- Là một dạng khai báo đối tượng view trở thành một thuộc tính trong lớp quản lý View Controller.
- Giúp sử dụng đối tượng view như một thuộc tính của lớp mà không cần phải đặt tag và nhớ tag.
- Tất cả các đối tượng view đều có thể kết nối IBOutlet





2.4 Xây dựng tương tác

- Khi kết nối thành công thì Xcode sẽ tự động sinh mã khai báo một thuộc tính dạng IBOutlet và vòng tròn cạnh thuộc tính này sẽ có tâm như hình bên dưới:

Kết nối thành công

```
8
9  #import <UIKit/UIKit.h>
10
11 @interface ViewController :
    UIViewController
12 @property (strong, nonatomic)
    IBOutlet UILabel *lblMessage;
13 @end
14
--
```

```
10
11 @interface ViewController :
    UIViewController
12 @property (strong, nonatomic)
    IBOutlet UILabel *lblMessage;
13
14 @end
```

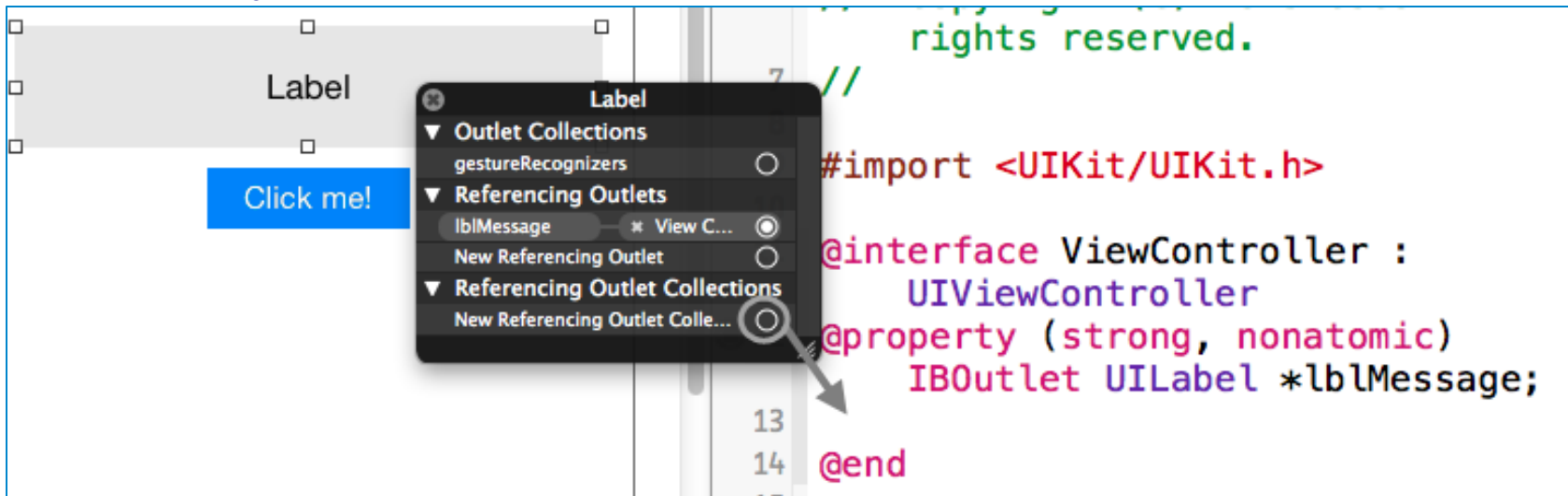
Không thành công



2.4 Xây dựng tương tác

❑ IBOutletCollection:

- Là một tập hợp các đối tượng view nhằm hỗ trợ việc xử lý một công việc chung cho nhiều đối tượng cùng một lúc.
- Một đối tượng view có thể kết nối với nhiều IBOutletCollection khác nhau.
- Thuộc tính này là một danh sách các đối tượng view được lưu theo kiểu NSArray





2.4 Xây dựng tương tác

❑ IBAction:

- Một loại kết nối hỗ trợ tạo ra phương thức xử lý khi sự kiện xảy ra trên đối tượng view
- IBAction chỉ hỗ trợ cho những đối tượng nào kế thừa từ lớp UIControl bao gồm: UIButton, UIDatePicker, UIPageControl,...





2.4 Xây dựng tương tác

The screenshot shows the Xcode interface. On the left, a UIButton is visible with the text 'Click me!'. A context menu is open over the button, listing various touch events. The 'Touch Up Inside' event is selected, and an arrow points from it to the '@end' line in the code editor on the right.


Code in the editor:

```
5 // Created by
   on 4/16/15.
6 // Copyright (c) 2015 cscom.
   All rights reserved.
7 //
8
9 #import <UIKit/UIKit.h>
10
11 @interface ViewController :
   UIViewController
12 @property (strong, nonatomic)
   IBOutlet UILabel *lblMessage
13 ;
14
15
16 @end
17
18
```



2.4 Xây dựng tương tác

```
13 |
14 | #import <UIKit/UIKit.h>
15 |
16 | @interface ViewController :
    UIViewController
17 | @property (strong, nonatomic)
    IBOutlet UILabel *lblMessage;
18 | - (IBAction)clickButton:(id)
    sender;
19 | @end
20 |
21 | - (void)didReceiveMemoryWarning {
22 |     [super didReceiveMemoryWarning];
23 |     // Dispose of any resources that can
    be recreated.
24 | }
25 |
26 | - (IBAction)clickButton:(id)sender {
27 |
28 | }
29 |
```





Thảo luận

