**BỘ THÔNG TIN VÀ TRUYỀN THÔNG**

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

------------------------------



**BÁO CÁO THỰC TẬP**

**TỐT NGHIỆP ĐẠI HỌC**

***Đề tài*:**

**PHÁT TRIỂN ỨNG DỤNG CUNG CẤP DỊCH VỤ TÌM KIẾM NHÀ TRỌ & MỘT SỐ DỊCH VỤ LIÊN QUAN KHÁC**

**Giáo viên hướng dẫn : TS. GVC VÕ XUÂN THỂ**

**Sinh viên thực hiện : TẠ THỊ YẾN NHI**

**Mã sinh viên : N14DCCN237**

**Lớp : D14CQCP01-N**

**Khóa : 2014 - 2019**

**Hệ : ĐẠI HỌC CHÍNH QUI**

**TP.HCM, tháng 08/2018**

**LỜI CẢM ƠN**

Ngày nay Khoa học kỹ thuật nói chung, công nghệ thông tin nói riêng đang trên đà phát triển và nâng cao về mọi mặt. Song song với đó, công nghệ Internet, điện thoại di động thông minh cũng như cái dịch vụ công nghệ đang trên đà phát triển rất cao. Và với công nghệ phát triển như hiện nay, con người đang được giúp sức rất nhiều về tiết kiệm chi phí cũng như công sức thông qua các chiếc điện thoại thông minh và Internet. Để đáp ứng được như cầu ngày càng hiện đại hoá hơn thì các ứng dụng thông minh được ra đời ngày càng nhiều hơn. Bắt nhịp được xu thế đó thì “Ứng dụng cung cấp dịch vụ nhà trọ & một số dịch vụ liên quan khác” được ra đời dựa trên nền tảng iOS được ra đời.

Qua đây, em xin chân thành cảm ơn quý thầy cô ở khoa Công Nghệ Thông Tin 2 - Học viện Công Nghệ Bưu Chính Viễn Thông cơ sở tại TP.HCM đã nhiệt tình giảng dạy cũng như truyền đạt cho chúng em rất nhiều kiến thức quý báu trong suốt bốn năm đại học của chúng em. Em xin chân thành cảm ơn phòng kỹ thuật & ban giám đốc công ty SAIGONLAB đã tạo điều kiện và giúp đỡ em trong thời gian em thực tập. Đặt biệt, em chân trọng gửi lời cảm ơn sâu sắc đến thầy TS. GVC Võ Xuân Thể đã đồng hành cùng em, hướng dẫn, góp ý và giúp em định hướng trong thời gian vừa qua.

Bước đầu đi vào thực tế, mọi thứ với em còn nhiều mới mẻ và việc thiếu sót là không thể tránh khỏi. Em rất mong nhận được những ý kiến đóng góp quý báu của quý Thầy Cô và các bạn học cùng lớp để kiến thức của em hoàn thiện hơn.

Em xin chân thành cảm ơn!

TP.HCM, ngày 5 tháng 7 năm 2018

Sinh viên

Tạ Thị Yến Nhi

**MỤC LỤC**

[CHƯƠNG 1 : GIỚI THIỆU ĐỀ TÀI 1](#_Toc521059532)

[1.1 Giới thiệu 1](#_Toc521059533)

[1.2 Mục tiêu 1](#_Toc521059534)

[1.3 Lĩnh vực 1](#_Toc521059540)

[1.4 Nền tảng được sử dụng 1](#_Toc521059541)

[1.5 Nội dung trình bày đề tài 1](#_Toc521059542)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT CỦA ĐỀ TÀI 2](#_Toc521059543)

[2.1 NỀN TẢNG CÔNG NGHỆ 2](#_Toc521059544)

[2.2 KIẾN TRÚC ỨNG DỤNG IOS 2](#_Toc521059545)

[2.2.1 Cocoa Touch 2](#_Toc521059546)

[2.2.2 Media 2](#_Toc521059547)

[2.2.3 Core Service 2](#_Toc521059548)

[2.2.4 Core OS 3](#_Toc521059549)

[2.3 QUI TRÌNH XÂY DỰNG ỨNG DỤNG 3](#_Toc521059550)

[2.3.1 Thiết kế giao diện 3](#_Toc521059551)

[2.3.2 Tổ chức dữ liệu 4](#_Toc521059552)

[2.3.4 Xây dựng tương tác: 4](#_Toc521059553)

[2.3.5 Kiểm định ứng dụng và đưa ứng dụng lên Apple Store 4](#_Toc521059554)

[2.4 GIỚI THIỆU NGÔN NGỮ LẬP TRÌNH SWIFT 4](#_Toc521059555)

[2.5 BẢO MẬT 5](#_Toc521059556)

[2.5.1 Bảo mật hệ thống (System security) 5](#_Toc521059557)

[2.5.2 Mã hoá và bảo vệ dữ liệu (Encryption and data protection) 5](#_Toc521059558)

[2.5.3 Bảo mật ứng dụng(App security) 6](#_Toc521059559)

[2.5.4 Bảo mật mạng (Network security) và các dịch vụ sử dụng kết nối mạng (Internet services) 7](#_Toc521059560)

[2.5.5 Bảo mật điều khiển thiết bị (Device controls) 7](#_Toc521059561)

[2.6 GIỚI THIỆU VỀ MAP KIT 8](#_Toc521059562)

[CHƯƠNG 3: PHÂN TÍCH HỆ THỐNG THÔNG TIN 9](#_Toc521059563)

[3.1 LƯỢC ĐỒ USE CASE 9](#_Toc521059564)

CHƯƠNG 4: GIỚI THIỆU SẢN PHẨM MINH HỌA CỦA ĐỀ TÀI 9

[4.1 Màn hình trang chủ 10](#_Toc521059565)

[4.2 Màn hình Của tôi. 12](#_Toc521059575)

[4.3 Màn hình Đăng Nhập 13](#_Toc521059577)

[4.4 Màn hình Đăng Ký 15](#_Toc521059578)

[4.5 Màn hình Nhập số điện thoại (Đăng tin) 16](#_Toc521059579)

[4.6 Màn hình Loại (Đăng tin) 17](#_Toc521059580)

[4.7 Màn hình Diện tích & Gía (Đăng tin) 18](#_Toc521059582)

[4.8 Màn hình Hình ảnh (Đăng tin) 19](#_Toc521059583)

[CHƯƠNG 5: KẾT LUẬN, HƯỚNG PHÁT TRIỂN CỦA ĐỀ TÀI 21](#_Toc521059584)

[5.1 KẾT LUẬN 21](#_Toc521059585)

[5.1.1 Kết quả đạt được 21](#_Toc521059586)

[5.1.2 Hạn chế 21](#_Toc521059587)

[5.1.3 Hướng khắc phục các hạn chế : 21](#_Toc521059588)

[5.1.4 Hướng mở rộng đề tài. 21](#_Toc521059589)

[DANH MỤC TÀI LIỆU THAM KHẢO 22](#_Toc521059590)

**DANH MỤC CÁC BẢNG, SƠ ĐỒ, HÌNH**

**KÍ HIỆU CÁC CỤM TỪ VIẾT TẮT**

**CHƯƠNG 1 : GIỚI THIỆU ĐỀ TÀI**

## Giới thiệu

Find Home là ứng dụng dành cho mọi người, giúp cho người muốn cho thuê nhà dễ dàng tiếp cận tới người có nhu cầu, còn người có nhu cầu thuê nhà dễ dàng tìm kiếm phòng trọ/nhà phù hợp cho bản thân, phù hợp với hoàn cảnh. - Find Home là phần mềm tiện ích, giúp cho mọi người tiết kiệm được thời gian đi lại xem căn nha, liên lạc chủ nhà hỏi tình hình, giá cả… Chỉ cần có 1 chiếc smartphone chạy hệ diều hành iOS người dung có thể ngồi tại nhà để tìm kiếm phòng trọ/nhà, xem mô tả, hình ảnh, diện tích, địa chỉ, giá cả của phòng trọ/nhà ở một cách nhanh chóng, chân thật. Tiết kiệm chi phí đi lại đáng kể.

## 1.2 Mục tiêu

### Phát triển ứng dụng cung cấp dịch vụ tìm kiếm nhà trọ - Find Home bằng ngôn ngữ Swift.

### Biết được các bước, quy trình phát triển một ứng dụng trên nền tảng iOS.

### Biết được cơ chế bảo mật của iOS.

### Biết được cách sử dụng bản đồ, định vị trên ứng dụng iOS

### - Hoàn thành ứng dụng Find Home.

## 1.3 Lĩnh vực

* Chuyên ngành: công nghệ phần mềm
* Chuyên môn: lập trình ứng dụng di động. Sử dụng ngôn ngữ Swift xây dựng ứng Find Home bằng công cụ Xcode trên nền tảng IOS.

### 1.4 Nền tảng được sử dụng

* Hệ điều hành Mac OS.
* Công cụ lập trình Xcode.

**1.5 Nội dung trình bày đề tài**

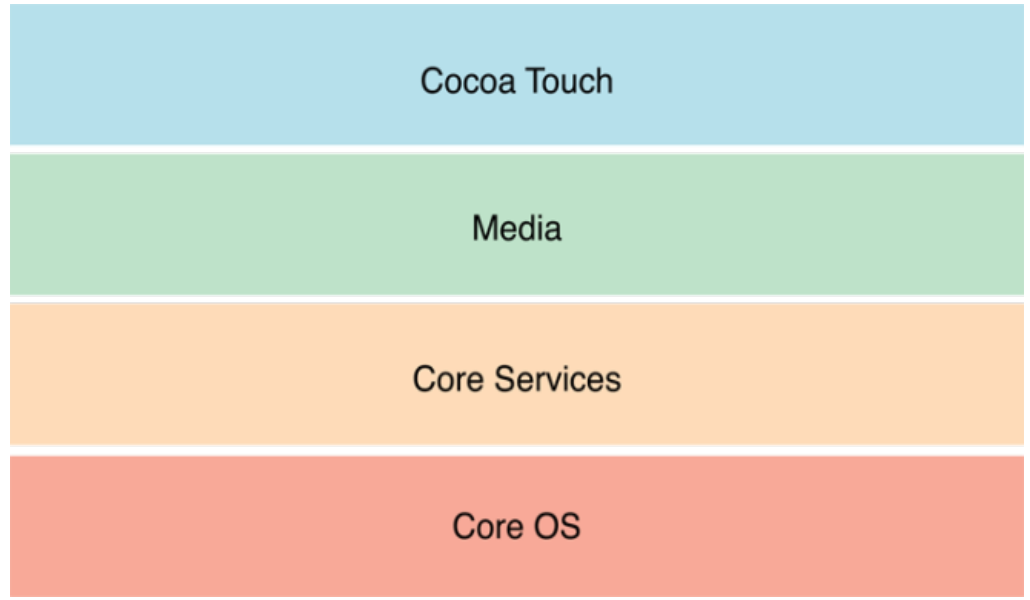
* Chương 1: Giới thiệu đề tài
* Chương 2: Cơ sở lý thuyết đề tài
* Chương 3: Phân tích hệ thống thông tin.
* Chương 4: Giới thiệu sản phẩm minh hoạ.
* Chương 5: Kết luận, hướng phát triển của đề tài

**CHƯƠNG 2: CƠ SỞ LÝ THUYẾT CỦA ĐỀ TÀI**

* 1. **NỀN TẢNG CÔNG NGHỆ**
* 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.
* 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 11.4 được Apple ra mắt vào ngày 30/5/2018. Hiện này đã có bản 12 Beta (bản dung thử)

**2.2 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.



**2.2.1 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 như: Map Kit, UIKit

**2.2.2 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...

**2.2.3 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, điên toán đám mây...

**2.2.4 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.

**2.3 QUI 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

**2.3.1 Thiết kế giao diện**

* Trong thiết kế giao diện ứng dụng iOS, View được xem là thành phần quan trọng nhất, cần lên trước kế hoạch các đối tượng view sẽ sử dụng, vì nó ảnh hưởng toàn bộ quá trình phát triển về sau. View cho phép cấu thành nên tất cả các thuộc tính một màn hình giao diện cần có.
* 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 | Nhập dữ liệu được nhập từ người dung | TextField, 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, AlertView |

**2.3.2 Tổ chức dữ liệu**

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

VD: Một sinh viên bao gồm các thuộc tính: Tên, mã sinh viên, lớp, ngày tháng năm sinh, giới tính…

**2.3.4 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.
* Kết nối IBOutlet là một dạng khai báo một đối tượgn view trở thành thuộc tính trong lớp quản lý View Controller. Tất cả các đối tượng view đều có thể kêt nối IBOutlet
* Kết nối IBAction là một loại kết nối hỗ trợ tạo ra phương thúc 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…

**2.3.5 Kiểm định ứng dụng và đưa ứng dụng lên Apple Store**

**2.4 GIỚI THIỆU NGÔN NGỮ LẬP TRÌNH SWIFT**

* Swift là ngôn ngữ lập trình mới được phát triển bởi Apple Inc với mục đích hỗ trợ lập trình viên trong việc phát triển các ứng dụng chạy trên các hệ điều hành như iOS, OSX và watchOS. [Swift][1] được xây dựng dựa trên việc kế thừa các tính năng của 2 ngôn ngữ C và Objective-C.
* Ưu điểm của ngôn ngữ lập trình Swift
* Code ngắn gọn, dễ đọc. Theo như Apple, 3 dòng code của Ojective C có thể được gói gọn trong 1 dòng code của Swift, vì vậy, việc viết ứng dụng sẽ tiết kiệm thời gian hơn, lỗi xuất hiện ít hơn và tang hiệu quả cho quá trình bảo trì hay dễ dàng vá lỗi Swift trong tương lai.
* Việc không cần chạy các chương trình Copliler và các chương trình Test quá thường xuyên sẽ giúp nhà phát triển tiết kiệm phần cứng.
* Giống như một phiên bản đơn giản của Ojective-C, Swift có các lớp đối tượng được tối ưu nhất với các dòng mã được sắp xếp một cách hợp lý, ở đó, các nhà phả triển sẽ có một môi trường Runtime nên rất quen thuộc và dễ dàng quản lý.
* Tránh sự cố tràn bộ nhớ là điều tuyệt vời mà Swift đem lại? tại sao ư, đó là vì XCode Runtime được thực hiện trong thời gian thực, tức là khi lập trình viên viết một dòng mã và nhận kết quả trong thời gian thực, hình ảnh có thể được xem trong IDE, ngoài ra Xcode cũng sẽ dựng các kịch bản thực hiện trong các ứng dụng nên lập trình viên có thể quản lý ứng dụng tốt hơn.
* Dựa trên nền tảng Ojective-C, vì vậy các ứng dụng dùng Ojective-C có thể được cập nhật bằng Swift, tức là các công cụ hữu ích của Ojective –C có thể được sử dụng ngay trên Swift.

**2.5 BẢO MẬT**

Theo hướng dẫn bảo mật của Apple, bảo mật trên IOS có thể được thực hiện qua các lớp:

* Bảo mật hệ thống(System security).
* Mã hoá và bảo vệ dữ liệu(Encryption and data protection).
* Bảo mật ứng dụng(App security).
* Bảo mật mạng (Network security) và các dịch vụ sử dụng kết nối mạng (Internet services).
* Bảo mật điều khiển thiết bị (Device controls)

**2.5.1 Bảo mật hệ thống (System security)**

* Bảo mật hệ thống được thiết kế sao cho cả phần mềm và phần cứng đều an toàn trên tất cả các thành phần cốt lõi của tất cả thiết bị iOS. Điều này bao gồm quá trình khởi động, cập nhật phần mềm và "Secure Enclave". Kiến trúc này là trung tâm của bảo mật trong iOS và không bao giờ bị cản trở khả năng sử dụng của thiết bị.
* Việc tích hợp chặt chẽ phần cứng, phần mềm và dịch vụ trên thiết bị iOS đảm bảo rằng mỗi thành phần của hệ thống được tin cậy và xác thực toàn bộ hệ thống. Từ bản cập nhật phần mềm khởi động cho iOS cho các ứng dụng của bên thứ ba, mỗi bước được phân tích và xem xét để giúp đảm bảo rằng phần cứng và phần mềm đang hoạt động tối ưu cùng nhau và sử dụng tài nguyên đúng cách

**2.5.2 Mã hoá và bảo vệ dữ liệu (Encryption and data protection)**

* Chuỗi khởi động an toàn, chữ ký trong các đoạn mã và bảo mật quy trình thời gian chạy tất cả đều giúp đảm bảo rằng chỉ có mã và ứng dụng đáng tin cậy mới có thể chạy trên thiết bị. iOS có các tính năng mã hóa và bảo vệ dữ liệu bổ sung để bảo vệ dữ liệu người dùng, ngay cả trong trường hợp các phần khác của cơ sở hạ tầng bảo mật bị xâm phạm (ví dụ: trên thiết bị có sửa đổi trái phép). Điều này mang lại lợi ích quan trọng cho cả người dùng và quản trị viên hệ thống, bảo vệ thông tin cá nhân và công ty mọi lúc và cung cấp phương pháp xóa từ xa ngay lập tức và hoàn toàn trong trường hợp mất cắp thiết bị.
* Ngoài các tính năng mã hóa phần cứng được tích hợp vào thiết bị iOS, Apple còn sử dụng công nghệ gọi là Data Protection để bảo vệ dữ liệu được lưu trữ trong bộ nhớ flash trên thiết bị. Bảo vệ dữ liệu cho phép thiết bị phản hồi các sự kiện phổ biến như cuộc gọi điện thoại đến, nhưng cũng cho phép mức mã hóa cao cho dữ liệu người dùng. Các ứng dụng chính của hệ thống như tin nhắn, lịch, thư, danh sách liên hệ, ảnh và dữ liệu sức khoẻ sử dụng phương thức bảo vệ dữ liệu theo mặc định và các ứng dụng của bên thứ 3 được cài đặt trên iOS 7 trở lên sẽ tự động nhận được sự bảo vệ này.
* Bảo vệ dữ liệu được thực hiện bằng cách xây dựng và quản lý một hệ thống phân cấp các khóa, và xây dựng trên các công nghệ mã hóa phần cứng được tích hợp vào mỗi thiết bị iOS. Bảo vệ dữ liệu được kiểm soát trên cơ sở từng tệp bằng cách gán từng tệp cho một lớp; khả năng truy cập được xác định bởi các khóa lớp đã được mở khóa hay chưa. Với sự ra đời của Hệ thống tệp Apple (APFS), hệ thống tệp hiện có thể phân chia thêm các khóa thành một cơ sở theo mức độ (các phần của tệp có thể có các khóa khác nhau).

**2.5.3 Bảo mật ứng dụng(App security)**

* Ứng dụng là một trong những yếu tố quan trọng nhất của kiến trúc bảo mật di động hiện đại. Mặc dù các ứng dụng cung cấp các lợi ích về năng suất tuyệt vời cho người dùng, nhưng chúng cũng có khả năng tác động tiêu cực đến bảo mật hệ thống, tính ổn định và dữ liệu người dùng nếu chúng không được xử lý đúng cách. Do đó, iOS cung cấp các lớp bảo vệ để đảm bảo rằng các ứng dụng được ký và xác minh, và được đóng gói để bảo vệ dữ liệu người dùng. Những yếu tố này cung cấp nền tảng ổn định, an toàn cho ứng dụng, cho phép hàng nghìn nhà phát triển phân phối hàng trăm nghìn ứng dụng trên iOS mà không ảnh hưởng đến tính toàn vẹn của hệ thống. Và người dùng có thể truy cập các ứng dụng này trên thiết bị iOS của họ mà không sợ vi rút, phần mềm độc hại hoặc các cuộc tấn công trái phép.
* Bảo mật dữ liệu trong ứng dụng IOS
* Bộ phát triển phần mềm iOS (SDK) cung cấp bộ API đầy đủ giúp nhà phát triển bên thứ ba và trong nhà dễ dàng sử dụng Bảo vệ dữ liệu và giúp đảm bảo mức độ bảo vệ cao nhất trong ứng dụng của họ. Bảo vệ dữ liệu có sẵn cho các tệp API và cơ sở dữ liệu, bao gồm NSFileManager, CoreData, NSData và SQLite. Cơ sở dữ liệu ứng dụng Mail (bao gồm tệp đính kèm), sách được quản lý, dấu trang Safari, hình ảnh khởi chạy ứng dụng và dữ liệu vị trí cũng được mã hóa bằng các khóa được bảo vệ bằng mật mã của người dùng trên thiết bị của họ. Lịch (không bao gồm tệp đính kèm), Danh bạ, Lời nhắc, Ghi chú, Tin nhắn và Ảnh triển khai Được bảo vệ cho đến khi xác thực người dùng đầu tiên. Các ứng dụng do người dùng cài đặt không chọn tham gia vào lớp Bảo vệ dữ liệu cụ thể sẽ nhận được Protected Until First User Authentication theo mặc định.
* Hầu hết các ứng dụng IOS hiện nay sử dụng phương thức lưu trữ đám mây trên internet, các ứng dụng di động thao tác với người dùng qua một remote server, điều này dẫn tới việc mất an toàn bảo mật khi sử dụng ứng dụng.
* Để khắc phục và đảm bảo vấn đề bảo mật Apple đã đưa ra khái niệm "App Transport Security". Khi ứng dụng sử dụng các yêu cầu liên quan đến truy vấn dữ liệu thông qua kết nối internet, "App Transport Security" sẽ kích hoạt buộc ứng dụng gửi truy vấn qua một kết nối bảo mật mặc định qua giao thức HTTPS nhằm đảm bảo thông tin của người dùng sẽ hạn chế tối đa bị tấn công.
* Ngoài ra, khi phát triển ứng dụng IOS cũng cần nên cân nhắc sử dụng các thư viện mã nguồn mở, việc lựa chọn thư viện phù hợp sẽ giúp tăng cường tính bảo mật trên ứng dụng, ngược lại sẽ gây nên các lỗ hổng trong ứng dụng khiến cho ứng dụng trở nên mất an toàn và thông tin của người dùng có thể bị khai thác một cách dễ dàng.

**2.5.4 Bảo mật mạng (Network security) và các dịch vụ sử dụng kết nối mạng (Internet services)**

* Ngoài các biện pháp bảo vệ tích hợp mà Apple sử dụng để bảo vệ dữ liệu được lưu trữ trên thiết bị iOS, có nhiều biện pháp bảo mật mạng mà tổ chức có thể thực hiện để giữ an toàn cho thông tin khi nó di chuyển đến và đi từ thiết bị iOS. Người dùng thiết bị di động phải có khả năng truy cập mạng công ty từ mọi nơi trên thế giới, vì vậy điều quan trọng là phải đảm bảo rằng họ được ủy quyền và dữ liệu của họ được bảo vệ trong quá trình truyền. iOS sử dụng — và cung cấp quyền truy cập của nhà phát triển cho các giao thức mạng chuẩn cho các giao tiếp được xác thực, ủy quyền và được mã hóa. Để thực hiện các mục tiêu bảo mật này, iOS tích hợp các công nghệ đã được chứng minh và các tiêu chuẩn mới nhất cho cả kết nối mạng Wi-Fi và dữ liệu di động. Trên các nền tảng khác, phần mềm tường lửa là cần thiết để bảo vệ các cổng giao tiếp mở chống lại sự xâm nhập. Vì iOS đạt được bề mặt tấn công giảm bằng cách hạn chế các cổng nghe và loại bỏ các tiện ích mạng không cần thiết như telnet, shell hoặc máy chủ web, không cần phần mềm tường lửa bổ sung trên thiết bị iOS.
* Apple đã xây dựng một bộ dịch vụ mạnh mẽ để giúp người dùng có được nhiều tiện ích và năng suất hơn nữa từ thiết bị của họ, bao gồm iMessage, FaceTime, Đề xuất Siri, iCloud, iCloud Backup và iCloud Keychain. Các dịch vụ Internet này đã được xây dựng với cùng mục tiêu bảo mật mà iOS quảng bá trên toàn bộ nền tảng. Những mục tiêu này bao gồm xử lý dữ liệu an toàn, cho dù ở phần còn lại trên thiết bị hoặc chuyển tiếp qua mạng không dây; bảo vệ thông tin cá nhân của người dùng; và bảo vệ mối đe dọa chống lại việc truy cập thông tin và dịch vụ độc hại hoặc trái phép. Mỗi dịch vụ sử dụng kiến trúc bảo mật mạnh mẽ của riêng mình mà không ảnh hưởng đến tính dễ sử dụng của iOS.

**2.5.5 Bảo mật điều khiển thiết bị (Device controls)**

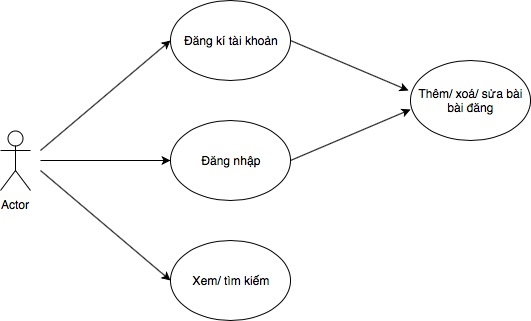
* Theo mặc định, mật mã của người dùng có thể được định nghĩa là mã PIN số. Trên các thiết bị có ID Touch hoặc Face ID, độ dài mật mã tối thiểu là sáu chữ số. Trên các thiết bị khác, độ dài tối thiểu là bốn chữ số. Người dùng có thể chỉ định mật mã chữ số dài hơn bằng cách chọn Tùy chỉnh
* Mã chữ và số trong tùy chọn mật mã trong Cài đặt> Mật khẩu. Mật mã dài hơn và phức tạp hơn khó đoán hoặc tấn công hơn và được khuyến nghị.
* Quản trị viên có thể thực thi các yêu cầu mật mã phức tạp và các chính sách khác bằng MDM hoặc Exchange ActiveSync hoặc bằng cách yêu cầu người dùng cài đặt cấu hình cấu hình theo cách thủ công.
* iOS hỗ trợ MDM cho phép các doanh nghiệp định cấu hình và quản lý an toàn iPhone, iPad, Apple TV và triển khai Mac trên các tổ chức của họ. Khả năng MDM được xây dựng trên các công nghệ iOS hiện có như cấu hình cấu hình, đăng ký qua mạng và dịch vụ Thông báo đẩy của Apple. Ví dụ, APN được sử dụng để đánh thức thiết bị để nó có thể giao tiếp trực tiếp với giải pháp MDM của nó qua kết nối bảo mật. Không có thông tin bí mật hoặc độc quyền nào được truyền qua APN.`

**2.6 GIỚI THIỆU VỀ MAP KIT**

* MapKit là thư viện của iOS dùng để hỗ trợ về bản đồ cho các thiết bị iOS bao gồm tổng quan trên ban đồ. Các thư viện hỗ trợ tìm kiếm, tìm đường đi,... trên bản đồ.
* MapKit Framework được xây dựng dựa trên data và APIs của Apple Maps, cung cấp cho các iOS developers một cơ chế đơn giản để tích hợp Maps trong các ứng dụng.

## CHƯƠNG 3: PHÂN TÍCH HỆ THỐNG THÔNG TIN

**3.1 LƯỢC ĐỒ USE CASE**



**CHƯƠNG 4: GIỚI THIỆU SẢN PHẨM MINH HỌA CỦA ĐỀ TÀI:**

### 4.1 Màn hình trang chủ

### Trang chủ: là màn hình chưa toàn bộ bài đăng của các tài khoản khác nhau và được sắp xếp theo ngày đăng giúp cho người đọc tin dễ dàng biết được khoản thời gian nào là hợp lí.

### Navigation Bar: Hiển thị tên trang “Trang chủ”

### Textfield search: cho phép người dùng nhập nội dùng tìm kiếm như. Chức năng này sẽ tìm kiếm dựa trên tiêu đề, địa chỉ(quận, thành phố), giá, và loại.

### Ngoài ra còn có các Button cho phép người dùng tìm kiếm dựa vào loại, địa chỉ và giá cả.

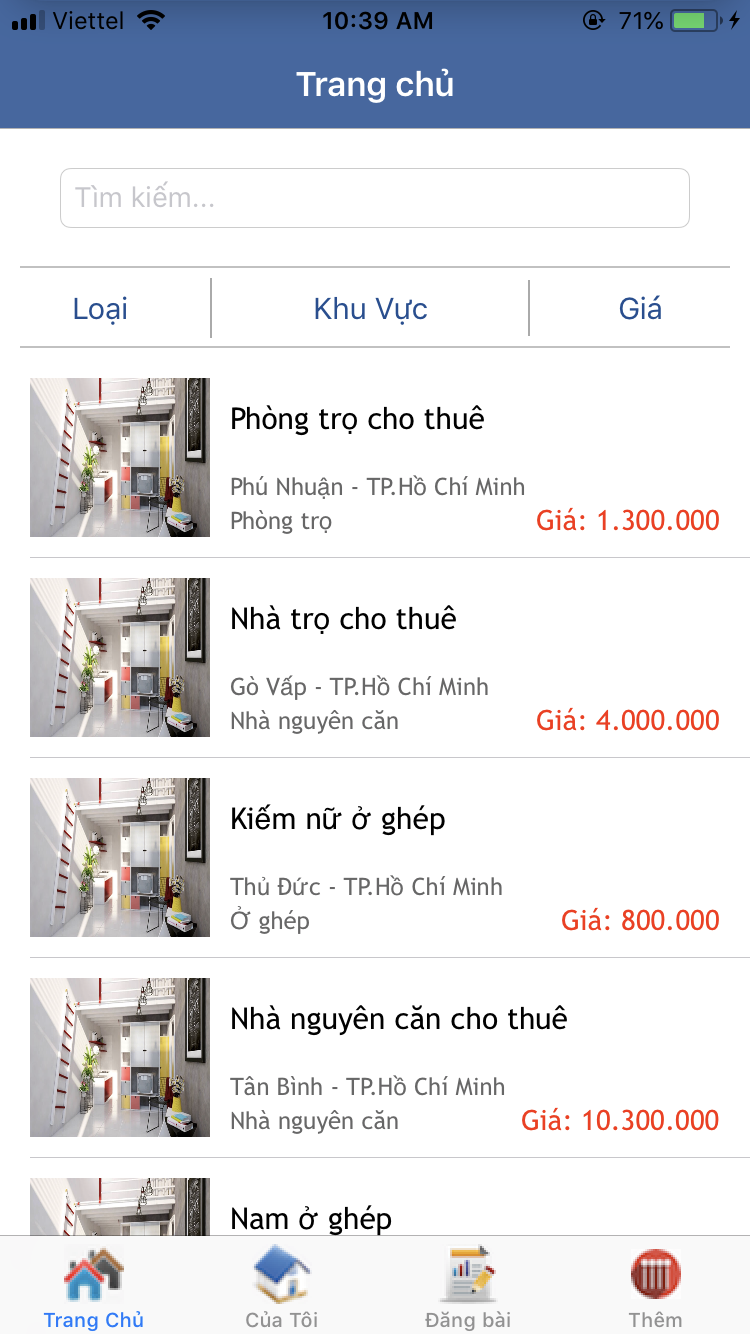
### Dưới dung màn hình là 1 tab bar bao gồm các Button:

### Trang chủ: Hiển thị màn hình Trang chủ

### Của tôi: Hiển thị màn hình chưa các bài đăng là User đó đã đăng,

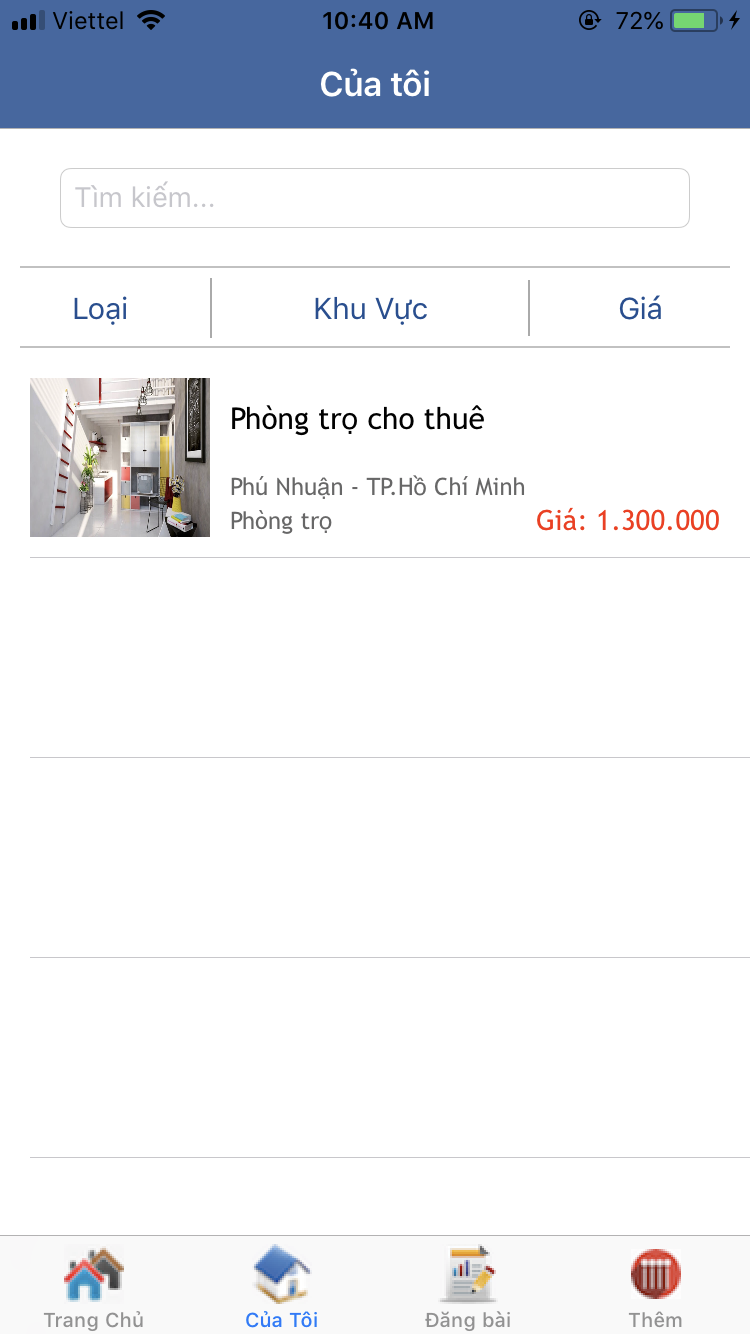
### Đăng tin: Bắt đầu quá trình đăng bài

### Thêm: Hiển thị màn hình thêm gồm các chức năng mở rộng

****

### 4.2 Màn hình Của tôi.

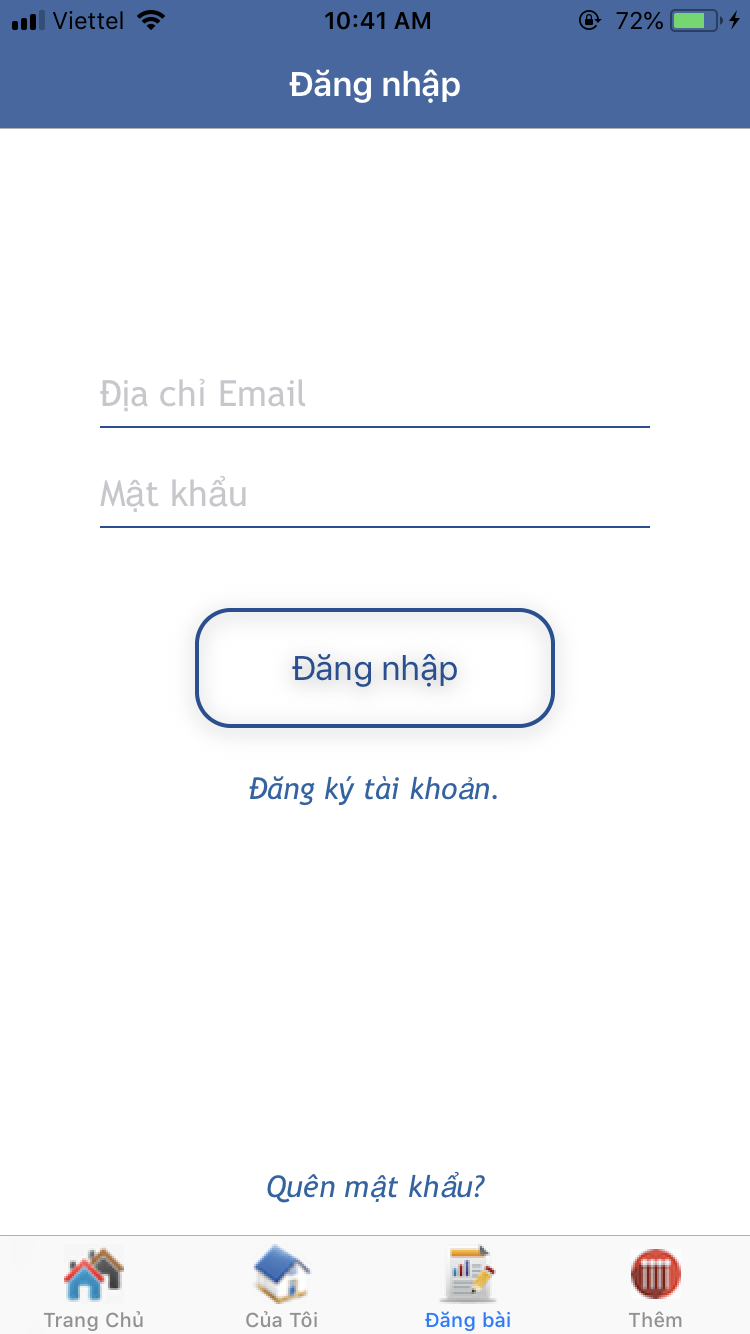
### - Có thiết kế tương tự như màn hình Trang chủ, nhưng màn hình này sẽ hiển thị lịch sử bài đăng của User.



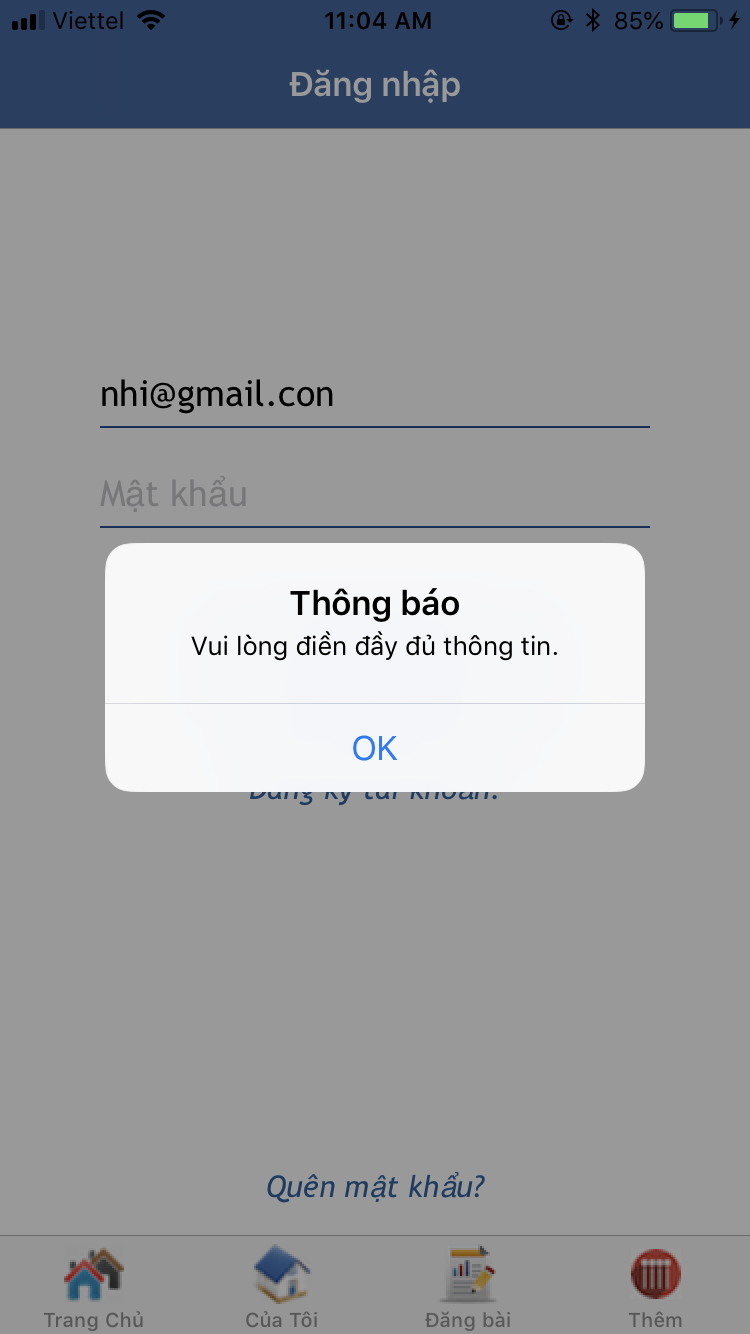
**4.3 Màn hình Đăng Nhập**

- Khi nhấn vào button Đăng tin trên Tabbar, nếu người dùng chưa đăng nhập thì ứng dụng sẽ hiển thị màn hình Đăng nhập.

- Nếu chưa có tài khoản người dùng sẽ đăng kí bằng kí chạm bào button Đăng kí tài khoản.

****

*Màn hình đăng nhập*



*Báo lỗi khi người dung chưa nhập đầy đủ thông tin.*

**4.4 Màn hình Đăng Ký**

- Màn hình bao gồm các textfield người dùng phải nhập đầy đủ thông tin như:

* Số điện thoại.
* Họ và tên.
* Mật khẩu lần 1
* Mật khẩu lần 2



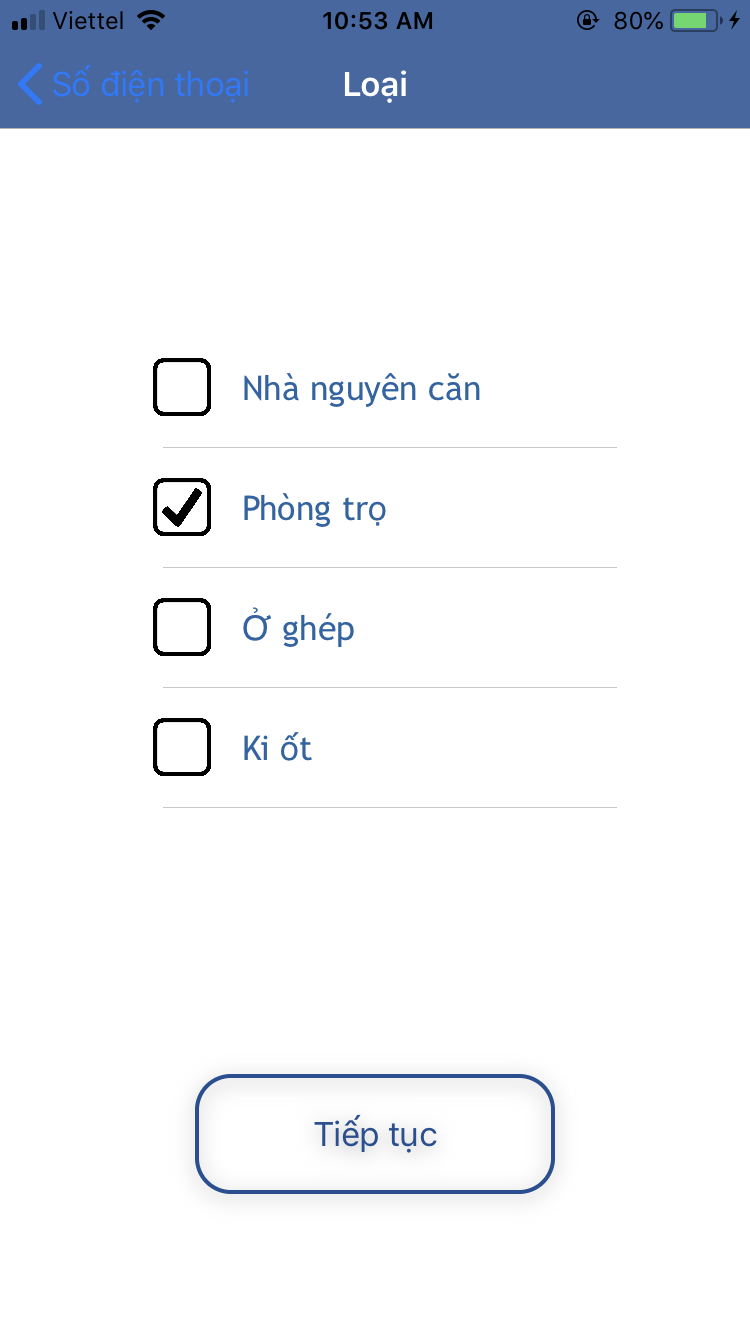
*Màn hình đăng ký*

**4.5 Màn hình Nhập số điện thoại (Đăng tin)**

- Nhập số điện thoại

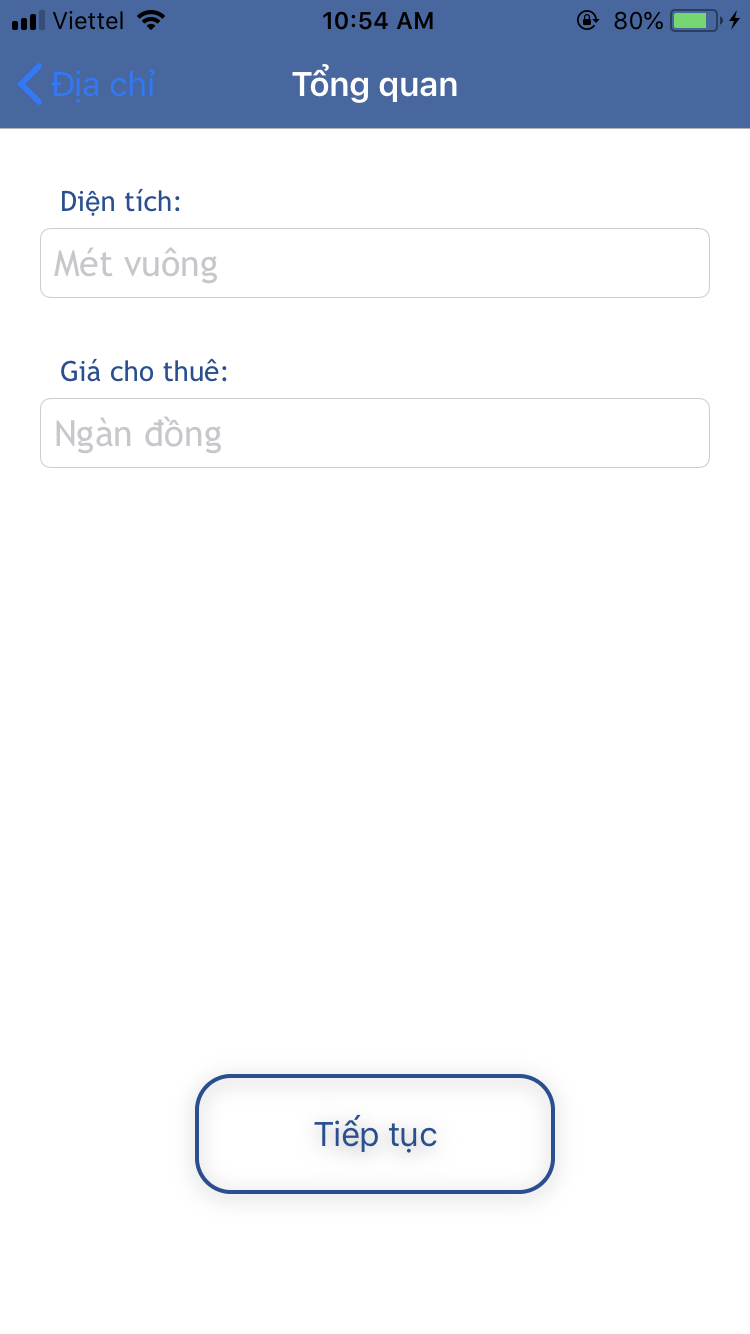
****

**4.6 Màn hình Loại (Đăng tin)**

****

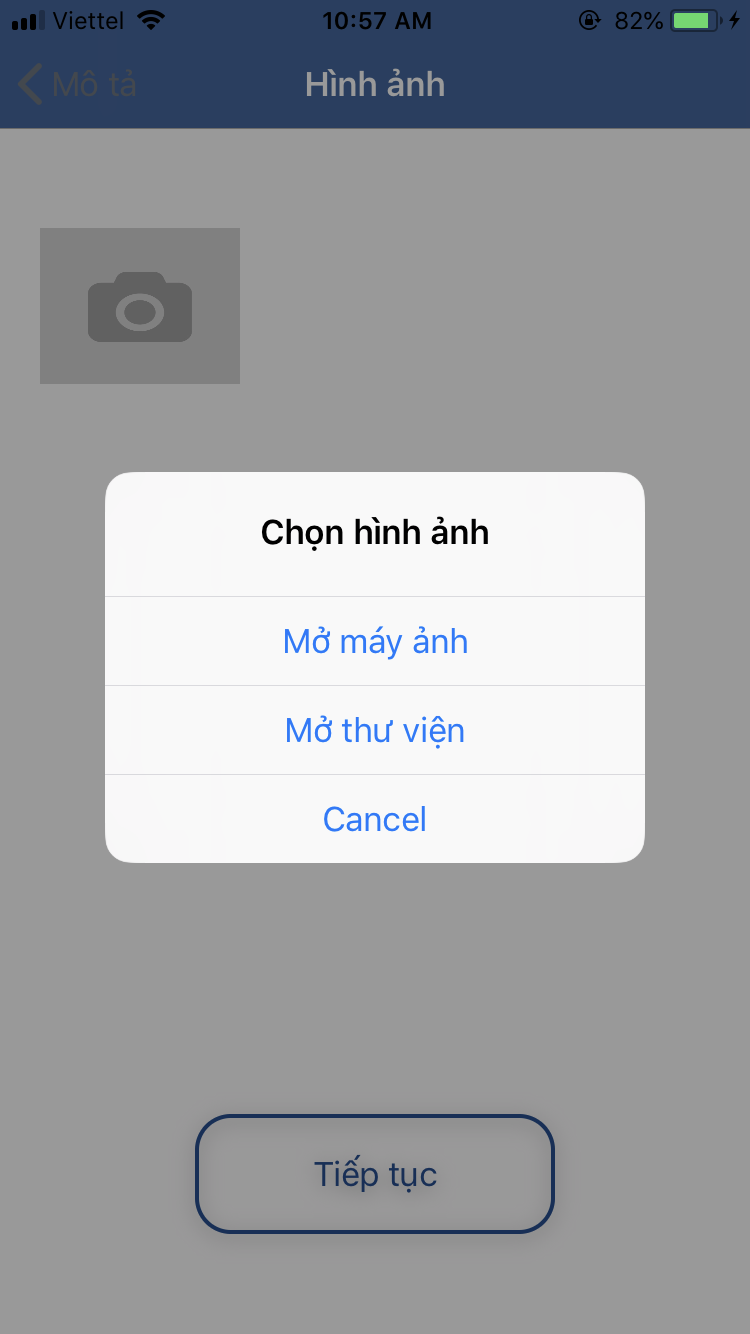
# *Màn hình loại*

**4.7 Màn hình Diện tích & Gía (Đăng tin)**



*Màn hình điềngdiện tích và giá.*

**4.8 Màn hình Hình ảnh (Đăng tin)**

**

*Chọn hình ảnh từ Máy ảnh, thư viện*

**

*Màn hình chi tiết*

**CHƯƠNG 5: KẾT LUẬN, HƯỚNG PHÁT TRIỂN CỦA ĐỀ TÀI**

**5.1 KẾT LUẬN**

**5.1.1 Kết quả đạt được**

- Hiểu được qui trình, phát triển ứng dụng trên nền tảng iOS

- Hiểu và sử dụng được ngôn ngữ lập trình Swift.

- Biết được các cơ chế bảo mật của iOS

- Các chức năng đã được đưa vào app:

+ Đăng kí/ đăng nhập tài khoản

+ Đăng tin

+ Quản lý bài đăng

+ Quản lý thông tin người dung.

**5.1.2 Hạn chế**

- Chưa hiểu hết được bảo mật của nền tảng iOS.

- Chưa đăng được bài đăng(sẽ hoàn thiện trong 1 vài ngày tới).

- Chưa xem được bản đồ dựa trên địa chỉ.

**5.1.3 Hướng khắc phục các hạn chế :**

- Tìm hiểu và đọc nhiều hơn các vấn đề liên quan tới bảo mật của iOS.

- Tìm hiểu nhiều hơn về cách tổ chức dữ liệu của Database.

- Tìm hiểu cách sử dụng MapKit

**5.1.4 Hướng mở rộng đề tài.**

- Dùng GPS và MapKit cho phép người dung thấy được những địa điểm cho thuê gần mình nhất.

- Người chủ nhà quản lí được thông tin của ngừoi thuê nhà (Thông tin người thuê, giá tiền hằng tháng bao gồm tiền nhà và điện nước, nếu là dãy trọ thì quản lí được tình hình dãy trọ đã cho thuê những phòng nào và còn những phòng nào)

- Thêm chức năng đẩy tin giúp bài đăng tiếp cận được nhiều người xem hơn, việc tìm kiếm khách hạn có nhu cầu thuê nhà dễ dàng hơn.

**DANH MỤC TÀI LIỆU THAM KHẢO**

* Giáo trình Lập trình iOS trung tâm tin học KHTN .
* https://developer.apple.com
* https://github.com
* http://stackoverflow.com
* https://firebase.google.com