

MỤC LỤC

MỤC LỤC.....	1
PHẦN MỞ ĐẦU	1
1. Tên đề tài.....	1
2. Lý do chọn đề tài.....	1
3. Lịch sử nghiên cứu.....	2
3.1 Trên thế giới.....	2
3.2 Trong nước.....	2
4. Mục tiêu nghiên cứu.....	3
5. Phương pháp nghiên cứu.....	3
6. Những đóng góp mới của đề tài và những vấn đề chưa thực hiện được.....	3
6.1 Những đóng góp mới của đề tài.....	3
6.2 Những vấn đề chưa thực hiện được.....	4
7. Kết luận của đề tài.....	4
CHƯƠNG 1: GIỚI THIỆU VỀ HỆ ĐIỀU HÀNH iOS VÀ NGÔN NGỮ LẬP TRÌNH OBJECTIVE-C.....	6
1.1 Giới thiệu hệ điều hành iOS.....	6
1.1.1 Hệ điều hành iOS là gì?.....	6
1.1.2 Kiến trúc của hệ điều hành iOS.....	6
1.2 Giới thiệu ngôn ngữ lập trình Objective-C.....	8
1.3 Tiểu kết.....	9
CHƯƠNG 2: GIỚI THIỆU VỀ BỘ CÔNG CỤ XCODE VÀ IOS SIMULATOR	10
2.1 Tìm hiểu bộ công cụ Xcode.....	10
2.1.1 Giới thiệu về Xcode.....	10
2.1.2 Một số đặc điểm của Xcode.....	10
2.2 Tìm hiểu iOS Simulator.....	13
2.2.1 Giới thiệu về iOS Simulator.....	13
2.2.2 Một số điểm hạn chế của iOS Simulator.....	15
2.3 Tiểu kết.....	15
CHƯƠNG 3: THAO TÁC XÂY DỰNG ỨNG DỤNG VÀ ĐƯA ỨNG DỤNG LÊN IPHONE.....	16
3.1 Giới thiệu về mô hình MVC.....	16
3.2 Thao tác xây dựng ứng dụng đơn giản.....	17
3.3 Đưa ứng dụng lên iPhone.....	25
3.3.1 Giới thiệu.....	25
3.3.2 Chuẩn bị.....	25
3.3.3 Tiến hành.....	27
3.4 Tiểu kết.....	34
CHƯƠNG 4: MỘT VÀI ỨNG DỤNG ĐÃ ĐƯỢC XÂY DỰNG.....	35
4.1 Ứng dụng tìm kiếm địa điểm xung quanh (PlacesNearMe).....	35
4.1.1 Giới thiệu phần mềm.....	35
4.1.2 Cấu trúc của phần mềm.....	35
4.1.3 Cơ chế hoạt động của phần mềm.....	36
4.1.4 Tính năng của phần mềm.....	37

4.1.5 Ưu điểm và những hạn chế của phần mềm.....	40
4.2 Ứng dụng tra cứu mã thông tin mã VIN.....	40
4.2.1 Giới thiệu phần mềm.....	40
4.2.2 Cấu trúc của phần mềm.....	42
4.2.3 Cơ chế hoạt động của phần mềm.....	43
4.2.4 Tính năng của phần mềm.....	44
4.2.5 Ưu điểm và những hạn chế của phần mềm.....	47
4.3 Tiêu kết.....	48
CHƯƠNG 5: TÀI LIỆU HƯỚNG DẪN XÂY DỰNG ỨNG DỤNG TRÊN IPHONE.....	49
5.1 Nhu cầu thực tiễn.....	49
5.2 Cơ sở nền tảng để xây dựng hướng dẫn.....	49
5.3 Xây dựng tài liệu hướng dẫn xây dựng ứng dụng trên iPhone.....	50
5.3.1 Bố cục tài liệu hướng dẫn.....	50
5.3.2 Dạng sách điện tử của tài liệu hướng dẫn.....	53
5.4 Tiêu kết.....	54
CHƯƠNG 6: ĐÁNH GIÁ KẾT QUẢ ĐẠT ĐƯỢC.....	54
6.1 Đóng góp của đề tài vào thực tiễn.....	54
6.2 Ưu điểm và hạn chế của đề tài.....	54
KẾT LUẬN.....	56
TÀI LIỆU THAM KHẢO.....	58

PHẦN MỞ ĐẦU

1. Tên đề tài

“XÂY DỰNG ỨNG DỤNG TRÊN IPHONE”

2. Lý do chọn đề tài

Ngày nay xu hướng sử dụng Smartphone và máy tính bảng đang gia tăng nhanh chóng trên thế giới nói chung và Việt Nam nói riêng, trong đó Việt Nam hiện đang đứng thứ hai thế giới về tốc độ tăng trưởng smartphone và máy tính bảng với tốc độ tăng trưởng 266%. Android, iOS, Windows Phone là những hệ điều hành chạy trên Smartphone và máy tính bảng phổ biến nhất thế giới: Android 75%, iOS 17,3%, Windows Phone 3,2%. Tại Việt Nam, theo nghiên cứu của IDC, vào thời điểm quý 2/2013, iOS đang chiếm tỉ lệ 1.6% trên tổng số thiết bị phân phối tại Việt Nam, đứng thứ ba sau Android và Windows Phone. [8],[12]

Cùng với sự tăng trưởng của Smartphone và các hệ điều hành chạy trên Smartphone, số lượng ứng dụng cho các hệ điều hành ngày càng tăng, tính cho đến hết năm 2012, số lượng ứng dụng iOS trên Apple App Store đã hơn 775.000 ứng dụng và Google Play đã có hơn 700.000 ứng dụng. Với sự phát triển quy mô lớn của ứng dụng, nhu cầu tìm hiểu về lập trình ứng dụng cho các hệ điều hành cũng tăng dần. [4]

Tuy nhiên, thực tiễn cho thấy, việc tìm hiểu cũng như tham gia các lớp học về lập trình ứng dụng iPhone ở Việt Nam còn nhiều hạn chế và khó khăn. Các lớp dạy lập trình ứng dụng iPhone chỉ mới xuất hiện nhiều trong thời gian gần đây, nhưng số lượng vẫn còn hạn chế.

Bên cạnh đó nguồn tài liệu tiếng Việt còn ít, việc tìm hiểu và sử dụng công cụ lập trình, cũng như tham khảo tài liệu tiếng Anh về lập trình ứng dụng iPhone đòi hỏi phải tiêu tốn một khoảng thời gian dài cũng như một ít hiểu biết về lập trình và đọc hiểu tiếng Anh tốt.

Vì vậy, với mong muốn tìm hiểu cách xây dựng ứng dụng iPhone để có thêm kiến thức mới, giúp ích cho quá trình làm việc sau khi ra trường cũng như giảm bớt những khó khăn cho người mới bắt đầu tìm hiểu về lập trình ứng dụng trên iPhone, nhóm thực hiện đề tài đã thực hiện nghiên cứu, xây dựng một số ứng dụng trên iPhone

dựa trên kiến thức tìm hiểu được, từ đó tổng hợp và xây dựng thành tài liệu hướng dẫn xây dựng ứng dụng trên iPhone.

3. Lịch sử nghiên cứu

3.1 Trên thế giới

Trên thế giới hiện tại đã có nhiều tài liệu hướng dẫn liên quan đến việc xây dựng ứng dụng trên iPhone cũng như nhiều trường Đại học ở Mỹ cũng đã đưa vào chương trình học môn học về lập trình ứng dụng cho iPhone – iPad. Một số quyển sách tiêu biểu trong số nhiều tài liệu hướng dẫn như:

- iPhone Application Programming WS (RWTH Aachen University – 21/10/2013).
- Developing iOS 7 Apps for iPhone and iPad (Standford University – 28/10/2013).
- Coding Together: Developing iOS 6 (Standford University – 15/1/2013).
- iPhone Application Programming (RWTH Aachen University – 11/10/2012).
- Programming iOS 6, 3rd Edition (Matt Neuburg – 3/2013).
- Head First iPhone And iPad Development, 2nd edition (O'Reilly – 2011).
- iOS 6 Application Development for Dummies (Neal Goldstein – Dave Wilson – 2013).
- iPhone and iPad Apps for Absolute Beginners, 3rd edition (Apress – 2013).

3.2 Trong nước

Theo tìm hiểu của nhóm thì hiện nay ở Việt Nam đã có nhiều công trình nghiên cứu về xây dựng ứng dụng iPhone. Tuy nhiên mục đích nghiên cứu của các công trình nghiên cứu này chỉ nghiên cứu việc xây dựng ứng dụng iPhone để phục vụ cho việc xây dựng một ứng dụng cụ thể chạy trên nền tảng iOS. Một vài công trình nghiên cứu về xây dựng ứng dụng iPhone như:

- Xây dựng hệ thống quản lý thông tin cá nhân trên iPhone (Trương Quốc Phú – Đại Học Bách Khoa Hà Nội – 6/2010).
- Xây dựng ứng dụng tìm kiếm đa phương tiện trên điện thoại iPhone (An Thị Hồng – Đại Học Bách Khoa Hà Nội – 6/2010).
- Nghiên cứu xây dựng ứng dụng cho hệ điều hành iOS (Hoàng Kim Ngọc – Đại Học Dân Lập Hải Phòng – 7/2012).

4. Mục tiêu nghiên cứu

Tìm hiểu cách thức lập trình mô phỏng với bộ công cụ Xcode để tạo ứng dụng chạy trên hệ điều hành iOS.

Xây dựng ứng dụng cơ bản cho iPhone bằng Xcode.

Xây dựng tài liệu hướng dẫn xây dựng ứng dụng iPhone bằng bộ công cụ Xcode.

5. Phương pháp nghiên cứu

Tìm hiểu và thu thập tài liệu hướng dẫn tiếng Anh và tiếng Việt về xây dựng ứng dụng trên iPhone bằng công cụ Xcode qua sách và trên mạng internet.

Tìm hiểu về ngôn ngữ lập trình Objective-C để hiểu hơn về ngôn ngữ lập trình ứng dụng iPhone được sử dụng nhiều nhất, nắm bắt được cách lập trình bằng ngôn ngữ này.

Tìm hiểu các đối tượng trong Xcode và sử dụng các đối tượng này để xây dựng ứng dụng cơ bản trên iPhone.

Tổng hợp kinh nghiệm từ quá trình sử dụng các đối tượng của Xcode để xây dựng ứng dụng iPhone, từ đó xây dựng thành tài liệu hướng dẫn.

6. Những đóng góp mới của đề tài và những vấn đề chưa thực hiện được

6.1 Những đóng góp mới của đề tài

Tìm hiểu được cách thức lập trình mô phỏng với Xcode từ đó xây dựng được hướng dẫn sử dụng công cụ Xcode để xây dựng ứng dụng trên iPhone bằng tiếng Việt giúp cho người tìm hiểu về xây dựng ứng dụng trên iPhone dễ tìm hiểu, tiếp cận và thực hành.

Hiện thực được quá trình tìm hiểu về xây dựng ứng dụng trên iPhone bằng hai ứng dụng hoàn chỉnh có tính ứng dụng vào thực tiễn là ứng dụng Tìm kiếm địa điểm xung quanh và ứng dụng Kiểm tra mã VIN.

Với tài liệu hướng dẫn bằng tiếng Việt có thể giúp những người mới bắt đầu tìm hiểu về xây dựng ứng dụng trên iPhone dễ dàng tìm hiểu, làm quen với công cụ Xcode và các đối tượng cơ bản trong Xcode được sử dụng để xây dựng ứng dụng trên iPhone. Điều này tiết kiệm được thời gian tìm hiểu cũng như việc tiếp thu đạt hiệu quả tốt hơn với những kiến thức, kinh nghiệm được đúc kết lại trong hướng dẫn, cũng như có thể thao tác theo hướng dẫn một cách dễ dàng với các hướng dẫn minh họa chi tiết, rõ ràng.

6.2 Những vấn đề chưa thực hiện được

Các đối tượng trong Xcode cũng như các vấn đề liên quan đến xây dựng ứng dụng iPhone rất đa dạng, do đó việc tìm hiểu và nghiên cứu xây dựng ứng dụng iPhone cần thực hiện trong thời gian dài mới có thể đạt được mức hiểu biết tương đối trong việc xây dựng ứng dụng iPhone. Do thời gian có hạn nên nhóm chỉ tìm hiểu được một vài đối tượng cơ bản trong Xcode, cũng như cách sử dụng các đối tượng đó để xây dựng ứng dụng iPhone.

Nhóm vẫn chưa đi sâu vào tìm hiểu được nhiều đối tượng khác cũng như tất cả các chức năng mà công cụ Xcode hỗ trợ người dùng để xây dựng ứng dụng iPhone.

7. Kết cấu của đề tài

Luận văn được chia thành ba phần: phần mở đầu, phần nội dung, phần kết luận.

Phần mở đầu: Trình bày lý do chọn đề tài, tình hình nghiên cứu trong và ngoài nước, mục tiêu và phương pháp nghiên cứu của đề tài, cũng như những đóng góp mới và những mặt hạn chế của đề tài.

Phần nội dung:

Chương 1: Giới thiệu tổng quan về hệ điều hành iOS và một số đặc điểm của hệ điều hành này. Bên cạnh đó còn giới thiệu cơ bản về ngôn ngữ lập trình Objective-C – ngôn ngữ được sử dụng phổ biến trong xây dựng ứng dụng trên iPhone.

Chương 2: Giới thiệu về bộ công cụ Xcode và iOS Simulator. Trong đó trình bày tổng quan về Xcode và iOS Simulator, trình bày một số đặc điểm và tính năng của Xcode, iOS Simulator.

Chương 3: Giới thiệu mô hình MVC trong xây dựng ứng dụng trên iPhone. Thêm vào đó sẽ có một minh họa căn bản về mô hình MVC và cách thức đưa ứng dụng lên iPhone.

Chương 4: Giới thiệu vài ứng dụng tiêu biểu do nhóm xây dựng dựa trên kiến thức tìm hiểu được. Trong đó trình bày tổng quan về giao diện, tính năng, một số điểm nổi bật cũng như hạn chế của ứng dụng.

Chương 5: Trình bày về cuốn tài liệu hướng dẫn xây dựng ứng dụng trên iPhone. Trong đó nêu lên nhu cầu thực tiễn của hướng dẫn, cơ sở lý thuyết để xây dựng nên tài liệu cũng như bối cảnh của tài liệu hướng dẫn. Ngoài ra còn đánh giá một số điểm hạn chế của tài liệu hướng dẫn.

Chương 6: Đánh giá kết quả đạt được của đề tài, những đóng góp mới do đề tài mang đến và ưu - khuyết điểm còn tồn tại.

Phản kết luận: Trình bày những kết luận về đề tài, kiến nghị về hướng nghiên cứu tiếp theo của đề tài.

CHƯƠNG 1: GIỚI THIỆU VỀ HỆ ĐIỀU HÀNH iOS VÀ NGÔN NGỮ LẬP TRÌNH OBJECTIVE-C

1.1 Giới thiệu hệ điều hành iOS

1.1.1 Hệ điều hành iOS là gì?

iOS là hệ điều hành trên các thiết bị di động của Apple, ban đầu hệ điều hành này chỉ được phát triển để chạy trên iPhone (gọi là iPhone OS), nhưng sau đó được mở rộng để chạy trên các thiết bị của Apple như iPod Touch, iPad và Apple TV. Tính cho đến hết năm 2012, số lượng ứng dụng iOS trên Apple App Store đã hơn 775.000 ứng dụng. [4],[5]

Giao diện người dùng của iOS dựa trên sự thao tác bằng tay, người dùng có thể thao tác với hệ điều hành này thông qua màn hình cảm ứng của các thiết bị Apple với rất nhiều động tác bằng tay.

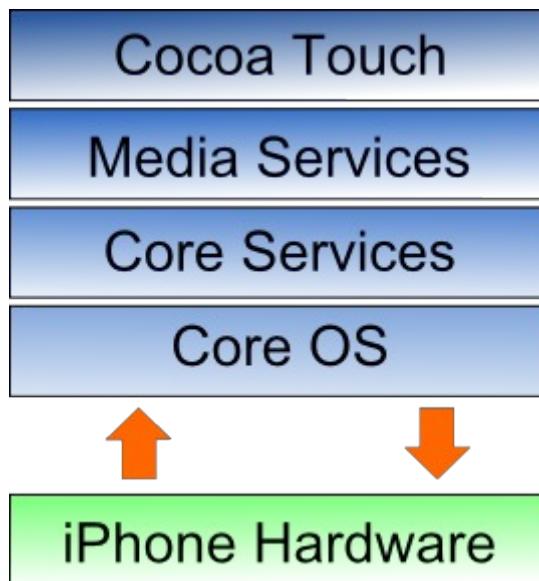
Phiên bản mới nhất của iOS hiện nay là 7.0 được phát hành ngày 18/9/2013 dành cho các thiết bị kể từ iPhone 4, iPod 5, iPad 2 trở lên. Giao diện này được xem là một sự thay đổi lớn khi giao diện của hệ điều hành được thiết kế theo phong cách phẳng. [5]



Hình 1.1: Giao diện iOS 7

1.1.2 Kiến trúc của hệ điều hành iOS

Kiến trúc của hệ điều hành iOS gồm có bốn lớp sau [10]:



Hình 1.2: Kiến trúc hệ điều hành iOS

Lớp **Media** cung cấp cho hệ điều hành iOS các dịch vụ đa phương tiện như âm thanh, video, hình ảnh, đồ họa, bao gồm một số thành phần sau:

- Core Graphics Framework
- Quartz Core Framework
- ES Framework
- Audio
- AV Foundation Framework
- Core Audio Framework
- Open Audio Library
- Media Player Framework

Lớp **Core Services** cung cấp một số dịch vụ cơ bản cho hệ điều hành và các lớp khác, bao gồm:

- Address book Framework
- Core Data Framework
- Core Foundation Framework
- Foundation Framework
- Core Location Framework

- Store Kit Framework
- SQLite Library

Lớp **Core OS** cung cấp một số dịch vụ hệ thống như memory manager, networking, file system, ... bao gồm:

- CFNetwork Framework
- External Accessory Framework
- Security Framework
- System

Lớp **Cocoa Touch** cung cấp các framework chung cho các lập trình viên phát triển ứng dụng, bao gồm:

- UIKit Framework
- Map Kit Framework
- Push Notification Service
- Message UI Framework
- Address Book UI Framework
- Game Kit Framework

1.2 Giới thiệu ngôn ngữ lập trình Objective-C

Ngôn ngữ Objective-C được tạo ra bởi Brad Cox và Tom Love vào năm 1980 tại công ty Stepstone. Từ năm 1988, công ty NeXT Software nắm giữ bản quyền của ngôn ngữ Objective-C. Họ đã phát triển các bộ thư viện và cả môi trường phát triển cho nó có tên là NEXTSTEP. [7]

Đến cuối tháng 12 năm 1996, hãng Apple đã mua lại công ty NeXT Software, môi trường NEXTSTEP/OPENSTEP đã trở thành phần cốt lõi của hệ điều hành OS X mà Apple giới thiệu sau này. Phiên bản chính thức của môi trường phát triển này do Apple giới thiệu ban đầu có tên là Cocoa.

Bằng việc hỗ trợ sẵn ngôn ngữ Objective-C, đồng thời tích hợp một số công cụ phát triển khác như Project Builder (đây là tiền thân của Xcode) và Interface Builder, Apple đã tạo ra một môi trường mạnh mẽ để phát triển ứng dụng trên Mac OS X. Đến

năm 2007, Apple tung ra bản nâng cấp cho ngôn ngữ Objective-C và gọi đó là Objective-C 2.0. [11]

Ngôn ngữ lập trình Objective-C dựa trên nền tảng ngôn ngữ C nhưng bổ sung thêm hỗ trợ lập trình hướng đối tượng. Objective-C là ngôn ngữ lập trình sử dụng để viết ứng dụng cho Apple's iOS và hệ điều hành Mac OS. [3]

1.3 Tiêu kết

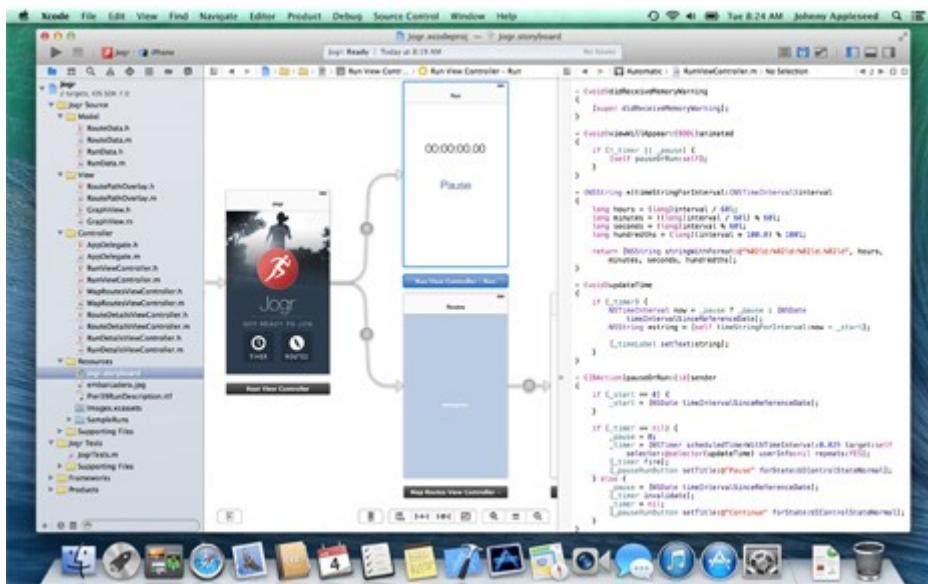
Chương này giới thiệu tổng quan về hệ điều hành iOS và kiến trúc của hệ điều hành này. Bên cạnh đó, nội dung chương này còn giới thiệu sơ lược về ngôn ngữ lập trình Objective-C - ngôn ngữ cơ bản trong lập trình ứng dụng iPhone.

CHƯƠNG 2: GIỚI THIỆU VỀ BỘ CÔNG CỤ XCODE VÀ IOS SIMULATOR

2.1 Tìm hiểu bộ công cụ Xcode

2.1.1 Giới thiệu về Xcode

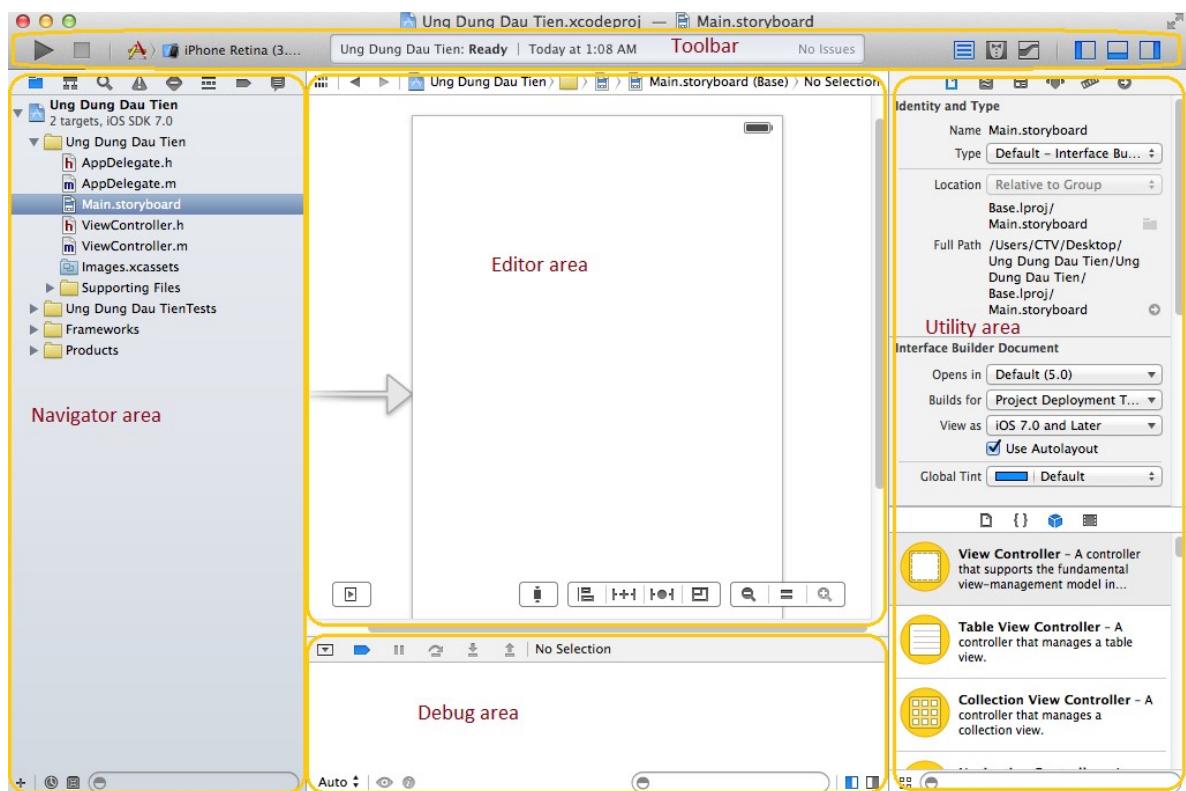
Phần mềm Xcode là một IDE tích hợp các công cụ phát triển phần mềm do Apple phát triển và cung cấp cho các lập trình viên để phát triển các ứng dụng cho các thiết bị chạy hệ điều hành của Apple (Mac OS, iOS). Xuất hiện lần đầu tiên năm 2003, phiên bản mới nhất hiện nay của Xcode là bản Xcode 5.0 và được miễn phí trên Mac App Store cho người dùng Mac OS X Lion và OS X Mountain Lion. Trình biên dịch của Xcode gọi là Apple LLVM Compiler. Trình biên dịch này hỗ trợ các ngôn ngữ lập trình như C, C++, Objective-C, Objective-C++, Java, AppleScript, Python và Ruby ... [13]



Hình 2.1: Phần mềm Xcode

2.1.2 Một số đặc điểm của Xcode

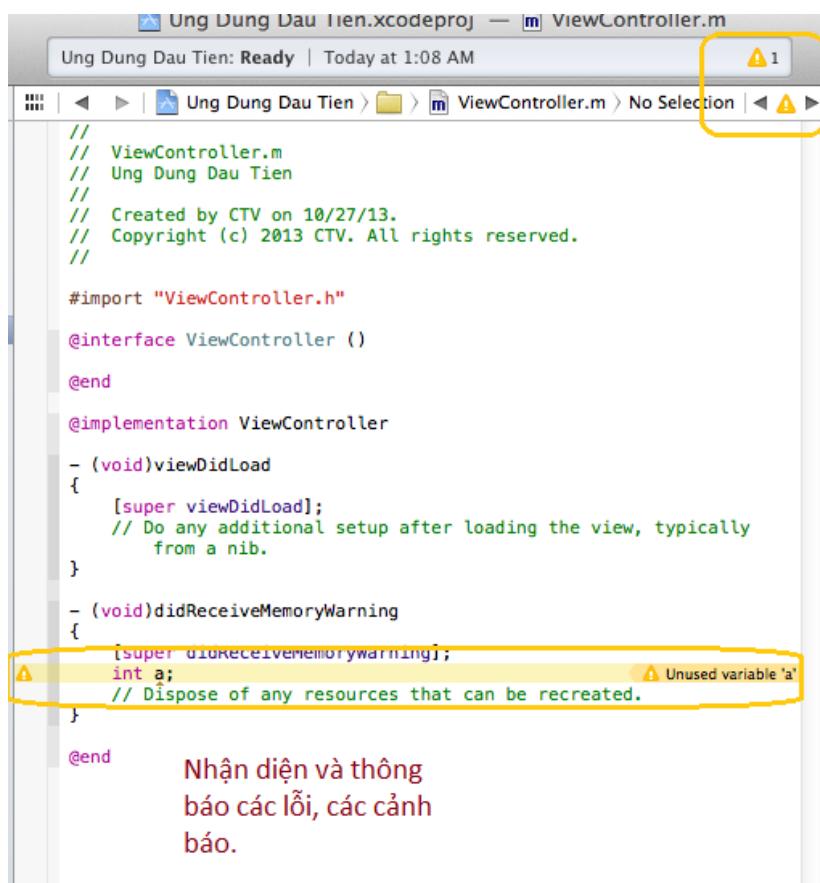
Giao diện làm việc của Xcode gồm có 5 phần chính : **Toolbar**, **Editor area**, **Navigator area**, **Debug area**, **Utility area**.



Hình 2.2: Giao diện Xcode

- **Debug area**: đây là vùng hỗ trợ người dùng trong quá trình debug lỗi của chương trình.
- **Toolbar area**: vùng chứa các công cụ tiện ích giúp người dùng có thể đơn giản trong việc chạy, debug ứng dụng, lựa chọn iOS Simulator, đóng mở các vùng khác, ...
- **Editor area**: vùng để thiết kế giao diện, viết và chỉnh sửa code của chương trình.
- **Utility area**: vùng này cho phép người dùng tùy chỉnh các tham số, giá trị của các đối tượng trên giao diện, cũng như cho phép kéo thả và sử dụng các đối tượng có sẵn của Xcode như Button, Label, Slider, ... hay các đoạn code mẫu (If, Switch, ...) vào trong project.
- **Navigator area**: cung cấp một cách nhìn trực quan, tiện lợi cho việc quản lý ứng dụng, xem thông báo lỗi, tìm kiếm một đoạn code trong chương trình hay kiểm tra mức độ hoạt động của RAM, CPU khi chạy ứng dụng, ...

Xcode cũng cung cấp một chế độ gõ lỗi thông minh hỗ trợ người dùng trong việc phát hiện lỗi, cảnh báo lỗi. Xcode còn có tính năng sửa chữa các lỗi tự động. Trong nhiều trường hợp Xcode sẽ không chỉ báo cáo một lỗi nó sẽ trình bày một giải pháp tốt để khắc phục lỗi bằng cách nhấp chuột vào lỗi để sửa chữa. Ví dụ như thêm một dấu chấm phẩy còn thiếu ... Một phím tắt ngay lập tức sẽ sửa chữa lỗi và cho phép người dùng tiếp tục viết ứng dụng.



The screenshot shows the Xcode interface with the code editor displaying `ViewController.m`. The status bar at the top right shows '1' in a yellow box, indicating one warning. The code editor shows the following code:

```

// ViewController.m
// Ung Dung Dau Tien
//
// Created by CTV on 10/27/13.
// Copyright (c) 2013 CTV. All rights reserved.
//

#import "ViewController.h"

@interface ViewController : UIViewController

@end

@implementation ViewController

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

- (void)didReceiveMemoryWarning
{
    [super didReceiveMemoryWarning];
    int a; // Unused variable 'a'
    // Dispose of any resources that can be recreated.
}

@end

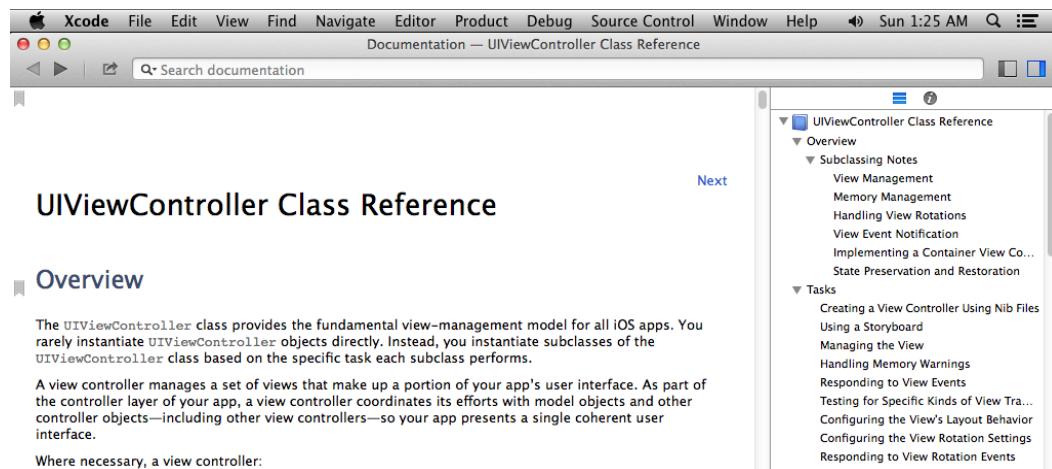
```

A red box highlights the line `int a;` in the `- (void)didReceiveMemoryWarning` block, with a tooltip 'Unused variable 'a'' appearing above it. A red box also highlights the status bar area showing '1'.

Nhận diện và thông báo các lỗi, các cảnh báo.

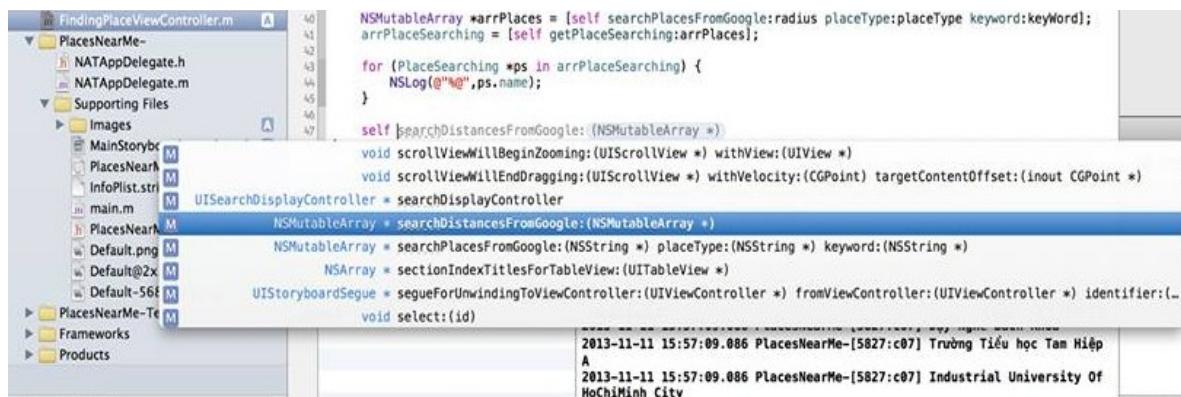
Hình 2.3: Xcode nhận diện và cảnh báo

Hơn thế nữa, kèm theo Xcode là một bộ tài liệu hướng dẫn từng bước, chi tiết và tiện lợi nhằm hỗ trợ người dùng trong việc lập trình. Trong quá trình viết ứng dụng, nếu bạn muốn tìm hiểu thêm một đối tượng, bạn có thể sử dụng tới bộ tài liệu này để có được hướng dẫn, ví dụ minh họa dễ hiểu.



Hình 2.4: Bộ tài liệu hướng dẫn kèm theo Xcode

IDE trong Xcode cung cấp nhiều công cụ và tính năng giúp người dùng dễ dàng lập trình hơn nhiều. Một trong những tính năng là mã Sense hiển thị một danh sách popup, hiển thị các lớp và các thành viên sẵn có, chẳng hạn như phương thức, thuộc tính ... [1]



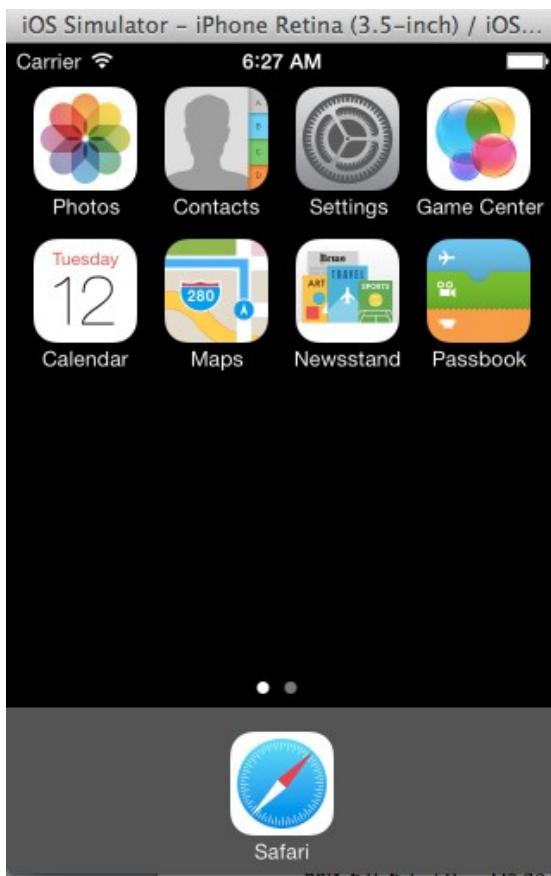
Hình 2.5: Danh sách popup hiện ra

Xcode hoạt động trên một cửa sổ đơn. Trong Xcode các cửa sổ sử dụng để thực hiện các nhiệm vụ như thiết kế giao diện hay viết code đều diễn ra trên một cửa sổ duy nhất.

2.2 Tìm hiểu iOS Simulator

2.2.1 Giới thiệu về iOS Simulator

iOS Simulator là một phần trong bộ công cụ kèm theo của phần mềm Xcode. iOS Simulator chứa iOS SDK cho phép bạn chạy trên Mac OS để giả lập môi trường iPhone, iPad nhằm phục vụ cho việc kiểm tra và tìm lỗi của ứng dụng được viết ra trước khi kiểm thử ứng dụng trên thiết bị thật.



Hình 2.6: iOS Simulator 7.0

iOS Simualtor cho phép cài đặt mô phỏng nhiều thiết bị iOS khác nhau như iPhone, iPhone Rentina, iPad, iPad Rentina ... với nhiều phiên bản iOS SDK khác nhau như 6.0, 6.1, 7.0 ... Do đó người lập trình có thể dễ dàng xây dựng và kiểm thử ứng dụng của mình dành cho phiên bản iOS mới (iOS 7) cũng như trên phiên bản iOS cũ (iOS 6.1, iOS 6.0, iOS 5).

Với iOS Simulator, người dùng có thể kiểm thử ứng dụng của mình về thiết kế giao diện, về tính năng của ứng dụng, từ đó có thể khắc phục các lỗi phát sinh, tối ưu hóa ứng dụng trước khi bạn đem ứng dụng lên thiết bị thật. iOS Simulator có thể chạy chung với phần mềm Xcode hoặc chạy độc lập đều được. Người dùng có thể tương tác với iOS Simulator thông qua bàn phím, chuột để nhập dữ liệu cũng như điều khiển các sự kiện của người dùng.

Mặc định sau khi cài Xcode 5, iOS Simulator kèm theo đã được cài đặt để hỗ trợ các thiết bị iPhone Rentina, iPad Rentina và iOS 7.0. Tuy nhiên nếu người dùng muốn iOS Simulator chạy các thiết bị iPhone, iPad thông thường và các phiên bản iOS thấp hơn như iOS 6.0, iOS 6.1 thì cần phải tải và cài đặt thêm iOS SDK cho các phiên

bản của iOS đó. Để tải thêm, người dùng vào **Xcode → Preferences → Download**. Sau đó lựa chọn phiên bản iOS cần cài đặt thêm và tải về.



Hình 2.7: *Bổ sung thêm các iOS Simulator và tài liệu trong Downloads*

2.2.2 Một số điểm hạn chế của iOS Simulator

Mặc dù iOS Simulator rất hữu ích trong việc kiểm thử ứng dụng trước khi đưa lên thiết bị thật, tuy nhiên bản thân iOS Simulator vẫn còn một số hạn chế nhất định. Đối với phần cứng, iOS Simulator vẫn còn khiếm khuyết ở một số điểm như không có camera, không có microphone, ... Ngoài ra còn một số framework không được hỗ trợ như Media player, Messenger UI ... [6]

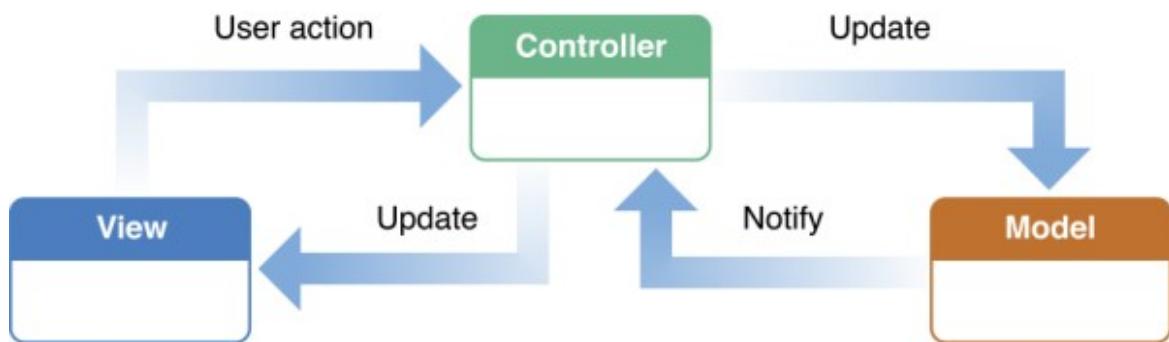
2.3 Tiêu kết

Chương này giới thiệu tổng quan về giao diện và tính năng của công cụ Xcode và iOS Simulator. Bên cạnh đó cũng trình bày một số thao tác trên Xcode và iOS Simulator.

CHƯƠNG 3: THAO TÁC XÂY DỰNG ỦNG DỤNG VÀ ĐUA ỦNG DỤNG LÊN IPHONE

3.1 Giới thiệu về mô hình MVC

Model – View – Controller (MVC) là một chuẩn mực thiết kế ứng dụng trong quá trình phát triển ứng dụng iOS mà theo đó các đối tượng tạo ra sẽ gắn cho một trong ba nhóm vai trò sau: model, view, controller. MVC không chỉ định nghĩa vai trò đối tượng đảm nhiệm mà còn định nghĩa cách giao tiếp giữa các đối tượng với nhau. Tập hợp các đối tượng của nhóm trong mô hình MVC được biết đến như một layer.



Hình 3.1: Mô hình MVC

Đối tượng View là một đối tượng trong ứng dụng mà người dùng có thể nhìn thấy được. Mục đích chính của đối tượng View là hiển thị dữ liệu từ đối tượng Model và cho phép thay đổi dữ liệu đó. Trong iOS, UIKit Framework cung cấp một tập hợp các lớp View còn trong Mac OS, AppKit Framework cũng cung cấp một tập hợp tương tự.

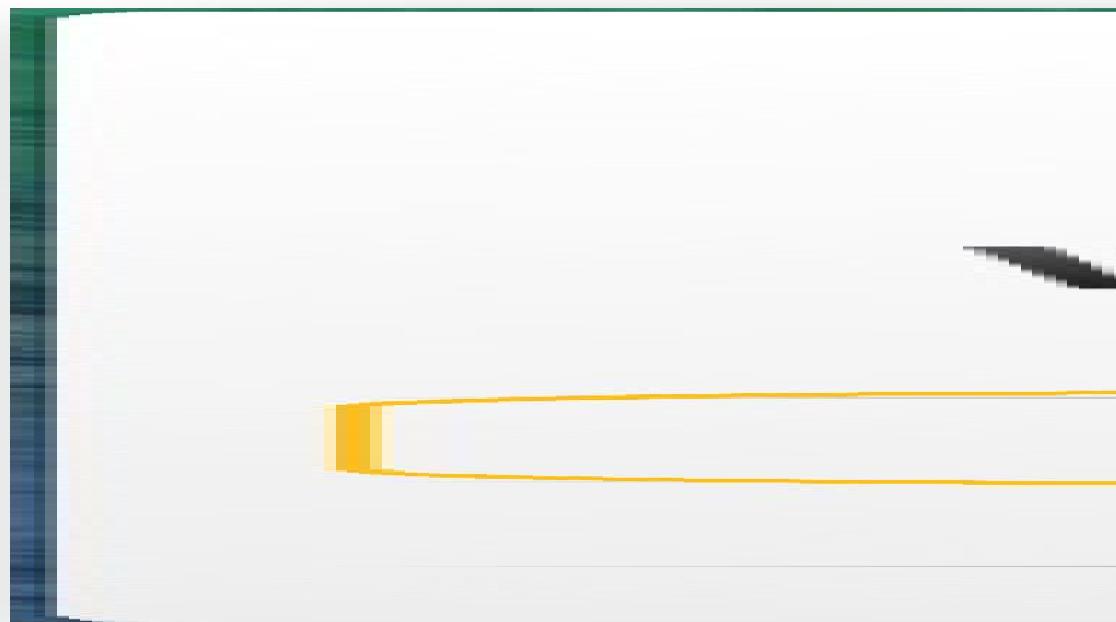
Đối tượng Controller là đối tượng đóng vai trò trung gian giữa một hoặc nhiều đối tượng View với một hoặc nhiều đối tượng Model. Đối tượng Controller sẽ phiên dịch hành động của người dùng được thực hiện trên đối tượng View, giao tiếp dữ liệu mới hoặc thay đổi tới các lớp Model. Khi một đối tượng Model thay đổi, một đối tượng Controller thông báo có dữ liệu mới cho các đối tượng View để các đối tượng này hiển thị dữ liệu đó.

Đối tượng Model đóng gói dữ liệu của ứng dụng, định nghĩa logic hay các thao tác tính toán với dữ liệu đó. Hành động của người dùng trong lớp View mà tạo hoặc thay đổi dữ liệu sẽ kết nối thông qua đối tượng Controller và kết quả là tạo hoặc thay

đổi dữ liệu của đối tượng Model. Khi đổi tượng Model thay đổi, nó sẽ thông báo cho đối tượng Controller và những thay đổi này sẽ hiển thị trong đối tượng View.

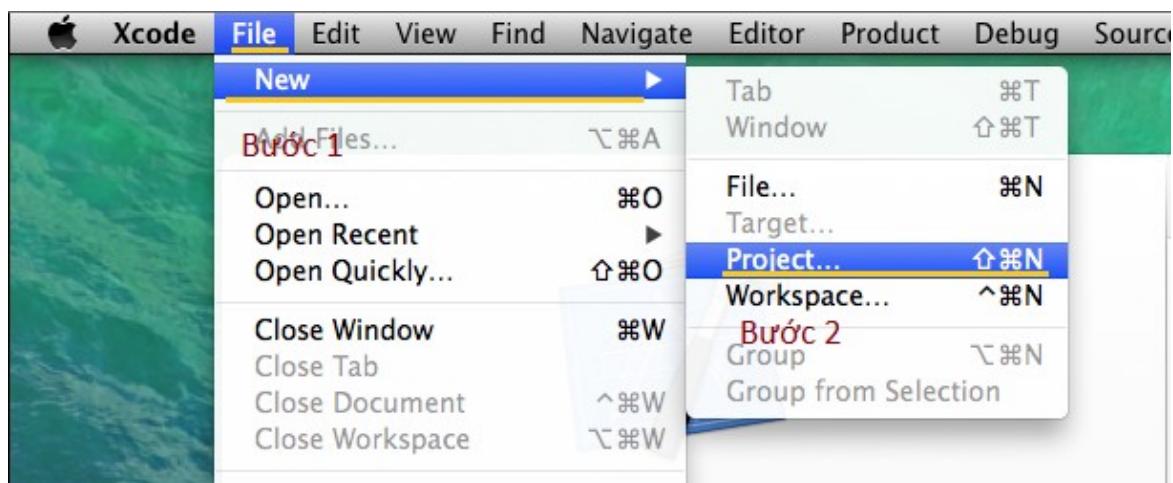
3.2 Thao tác xây dựng ứng dụng đơn giản

Bước 1: Khởi động phần mềm Xcode lên.



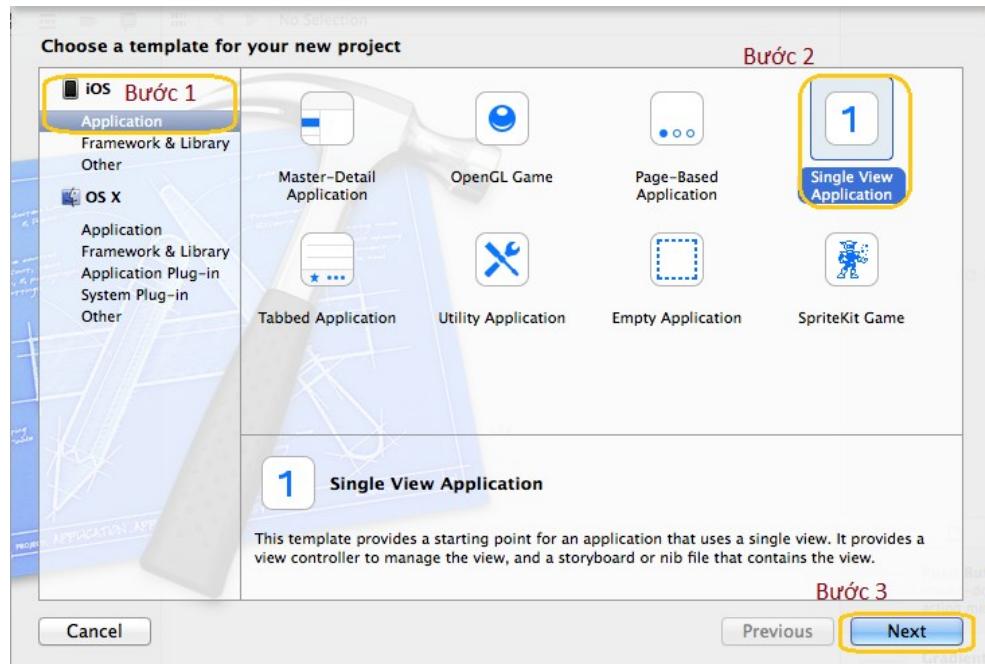
Hình 3.2: Giao diện Xcode khi mở lên

Bước 2: Tại giao diện Xcode, bạn có thể tạo mới một project bằng cách chọn Create new project. Ngoài ra bạn có thể tạo project mới bằng cách chọn **File > New > Project**.



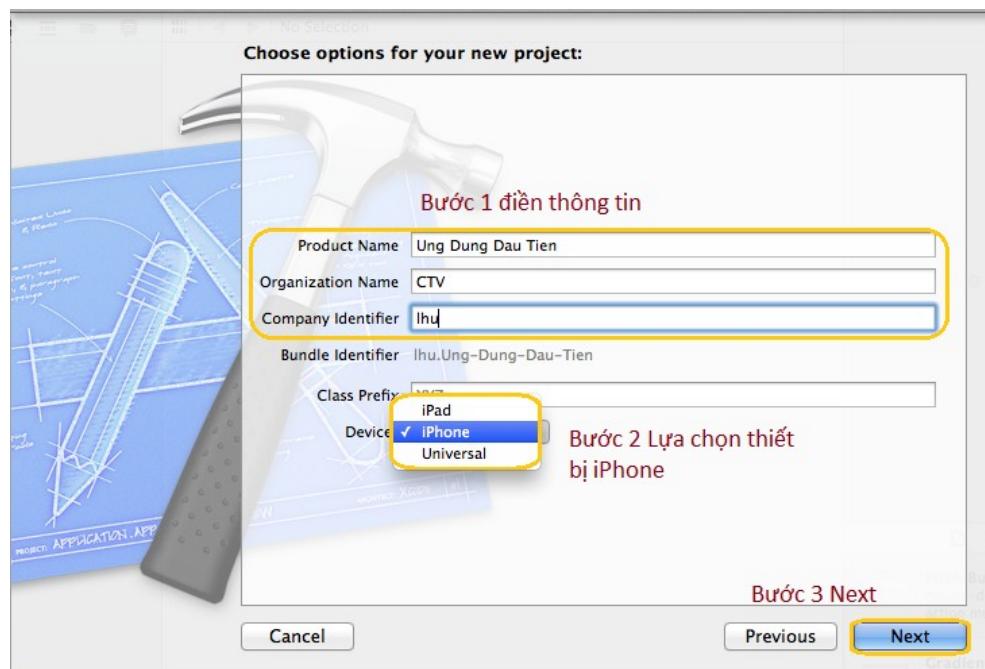
Hình 3.3: Tạo project bằng menu

Bước 3: Sau khi chọn New Project, Xcode sẽ yêu cầu bạn lựa chọn một hình thức cho Project này (ứng dụng cho iPhone hay Mac OS, Single View hay Empty View, ...). Cách đơn giản nhất là bạn chọn **Single View**.



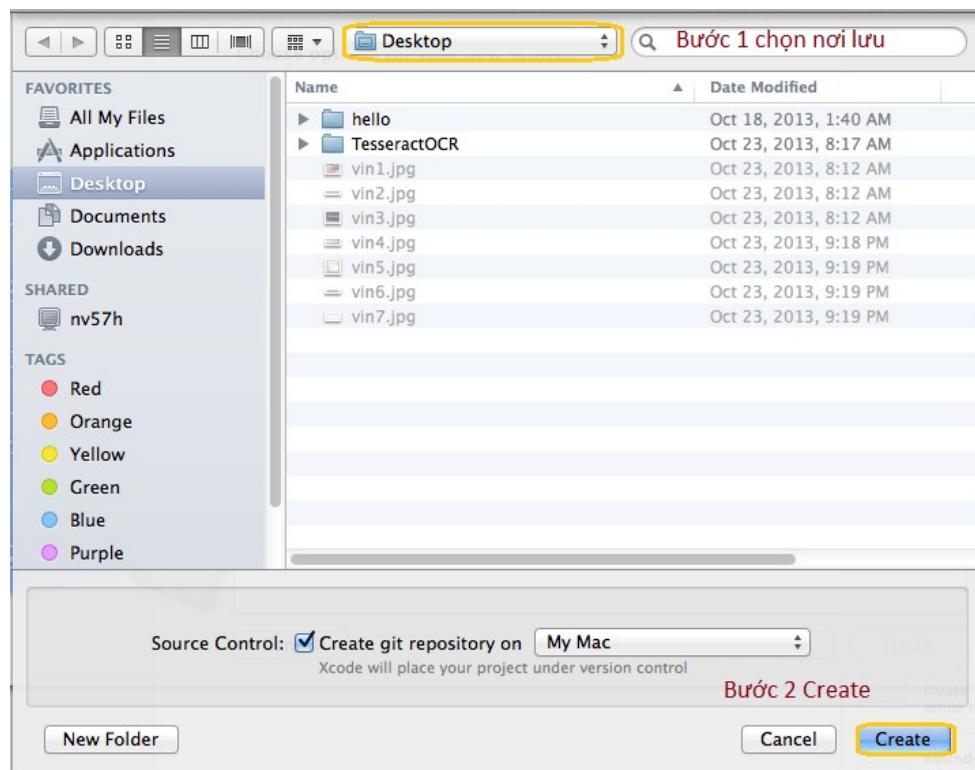
Hình 3.4: Chọn lựa mẫu cho project

Bước 4: Tiếp theo đó bạn điền thêm một vài thuộc tính của Project như **Product Name**, **Organization Name**, **Company Identifier**. Sau đó bạn tiến hành lựa chọn Devices cho Project (iPhone, iPad hay Universal để viết ứng dụng cho cả hai).

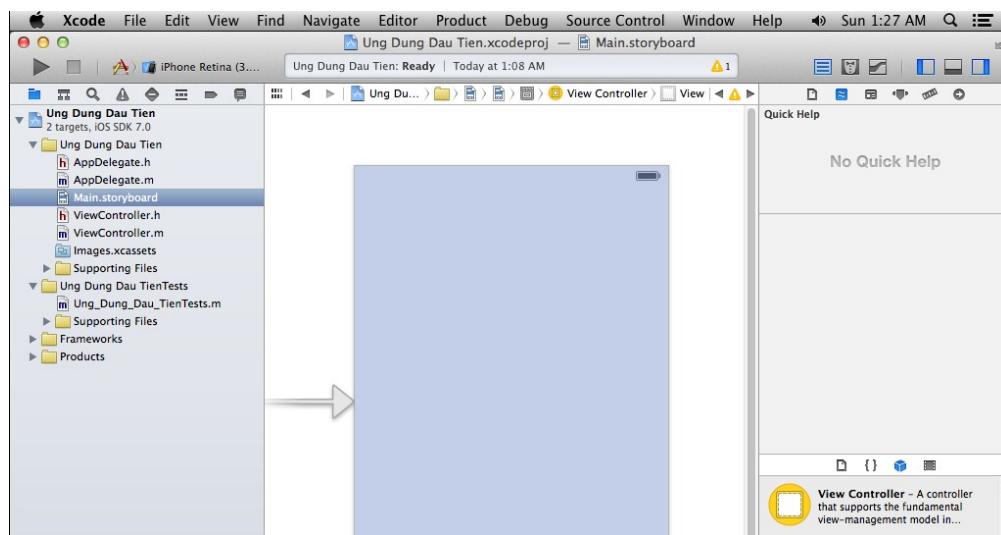


Hình 3.5: Điền thông tin cho project

Bước 5: Tiếp theo bạn chọn nơi lưu trữ Project trên máy tính để lưu Project và chọn **Create**.

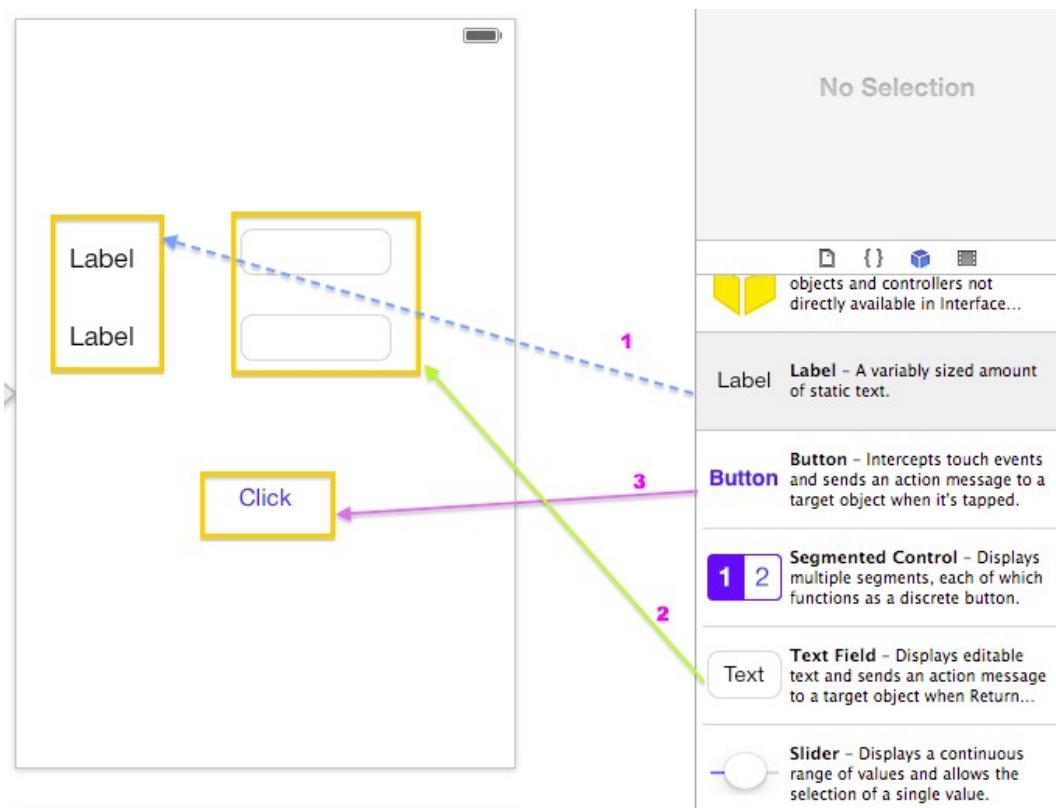


Hình 3.6: Chọn nơi lưu trữ project



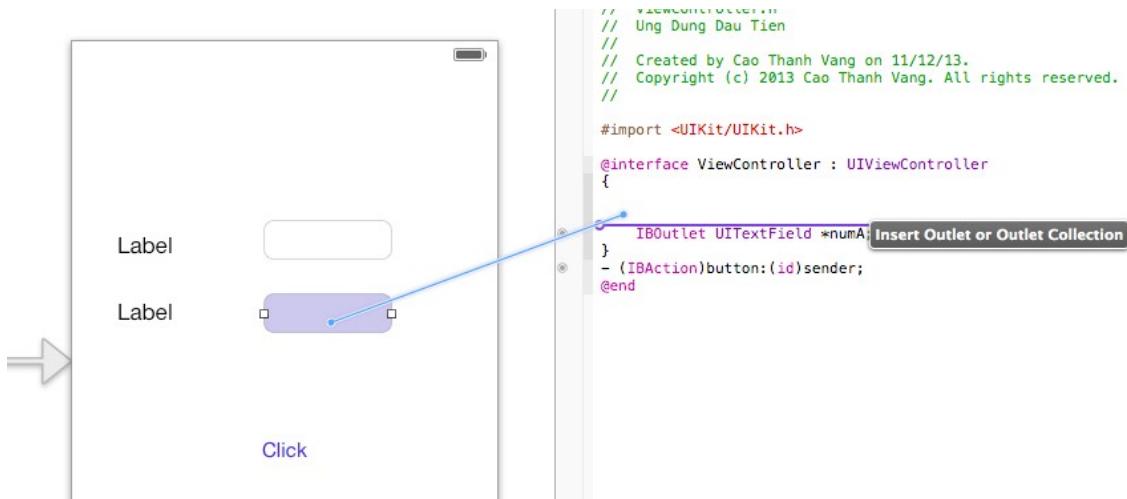
Hình 3.7: Giao diện Project mới tạo

Bước 6: Kéo thả đối tượng Button, Label và Text Field vào giao diện và thiết kế giao diện như hình 18. Đặt title cho button là Click.



Hình 3.8: Kéo thả đối tượng vào giao diện

Bước 7: Ánh xạ các đối tượng vào tập tin .h để khai báo với controller.



Hình 3.9: Giữ Ctrl và kéo thả đối tượng vào tập tin .h

```

// ViewController.h
// Ung Dung Dau Tien
//
// Created by Cao Thanh Vang on 11/12/13.
// Copyright (c) 2013 Cao Thanh Vang. All rights reserved.
//

#import <UIKit/UIKit.h>

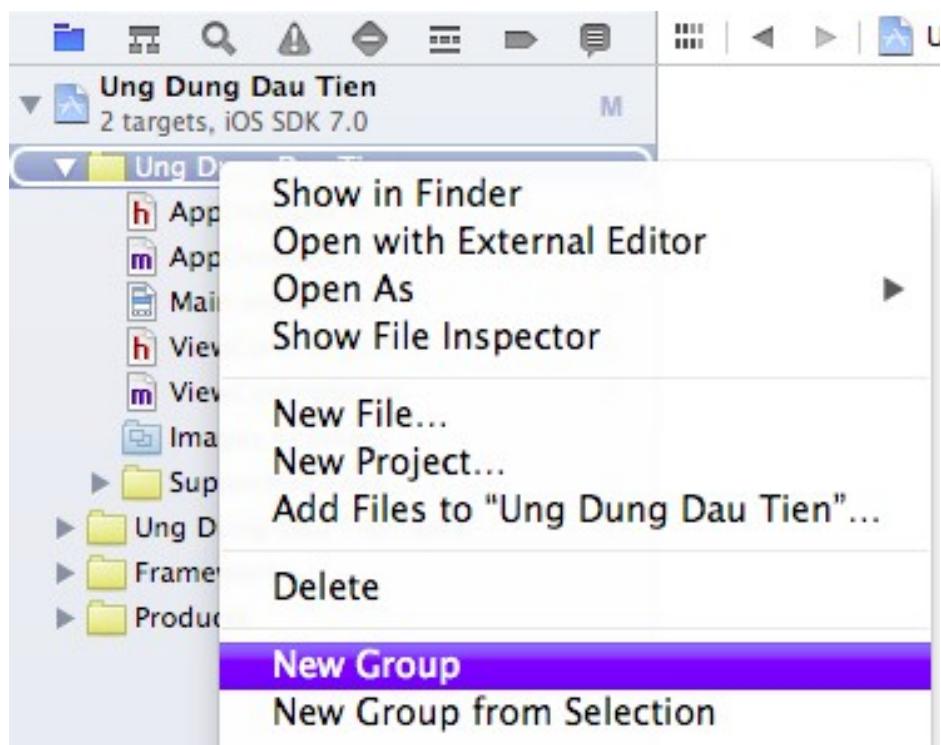
@interface ViewController : UIViewController
{

    IBOutlet UITextField *numB;
    IBOutlet UITextField *numA;
}
- (IBAction)button:(id)sender;
@end

```

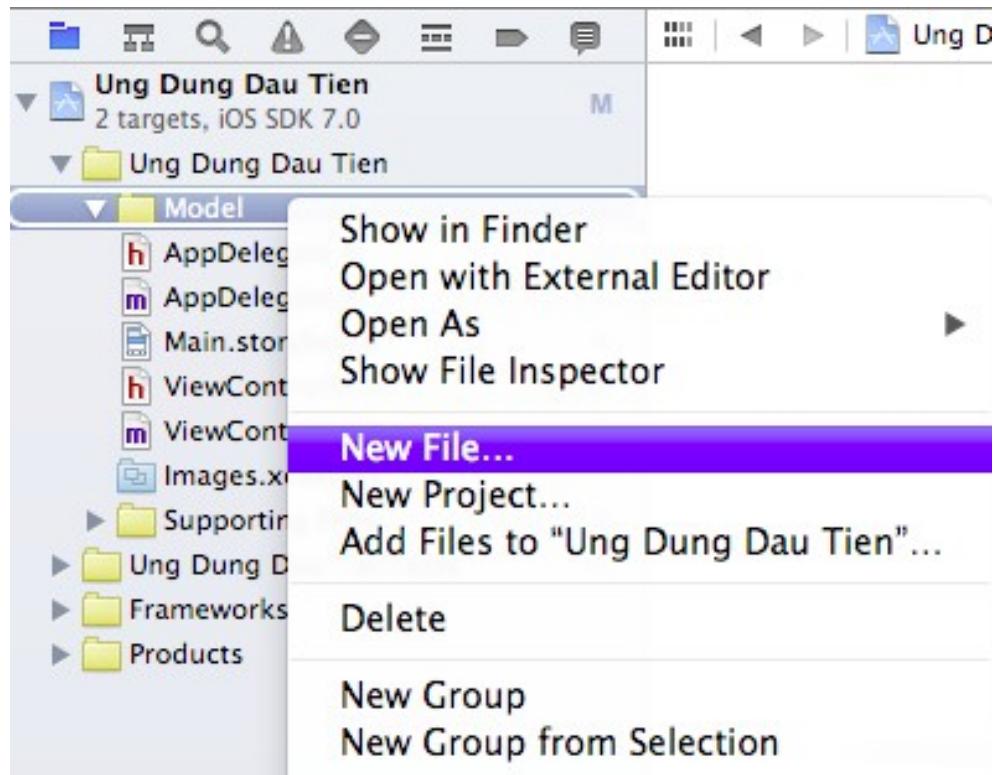
Hình 3.10: Hoàn tất ánh xạ các đối tượng

Bước 8: Thêm một group mới tên là Model.



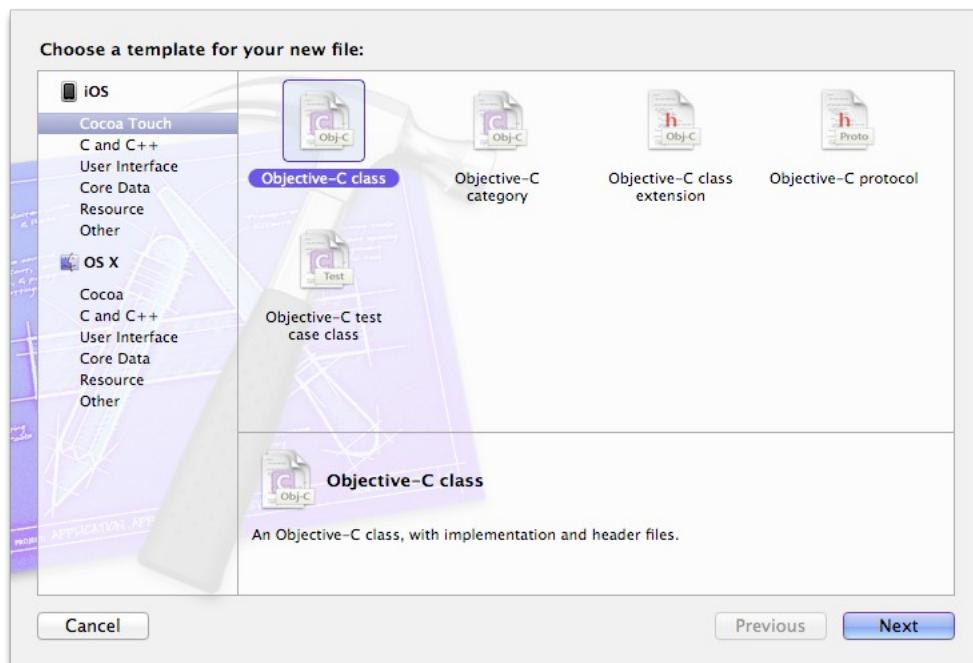
Hình 3.11: New Group

Bước 9: Thêm mới File vào project.



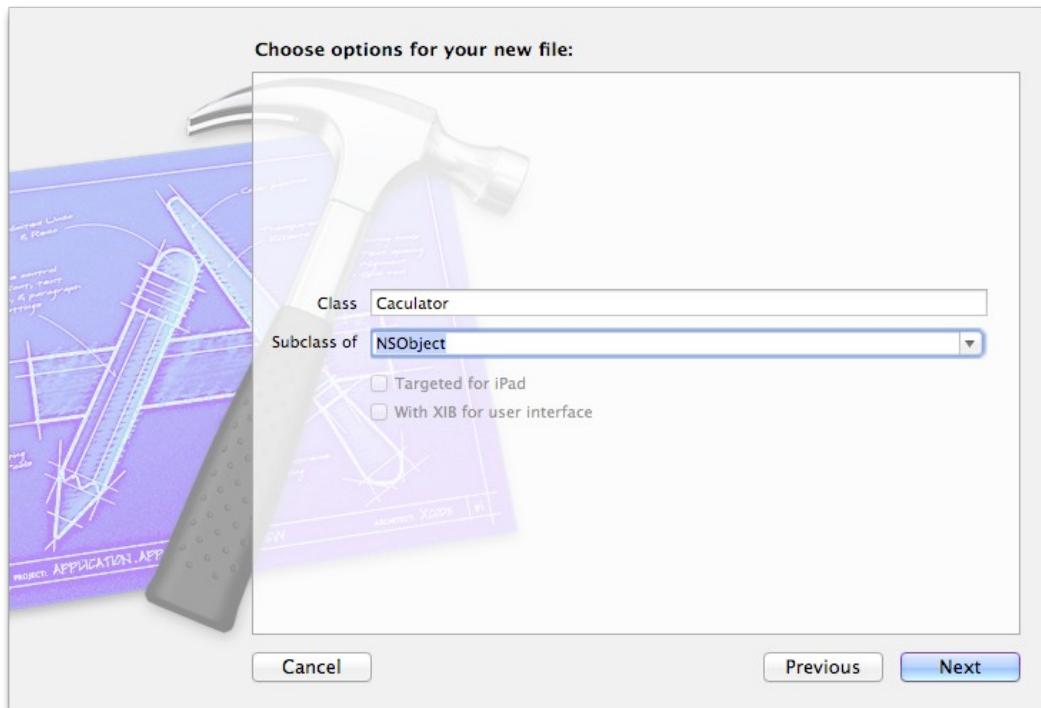
Hình 3.12: New File

Bước 10: Chọn loại File thêm mới vào là Objective-C Class.



Hình 3.13: Chọn Objective-C Class

Bước 11: Đặt tên class mới là Caculator và chọn subclass là **NSObject** → Next → Create.



Hình 3.14: Đặt tên class mới

Bước 12: Khai báo biến và hàm “Cong” trong tập tin .h

```

//  

//  Caculator.h  

//  Ung Dung Dau Tien  

//  

//  Created by Cao Thanh Vang on 11/12/13.  

//  Copyright (c) 2013 Cao Thanh Vang. All rights reserved.  

//  

#import <Foundation/Foundation.h>  

@interface Caculator : NSObject  

@property(nonatomic) int numberA;  

@property(nonatomic) int numberB;  

-(int)Cong;  

@end

```

Hình 3.15: Khai báo biến và hàm

Bước 13: Viết code cho hàm “Cong”.

```

// Caculator.m
// Ung Dung Dau Tien
//
// Created by Cao Thanh Vang on 11/12/13.
// Copyright (c) 2013 Cao Thanh Vang. All rights reserved.
//

#import "Caculator.h"

@implementation Caculator

-(int)Cong
{
    return self.numberA + self.numberB;
}

@end

```

Hình 3.16: Viết code cho hàm “Cong”

Bước 14: Trong tập tin ViewController.m, viết code cho button.

```

// ViewController.m
// Ung Dung Dau Tien
//
// Created by Cao Thanh Vang on 11/12/13.
// Copyright (c) 2013 Cao Thanh Vang. All rights reserved.
//

#import "ViewController.h"
#import "Caculator.h"

@interface ViewController : UIViewController

@end

@implementation ViewController

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

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

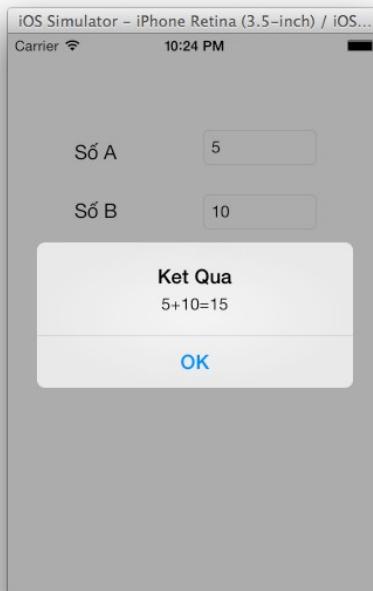
- (IBAction)button:(id)sender {
    Caculator *calc = [[Caculator alloc] init];
    calc.numberA = [numA.text intValue];
    calc.numberB = [numB.text intValue];
    NSString *kq = [NSString stringWithFormat:@"%@+%@", numA.text, numB.text, calc.Cong];

    UIAlertView *notice = [[UIAlertView alloc] initWithTitle:@"Kết Quả" message:kq delegate:self cancelButtonTitle:@"OK" otherButtonTitles:nil, nil];
    [notice show];
}
@end

```

Hình 3.17: Viết code cho button

Bước 15: Chạy ứng dụng và kiểm tra.



Hình 3.18: Kết quả

3.3 Đưa ứng dụng lên iPhone

3.3.1 Giới thiệu

Để build ứng dụng lên iPhone (máy thật), Apple yêu cầu lập trình viên phải có tài khoản Apple Developer và chi phí cho 1 tài khoản như vậy là 99\$/năm, nếu chúng ta chưa có đủ điều kiện để sắm 1 tài khoản 99\$ thì có cách nào khác để build ứng dụng lên iPhone hay không. JailCoder giúp chúng ta làm được điều này mà không phải mất tài khoản 99\$ và đương nhiên đây là một cách không chính thống do đó vẫn còn nhiều nhược điểm khi sử dụng JailCoder.

3.3.2 Chuẩn bị

Bước 1: Download công cụ JailCoder tại

<http://oneiros.altervista.org/jailcoder/>

JailCoder được cài đặt trên các hệ điều hành OSX như Mac OSX Leopard, Snow Leopard, Lion, Lion Mountain, ...

Bước 2: Điều kiện như sau:

- Sử dụng Xcode 3 đến Xcode 4

- Đã cài đặt AppSync, vào Cydia để tìm và cài đặt phần mềm AppSync. Lưu ý nên cài đặt AppSync tương đương với phiên bản iOS chúng ta đang sử dụng. VD: iPhone đang dùng iOS 6.x.x thì sẽ cài “AppSync for iOS6”.



Hình 3.19: Phần mềm AppSync

- iPhone của chúng ta phải là iPhone đã được Jailbreak, những iPhone đã Jailbreak hầu hết đều có icon Cydia trên giao diện home của iPhone.



Hình 3.20: Cydia trên iPhone

- Thoát hoàn toàn Xcode và các iOS Simulator đang chạy trước khi tiến hành JailCoder. Phiên bản iOS đang sử dụng trên iPhone phải lớn hơn hoặc bằng phiên bản iOS mà ứng dụng chúng ta đang sử dụng. VD: iPhone đang sử dụng iOS 6, muốn build ứng dụng nào đó lên iPhone thì ứng dụng đó phải sử dụng iOS SDK 6 trở xuống.

3.3.3 Tiến hành

Bước 1: mở JailCoder → Click **Guided Path**



Hình 3.21: Chọn Guided Patch

→ Click “Got it”

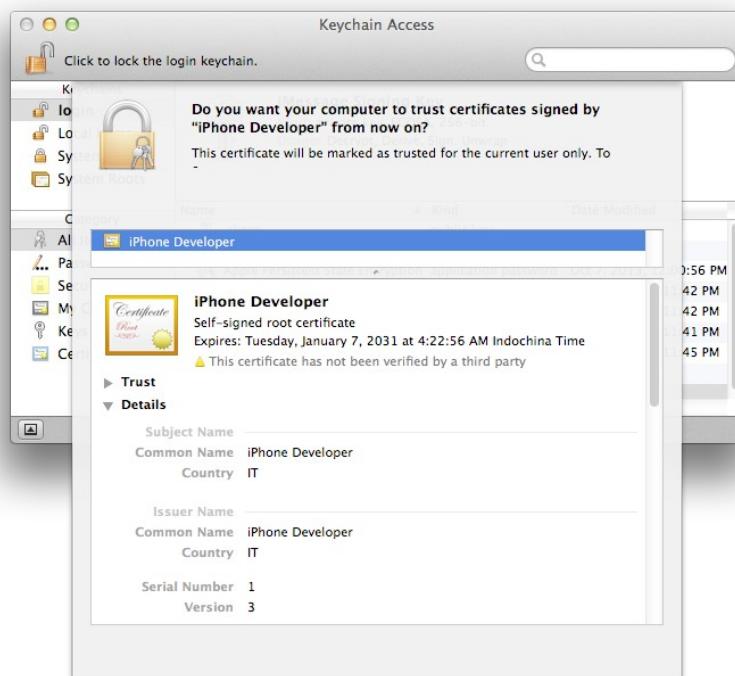


Hình 3.22: Chọn Got it

→ Click “Certificate Root” → xuất hiện cửa sổ “Keychain Access”



Hình 3.23: Chọn Certificate Root



Hình 3.24: Cửa sổ Keychain Access

→ Click “Always Trust” để trust Certificate “iPhone Developer”, cửa sổ yêu cầu nhập password xuất hiện, nhập password hiện tại của máy Mac → chọn “Update Setting”.



Hình 3.25: Cửa sổ yêu cầu nhập mật khẩu

→ Quay lại cửa sổ “JailCoder” click “**Certificate Private**”, tương tự nhập password (3 lần), sẽ hiển thị thông báo lỗi nhưng không sao → click **Ok**.



Hình 3.26: Chọn Certificate Private



Hình 3.27: Cửa sổ yêu cầu nhập mật khẩu



Hình 3.28: Thông báo lỗi xuất hiện

→ Nếu thành công chúng ta sẽ thấy Certificate “iPhone Developer” đã được thêm vào Keychain Access như sau:



Hình 3.29: iPhone Developer Keychain

→ Quay lại cửa sổ “JailCoder” → click “Next”



Hình 3.30: Chọn next



Hình 3.31: Chọn Patch my Xcode

→ Click “Patch my Xcode” xuất hiện cửa sổ yêu cầu nhập password



Hình 3.32: Cửa sổ yêu cầu nhập mật khẩu

→ Nếu thành công sẽ xuất hiện giao diện như sau:



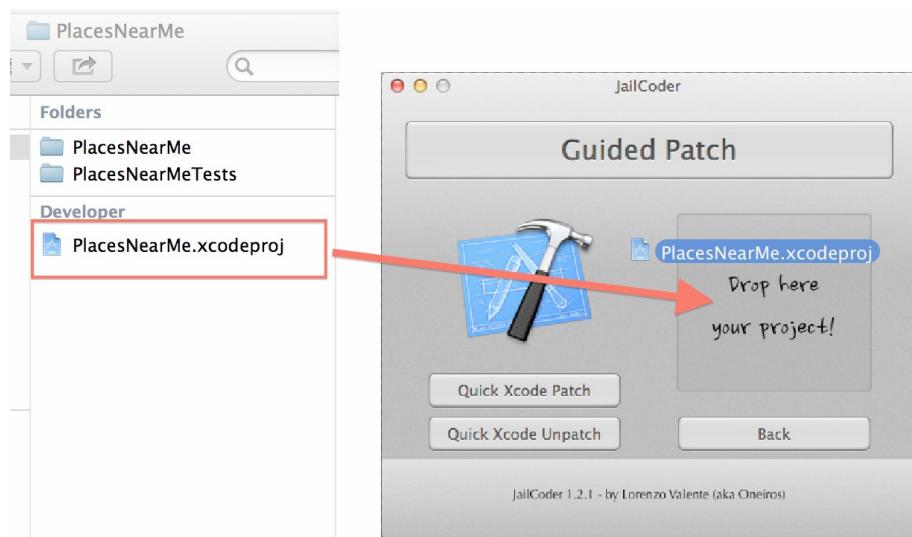
Hình 3.33: Hoàn tất Patch Xcode

→ Click “**Back to Main Menu**” để patch project mà chúng ta cần build lên iPhone.



Hình 3.34: Chọn Patch my Project

→ Click “**Patch My Project**” sau đó kéo thả tập tin “*.xcodeproj” vào khung “**Drop here your project!**”



Hình 3.35: Kéo thả Project vào khung Patch

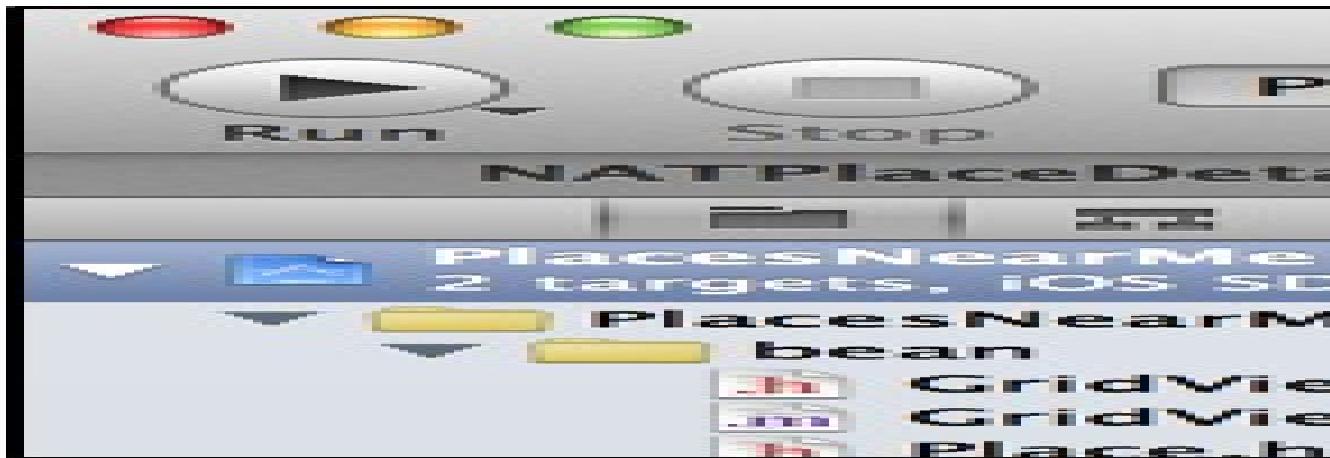
→ Nếu thành công chúng ta sẽ thấy JailCoder thông báo “**Pathed!**”



Hình 3.36: Patch thành công

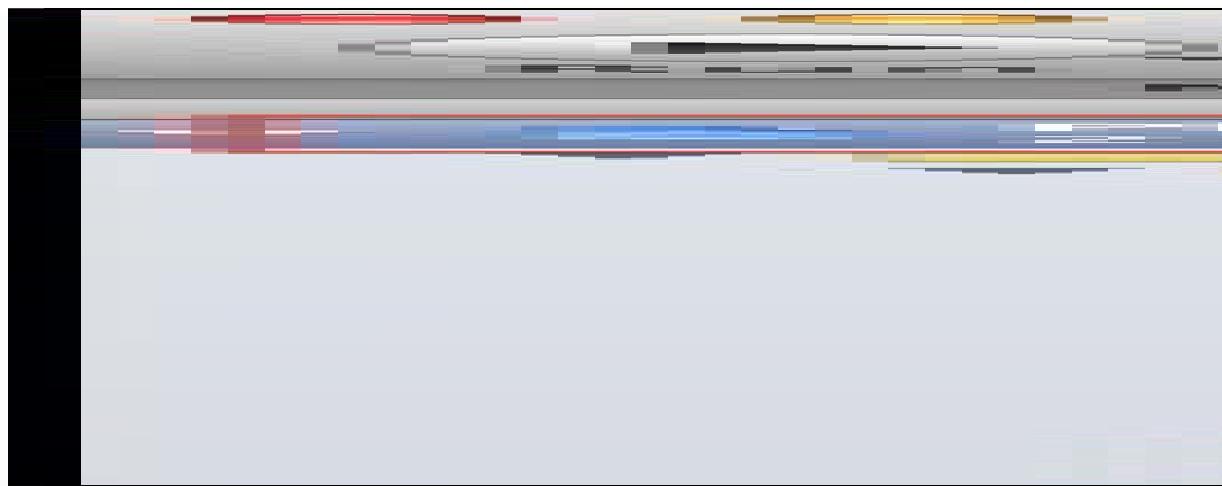
Bước 2: Kết nối iPhone vào máy tính, nếu iPhone đặt password thì phải nhập password để mở iPhone hoặc tốt hơn nên bỏ password của iPhone.

Bước 3: Mở Project mà chúng ta đã “path” trước đó, chờ khoảng 1 phút (để check ID và những thứ linh tinh) sẽ thấy xuất hiện thiết bị của chúng ta.



Hình 3.37: Thiết bị iPhone được nhận diện trong Xcode

Bước 4: Mở “Build Phases” → tab “Run Script” → check vào “Run script only when installing”



Hình 3.38: Chọn Run script only when installing

Bước 5: Run ứng dụng, nếu thành công sẽ thấy ứng dụng của chúng ta trên iPhone.



Hình 3.39: Ứng dụng đã build thành công trên iPhone

3.4 Tiêu kết

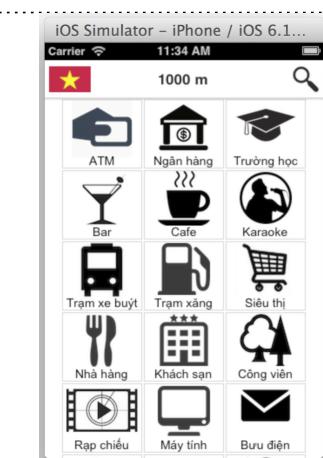
Nội dung chương này trình bày về mô hình MVC trong lập trình ứng dụng iPhone và ví dụ minh họa cho mô hình này. Ngoài ra nội dung chương này cũng trình bày cách thức để đưa ứng dụng lên iPhone đã Jailbreak bằng công cụ JailCoder.

CHƯƠNG 4: MỘT VÀI ÚNG DỤNG ĐÃ ĐƯỢC XÂY DỰNG

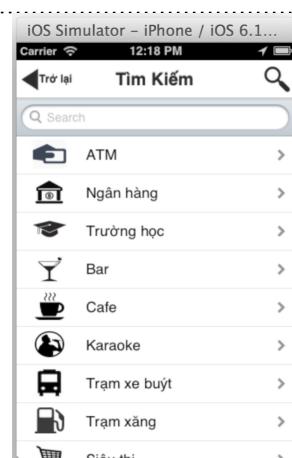
4.1 Úng dụng tìm kiếm địa điểm xung quanh (PlacesNearMe)

4.1.1 Giới thiệu phần mềm

PlacesNearMe giúp người dùng tìm kiếm bất kỳ địa điểm nào ở gần vị trí của họ, chẳng hạn: ATM, Hotels, Banks, School, Park, Karaoke ... Úng dụng sẽ hiển thị bản đồ, các thông tin về nơi cần tìm như: số điện thoại, website, Google plus, hình ảnh ... các giao diện chính của ứng dụng như sau:



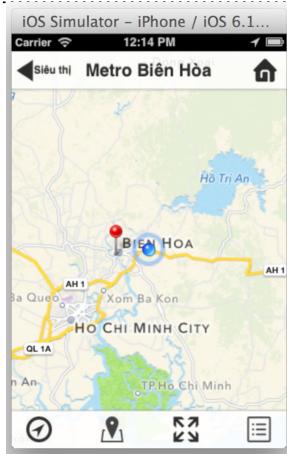
Hình 4.1: Giao diện chính



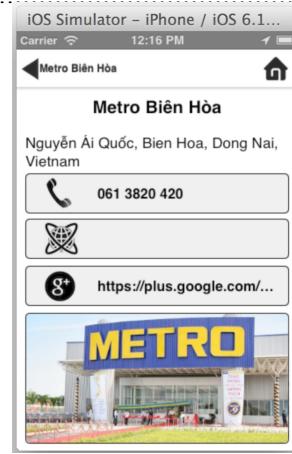
Hình 4.2: Giao diện tìm kiếm



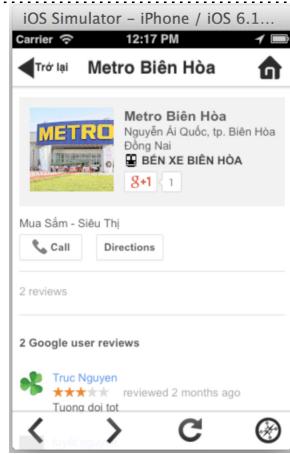
Hình 4.3: Giao diện kết quả tìm kiếm



Hình 4.4: Giao diện bản đồ



Hình 4.5: Giao diện thông tin chi tiết của nơi tìm kiếm



Hình 4.6: Giao diện duyệt web

4.1.2 Cấu trúc của phần mềm

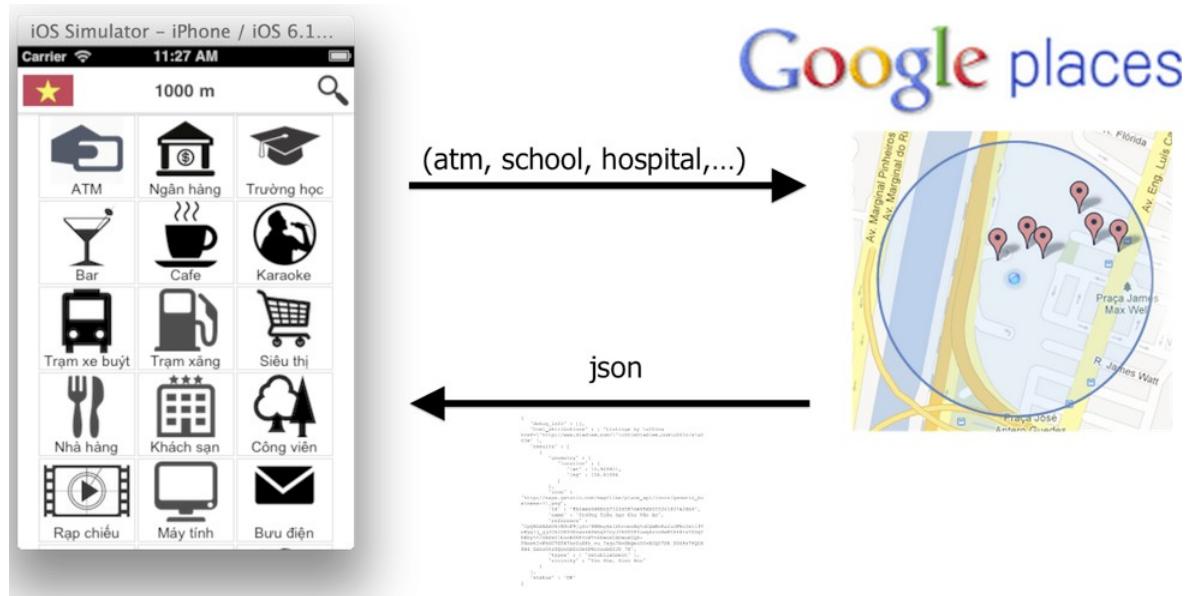
PlacesNearMe có cấu trúc gồm hai phần: lấy dữ liệu (Places API) và hiển thị dữ liệu (Các đối tượng trong Xcode).

Phần lấy dữ liệu sử dụng một dịch vụ do Google cung cấp cho phép tìm kiếm các địa điểm xung quanh một vị trí nào đó, dịch vụ này được gọi là Places API. Places API sẽ nhận vào các từ khóa tìm kiếm như: tọa độ, địa điểm, khoảng cách cần tìm, ... và trả về một dạng văn bản có cấu trúc được gọi là json.

Phần hiển thị dữ liệu dựa vào các đối tượng hỗ trợ trong công cụ lập trình Xcode, lập trình viên sẽ xử lý và hiển thị dữ liệu json lên giao diện. Các đối tượng điển hình trong Xcode mà PlacesNearMe sử dụng: Map View, Web View, Alert View, Image View, Table View Controller ...

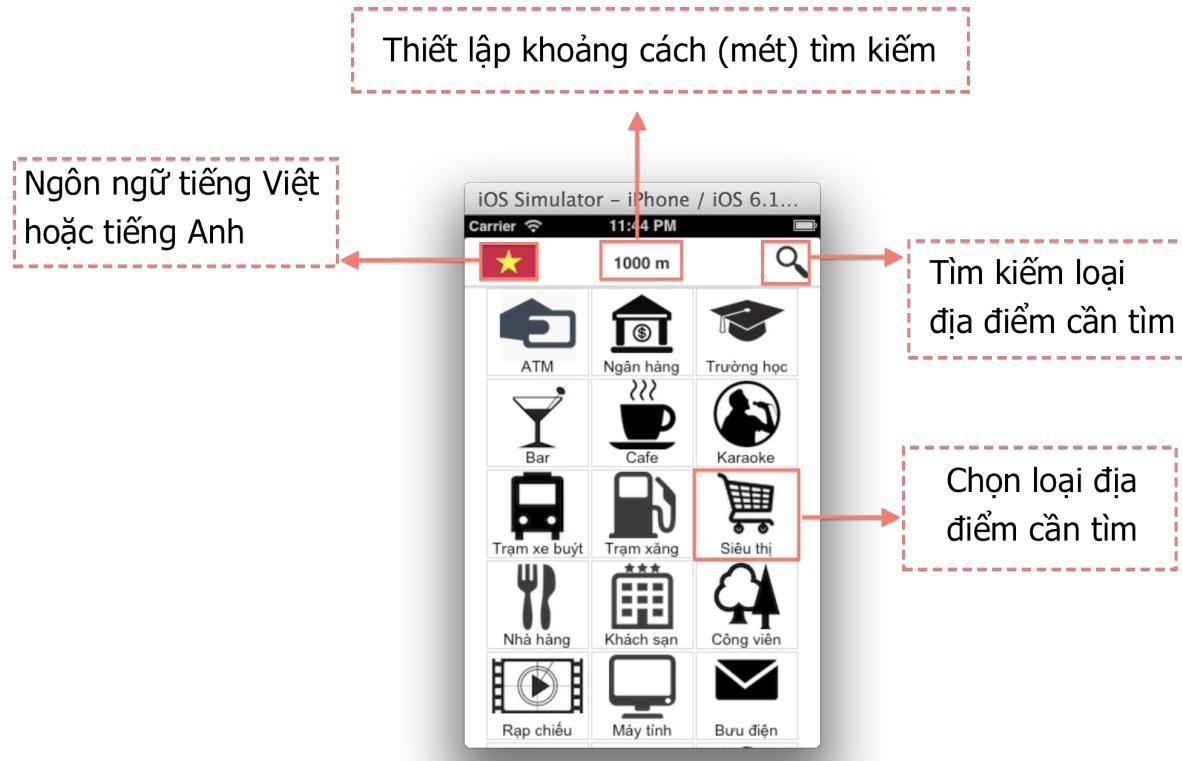
4.1.3 Cơ chế hoạt động của phần mềm

PlacesNearMe sẽ gửi Google các từ khóa tìm kiếm (ATM, School, Hospital, ...). Google sẽ trả về một dạng text có cấu trúc (json), dựa vào nguồn dữ liệu đó lập trình viên sẽ xử lý và thể hiện lên iPhone.

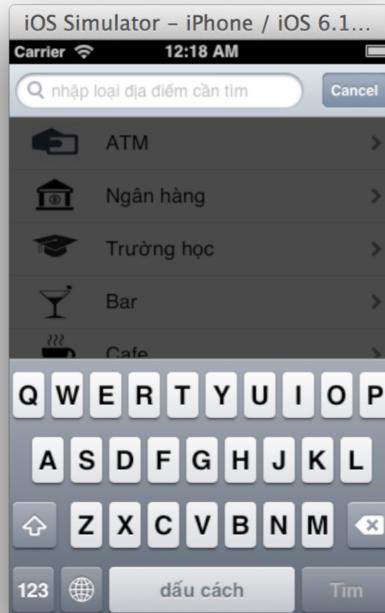


Hình 4.7: Cơ chế hoạt động của PlacesNearMe

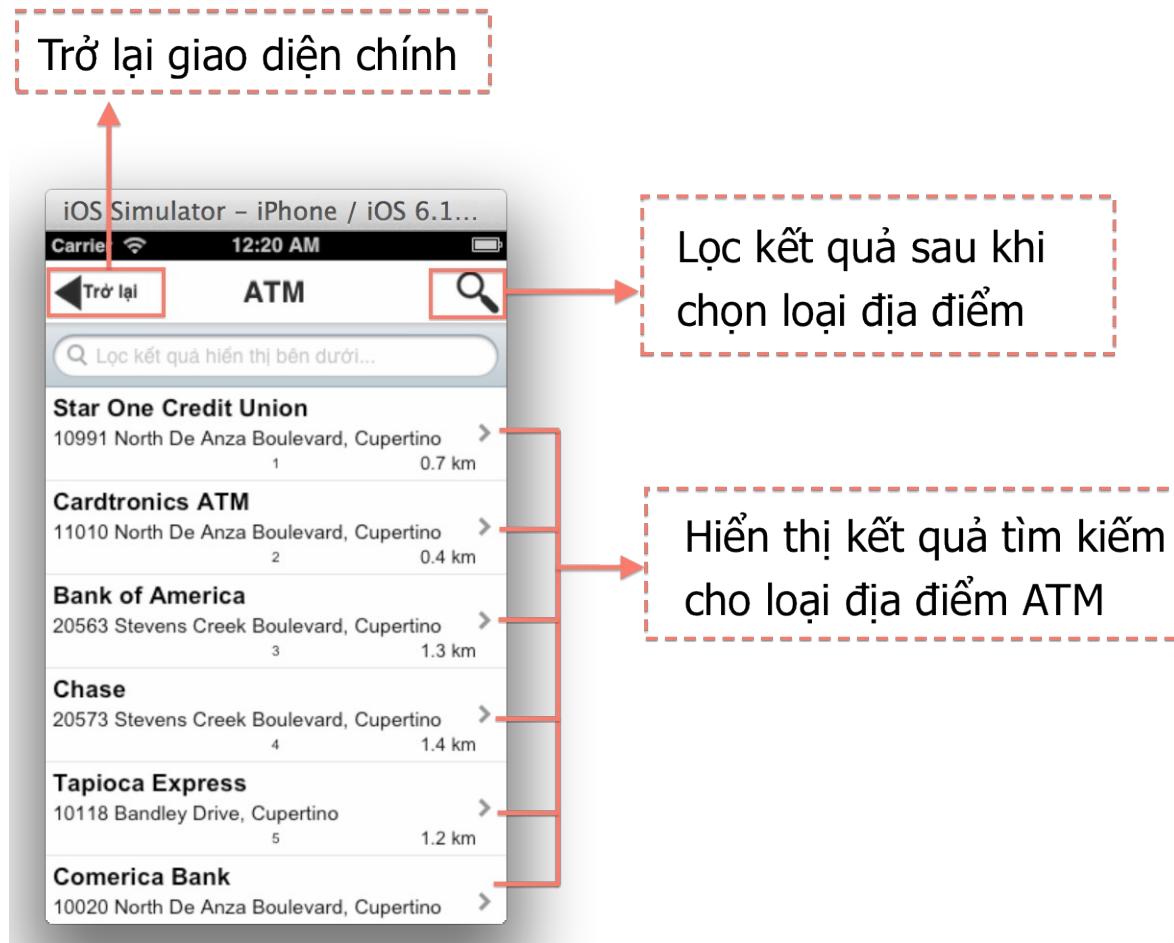
4.1.4 Tính năng của phần mềm



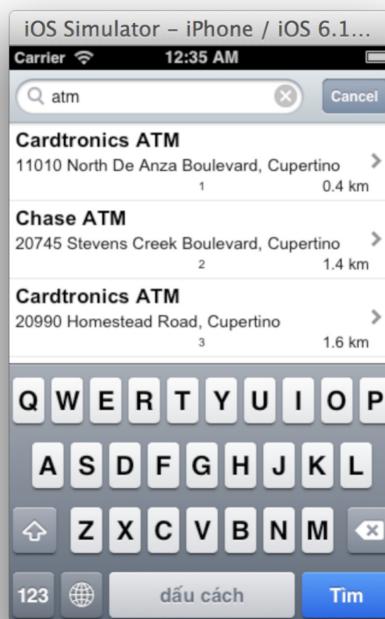
Hình 4.8: Các tính năng của phần mềm trên giao diện chính



Hình 4.9: Giao diện của tính năng “tìm kiếm loại địa điểm cần tìm”



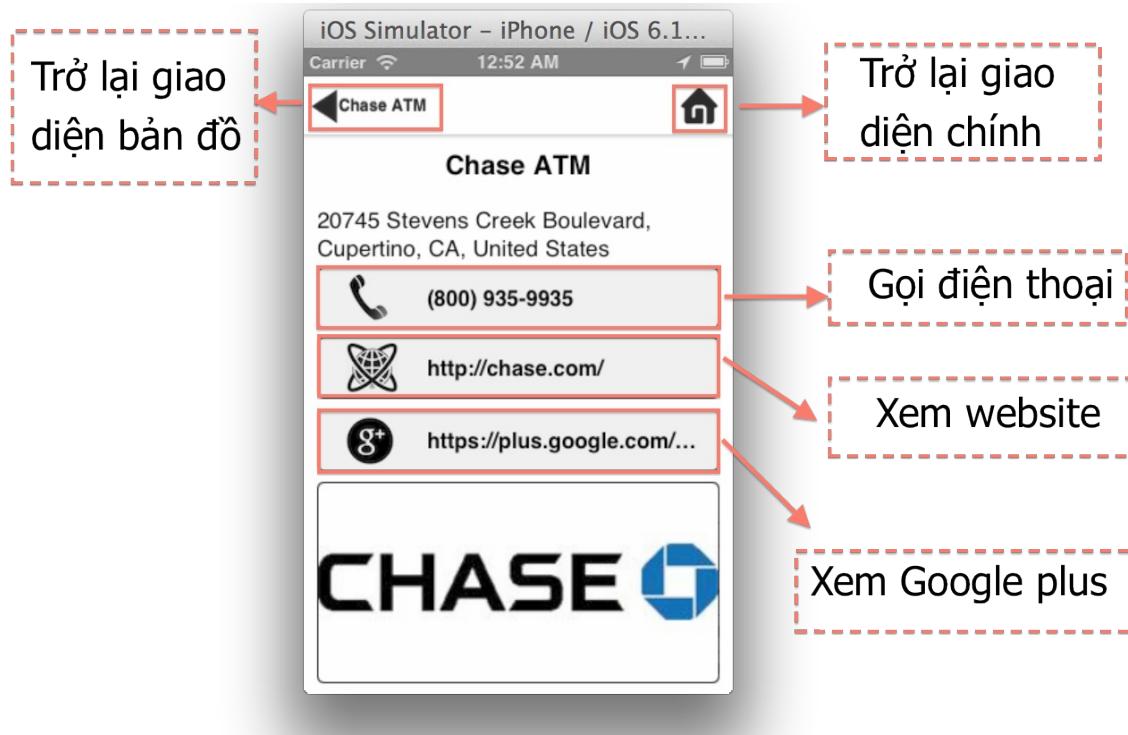
Hình 4.10: Các tính năng trên giao diện kết quả tìm kiếm



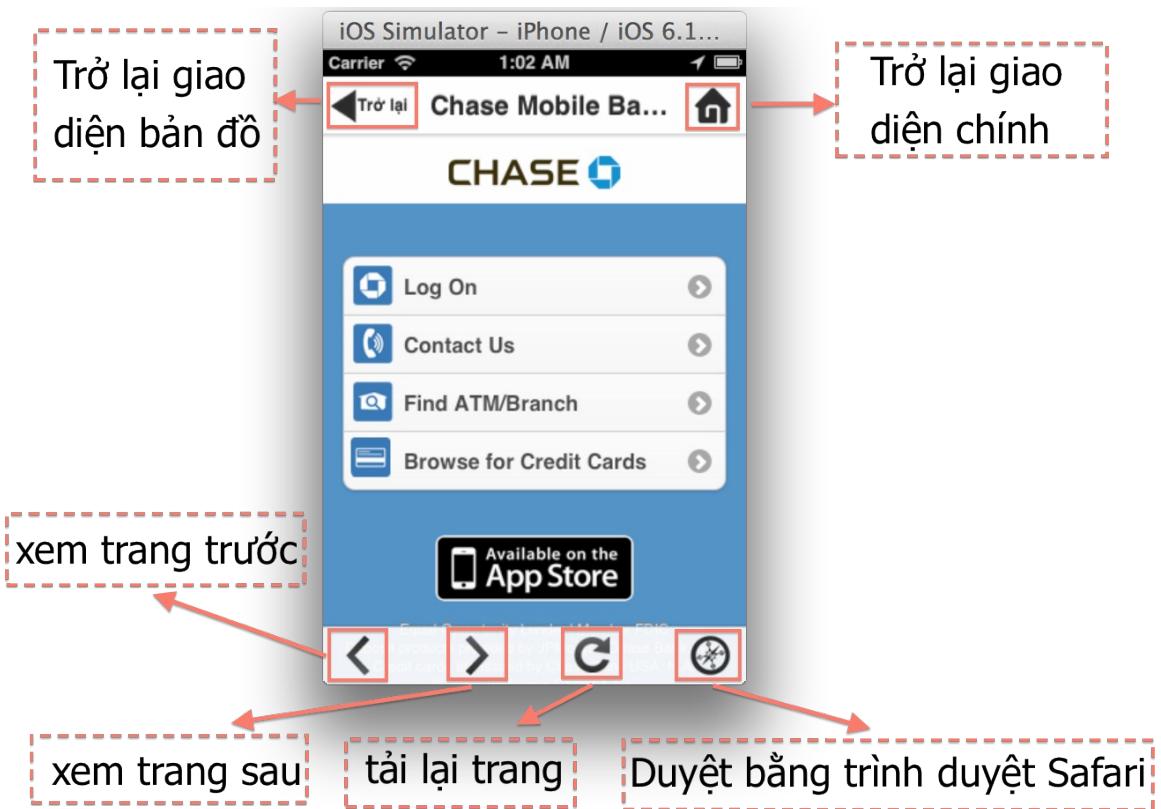
Hình 4.11: Giao diện của tính năng “lọc kết quả sau khi đã chọn địa điểm”



Hình 4.12: Các tính năng trên giao diện bản đồ



Hình 4.13: Các tính năng trên giao diện thông tin chi tiết của nơi tìm kiếm



Hình 4.14: Các tính năng trên giao diện duyệt web

4.1.5 Ưu điểm và những hạn chế của phần mềm

- **Ưu điểm:**

- + Giao diện thân thiện, dễ sử dụng, tìm kiếm nhanh và hiệu quả.
- + Sử dụng API của Google nên độ chính xác cao và xử lý nhanh.

- **Hạn chế:**

- + Chỉ hỗ trợ tối đa 1000 request/ngày do đang dùng dịch vụ Google Places Api miễn phí.
- + Yêu cầu phải kết nối internet khi sử dụng.
- + Chưa hỗ trợ tìm đường đi.

4.2 Ứng dụng tra cứu mã thông tin mã VIN

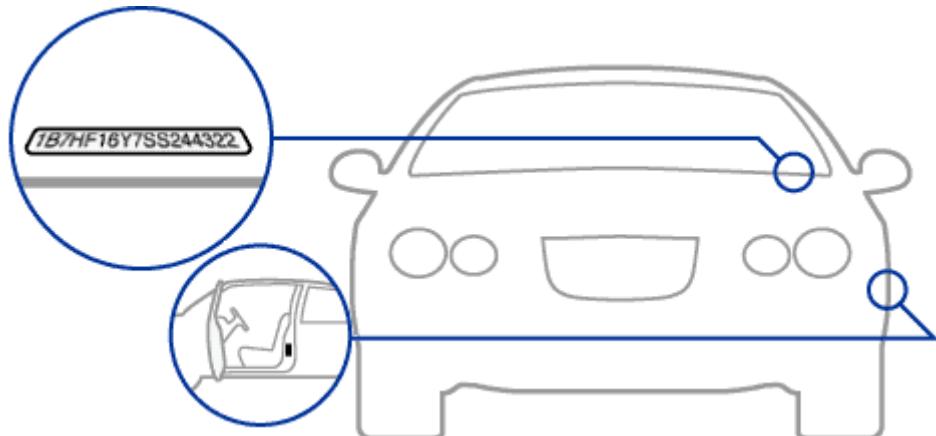
4.2.1 Giới thiệu phần mềm

Số VIN là từ viết tắt của “Vehicle Identification Number”, bao gồm 17 ký tự và được đánh số theo nhiều cách khác nhau. [9]



Hình 4.15: Một số VIN trên xe

Số VIN thường được sử dụng để xác định các thông số của xe như loại xe, nơi sản xuất, hãng sản xuất, năm sản xuất, nhiêu liệu, động cơ, ... VIN có thể nằm ở một số vị trí trên chiếc xe, nhưng phổ biến nhất trên bảng điều khiển (Bạn có thể nhìn thấy qua kính chắn gió) và nhãn dán bên cửa của tài xế. Trong một số xe, số VIN cũng được đặt trên động cơ, mui xe và các bộ phận khác. [13]

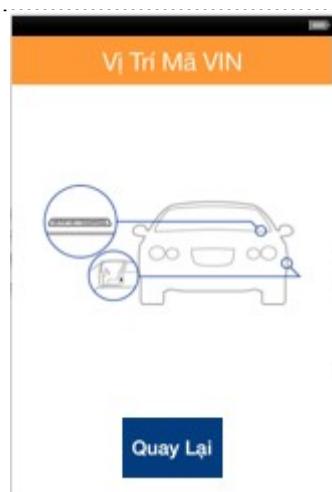


Hình 4.16: Một số vị trí của VIN

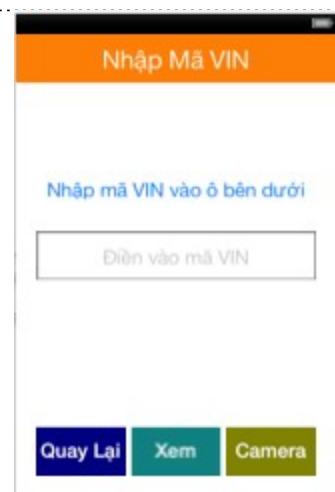
Các giao diện chính của ứng dụng:



Hình 4.17: Giao diện chính



Hình 4.18: Giao diện vị trí mã VIN



Hình 4.19: Giao diện nhập mã VIN



Hình 4.20: Giao diện camera



Hình 4.21: Giao diện xử lý hình ảnh



Hình 4.22: Giao diện kết quả

4.2.2 Cấu trúc của phần mềm

Phần mềm Kiểm tra mã VIN có cấu trúc gồm 3 phần: cơ sở dữ liệu về mã VIN được lưu trữ bằng SQLite, giao diện nhập dữ liệu và hiển thị dữ liệu (sử dụng các đối tượng trong Xcode) và xử lý nhận diện mã VIN trong hình ảnh thông qua Cloud OCR SDK API của hãng ABBYY.

Cơ sở dữ liệu mã VIN sử dụng cơ sở dữ liệu mã VIN năm 2013 của hãng Ford. Cơ sở dữ liệu này sẽ được lưu trữ thông qua hệ quản trị cơ sở dữ liệu SQLite. Với việc sử dụng SQLite, cơ sở dữ liệu sẽ dễ dàng truy cập bằng các câu lệnh truy vấn của SQL như SELECT, CREATE, UPDATE ... Việc quản lý SQLite dễ dàng và đơn giản thông qua plugin SQLite Manager của Firefox và có thể chạy hoàn toàn độc lập mà không cần đến server.

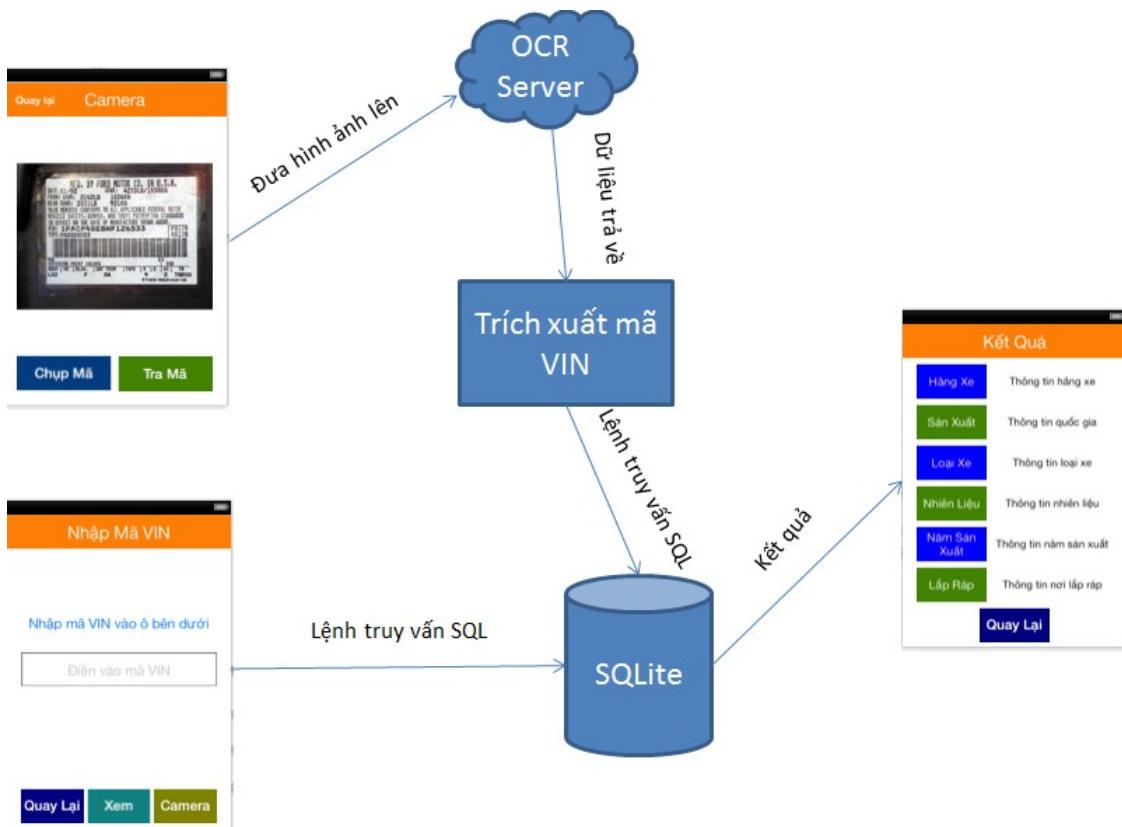
Giao diện nhập dữ liệu và hiển thị dữ liệu sử dụng các đối tượng của Xcode như Button, Label, Text Field, View Controller ... cũng như Camera của iPhone để nhận dữ liệu do người dùng nhập vào, đồng thời hiển thị dữ liệu đã được truy xuất từ SQLite.

Xử lý nhận diện mã VIN trong hình ảnh sử dụng OCR SDK API của hãng ABBYY. Hình ảnh sẽ được đưa lên server, server sẽ xử lý nhận diện và trả về một đoạn text nhận diện được từ hình ảnh. Phần mềm sẽ xử lý đoạn text trả về từ server để tách mã VIN ra và đưa vào SQLite để tra cứu dữ liệu.

4.2.3 Cơ chế hoạt động của phần mềm

Phần mềm hoạt động theo hai cơ chế chính. Cơ chế thứ nhất là người dùng sẽ nhập vào mã VIN bằng tay, chương trình sẽ khởi tạo các câu lệnh truy vấn SQL và gửi đến SQLite để truy xuất dữ liệu và đem kết quả hiển thị ra giao diện.

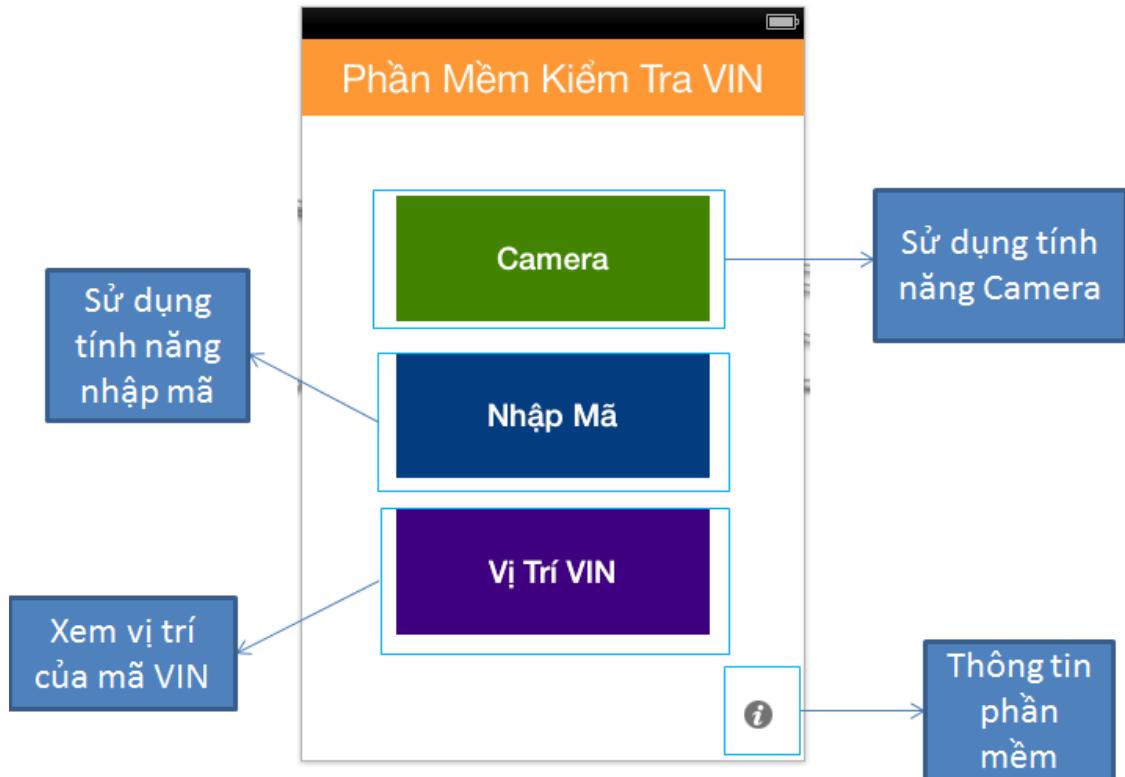
Cơ chế thứ hai sẽ cho phép người dùng sử dụng camera của iPhone chụp lại mã VIN và đưa hình ảnh đó lên OCR Server. Server sẽ xử lý và trả về một đoạn text kết quả sau khi nhận diện ký tự trong hình ảnh. Phần mềm sẽ phải xử lý đoạn text trả về để lọc ra được mã VIN. Nếu hình ảnh chính xác, mã VIN sẽ được lấy ra và tạo câu lệnh truy vấn gửi đến SQLite. SQLite sẽ truy xuất dữ liệu và trả về kết quả. Kết quả sẽ hiển thị ra giao diện.



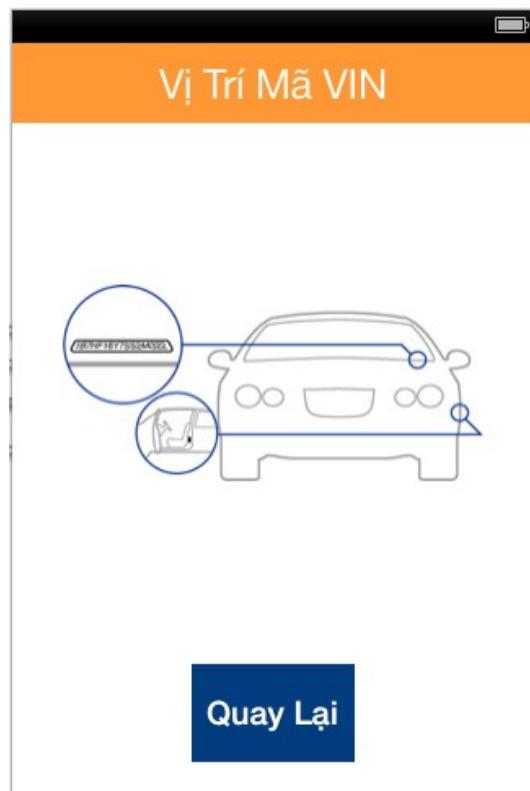
Hình 4.23: Cơ chế hoạt động của phần mềm

4.2.4 Tính năng của phần mềm

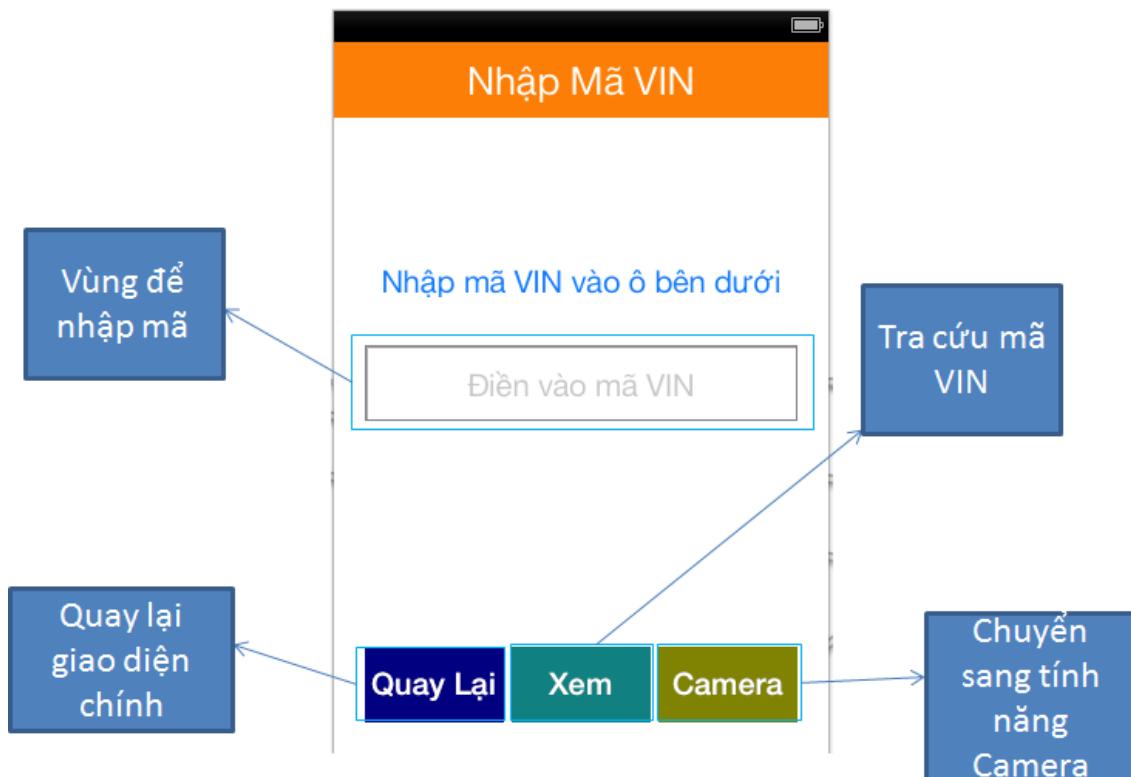
Phần mềm Kiểm tra mã VIN cho phép người dùng có thể kiểm tra mã VIN của xe, từ đó có được một số thông tin về xe nhưng hãng xe, loại xe, nơi sản xuất, năm sản xuất, ... Ngoài ra phần mềm cũng hỗ trợ người dùng tìm kiếm theo các phương án như nhập vào số VIN hoặc sử dụng camera của iPhone để chụp lại mã VIN rồi tra cứu.



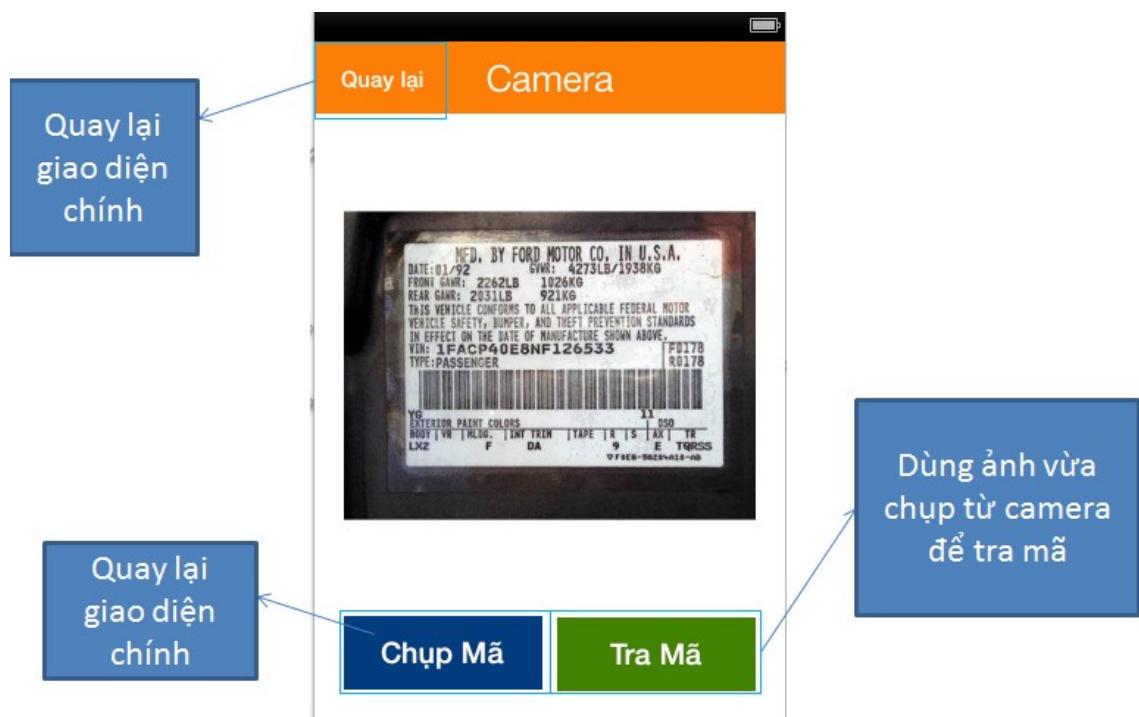
Hình 4.24: Giao diện tổng quan của phần mềm



Hình 4.25: Vị trí mã VIN



Hình 4.26: Tính năng nhập mã VIN để tra cứu



Hình 4.27: Giao diện camera



Hình 4.28: Giao diện xử lý hình ảnh



Hình 4.29: Giao diện kết quả

4.2.5 Ưu điểm và những hạn chế của phần mềm

- **Ưu điểm:**

- + Phần mềm cho phép tra cứu mã VIN một cách nhanh chóng, sử dụng cơ sở dữ liệu của SQLite nên dễ sử dụng, hoạt động độc lập không cần tới server.
- + Hơn nữa phần mềm còn có thêm tính năng chụp ảnh mã VIN để tra cứu dựa trên công nghệ OCR của hãng ABBYY giúp nhận diện mã VIN với độ chính xác cao từ hình ảnh mã VIN chụp được.

- **Hạn chế:**

- + Cơ sở dữ liệu mã VIN hiện tại chỉ hỗ trợ cho hãng Ford và Lincoln.
- + Độ chính xác của mã VIN khi sử dụng tính năng camera còn phụ thuộc vào hình ảnh mã VIN được chụp lại.
- + Chưa cho phép lưu hình ảnh chụp được vào thư viện Photo của iPhone cũng như hỗ trợ sử dụng hình ảnh trong Photo để tra cứu.
- + Nếu sử dụng chức năng camera để chụp ảnh cần phải kết nối mạng (3G hoặc wifi) để gửi ảnh về server xử lý trước khi lấy kết quả trả về.

4.3 Tiêu kết

Chương này trình bày về một số ứng dụng nhóm xây dựng được về đặc điểm, tính năng, giao diện. Bên cạnh đó nội dung chương này còn đánh giá những ưu và khuyết điểm còn tồn tại của phần mềm.

CHƯƠNG 5: TÀI LIỆU HƯỚNG DẪN XÂY DỰNG ỨNG DỤNG TRÊN IPHONE

5.1 Nhu cầu thực tiễn

Theo nhóm tìm hiểu, hiện nay trên thế giới đã có nhiều tài liệu hướng dẫn xây dựng ứng dụng trên iPhone bằng tiếng Anh như “**iOS 6 Application Development for Dummies**” (Neal Goldstein – Dave Wilson – 2013), “**iPhone and iPad Apps for Absolute Beginners, 3rd edition**” (Apress – 2013), “**Head First iPhone And iPad Development, 2nd edition**” (O'Reilly – 2011), “**Programming iOS 6, 3rd Edition**” (Matt Neuburg – 3/2013) ... Ngoài ra một số trường đại học trên thế giới cũng đã đưa bộ môn lập trình iPhone vào giảng dạy và thường xuyên cập nhật chia sẻ giáo trình lên mạng như “**Developing iOS 7 Apps for iPhone and iPad**” (Standford University – 28/10/2013), “**iPhone Application Programming WS**” (RWTH Aachen University – 21/10/2013) ...

Ở Việt Nam, việc tìm tài liệu hướng dẫn xây dựng ứng dụng iPhone bằng tiếng Việt không hề dễ dàng. Chủ yếu người tìm kiếm thường tìm kiếm trên mạng internet, các diễn đàn, nhóm về lập trình, lập trình iPhone nhưng đa phần là giới thiệu các tài liệu của nước ngoài, các bài hướng dẫn bằng tiếng Việt thì ít. Bên cạnh đó, các giáo trình của các trung tâm đào tạo lập trình iPhone tại Việt Nam lại ít được chia sẻ rộng rãi mà chỉ lưu hành nội bộ là chủ yếu gây khó khăn cho việc tiếp cận tài liệu của người tìm kiếm.

Vì thế, nhóm đã thực hiện tìm hiểu, nghiên cứu và tổng hợp những kiến thức, kết quả đạt được để xây dựng nên tài liệu hướng dẫn xây dựng ứng dụng iPhone bằng tiếng Việt với mong muốn hỗ trợ tốt hơn cho mọi người trong việc tìm hiểu xây dựng ứng dụng iPhone.

5.2 Cơ sở nền tảng để xây dựng hướng dẫn

Khi bắt đầu thực hiện đề tài, nhóm đã tiến hành tìm hiểu các tài liệu trong nước và ngoài nước, các nguồn tài liệu từ sách ebook, tài liệu hướng dẫn của Apple hay từ video, thực hành làm theo các ví dụ. Qua đó, nhóm nắm được cách sử dụng một số đối tượng trong Xcode và thực hành sử dụng các đối tượng đó vào việc xây dựng ứng dụng có tính thực tiễn. Từ các ứng dụng được hoàn thiện, nhóm đã tổng kết các đối

tượng, kinh nghiệm đạt được trong quá trình tìm hiểu và xây dựng ứng dụng, từ đó biên soạn thành tài liệu hướng dẫn bằng tiếng Việt.

5.3 Xây dựng tài liệu hướng dẫn xây dựng ứng dụng trên iPhone

5.3.1 Bố cục tài liệu hướng dẫn

Bố cục tài liệu hướng dẫn gồm tám phần:

Phần 1: Chuẩn bị trước khi bắt đầu xây dựng ứng dụng iPhone.

- Giới thiệu về quá trình chuẩn bị một hệ điều hành Mac OS làm nền tảng trước khi cài đặt công cụ Xcode. Trình bày cho người đọc các phương án có thể sử dụng để sở hữu được một hệ điều hành Mac OS, từ đó có được lựa chọn thích hợp.

- Giới thiệu các cách thức để cài đặt công cụ Xcode lên hệ điều hành Mac OS.

Phần 2: Tìm hiểu phần mềm Xcode và iOS Simulator.

- Tìm hiểu về công cụ Xcode.

- + Giới thiệu về công cụ lập trình Xcode phiên bản 5 (phiên bản mới nhất hiện nay), giao diện cũng như một số tính năng nổi bật.

- + Thao tác tạo một ứng dụng mới.

- + Trình bày chi tiết giao diện Xcode và tính năng của một số vùng trên giao diện.

- **Navigator area:** cho phép quản lý các tập tin, thư mục, các thông báo lỗi, quá trình debug, ... của ứng dụng.

- **Editor area:** cho phép thiết kế giao diện cho ứng dụng, viết code cho các sự kiện, ...

- **Toolbar area:** chứa các button cho phép thao tác nhanh chóng một số tính năng như chạy ứng dụng, lựa chọn phiên bản iOS SDK phù hợp, đóng mở các vùng của giao diện Xcode, ...

- **Utility area:** được sử dụng để thay đổi thuộc tính của các đối tượng Xcode, chia các đối tượng của Xcode dùng để thiết kế giao diện, ...

- + Cách thức thiết kế giao diện cho một ứng dụng.

- + Thao tác viết code cho ứng dụng bao gồm việc ánh xạ các đối tượng vào tập tin quản lý, cũng như viết code cho các sự kiện của đối tượng.
- + Quá trình chạy và kiểm tra lỗi của ứng dụng.
- Tìm hiểu về công cụ iOS Simulator.
 - + Giới thiệu về iOS Simulator và tính năng hỗ trợ mô phỏng iPhone, iPad với iOS SDK.
 - + Trình bày một số thao tác cơ bản trên iOS Simulator như:
 - Thao tác đóng/mở iOS Simulator.
 - Thao tác xoay màn hình Simulator.
 - Ân/hiện bàn phím trong Simulator.
 - Cài đặt/gỡ bỏ ứng dụng trên Simulator.
 - Thao tác bổ sung thêm các phiên bản iOS SDK của các phiên bản iOS 6, iOS 6.1, ...
 - Thao tác chụp ảnh màn hình, copy/paste trong Simulator.
 - Trình bày một số điểm hạn chế của iOS Simulator.

Phần 3: Tìm hiểu về ngôn ngữ lập trình Objective-C.

- Giới thiệu về ngôn ngữ lập trình Objective-C.
- Trình bày cú pháp, khai báo, kiểu dữ liệu, cách viết hàm trong Objective-C.
- Hướng dẫn một số cấu trúc, đối tượng trong Objective-C kèm theo ví dụ minh họa.
 - + Cấu trúc điều kiện: If-Else, Switch-Case.
 - + Vòng lặp: For, While, Do-While.
 - + Mảng.
 - + Đối tượng NSString.

Phần 4: Hướng dẫn cơ bản.

- Hướng dẫn thêm App icon, Launch image cho ứng dụng.

- Thay đổi tên ứng dụng trường hợp tên ứng dụng quá dài.
- Cách ẩn status bar khi chạy ứng dụng.
- Hướng dẫn thay đổi background cho ứng dụng.
- Cách thêm một framework vào project.
- Thao tác ẩn bàn phím cho text field.
- Cách thực hiện để project hỗ trợ xoay màn hình.

Phần 5: Một số đối tượng cơ bản.

Hướng dẫn một số đối tượng cơ bản trong Xcode dựa trên các đối tượng đã tìm hiểu và ứng dụng vào xây dựng phần mềm. Với mỗi đối tượng đều có ví dụ minh họa chi tiết, đơn giản, dễ hiểu. Một số đối tượng được hướng dẫn như:

- Label – Button – Text field.
- Sử dụng camera iPhone.
- Kết nối cơ sở dữ liệu với SQLite.
- Activity Indicator View.
- TableView Controller.
- Action Sheet.
- Slider.
- Truyền dữ liệu giữa các Views.
- Map View.
- Image View.
- Web View.
- Alert View.
- Search Bar.

Phần 6: Hướng dẫn xây dựng ứng dụng Tìm kiếm địa điểm xung quanh và Kiểm tra mã VIN.

- Giới thiệu phần mềm, tính năng của từng phần mềm.

- Hướng dẫn xây dựng phần mềm theo từng bước giúp người đọc có thể thực hiện theo hướng dẫn và tự mình xây dựng được hai ứng dụng Tìm kiếm địa điểm xung quanh và Kiểm tra mã VIN.
- Đánh giá ưu và nhược điểm của từng phần mềm.

Phần 7: Sử dụng JailCoder đưa ứng dụng lên iPhone đã Jailbreak.

- Giới thiệu về JailCoder.
- Quá trình chuẩn bị trước khi đưa ứng dụng lên iPhone
- Thao tác từng bước để đưa ứng dụng xây dựng được lên thiết bị iPhone đã được Jailbreak.

Phần 8: Một số vấn đề khác trong xây dựng ứng dụng trên iPhone.

- Mặc dù iOS 7 đã chính thức ra mắt, nhưng hiện nay đa phần người dùng vẫn còn sử dụng phiên bản iOS 6. Phần này hướng dẫn người đọc sử dụng công cụ Xcode 5 để xây dựng ứng dụng chạy trên phiên bản iOS 6.1, iOS 6.0.
- Ngoài ra, phần này còn hướng dẫn thêm cách thức để xây dựng ứng dụng có thể chạy được trên cả phiên bản iOS 7 và phiên bản iOS thấp hơn như iOS 6.

5.3.2 Dạng sách điện tử của tài liệu hướng dẫn

Tài liệu hướng dẫn được chuyển sang dạng ebook để thuận tiện cho việc chia sẻ lên mạng internet cũng như dễ dàng đọc trên các thiết bị như PC, Laptop, Smartphone

...

Một số đặc điểm của tài liệu dạng ebook:

- Dung lượng nhẹ.
- Chia sẻ nhanh chóng trên mạng internet thông qua các diễn đàn, nhóm trên Facebook.
- Có thể đọc được trên nhiều thiết bị như: PC, Laptop, Smartphone, Máy tính bảng, ...
- Không tốn kém chi phí cho việc in ấn hoặc vận chuyển.
- Có hệ thống mục lục rõ ràng, dễ dàng tra cứu, tìm kiếm nội dung.

5.4 Tiêu kết

Trong chương này giới thiệu về tài liệu hướng dẫn xây dựng ứng dụng trên iPhone bằng công cụ Xcode mà nhóm đã biên soạn. Nêu lên cơ sở lý thuyết để thực hiện tài liệu hướng dẫn, bối cảnh nội dung được trình bày trong tài liệu hướng dẫn.

CHƯƠNG 6: ĐÁNH GIÁ KẾT QUẢ ĐẠT ĐƯỢC

6.1 Đóng góp của đề tài vào thực tiễn

Nhóm đã xây dựng được hai ứng dụng có tính ứng dụng thực tế, đáp ứng nhu cầu của người sử dụng. Ứng dụng tìm kiếm địa điểm xung quanh hỗ trợ người dùng tìm kiếm các địa điểm như trường học, ATM, ăn uống, ... trong phạm vi tùy chỉnh một cách nhanh chóng, dễ dàng, chính xác. Ứng dụng tra cứu mã VIN hỗ trợ người dùng nắm bắt được thông tin của xe như hãng xe, năm sản xuất, loại xe, nơi sản xuất, ... thông qua việc kiểm tra số VIN của xe chỉ bằng một chiếc iPhone mà không cần phải truy cập vào các website trên internet để tìm kiếm.

Công cụ Xcode là công cụ duy nhất do Apple cung cấp để các lập trình viên sử dụng vào việc xây dựng ứng dụng cho thiết bị chạy hệ điều hành của Apple (Mac OS, iOS) nói chung và của iPhone nói riêng. Nhóm đã tìm hiểu và xây dựng tài liệu hướng dẫn sử dụng một số đối tượng cơ bản của Xcode vào xây dựng ứng dụng trên iPhone, với mong muốn tạo điều kiện dễ dàng hơn cho những ai đam mê, muốn tìm hiểu về xây dựng ứng dụng trên iPhone.

Ngoài tài liệu được biên soạn thành sách in, nhóm còn triển khai xây dựng phiên bản ebook cho hướng dẫn để dễ dàng chia sẻ lên mạng internet, tiếp cận được với nhiều người dùng hơn.

Việc xây dựng được tài liệu hướng dẫn xây dựng ứng dụng iPhone bằng tiếng Việt sẽ hỗ trợ tốt hơn cho mọi người muốn tìm hiểu về lập trình iPhone, qua đó tiết kiệm hơn về thời gian, cũng như giảm bớt lo ngại về khả năng đọc hiểu tài liệu tiếng Anh của người khi tiến hành tìm hiểu về lập trình iPhone.

6.2 Ưu điểm và hạn chế của đề tài

Với những kết quả đạt được của đề tài, nhóm nghiên cứu xin đánh giá những ưu điểm và hạn chế của đề tài như sau:

- Ưu điểm:

+ Xây dựng được hai ứng dụng có tính ứng dụng vào thực tiễn. Ứng dụng “Tìm kiếm địa điểm xung quanh” hỗ trợ người dùng tìm kiếm các địa điểm theo mục đích tìm kiếm trong phạm vi tuỳ chỉnh một cách nhanh chóng, dễ dàng, chính xác. Ứng dụng “Tra cứu mã VIN” hỗ trợ người dùng nắm bắt được thông tin của xe dựa theo số VIN của xe chỉ bằng một chiếc iPhone mà không cần phải tìm kiếm trên mạng internet.

+ Tài liệu hướng dẫn xây dựng ứng dụng trên iPhone là tài liệu tiếng Việt, hoàn toàn miễn phí, hướng dẫn chi tiết cách sử dụng một số đối tượng trong Xcode như Label, Button, Text Field, Table View, ... để xây dựng ứng dụng trên iPhone, cũng như có ví dụ minh họa chi tiết. Ngoài ra tài liệu hướng dẫn còn được chuyển sang định dạng ebook để dễ dàng chia sẻ rộng rãi đến người dùng trên các diễn đàn, nhóm chuyên về lập trình, lập trình iPhone.

- Hạn chế:

+ Do hiểu biết và khả năng có hạn nên các ứng dụng vẫn còn tồn tại nhiều hạn chế về giao diện cũng như tính năng.

+ Thời gian ngắn nên số lượng đối tượng được tìm hiểu và đưa vào xây dựng hướng dẫn còn ít, việc chia sẻ hướng dẫn lên mạng internet chưa lâu nên chưa tổng hợp được phản hồi đóng góp từ mọi người từ đó có được đánh giá khách quan về tài liệu hướng dẫn.

KẾT LUẬN

Đề tài nghiên cứu với mục tiêu tìm hiểu cách thức lập trình mô phỏng với bộ công cụ Xcode để tạo ứng dụng chạy trên hệ điều hành iOS, thông qua kết quả tìm hiểu tiến hành xây dựng ứng dụng thực tiễn chạy trên iPhone và xây dựng tài liệu hướng dẫn sử dụng các đối tượng trong Xcode để lập trình ứng dụng iPhone. Với những mục tiêu nghiên cứu đề ra, nhóm đã nghiên cứu và đạt được hai kết quả chính.

Kết quả thứ nhất là xây dựng được hai ứng dụng chạy trên iPhone có tính ứng dụng thực tiễn từ các đối tượng tìm hiểu được trong Xcode. Ứng dụng đầu tiên là ứng dụng “Tìm kiếm địa điểm xung quanh” hỗ trợ người dùng tìm kiếm các địa điểm như trường học, ATM, ăn uống, ... trong phạm vi tuỳ chỉnh một cách nhanh chóng, dễ dàng, chính xác. Ứng dụng thứ hai là ứng dụng “Tra cứu mã VIN” hỗ trợ người dùng nắm bắt được thông tin của xe như hãng xe, năm sản xuất, loại xe, nơi sản xuất, ... thông qua việc kiểm tra số VIN của xe chỉ bằng một chiếc iPhone mà không cần phải truy cập vào website trên internet để tìm kiếm.

Kết quả thứ hai là hoàn thiện được tài liệu hướng dẫn sử dụng các đối tượng trong Xcode để lập trình ứng dụng iPhone. Tài liệu sẽ hướng dẫn người đọc quá trình chuẩn bị trước khi bắt đầu tìm hiểu về lập trình ứng dụng trên iPhone, hướng dẫn một số đối tượng của Xcode như Label, Button, Text Field, Table View, SQLite, ... với các ví dụ minh họa cho từng đối tượng. Ngoài ra tài liệu cũng hướng dẫn người đọc cách thức để xây dựng được hai ứng dụng “Tìm kiếm địa điểm xung quanh” và “Tra cứu mã VIN” với thao tác từng bước, hình ảnh minh họa cụ thể.

Do thời gian có hạn và hiểu biết còn ít nên nhóm chỉ dừng lại ở việc tìm hiểu được một vài đối tượng cơ bản trong Xcode cũng như cách sử dụng các đối tượng đó để xây dựng ứng dụng iPhone chứ vẫn chưa đi sâu vào tìm hiểu thêm được nhiều đối tượng khác cũng như tất cả các chức năng mà công cụ Xcode hỗ trợ người dùng để xây dựng ứng dụng iPhone. Các ứng dụng xây dựng được vẫn còn tồn tại một số hạn chế về giao diện và tính năng.

Hướng phát triển nghiên cứu tiếp theo:

Qua những kết quả đã đạt được và chưa đạt được, nhóm nghiên cứu nhận thấy rằng khả năng và kiến thức của nhóm còn nhiều hạn chế, để kết quả nghiên cứu hoàn

thiện hơn và mang lại tính ứng dụng cao hơn trong thực tiễn, nhóm nghiên cứu nhận thấy cần phải tìm hiểu thêm nhiều đối tượng khác của Xcode từ cơ bản đến nâng cao, từ đó bổ sung và hoàn thiện hơn tài liệu hướng dẫn mà nhóm đã biên soạn. Đồng thời dựa trên những đối tượng mới được tìm hiểu để xây dựng thêm một vài ứng dụng có tính ứng dụng thực tiễn.

TÀI LIỆU THAM KHẢO

TIẾNG VIỆT

[1] Hoàng Kim Ngọc (2012), *Nghiên cứu xây dựng ứng dụng cho hệ điều hành iOS*, Đề án tốt nghiệp Đại học hệ chính quy ngành Công Nghệ Thông Tin, Trường Đại học Dân lập Hải Phòng.

[2] Trung tâm Nhất Nghệ (2013), *Lập Trình Ứng Dụng iPhone Cơ Bản*, Hồ Chí Minh.

TIẾNG ANH

[3] Aaron Hillegas (2011), *Objective-C Programming: The Big Nerd Ranch Guide*, Big Nerd Ranch Inc.

INTERNET

[4] *Apple App Store: 40B Downloads And Counting; 2B+ In December, ~20B In 2012*, TechCrunch, truy cập ngày 9/11/2013, <<http://techcrunch.com/2013/01/07/apple-app-store-40bn-downloads-and-counting-2bn-in-december-20bn-in-2012/>>.

[5] *iOS (Apple)*, Wikipedia, truy cập ngày 9/11/2013, <http://vi.wikipedia.org/wiki/IOS_%28Apple%29>.

[6] *iOS Simulator User Guide*, iOS Developer Library, truy cập ngày 11/11/2013, <https://developer.apple.com/library/ios/documentation/IDEs/Conceptual/iOS_Simulator_Guide/TestingontheiOSSimulator/TestingontheiOSSimulator.html>.

[7] *Objective-C*, Wikipedia, truy cập ngày 10/11/2013, <<http://en.wikipedia.org/wiki/Objective-C>>.

[8] *Soi động thị trường kho ứng dụng cho di động*, Dân Trí, truy cập ngày 9/11/2013, <<http://dantri.com.vn/suc-manh-so/soi-dong-thi-truong-kho-ung-dung-cho-di-dong-760940.htm>>.

[9] *Số VIN*, Wikipedia, truy cập ngày 10/11/2013, <http://vi.wikipedia.org/wiki/S%C3%A1o_VIN>.

[10] *The iPhone OS Architecture and Frameworks*, Techotopia, truy cập ngày 10/11/2013, <http://www.techotopia.com/index.php/The_iPhone_OS_Architecture_and_Frameworks>.

ks>.

[11] *Tổng quan Objective-C Lập trình ứng dụng trên iOS*, Framgia's Blog, truy cập ngày 10/11/2013, <<http://tech.blog.framgia.com/vn/?p=1228>>.

[12] *Windows Phone Up-and-Coming Challenger to Android for Vietnam Smartphone Market*, IDC. International Data Corporation, truy cập ngày 9/11/2013, <<http://www.idc.com/getdoc.jsp?containerId=prVN24300613>>.

[13] *Xcode*, Wikipedia, truy cập ngày 11/11/2013, <<http://en.wikipedia.org/wiki/Xcode>>.