

TRƯỜNG ĐẠI HỌC LẠC HỒNG  
KHOA CÔNG NGHỆ THÔNG TIN

-----&📖&-----

# BÁO CÁO

## NGHIÊN CỨU KHOA HỌC

ĐỀ TÀI:

XÂY DỰNG HỆ THỐNG TÌM KIẾM ĐỊA ĐIỂM  
ĂN UỐNG TRÊN ANDROID

NGUYỄN VŨ HÙNG

BIÊN HÒA, THÁNG 11/2015

TRƯỜNG ĐẠI HỌC LẠC HỒNG  
KHOA CÔNG NGHỆ THÔNG TIN

-----📖-----

# BÁO CÁO

## NGHIÊN CỨU KHOA HỌC

ĐỀ TÀI:

**XÂY DỰNG HỆ THỐNG TÌM KIẾM ĐỊA ĐIỂM  
ĂN UỐNG TRÊN ANDROID**

Sinh viên thực hiện: NGUYỄN VŨ HƯNG  
Giáo viên hướng dẫn: VŨ ĐÌNH TRUNG

BIÊN HÒA, THÁNG 11/2015

## LỜI CẢM ƠN

*Tôi xin gửi lời cảm ơn chân thành và biết ơn sâu sắc tới ThS. Vũ Đình Trung, người thầy đã tận tình chỉ bảo và hướng dẫn cho tôi trong suốt quá trình thực hiện bài nghiên cứu khoa học này.*

*Tôi xin chân thành cảm ơn sự giúp đỡ, góp ý nhiệt tình của quý thầy cô trong Khoa Công nghệ thông tin Trường Đại học Lạc Hồng đã tạo điều kiện thuận lợi cho tôi để học tập, nghiên cứu và thực hiện được đề tài này.*

*Mặc dù tôi đã cố gắng hoàn thành tốt đề tài nhưng cũng không thể tránh khỏi những sai sót nhất định, rất mong được sự thông cảm của quý Thầy Cô.*

*Cuối cùng, tôi xin gửi lời cảm ơn tới gia đình, người thân và bạn bè, những người luôn ở cạnh tôi những lúc thử thách nhất, luôn luôn động viên, khuyến khích tôi nỗ lực hơn nữa trong cuộc sống và trong công việc.*

*Tôi xin chân thành cảm ơn xin gửi lời chúc sức khỏe và thành đạt tới tất cả quý thầy cô cùng các bạn!*

*Biên Hòa, ngày ... tháng... năm 2015*

*Nguyễn Vũ Hưng*

## MỤC LỤC

### Contents

LỜI CẢM ƠN .....	2
MỤC LỤC.....	3
DANH MỤC CÁC TỪ VIẾT TẮT .....	5
DANH MỤC HÌNH.....	6
DANH MỤC BẢNG.....	8
DANH MỤC BIỂU ĐỒ .....	9
PHẦN MỞ ĐẦU .....	1
CHƯƠNG 1: TỔNG QUAN.....	4
1.1 Đặt vấn đề.....	4
1.2 Tình hình nghiên cứu trong và ngoài nước .....	5
1.3 Khảo sát nhu cầu người dùng .....	7
1.4 Những tính năng đề xuất .....	10
1.5 Những yêu cầu của ứng dụng .....	11
CHƯƠNG 2: PHÂN TÍCH, THIẾT KẾ VÀ XÂY DỰNG HỆ THỐNG	12
2.1 Phân tích và thiết kế hệ thống.....	12
2.1.1 Mô hình Client – Server .....	12
2.1.2 Giao tiếp giữa Client và Server .....	13
2.1.3 Cơ sở dữ liệu.....	15
2.1.4 Lưu trữ hoạt động của người dùng.....	20
2.1.5 Thiết kế cấu trúc ứng dụng.....	20
2.1.6 Lưu trữ dữ liệu ở ứng dụng .....	23
2.1.7 Xác định vị trí và dẫn đường .....	23

2.2	Biểu đồ và kiến trúc hệ thống .....	24
2.2.1	Các biểu đồ hệ thống.....	24
2.3	Tìm hiểu công nghệ.....	29
2.3.1	Node.js .....	29
2.3.2	MongoDB .....	31
2.3.3	Amazon Web Services.....	33
2.3.4	Hệ điều hành Android.....	38
2.3.5	Google Map.....	40
2.3.6	Spring Framework.....	41
2.4	Xây dựng hệ thống .....	44
2.4.1	Tương thích đa dạng thiết bị .....	44
2.4.2	Xử lý và hiển thị hình ảnh.....	44
2.4.3	Hiện thực hệ thống .....	45
2.5	Triển khai server.....	62
CHƯƠNG 3: KẾT LUẬN VÀ ĐỀ NGHỊ.....		64
3.1	Kết quả đạt được và nhận xét.....	64
3.2	Hạn chế .....	64
3.3	Hướng phát triển.....	65
TÀI LIỆU THAM KHẢO .....		67

## **DANH MỤC CÁC TỪ VIẾT TẮT**

NPM	:	Node.js Package Manager
CRUD	:	Create, Read, Update, Delete
TCP	:	Transmission Control Protocol
HTTP	:	HyperText Transfer Protocol
HTML	:	HyperText Markup Language
XML	:	eXtensible Markup Language
JSON	:	Javascript Object Notation
NoSQL	:	Not Only SQL
CSDL	:	Cơ sở dữ liệu
EC2	:	Amazon Elastic Compute Cloud
POJOs	:	Plain Old Java Object
JPG	:	Joint Photographic Experts Group
DI	:	Dependency Injection
AOP	:	Aspect-Oriented Programming
APK	:	Android Application Package
I/O	:	Input/Output
SDK	:	Software Development Kit
AWS	:	Amazon Web Services
GPS	:	Global Positioning System
MVC	:	Model View Controller
RDBMS	:	Relational Database Management System
CPU	:	Central Processing Unit
SSH	:	Secure Shell
AMIS	:	Amazon Machine Images
IoC	:	Inverse of Control
IP	:	Internet Protocol
VPCs	:	Virtual Private Clouds

## DANH MỤC HÌNH

Hình 2.1 Mô hình Client – Server. ....	12
Hình 2.2 Sơ đồ lớp của tập tài liệu Locs. ....	16
Hình 2.3 Sơ đồ lớp của tập tài liệu News. ....	18
Hình 2.4 Sơ đồ lớp của tập tài liệu Orders. ....	18
Hình 2.5 Sơ đồ lớp của tập tài liệu Owners. ....	19
Hình 2.6 Sơ đồ thiết kế giao diện ứng dụng. ....	21
Hình 2.7 Sơ đồ use case của người dùng. ....	25
Hình 2.8 Sơ đồ use case của Admin. ....	25
Hình 2.9 Biểu đồ lớp Shop. ....	26
Hình 2.10 Biểu đồ sequence chức năng Search. ....	26
Hình 2.11 Biểu đồ sequence chức năng viết bình luận. ....	27
Hình 2.12 Biểu đồ sequence chức năng xem bình luận. ....	27
Hình 2.13 Biểu đồ sequence chức năng liệt kê địa điểm. ....	28
Hình 2.14 Biểu đồ sequence dẫn đường đến một địa điểm. ....	29
Hình 2.15 Cấu trúc trong MongoDB. ....	33
Hình 2.16 Cấu trúc các tầng Amazon Web Services. ....	35
Hình 2.17 Sơ đồ hoạt động Amazon EC2. ....	36
Hình 2.18 Giao diện tab News. ....	46
Hình 2.19 Giao diện tab Search. ....	47
Hình 2.20 Giao diện tab Map. ....	48
Hình 2.21 Giao diện tab Setting. ....	49
Hình 2.22 Giao diện màn hình thông tin địa điểm. ....	50
Hình 2.23 Giao diện thông tin tin tức. ....	52
Hình 2.24 Giao diện tìm kiếm. ....	53
Hình 2.25 Giao diện viết bình luận. ....	54
Hình 2.26 Giao diện xem bình luận và xem thực đơn. ....	55
Hình 2.27 Giao diện đặt hàng. ....	56

Hình 2.28 Giao diện dẫn đường đến một địa điểm ăn uống và hiển thị các địa điểm.	58
Hình 2.29 Giao diện quản lý địa điểm.	59
Hình 2.30 Giao diện Edit Shop.	59
Hình 2.31 Giao diện Comment.	60
Hình 2.32 Giao diện Dish.	60
Hình 2.33 Giao diện Order.	61
Hình 2.34 Giao diện News.	61
Hình 2.35 Giao diện My Account.	62



## **DANH MỤC BẢNG**

Bảng 2.1 Bảng miêu tả cấu trúc dữ liệu đối tượng Locs. ....	17
Bảng 2.2 Bảng miêu tả cấu trúc dữ liệu đối tượng News.....	18
Bảng 2.3 Bảng miêu tả cấu trúc dữ liệu đối tượng Orders.....	19
Bảng 2.4 Bảng miêu tả cấu trúc dữ liệu đối tượng Owners. ....	20
Bảng 2.5 Bảng mô tả tab News. ....	46
Bảng 2.6 Bảng mô tả tab Search.....	47
Bảng 2.7 Bảng mô tả tab Setting. ....	49
Bảng 2.8 Bảng mô tả chi tiết thông tin địa điểm. ....	51
Bảng 2.9 Bảng mô tả tính năng viết bình luận.....	54
Bảng 2.10 Bảng mô tả chi tiết viết bình luận.....	55
Bảng 2.11 Bảng mô tả chi tiết xem thực đơn.....	56
Bảng 2.12 Bảng mô tả chi tiết giao diện đặt hàng.....	57

## **DANH MỤC BIỂU ĐỒ**

Biểu Đồ 1.1 Tỷ lệ sử dụng điện thoại thông minh và dịch vụ 3G.....	8
Biểu Đồ 1.2 Tỷ lệ lựa chọn hình thức ăn uống và đối tượng thành viên trong nhóm.	9
Biểu Đồ 1.3 Tỷ lệ phương thức tìm hiểu thông tin khi tìm một địa điểm ăn uống. ..	10

## **PHẦN MỞ ĐẦU**

### **Lý do chọn đề tài**

Trong tất cả những mặt văn hóa Việt Nam, có lẽ ẩm thực là một trong số những mảng dễ tiếp cận nhất, đồng thời cũng thể hiện những nét văn hóa đặc trưng độc đáo của đất nước ta. Văn hóa ẩm thực là nét văn hóa tự nhiên hình thành trong cuộc sống. Đối với người Việt, ẩm thực không chỉ là nét văn hóa về vật chất mà còn là văn hóa về tinh thần. Cùng với sự phát triển về kinh tế, đời sống của người dân được cải thiện, thì nhu cầu ăn uống đang ngày càng tăng cao. Một trong những điều khó khăn mà thực khách phải đối mặt là sự thiếu hụt về thông tin. Trước đây, hay ngay cả trong thời điểm hiện tại, thực khách chủ yếu tìm hiểu về nơi mình muốn tới chủ yếu bằng cách truyền miệng, tham khảo trực tiếp từ bạn bè, hay qua các phương tiện cộng đồng như diễn đàn, mạng xã hội. Tuy nhiên, dù có chuẩn bị kỹ càng như thế nào, thì thực khách cũng phải đối mặt với những rủi ro do thiếu hụt thông tin và kinh nghiệm như lạc đường, không tìm được nơi mình muốn đến. Điều này đặt ra nhu cầu về một phương tiện hữu hiệu để người dùng có thể tra cứu, tìm hiểu những địa điểm, hoạch định cho chuyến đi của mình, cùng với các giải pháp hạn chế những rủi ro có thể gặp phải. Đây chính là những nền tảng cho sự phát triển của hệ thống tìm kiếm địa điểm ăn uống trên điện thoại thông minh.

Thực tế đề tài về hệ thống tìm kiếm địa điểm ăn uống trên điện thoại thông minh không hoàn toàn mới trên thế giới. Nhưng hầu hết hệ thống đều tập trung vào khía cạnh cung cấp và tra cứu thông tin địa điểm, mà không tính đến yếu tố xây dựng cộng đồng cũng như điều kiện thực tế và nhu cầu ăn uống của người dân Việt Nam. Đây cũng chính là cơ hội để các hệ thống tìm kiếm địa điểm ăn uống dành riêng cho thị trường Việt Nam được nghiên cứu và phát triển.

### **Mục tiêu nghiên cứu**

Từ thực tế đặt ra, nhu cầu xây dựng một hệ thống tìm kiếm địa điểm ăn uống là vô cùng cần thiết. Mục tiêu của khóa luận là xây dựng và triển khai hệ thống tìm kiếm địa điểm ăn uống trên Android.

#### **Mục tiêu của khóa luận:**

Xây dựng một hệ thống giúp người dùng có thể tìm kiếm địa điểm ăn uống ngay trên chiếc điện thoại thông minh của mình.

Người dùng có thể tìm kiếm địa điểm nhanh chóng nhờ vào tính năng dẫn đường trên ứng dụng của mình.

Người dùng có thể dễ dàng tìm kiếm thông tin về nhà hàng, món ăn ở Thành phố Hồ Chí Minh và Thành phố Biên Hòa chỉ trong vài phút và ngoài ra họ còn có thể gọi đặt món và lựa chọn địa điểm tốt nhất cho mình và bạn bè.

### **Đối tượng và phạm vi nghiên cứu**

Đối tượng và phạm vi nghiên cứu là các báo cáo nghiên cứu khoa học của ngành kỹ thuật phần mềm.

### **Phương pháp nghiên cứu**

- Ứng dụng NodeJS làm Web Service.
- Ứng dụng chuẩn định dạng dữ liệu JSON để xây dựng cơ sở dữ liệu.
- Ứng dụng MongoDB dùng làm hệ quản trị cơ sở dữ liệu.
- Ứng dụng hệ điều hành Android và công cụ lập trình trên hệ điều hành Android để phát triển ứng dụng.
- Tìm hiểu về cách sử dụng Google Map API để tích hợp vào ứng dụng.
- Tìm hiểu về dịch vụ Web Service của Amazon dùng để làm host cho Web Service của ứng dụng.
- Xây dựng chương trình quản lý cửa hàng bằng Spring MVC để người dùng quản lý cửa hàng của mình.

- Thu thập thông tin các quán ăn, món ăn trên địa bàn và đưa vào cơ sở dữ liệu.

### **Kết cấu đề tài**

Báo cáo này được trình bày thành 3 phần chính: Phần mở đầu, Phần nội dung và phần Kết luận.

Phần mở đầu:

Giới thiệu sơ lược về lý do chọn đề tài, lịch sử nghiên cứu, mục tiêu nghiên cứu, đối tượng và phạm vi nghiên cứu, phương pháp nghiên cứu từ đó đem lại cho mọi người một cái nhìn tổng quan nhất về đề tài.

Phần nội dung:

Chương 1: Tổng Quan: Đề cập tới nhu cầu của ứng dụng tìm kiếm địa điểm ăn uống, tóm tắt văn bản, tình hình nghiên cứu ứng dụng trong nước cũng như ngoài nước. Khảo sát và phân tích sẽ tập trung trình bày khảo sát về nhu cầu ẩm thực, nhu cầu sử dụng điện thoại thông minh để tìm kiếm các nhà hàng, hàng quán.

Chương 2: Phân tích, thiết kế và xây dựng hệ thống: dựa vào tính năng và yêu cầu của ứng dụng, từ đó phân tích, xác định hướng đi, chọn lựa các giải pháp, đề xuất thiết kế phù hợp, trình bày một số nền tảng công nghệ, các biểu đồ, kiến trúc hệ thống và những vấn đề trong khâu hiện thực ứng dụng.

Phần kết luận:

Chương 3: Kết luận và đề nghị: tổng hợp lại các công việc đã hoàn thành, những vấn đề chưa giải quyết được cũng như định hướng phát triển của đề tài trong tương lai.

## CHƯƠNG 1: TỔNG QUAN

### 1.1 Đặt vấn đề

Trong những năm gần đây, cùng với sự phát triển của kinh tế xã hội, điện thoại thông minh ngày càng trở nên phổ biến ở thị trường Việt Nam. Bên cạnh đó, cùng với sự tăng cao của nhu cầu ăn uống hiện nay, điện thoại thông minh với khả năng linh động, cùng sự hỗ trợ của mạng lưới 3G phủ khắp toàn quốc, đóng góp một phần không nhỏ trong sự phát triển của các phần mềm tìm kiếm và chia sẻ địa điểm nói chung, hay lớp ứng dụng dành cho ăn uống nói riêng.

Trong thực tế, những ứng dụng tìm kiếm địa điểm ăn uống hiện nay đóng vai trò như một cẩm nang ăn uống, phục vụ chủ yếu nhu cầu tìm kiếm, và tra cứu thông tin địa điểm. Tuy nhiên, ở thị trường Việt Nam, nhu cầu về ăn uống của người Việt mỗi ngày càng cao. Với đặc thù về cơ sở hạ tầng và phương tiện giao thông, việc tìm kiếm một địa điểm ăn uống phù hợp với sở thích và túi tiền gặp rất nhiều khó khăn. Điều này mở ra cơ hội cho một ứng dụng tìm kiếm địa điểm ăn uống phù hợp với điều kiện và hoàn cảnh của thị trường Việt Nam được nghiên cứu và phát triển.

Với định hướng như trên, đề tài đã nghiên cứu, thiết kế và xây dựng hệ thống tìm kiếm địa điểm ăn uống trên nền tảng Android. Bên cạnh các chức năng cơ bản của một ứng dụng ăn uống như tìm kiếm, xem thông tin, đánh giá, xem bản đồ,... ứng dụng còn hỗ trợ tính năng dẫn đường, xác định các quán ăn ở gần mình trong vòng bán kính 5km.

Hệ thống được thiết kế theo mô hình client – server, trong đó việc lưu trữ dữ liệu đều được thực hiện ở server. Truyền tải dữ liệu giữa client và server được thực hiện theo hai hướng: client chủ động gửi yêu cầu và lấy dữ liệu ở server; hay server sẽ gửi thông điệp phản hồi nội dung về cho client. Bên cạnh đó, đề tài còn xây dựng một website để dễ quản lý các đơn đặt hàng của khách hàng.

Trong phạm vi của đề tài, những yêu cầu về bảo mật và hiệu năng của hệ thống chỉ đáp ứng ở mức cơ bản. Đồng thời, dữ liệu mà đề tài cung cấp chỉ với mục

đích thử nghiệm. Tuy nhiên, với nền tảng mã đề tài xây dựng, các vấn đề này có thể được giải quyết dễ dàng trong tương lai.

## **1.2 Tình hình nghiên cứu trong và ngoài nước**

Liên quan đến lĩnh vực ứng dụng trên điện thoại di động nói chung và ứng dụng tìm kiếm địa điểm ăn uống trên Android nói riêng trong những năm qua đã được sự quan tâm của cộng đồng trong và ngoài nước. Một vài ứng dụng phổ biến hiện nay có thể kể như:

### **Trong nước:**

**Foody** (2012): Được xây dựng từ giữa năm 2012 tại Thành phố Hồ Chí Minh, Việt Nam, Foody là cộng đồng tin cậy cho mọi người có thể tìm kiếm, đánh giá, bình luận các địa điểm ăn uống: nhà hàng, quán ăn, cafe, bar, karaoke, tiệm bánh, khu du lịch... tại Việt Nam - từ website hoặc ứng dụng di động. Tất cả thành viên Foody kết nối những thực khách đến với các địa điểm ăn uống lớn nhỏ cả đất nước. Đến thời điểm hiện tại, Foody với hàng trăm ngàn địa điểm và hàng trăm ngàn bình luận, hình ảnh tại Việt Nam, ở hầu hết các tỉnh thành. Foody là cách dễ nhất để bạn có thể tìm kiếm và lựa chọn địa điểm tốt nhất cho mình và bạn bè.

### **Ưu điểm:**

- Số lượng cửa hàng đăng kí với Foody khá đông đảo.
- Số lượng người dùng lớn.
- Giao diện bắt mắt.

### **Nhược:**

- Giao diện còn phức tạp.
- Chưa tích hợp tính năng đặt hàng.

**Thánh Ăn** (2011): Là ứng dụng thuần việt được phát triển vào năm 2011, Ứng dụng cung cấp cho bạn nhiều chức năng đa dạng, nó giúp người dùng tìm kiếm, đánh giá và đưa ra nhận xét cho các món ăn cũng như các cửa hàng.

### **Ưu điểm:**

- Giao diện khá đẹp.
- Thân thiện với người dùng.

- Có khả năng gợi ý món ăn và cửa hàng.

**Nhược:**

- Chưa tích hợp tính năng đặt hàng.
- Cập nhật khá chậm.

**ClingMe (2009):** là một ứng dụng Việt thông minh với nhiều tính năng độc đáo. Ngoài việc trợ giúp tìm những địa điểm xung quanh, đưa ra gợi ý các hàng quán gần và phù hợp nhu cầu người dùng nhất dựa trên GPS, ứng dụng còn hỗ trợ tìm kiếm các cây xăng, ATM, bãi đỗ xe và cả tiệm thuốc ở khắp đất nước Việt Nam. ClingMe có giao diện khá trực quan, cách bố trí hợp lý giúp người dùng dễ làm quen ngay ở lần đầu tiên.

**Ưu điểm:**

- Giao diện bắt mắt.
- Thân thiện với người dùng.
- Không chỉ tìm kiếm nhà hàng còn có thể tìm kiếm ATM, cây xăng.

**Nhược:**

- Chưa tích hợp tính năng đặt hàng.
- Cập nhật khá chậm.

**Ăn Vặt Hà Nội (2010):** Đây là một ứng dụng thiết thực dành cho người dùng yêu thích các món ăn nhẹ độc đáo mang hương vị của Hà Nội. Ứng dụng như một cuốn sổ tay nhỏ, giúp bạn nhanh chóng tìm ra những địa điểm ăn ngon trên địa bàn Hà Nội.

**Ưu điểm:**

- Giao diện thiết kế đơn giản.
- Thân thiện với người dùng.
- Số lượng cửa hàng đăng kí khá lớn.

**Nhược:**

- Chưa tích hợp tính năng đặt hàng.
- Hiện tại chỉ hỗ trợ trên địa bàn thủ đô Hà Nội.



**Ngoài nước:**

**Foodspotting** (2009): Foodspotting là ứng dụng trực tuyến cho phép người dùng có thể xem và tìm kiếm các nhà hàng. Thay vì việc xem bình luận bạn hoàn toàn có thể giới thiệu nhà hàng cho bạn bè và nhận những giới thiệu từ bạn bè.

**Ưu điểm:**

- Giao diện khá đơn giản.
- Thân thiện với người dùng.

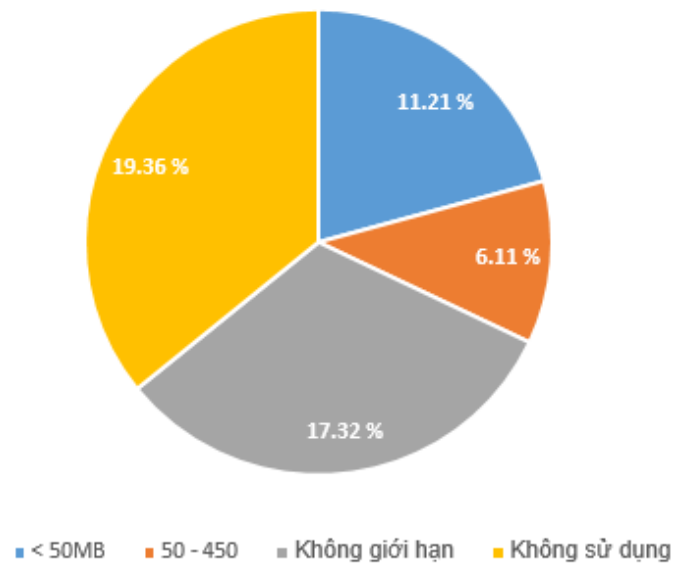
**Nhược:**

- Giao diện kém bắt mắt.
- Chưa thể tìm kiếm theo tên món ăn.
- Chưa tích hợp tính năng đặt hàng.

Nhìn thấy những thiếu sót về tính năng và sự phức tạp, phiền toái trong việc sử dụng các ứng dụng trên. Nhu cầu đặt ra là cần một ứng dụng có giao diện đơn giản, bắt mắt và dễ dàng sử dụng đồng thời tích hợp thêm một số tính năng như đặt hàng qua ứng dụng và tìm kiếm theo tên món ăn dẫn tới quyết định chọn và phát triển đề tài Xây dựng hệ thống tìm kiếm địa điểm ăn uống trên Android.

**1.3 Khảo sát nhu cầu người dùng**

Trong quá trình tìm hiểu nhu cầu ứng dụng, nhóm có tìm hiểu một số website điều tra về tình hình sử dụng smartphone hiện nay. Cụ thể như khảo sát ở trang web [smartinsights.com](http://smartinsights.com) thì có khoảng 80% người dùng trên thế giới sử dụng smartphone. Từ đó, suy ra số lượng người dùng smartphone ở Việt Nam cũng tầm khoảng đó. Kết quả điều tra gồm những phần sau:



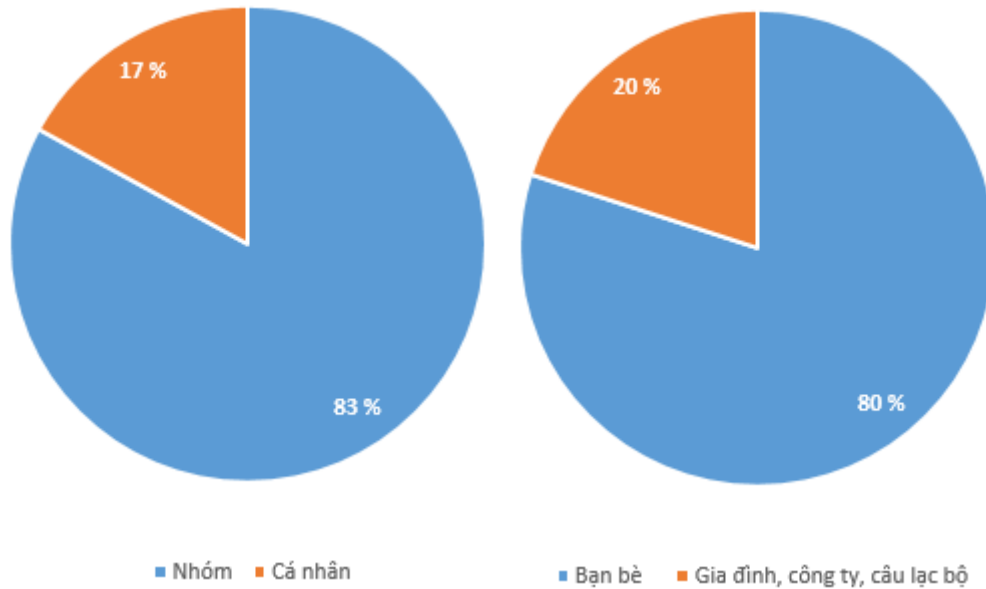
*Biểu Đồ 1.1 Tỷ lệ sử dụng điện thoại thông minh và dịch vụ 3G.*

### **Về tỉ lệ sử dụng điện thoại thông minh và dịch vụ 3G**

Theo khảo sát mới nhất của cộng đồng khảo sát trực tuyến Vinarsearch, dựa trên 4426 mẫu khảo sát online thì: cứ 100 thành viên của Vinarsearch thì có gần 80 người hiện đang sở hữu ít nhất một smartphone (chiếm 77.8%) và có 2/3 trong số đó có sử dụng dịch vụ 3G để truy cập Internet. Điều này thể hiện dịch vụ 3G là một thành phần không thể thiếu đối với đa số người dùng. Tuy nhiên, phần lớn người sử dụng dịch vụ 3G chỉ với gói cước giới hạn dung lượng, đây cũng chính là điểm đáng lưu ý khi phát triển ứng dụng.

### **Về nhu cầu ăn uống theo nhóm**

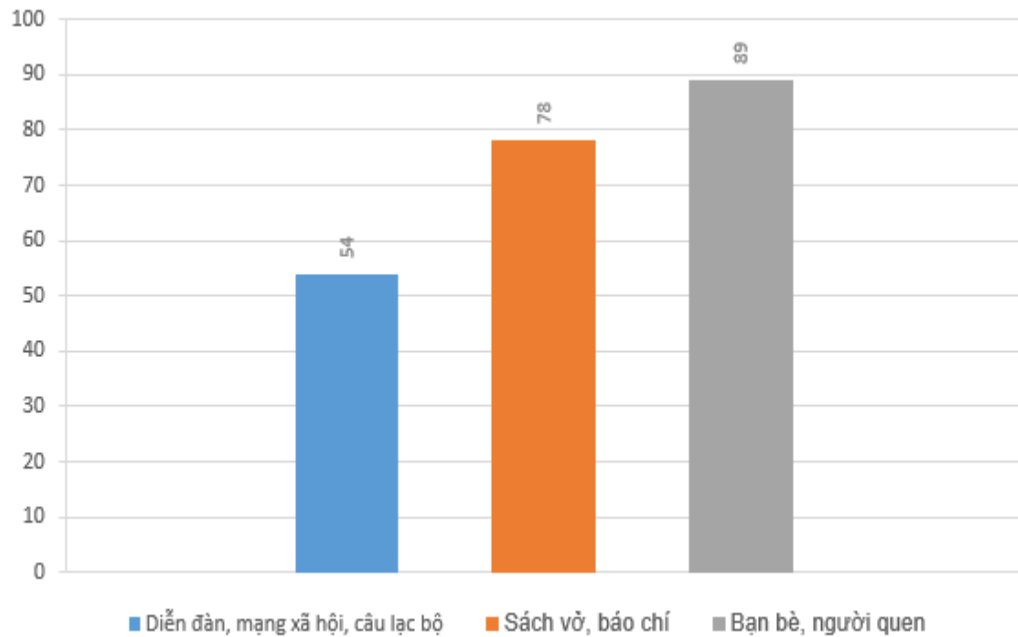
Theo điều tra, phần lớn người dùng thường chọn hình thức ăn uống ưa thích là ăn uống theo nhóm với đại bộ phận thành viên đến từ bạn bè của mình. Bên cạnh đó, cũng một phần không nhỏ thành viên trong gia đình, câu lạc bộ, công ty tổ chức tiệc tùng và có niềm đam mê về ăn uống.



*Biểu Đồ 1.2 Tỷ lệ lựa chọn hình thức ăn uống và đối tượng thành viên trong nhóm.*

### **Về phương thức tìm hiểu thông tin về một địa điểm ăn uống**

Kinh nghiệm từ bạn bè hay người quen là ưu tiên hàng đầu, tiếp đến là từ sách vở báo chí, và cuối cùng là tìm hiểu thông qua các diễn đàn, mạng xã hội. Tuy nhiên, những nguồn thông tin này thường rải rác và có độ khách quan không cao, đòi hỏi người dùng phải có kinh nghiệm tìm kiếm và sàng lọc thông tin, nhằm đưa ra những lựa chọn chính xác khi đi ăn uống.



*Biểu Đồ 1.3 Tỷ lệ phương thức tìm hiểu thông tin khi tìm một địa điểm ăn uống.*

## Kết luận

Từ những kết quả điều tra như trên, có thể kết luận rằng một ứng dụng tìm kiếm địa điểm ăn uống ở Việt Nam có thể đáp ứng được nhu cầu của người dùng phải có những đặc điểm sau:

- Cung cấp phương tiện tra cứu thông tin hiệu quả và chính xác.
- Cung cấp phương tiện kết nối giữa người dùng với người dùng.

### 1.4 Những tính năng đề xuất

Trên cơ sở phân tích kết quả khảo sát nhu cầu của người dùng ở phần trước, đề tài đề xuất các tính năng chính của ứng dụng như sau:

**Tìm kiếm địa điểm ăn uống, món ăn:** là tính năng không thể thiếu đối với ứng dụng tìm kiếm địa điểm ăn uống. Tính năng tìm kiếm cho phép người dùng tìm kiếm thông tin về món ăn, những địa điểm quán xá, nhà hàng ở những vị trí cụ thể. Tính năng này có thể lọc kết quả từ những từ riêng biệt nhằm tối ưu hóa trải nghiệm người dùng.

**Xem thông tin, đánh giá, bình luận:** tính năng hỗ trợ người dùng xem thông tin, bình luận, đánh giá về những địa điểm mà người dùng đến, đánh giá tương quan

với những người dùng khác. Đây chính là nguồn thông tin mà người dùng có thể tham khảo để lựa chọn món ăn, quán ăn phù hợp với nhu cầu và sở thích của mình.

Đặt hàng món ăn: tính năng hỗ trợ người dùng đặt món qua mạng, giúp người dùng không cần phải đến trực tiếp nơi đó để mua. Tính năng này vô cùng thuận tiện đối với những người không có thời gian để đi ăn uống, những người gặp khó khăn trong vấn đề di chuyển.

Liệt kê các địa điểm ăn uống trong phạm vi bán kính 5km xung quanh bạn: là tính năng quan trọng không thể thiếu đối với ứng dụng tìm kiếm địa điểm ăn uống. Tính năng này cho phép liệt kê các địa điểm quán ăn, nhà hàng trong phạm vi bán kính 5km xung quanh bạn, giao diện trực quan, giúp người dùng dễ dàng nhận biết và thao tác.

Chỉ dẫn đường đi sau khi chọn địa điểm quán ăn: sau khi đã tìm được địa điểm mình mong muốn, thì ứng dụng có tính năng dẫn đường, giúp cho người dùng có thể xác định được đường đi đến nơi cần đến. Đây cũng là một tính năng quan trọng của đề tài.

### **1.5 Những yêu cầu của ứng dụng**

Ứng dụng của đề tài phải đảm bảo được những yêu cầu sau đây:

Giao diện đơn giản, thân thiện, dễ sử dụng: đây là yêu cầu chung của các ứng dụng trên thiết bị di động.

Dữ liệu về những địa điểm ăn uống phải phong phú, đa dạng và chính xác. Tuy nhiên, trong giới hạn đề tài luận văn, số lượng địa điểm chỉ được giới hạn nhằm mục đích thử nghiệm.

Yêu cầu về hiệu năng: cung cấp và lựa chọn các giải pháp hiện thực nhằm tối ưu hóa thời lượng pin cũng như tối thiểu lưu lượng 3G mà người dùng sử dụng.

Các yêu cầu trên đây sẽ lần lượt được giải quyết thông qua quá trình phân tích, thiết kế cũng như hiện thực hệ thống ở chương tiếp theo.

## CHƯƠNG 2: PHÂN TÍCH, THIẾT KẾ VÀ XÂY DỰNG HỆ THỐNG

Trong chương này, báo cáo thực hiện việc phân tích các yêu cầu, chức năng của ứng dụng, xác định các hướng đi trước các vấn đề phát sinh, từ đó so sánh, chọn lựa và đưa ra các giải pháp, đề xuất thiết kế phù hợp, trình bày một số nền tảng công nghệ, các biểu đồ, kiến trúc hệ thống và những vấn đề trong khâu hiện thực ứng dụng.

### 2.1 Phân tích và thiết kế hệ thống

#### 2.1.1 Mô hình Client – Server

##### Phân tích

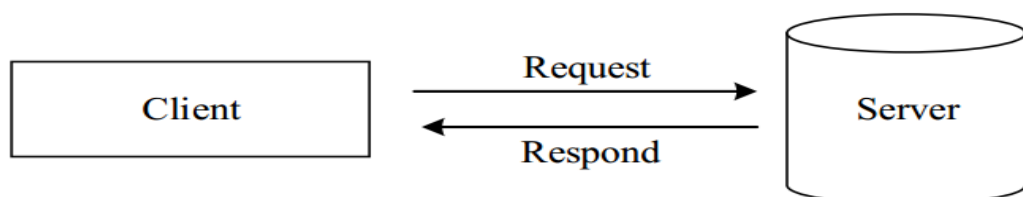
Một trong những yêu cầu đầu tiên của ứng dụng là phải có lượng thông tin phong phú và đa dạng về các địa điểm ăn uống nhằm đáp ứng nhu cầu của người dùng. Ngoài ra, các thông tin này cần phải được cập nhật liên tục và đồng bộ giữa các thiết bị của từng người dùng khác nhau.

Bên cạnh việc tìm kiếm và tra cứu thông tin, ứng dụng cũng bao gồm những chức năng đòi hỏi sự tương tác giữa người dùng với nhau như các tính năng bình luận, đánh giá.

Do đó, việc hiện thực ứng dụng trên từng thiết bị di động độc lập, vốn bị giới hạn về khả năng xử lý, bộ nhớ và thời lượng pin, là bất khả thi. Đòi hỏi phải có một hệ thống máy chủ đảm nhận công việc lưu trữ và đồng bộ lượng lớn dữ liệu, cũng như đảm nhận các công việc tính toán khi cần thiết.

##### Thiết kế

Ứng dụng cần có một server để giải quyết các vấn đề đã phân tích ở trên. Hệ thống sẽ hoạt động theo mô hình client – server được minh họa như sơ đồ sau:



Hình 2.1 Mô hình Client – Server.

Như sơ đồ trên, hệ thống sẽ gồm hai phần:

- Server: lưu trữ dữ liệu, thực hiện các thao tác trên dữ liệu, tính toán, tìm kiếm, và trả kết quả về cho client.
- Client: thiết bị có cài đặt. Ứng dụng có nhiệm vụ tiếp nhận các thao tác từ phía người dùng, gửi yêu cầu lên server và nhận kết quả trả về tương ứng.

Ngoài việc giải quyết các vấn đề về lưu trữ và đồng bộ dữ liệu giữa nhiều người dùng khác nhau, cũng như công việc tính toán khi cần thiết, mô hình client – server còn có những ưu điểm sau:

Dễ dàng phát triển, mở rộng trên các nền tảng khác: hầu hết các nền tảng di động như iOS, Window Phone, Android đều hỗ trợ các giao thức kết nối với server. Do đó, sử dụng mô hình client – server giúp việc mở rộng ứng dụng trên các nền tảng khác trở nên dễ dàng hơn, chỉ bao gồm công sức phát triển ứng dụng client mà không cần phải thiết kế lại server.

Dễ dàng nâng cấp và mở rộng server: hoàn toàn có thể nâng cấp và cải thiện hiệu suất của hệ thống bằng cách nâng cấp cấu hình của server.

### **2.1.2 Giao tiếp giữa Client và Server**

#### **Phân tích**

Đối với một hệ thống được thiết kế theo mô hình Client – Server, phương pháp giao tiếp giữa client và server là yếu tố cốt lõi cần được quan tâm trong quá trình thiết kế ứng dụng. Khác với nền tảng máy tính cá nhân vốn không bị giới hạn về thời lượng pin, cũng như lưu lượng sử dụng kết nối Internet, người dùng thiết bị di động rất chú trọng cách một ứng dụng sử dụng năng lượng cũng như bộ nhớ trên thiết bị.

Một cách tổng quát, việc giao tiếp giữa client và server có thể chia thành hai hướng:

- Client chủ động gửi yêu cầu lên server, server truy xuất thông tin và trả kết quả về cho client, client hiển thị kết quả lên ứng dụng.

- Server tập hợp những thay đổi từ lần cập nhật gần nhất của client, gửi về cho client, client thông báo cho người dùng hoặc thể hiện trực tiếp trên ứng dụng.

Hai hướng tiếp cận đều có điểm mạnh và điểm yếu riêng:

- Client chủ động: đòi hỏi phải thiết kế những thao tác do người dùng kích hoạt hoặc những tác vụ ngầm trên ứng dụng nhằm gửi yêu cầu cập nhật lên server sau khoảng thời gian nhất định. Khoảng thời gian cập nhật này thường được người phát triển định trước trong quá trình phân tích và thiết kế ứng dụng. Điểm mạnh của hướng tiếp cận này là hiện thực đơn giản, không yêu cầu cao về hệ thống server. Tuy nhiên, việc đòi hỏi phải có thao tác kích hoạt từ phía người dùng đồng nghĩa với việc người dùng phải thực hiện thao tác liên tục nhằm cập nhật thông tin. Bên cạnh đó, việc sử dụng tác vụ ngầm gửi yêu cầu cập nhật sau khoảng thời gian định trước cũng có mặt hạn chế. Nếu người phát triển định khoảng thời gian này quá ngắn, ứng dụng sẽ phải gửi yêu cầu và nhận kết quả trả về liên tục. Tuy việc định khoảng thời gian cập nhật ngắn có thể làm tăng trải nghiệm, giảm thời gian chờ của người dùng, nhưng việc này đòi hỏi ứng dụng phải luôn hoạt động, làm giảm thời lượng pin cũng như tiêu hao lưu lượng sử dụng 3G của người dùng một cách nhanh chóng, trong khi chưa có gì có thể khẳng định lần gửi yêu cầu đó là có nghĩa (có thông tin mới cần đồng bộ giữa client và server). Nếu nhà phát triển định khoảng thời gian này quá dài, tuy có thể tiết kiệm được thời lượng pin và lưu lượng 3G, nhưng với thời gian chờ quá lâu có thể làm giảm trải nghiệm người dùng, đặc biệt đối với những ứng dụng mang yếu tố cộng đồng như đề tài đang phát triển.
- Server chủ động: cách tiếp cận này tránh được những mặt hạn chế của phương pháp client chủ động. Server dựa trên sự thay đổi



trong dữ liệu và gửi yêu cầu cập nhật hay trực tiếp truyền thông tin về phía client. Tuy nhiên, phương hướng này đòi hỏi server phải hiện thực nhiều giải pháp nhằm có thể giữ kết nối với lượng lớn ứng dụng trên nhiều thiết bị khác nhau.

Như vậy, phương pháp client chủ động dù đơn giản về mặt hiện thực, nhưng gặp những hạn chế trong việc phụ thuộc vào thao tác kích hoạt cũng như xác định thời gian cập nhật trên client. Phương pháp server chủ động cho phép server gửi trực tiếp dữ liệu về client mà không thông qua kích hoạt từ client, nhưng phương pháp này lại đòi hỏi nhiều hiện thực ở phía server cũng như client nhằm đảm bảo duy trì kết nối một cách liên tục.

### **Thiết kế**

Với những đặc thù cũng như điểm mạnh và điểm yếu của cả hai phương pháp như trên, đề tài đề nghị sử dụng phương pháp trong hệ thống, cụ thể như sau:

Client chủ động: sử dụng khi có thao tác kích hoạt trực tiếp từ người dùng ở phía client hoặc sau khi nhận được tín hiệu yêu cầu cập nhật từ phía server.

#### **2.1.3 Cơ sở dữ liệu**

##### **Phân tích**

Có nhiều lựa chọn khác nhau cho việc lưu trữ dữ liệu của hệ thống, được chia thành hai nhóm chính như sau:

- Cơ sở dữ liệu quan hệ (relational database): cơ sở dữ liệu truyền thống, được sử dụng rất phổ biến trên thế giới. Nổi bật nhất là MySQL, SQL Server và Oracle.
- Cơ sở dữ liệu NoSQL (Not Only SQL): bắt đầu phổ biến trong khoảng vài năm trở lại đây, được sử dụng trong các hệ thống lớn với lượng dữ liệu khổng lồ (Google, Amazon,...), nổi bật với khả năng mở rộng và hiệu năng cao như Cassandra, Kyoto Cabinet, MongoDB.

Cơ sở dữ liệu quan hệ dù có những ưu điểm về khả năng mở rộng và hiệu năng so với cơ sở dữ liệu NoSQL, nhưng đối với đề tài ứng dụng cộng đồng có hỗ trợ web như tìm kiếm địa điểm ăn uống thì lại tỏ ra không phù hợp. Lượng dữ liệu đối với những ứng dụng cộng đồng là vô cùng lớn nên đòi hỏi phải có một cơ sở dữ liệu nào đó để lưu trữ, xử lý đến hàng petabytes dữ liệu với khả năng chịu tải, chịu lỗi cao những chỉ đòi hỏi về tài nguyên phần cứng thấp. Ngoài ra, cơ sở dữ liệu cần phải thiết kế đơn giản, nhẹ, gọn, tác vụ đọc-ghi cao, đồng nhất, dễ dàng thêm bớt các node mà không ảnh hưởng tới toàn hệ thống. Cơ sở dữ liệu NoSQL là phù hợp nhất.

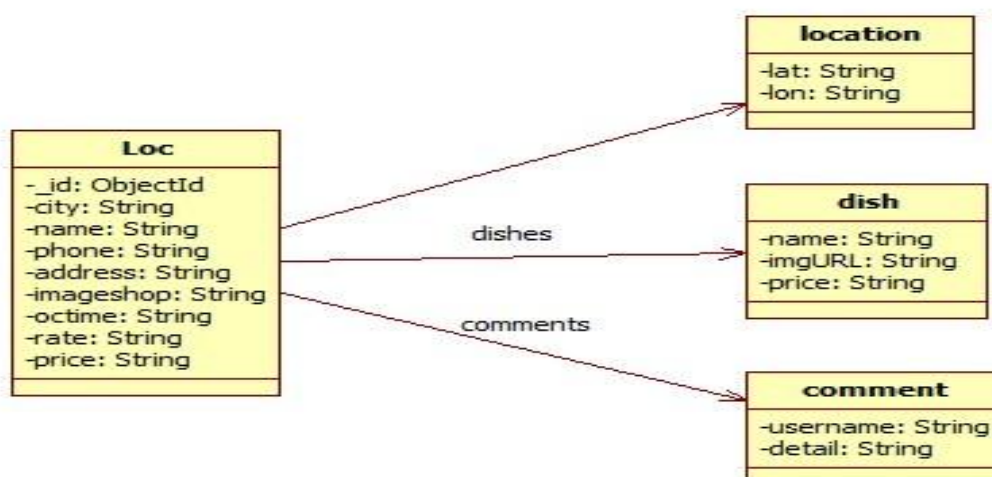
Trong những cái tên còn lại của cơ sở dữ liệu NoSQL, MongoDB được chọn làm cơ sở dữ liệu phía server của ứng dụng và vì nó lưu trữ dữ liệu dưới dạng binary JSON, khả năng truy vấn uyển chuyển, shell script tương tác bằng Javascript nên do đó nó hay đi chung với Node.js.

### Thiết kế

Cơ sở dữ liệu của hệ thống gồm những đối tượng chính như sau:

- Locs: lưu trữ thông tin về địa điểm ăn uống, bao gồm những thông tin về vị trí, các món ăn.

Sơ đồ lớp



Hình 2.2 Sơ đồ lớp của tập tài liệu Locs.

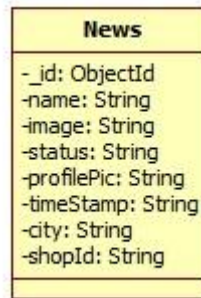
Bảng Locs có cấu trúc như sau:

*Bảng 2.1 Bảng miêu tả cấu trúc dữ liệu đối tượng Locs.*

Locs		
_id	ObjectId	: mã của tập tài liệu
city	String	: tên thành phố
name	String	: tên địa điểm ăn uống
phone	String	: số điện thoại
address	String	: địa chỉ
location	Object	
lat	String	: vĩ tuyến
lon	String	: kinh tuyến
imageshop	String	: url hình ảnh địa điểm
octime	String	: giờ mở cửa
rate	Int32	: đánh giá
dishes	Array	
name	String	: tên món ăn
imgURL	String	: url hình ảnh món ăn
price	String	: giá món ăn
Comments		
username	String	: tên người dùng
detail	String	: lời bình luận

- News: là nơi chứa các tin tức khuyến mãi về các địa điểm ăn uống trên địa bàn TP.HCM.

Sơ đồ lớp



Hình 2.3 Sơ đồ lớp của tập tài liệu News.

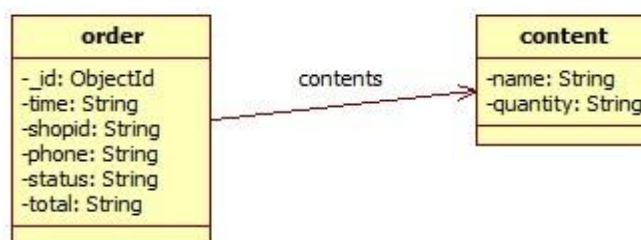
Bảng News có cấu trúc như sau:

Bảng 2.2 Bảng miêu tả cấu trúc dữ liệu đối tượng News.

News		
_id	ObjectId	: mã của tập tài liệu
name	String	: tên địa điểm ăn uống
image	String	: url hình ảnh của địa điểm
status	String	: dòng sự kiện của địa điểm
profilePic	String	: nơi chứa các tập dữ liệu hình ảnh
timeStamp	String	: thời gian diễn ra sự kiện
city	String	: nơi diễn ra sự kiện
shopId	String	: mã của địa điểm

- Orders: nơi chứa thông tin đặt hàng của khách hàng qua ứng dụng.

Sơ đồ lớp



Hình 2.4 Sơ đồ lớp của tập tài liệu Orders.

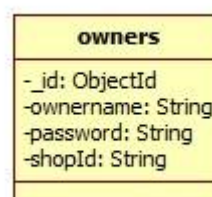
Bảng Orders có cấu trúc như sau:

*Bảng 2.3 Bảng miêu tả cấu trúc dữ liệu đối tượng Orders.*

Orders								
_id	ObjectId	: mã của tập tài liệu						
time	String	: thời gian đặt hàng						
shopid	String	: mã của địa điểm						
phone	String	: số điện thoại của khách hàng						
status	String	: ý kiến nhận xét của khách hàng						
total	String	: tổng chi phí đặt hàng của khách hàng						
content	<table border="1"> <tr> <td>name</td><td>String</td><td>: tên món ăn</td></tr> <tr> <td>quantity</td><td>String</td><td>: số lượng</td></tr> </table>		name	String	: tên món ăn	quantity	String	: số lượng
name	String	: tên món ăn						
quantity	String	: số lượng						

- Owners: là nơi để lưu trữ các tài khoản quản trị của hệ thống, những tài khoản này dùng để quản lý website đặt hàng.

Sơ đồ lớp



*Hình 2.5 Sơ đồ lớp của tập tài liệu Owners.*

Bảng Owners có cấu trúc như sau:

*Bảng 2.4 Bảng miêu tả cấu trúc dữ liệu đối tượng Owners.*

Owners		
_id	ObjectId	: mã của tập tài liệu
ownername	String	: tên người quản trị
password	String	: mật khẩu của người quản trị
shopId	String	: mã của địa điểm

#### 2.1.4 Lưu trữ hoạt động của người dùng

##### Phân tích

Hoạt động của người dùng bao gồm những hoạt động tương tác của người dùng đến các đối tượng trong ứng dụng. Hệ thống sẽ lưu trữ những hoạt động này nhằm phục vụ cho việc cập nhật tin tức của các người dùng khác nhau trong mạng xã hội.

Những hoạt động được lưu trữ bao gồm:

- Đánh giá về một địa điểm
- Bình luận về một địa điểm

Về cơ bản, những thông tin hoạt động đã được lưu đầy đủ trong cơ sở dữ liệu giữa người dùng với đối tượng.

##### Thiết kế

Đối với những ứng dụng mang tính cộng đồng như tìm kiếm địa điểm ăn uống thì việc thiết kế theo cơ sở dữ liệu NoSQL là vô cùng hợp lý. Vì chỉ với NoSQL thì nó mới có thể lưu trữ một lượng lớn dữ liệu hoạt động từ người dùng.

#### 2.1.5 Thiết kế cấu trúc ứng dụng

##### Phân tích

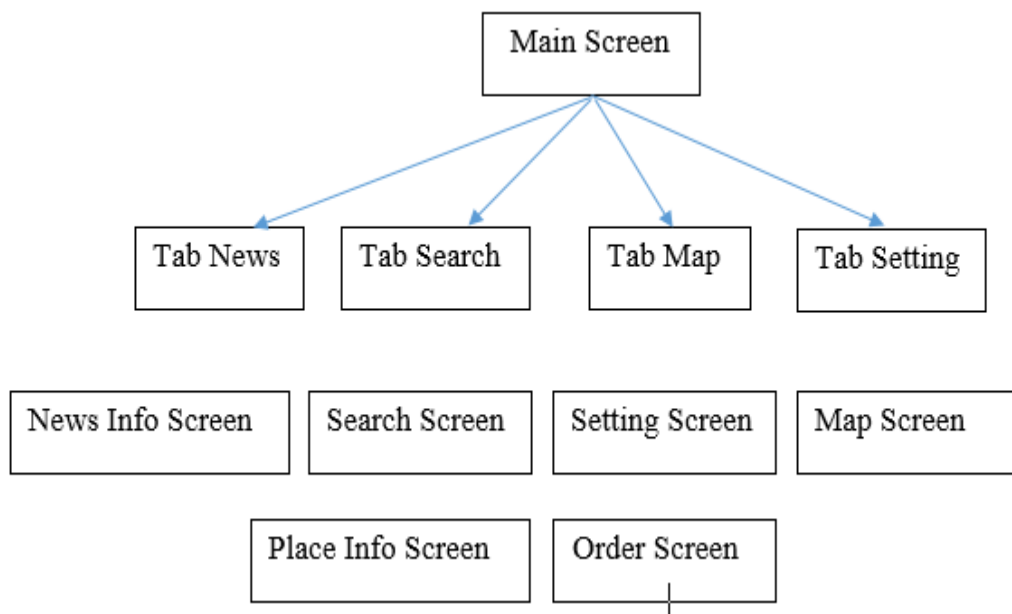
Thiết kế là yếu tố quan trọng hàng đầu trong việc xây dựng ứng dụng nói chung và ứng dụng trên thiết bị di động nói riêng. Với kích cỡ màn hình nhỏ, việc thiết kế ứng dụng phải đảm bảo tiêu chí thân thiện, dễ sử dụng, thao tác đơn

giản, điều hướng không quá sâu nhằm không tạo cảm giác rối rắm cho người dùng.

Phong cách thiết kế ứng dụng phụ thuộc khá nhiều vào nền tảng và kinh nghiệm của người phát triển. Vì lý do đó, việc thiết kế ứng dụng của đề tài chỉ đảm bảo đáp ứng các yêu cầu cơ bản và áp dụng các kỹ thuật mà đề tài cho là phù hợp với ứng dụng.

### Thiết kế

Với yêu cầu như trên, ứng dụng được thiết kế theo sơ đồ sau:



*Hình 2.6 Sơ đồ thiết kế giao diện ứng dụng.*

Theo sơ đồ trên, ứng dụng được chia ra thành những màn hình chính như sau:

Main Screen: đây là màn hình được thiết kế theo cấu trúc gồm nhiều tab, các tab có thể chuyển đổi qua lại với nhau tùy theo nhu cầu của người dùng. Nội dung của các tab như sau:

- Tab News: thể hiện danh sách tin tức về các địa điểm ăn uống ở trên địa bàn TP.HCM. Danh sách tin tức này bao gồm những thông tin về các nhà hàng, món ăn, các tin tức giảm giá, khuyến mãi.

- Tab Search: tìm kiếm và hiển thị địa điểm ăn uống trong địa bàn TP.HCM.
- Tab Map: thể hiện các địa điểm ăn uống trong vòng bán kính 5km ở vị trí hiện tại.
- Tab Setting: chứa các thông tin về ứng dụng và cài đặt phạm vi bán kính.
- Bên cạnh giao diện chính như trên, tùy theo thao tác của người dùng, ứng dụng có thể chuyển tới những màn hình như sau:
- News Info Screen: thể hiện thông tin khuyến mãi, giảm giá về một món ăn hoặc một địa điểm ăn uống. Thông tin này bao gồm những thông tin về tên địa điểm, địa chỉ, dòng trạng thái, thời gian diễn ra sự kiện.
- Search Screen: người dùng sẽ nhập tên của món ăn hoặc địa điểm vào thanh Search Bar, sau đó màn hình sẽ hiển thị danh sách các địa điểm hoặc món ăn mà người dùng vừa nhập vào.
- Setting Screen: màn hình sẽ hiển thị những thông tin về ứng dụng của nhà phát triển và là nơi để cài đặt phạm vi bán kính của người dùng (từ 1km – 5km).
- Map Screen: hiển thị thông tin vị trí về các địa điểm mà người dùng đã chọn trong phạm vi bán kính mà người dùng đã cài đặt. Người dùng có thể chọn một địa điểm trong số đó để thực hiện việc chỉ đường đến địa điểm.
- Place Info Screen: thể hiện thông tin về một địa điểm ăn uống, bao gồm thông tin về tên địa điểm, địa chỉ, đánh giá, mức giá dao động khi ăn ở đó, hình ảnh. Trong màn hình này chứa những tính năng như viết bình luận, xem bình luận, xem thực đơn của địa điểm, đặt hàng.
- Order Screen: đây là nơi để người dùng có thể đặt hàng món ăn, bao gồm những thông tin như giá cả, số điện thoại, thực đơn, số lượng.



Ngoài ra, ở từng màn hình giao diện như trên, tùy theo số lượng thông tin cần thể hiện và tính năng cho phép, ứng dụng sẽ thiết kế những giao diện con tương ứng, và người dùng có thể thông qua thao tác trên ứng dụng nhằm kích hoạt để xem thông tin cũng như thực hiện tính năng. Như vậy, giao diện ứng dụng có thể trở nên đơn giản, tập trung vào những nội dung chính, qua đó làm tăng trải nghiệm người dùng.

### **2.1.6 Lưu trữ dữ liệu ở ứng dụng**

#### **Phân tích**

Đối với ứng dụng thiết kế theo mô hình client – server, toàn bộ dữ liệu của ứng dụng được lưu trữ trong hệ thống cơ sở dữ liệu của server, và tùy theo nhu cầu của người dùng, dữ liệu tương ứng sẽ được gửi về client và hiển thị lên ứng dụng. Tuy nhiên, đối với những thông tin cần thiết mà người dùng có thể tiếp tục sử dụng trong tương lai, ứng dụng phải cung cấp cơ chế lưu trữ tạm thời những thông tin này và xóa bỏ khi người dùng không còn sử dụng nữa.

#### **Thiết kế**

Theo yêu cầu tính năng của ứng dụng đã miêu tả ở những phần trước, dữ liệu lưu trữ trong ứng dụng là loại dữ liệu lưu trữ tạm thời và xóa ngay khi có dữ liệu thay thế, gồm dữ liệu của những trang ở màn hình chính của ứng dụng. Khi người dùng chuyển sang trang bất kỳ, ứng dụng sẽ gửi yêu cầu lên server và nội dung của trang sẽ được làm mới để đảm bảo tất cả dữ liệu đều được đồng bộ và cập nhật.

Bên cạnh đó, việc quản lý dữ liệu trong ứng dụng phải đảm bảo yêu cầu đúng và đủ nhằm tránh trường hợp lưu trữ dữ liệu không chính xác cũng như lưu trữ quá nhiều dữ liệu không cần thiết có thể ảnh hưởng đến hiệu suất của toàn bộ hệ thống trên thiết bị của người dùng.

### **2.1.7 Xác định vị trí và dẫn đường**

#### **Phân tích**

Xác định vị trí và dẫn đường là một trong số những yêu cầu chính của ứng dụng. Các tính năng quan trọng của đề tài như định vị, xác định tuyến đường

giữa hai địa điểm, xác định những địa điểm gần người dùng đều liên quan trực tiếp đến vấn đề này.

Trên thực tế, đây là một bài toán phức tạp bao gồm nhiều thành phần khác nhau như lưu trữ và xử lý thông tin bản đồ, thuật toán tìm đường giữa hai địa điểm. Việc tự xây dựng hoàn toàn tính năng này sao cho hiệu quả là bất khả thi trong thời gian ngắn. Tuy nhiên, với sự phát triển của các dịch vụ bản đồ như Google Map, Bing Map, HERE Map, vấn đề này có thể được giải quyết gần như triệt để.

### **Thiết kế**

Đề tài sử dụng API của Google Map để giải quyết vấn đề định vị và dẫn đường trong hệ thống, Google Map có một số ưu điểm như sau:

- Hỗ trợ tốt ứng dụng trên nền tảng Android.
- Hệ thống xử lý nhanh, thông tin chính xác và đầy đủ.
- Hỗ trợ thuật toán tìm đường giữa hai địa điểm.
- Miễn phí với lượng yêu cầu thấp.

Mỗi địa điểm được lưu trong cơ sở dữ liệu của ứng dụng đều kèm theo địa chỉ GPS gồm kinh độ (longitude) và vĩ độ (latitude). Tọa độ này được sử dụng để đánh dấu địa điểm trên bản đồ, cũng như phục vụ cho chức năng xác định tuyến đường giữa hai địa điểm.

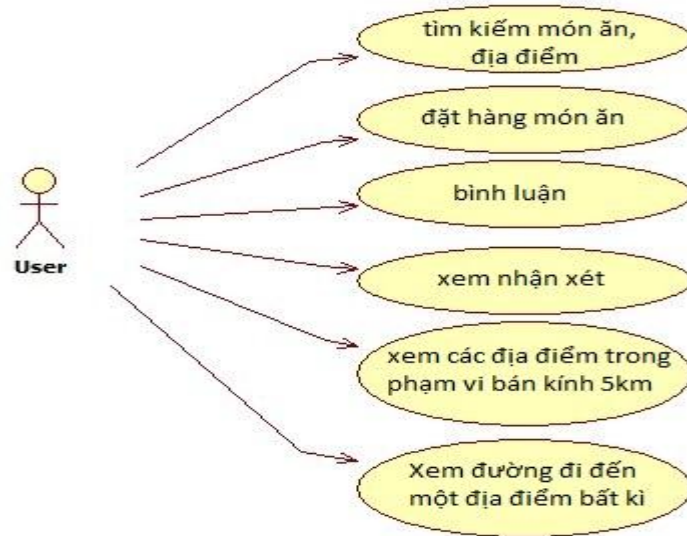
- Việc xác định tuyến đường giữa hai địa điểm được thực hiện qua:
- Xác định tọa độ của điểm đi và điểm đến.
- Gởi yêu cầu tìm đường đến API của Google Map với hai tọa độ được xác định như trên.
- Từ kết quả trả về phân tích ra dữ liệu của tuyến đường.
- Vẽ tuyến đường lên bản đồ trong ứng dụng.

## **2.2 Biểu đồ và kiến trúc hệ thống**

### **2.2.1 Các biểu đồ hệ thống**

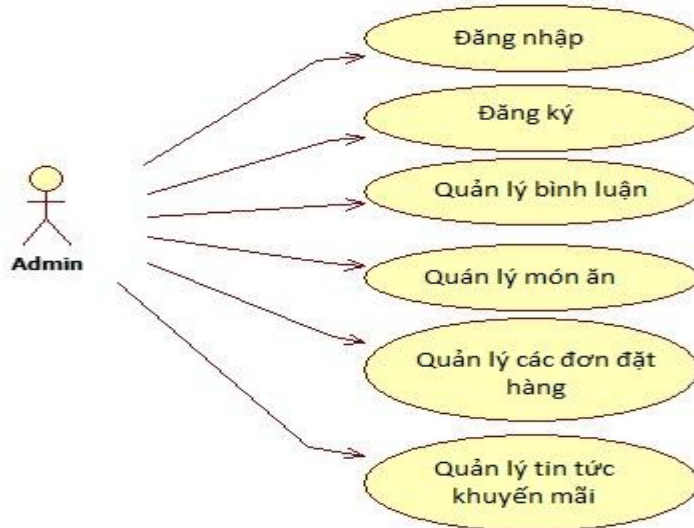
### 2.2.1.1 Biểu đồ Use case

User



Hình 2.7 Sơ đồ use case của người dùng.

Admin

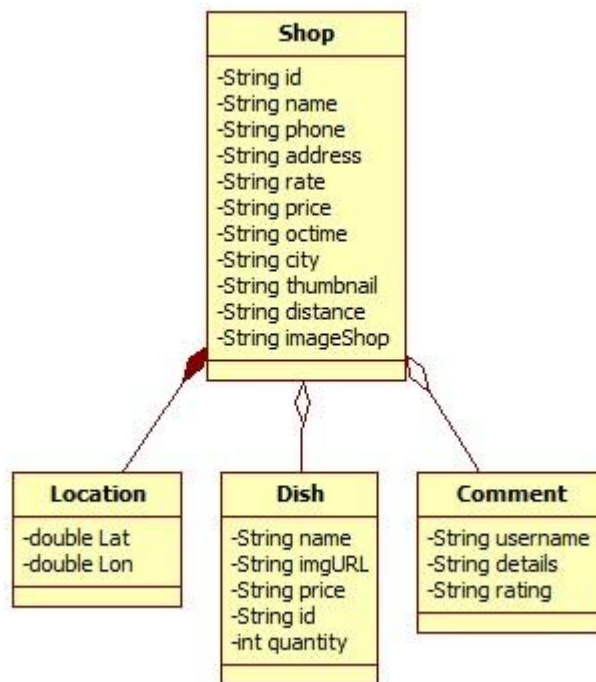


Hình 2.8 Sơ đồ use case của Admin.

### 2.2.1.2 Biểu đồ lớp

Biểu đồ lớp của Shop

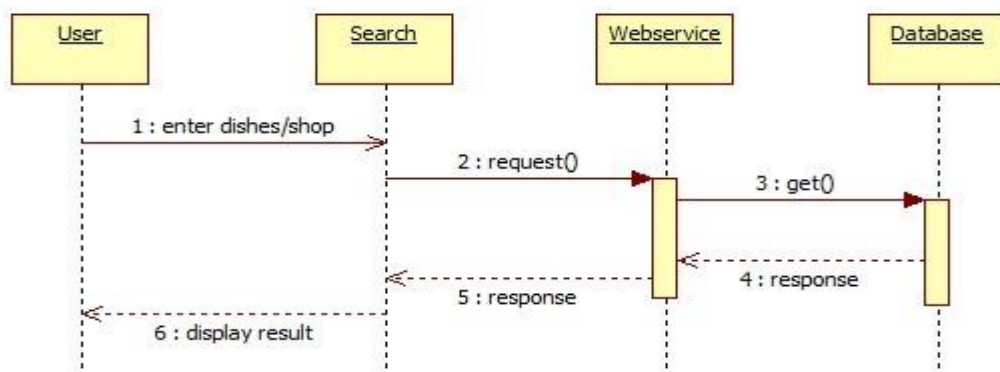
Thông tin chi tiết về một địa điểm ăn uống.



Hình 2.9 Biểu đồ lớp Shop.

### 2.2.1.3 Biểu đồ Sequence

#### Tìm kiếm món ăn/nhà hàng



Hình 2.10 Biểu đồ sequence chức năng Search.

Bước 1: Người dùng nhập tên món ăn/nhà hàng vào thanh search và nhấn nút Ok.

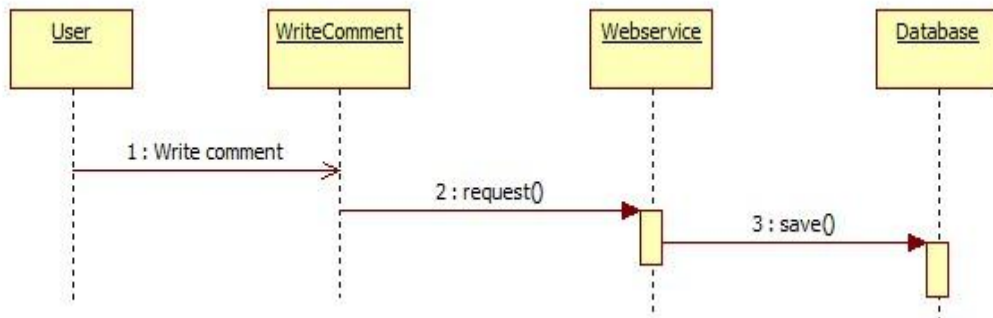
Bước 2: Client sẽ gửi request đến Webservice để lấy thông tin dữ liệu.

Bước 3 + 4: Webservice sẽ lấy dữ liệu từ database, database sẽ gửi dữ liệu về cho Webservice.

Bước 5: Webservice trả kết quả về cho client.

Bước 6: Client sẽ hiển thị kết quả tìm kiếm cho người dùng xem.

### Viết bình luận



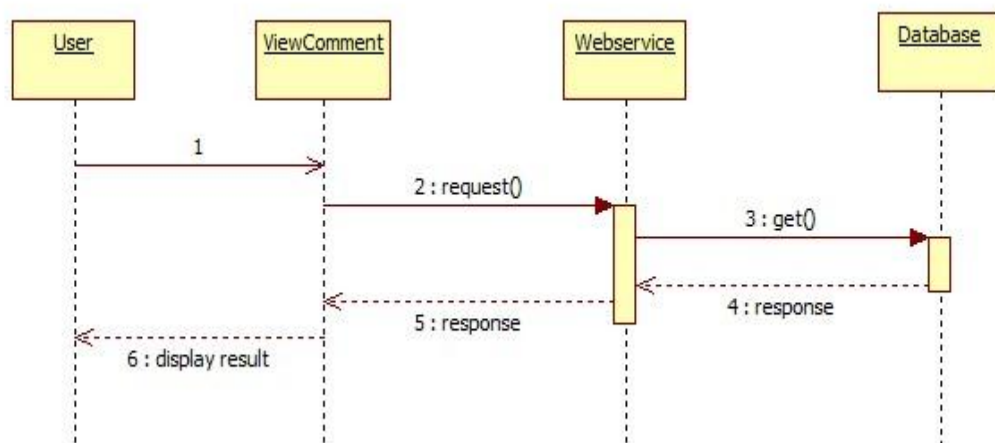
Hình 2.11 Biểu đồ sequence chức năng viết bình luận.

Bước 1: Người dùng sẽ nhập username, rồi sau đó tiến hành nhập bình luận. Sau khi nhập bình luận, người dùng nhấn Post.

Bước 2: Client sẽ gửi yêu cầu lên webservice.

Bước 3: Webservice sẽ xử lý và tiến hành lưu vào cơ sở dữ liệu.

### Xem bình luận



Hình 2.12 Biểu đồ sequence chức năng xem bình luận.

Bước 1: Người dùng sẽ chọn tính năng xem bình luận trên ứng dụng.

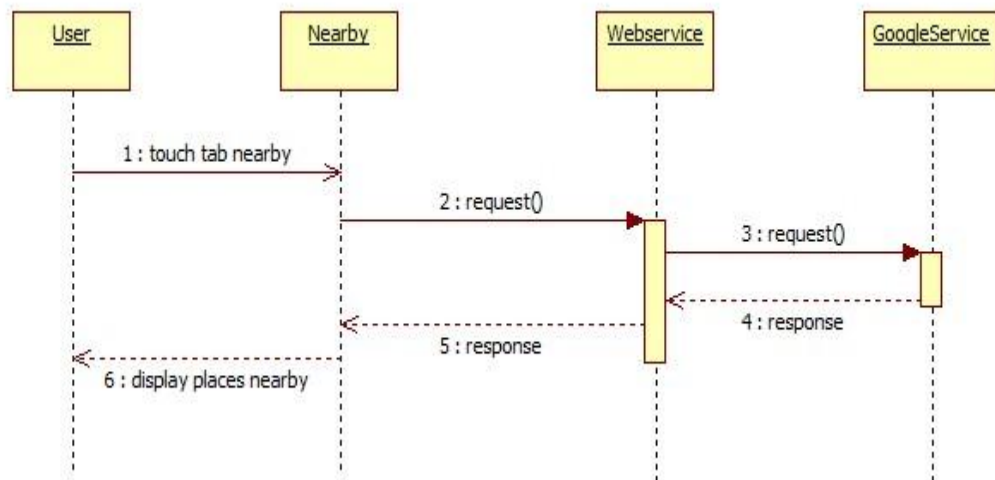
Bước 2: Ứng dụng sẽ gửi yêu cầu lên Webservice.

Bước 3 + 4: Webservice sẽ lấy dữ liệu từ database và database sẽ trả kết quả về cho webservice.

Bước 5: Webservice trả kết quả về cho ứng dụng.

Bước 6: Ứng dụng sẽ hiển thị kết quả cho người dùng xem.

### **Liệt kê địa điểm ăn uống trong phạm vi bán kính 5km xung quanh**



*Hình 2.13 Biểu đồ sequence chức năng liệt kê địa điểm.*

Bước 1: Người dùng sẽ chạm vào tab nearby của ứng dụng.

Bước 2: Ứng dụng sẽ gửi yêu cầu lên Webservice.

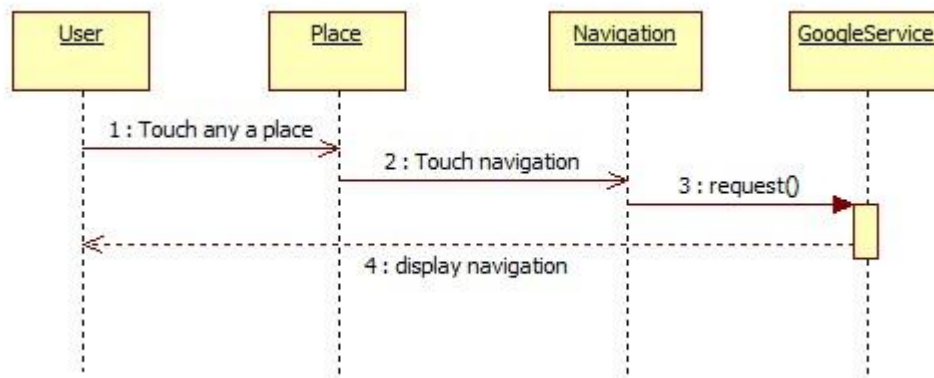
Bước 3: Webservice sẽ xử lý và gửi yêu cầu lên GoogleService để lấy địa điểm.

Bước 4: GoogleService trả về kết quả cho Webservice.

Bước 5: Webservice trả về kết quả cho Nearby.

Bước 6: Ứng dụng sẽ hiển thị các địa điểm trong phạm vi bán kính mà người dùng cài đặt ở xung quanh vị trí hiện tại.

### **Chỉ dẫn đường đi sau khi chọn địa điểm**



Hình 2.14 Biểu đồ sequence dẫn đường đến một địa điểm.

Bước 1: Người dùng chạm vào bất kì một địa điểm nào đó. Giao diện sẽ hiển thị navigation.

Bước 2: Tiếp theo người dùng chọn navigation.

Bước 3: Ứng dụng sẽ gửi request lên GoogleService để xử lí.

Bước 4: GoogleService sẽ gửi kết quả trả về cho ứng dụng. Ứng dụng sẽ hiển thị dẫn đường đến địa điểm mà người dùng vừa chọn.

## 2.3 Tìm hiểu công nghệ

### 2.3.1 Node.js

#### 2.3.1.1 Node.js là gì?

Node.js[5] là một nền tảng server side xây dựng trong JavaScript Engine của Google Chrome (V8 Engine). Nó được phát triển vào năm 2009 bởi Ryan Dahl và bản version gần đây nhất là v0.10.36. Node.js rất thích hợp để xây dựng các ứng dụng có tốc độ cao và dễ dàng mở rộng một cách dễ dàng. Do sử dụng cơ chế event-driven và áp dụng theo mô hình non-blocking I/O nên làm cho Node trở nên nhẹ nhàng và hiệu quả hơn rất thích hợp cho các ứng dụng thời gian thực.

Node.js là một mã nguồn mở, đa nền tảng chạy trong môi trường server và các ứng dụng mạng. Các ứng dụng Node.js được viết bằng ngôn ngữ JavaScript và có thể chạy trong các hệ điều hành OSX, MW, Linux.

### 2.3.1.2 Các đặc trưng cơ bản của Node.js

Node.js có một thư viện dồi dào với nhiều module JavaScript khác nhau tạo ra sự dễ dàng trong việc phát triển các ứng dụng web.

- Dưới đây là những điểm quan trọng làm cho Node.js là sự lựa chọn hàng đầu cho các kỹ sư phần mềm:
- Asynchronous and Event Driven: Tất cả các thư viện API của Node.js đều là asynchronous và được viết theo mô hình non-blocking. Về cơ bản nghĩa là Node.js sẽ không bao giờ đợi bất kỳ API nào thực hiện xong rồi trả về kết quả mà nó sẽ di chuyển đến API tiếp theo bằng cách dùng một cơ chế thông báo của Events nằm trong Node.js để giúp server có thể nhận phản hồi từ API trước đó.
- Very fast: Là một nền tảng được xây dựng trên Chrome's V8 JavaScript Engine. Thư viện của Node.js rất dễ dàng trong quá trình viết code.
- Single Threaded dễ mở rộng hơn: Node.js sử dụng mô hình single threaded với cơ chế event looping. Với cơ chế này nó sẽ giúp server đáp ứng một cách non-blocking và làm cho server mở rộng một cách dễ dàng trái ngược với các server trước kia là tạo ra các threads hạn chế để xử lý yêu cầu. Node.js sử dụng single threaded để tạo ra một số lượng request lớn hơn các server trước kia. VD: Apache HTTP Server.
- No Buffering: Các ứng dụng Node.js không bao giờ buffer bất kỳ dữ liệu nào hết. Các ứng dụng này chỉ đơn giản là output dữ liệu trong một vùng nhất định.
- Tooling: NPM viết tắt Node.js Package Manager là một công cụ dùng để quản lý gói cài đặt của Node.js. Nó được dùng để cài đặt các chương trình Node.js từ npm registry. Bằng cách tổ chức cài đặt và quản lý của bên thứ 3, nên nó sẽ giúp các developers xây dựng nhanh hơn. NPM không được dùng để load code thay vào đó nó



được sử dụng để cài đặt và quản lý mã phụ thuộc vào dòng lệnh. Các gói được tìm thấy trong npm registry có thể chạy từ các thư viện đơn giản như Underscore.js để thực hiện nhiệm vụ Grunt.

#### **2.3.1.3 Những trường hợp nên sử dụng Node.js**

- I/O bound Applications
- Data Streaming Applications
- Data Intensive Realtime Applications
- JSON APIs based Applications
- Single Page Applications

#### **2.3.1.4 Những trường hợp không nên sử dụng Node.js**

- Không nên sử dụng Node.js cho các ứng dụng hao tốn tài nguyên.
- Một ứng dụng chỉ toàn CRUD.

### **2.3.2 MongoDB**

#### **2.3.2.1 MongoDB là gì?**

MongoDB[4] là một mã nguồn mở và là một tập tài liệu dùng cơ chế NoSQL để truy vấn, nó được viết bởi ngôn ngữ C++. Chính vì được viết bởi C++ nên nó có khả năng tính toán với tốc độ cao chứ không giống như các hệ quản trị CSDL hiện nay. MongoDB có cấu trúc như JSON, chính vì thế nó có hiệu suất cao, tương tác nhanh và khả năng mở rộng tốt. Nó hoạt động trên khái niệm collection và document.

#### **2.3.2.2 Collection, Document là gì?**

Collection là một nhóm tài liệu (document), nó tương tự như một table trong CSDL, thông thường một Collection thuộc về một database duy nhất tuy nhiên nó không có ràng buộc nên khả năng truy suất rất nhanh. Vì vậy mỗi Collection có thể chứa nhiều dạng khác nhau chứ không ràng buộc field cố định như trong hệ quản trị CSDL.

Document trong MongoDB có cấu trúc tương tự như kiểu dữ liệu JSON, nghĩa là sẽ có các cặp (key => value) nên nó có tính năng động rất lớn. Document có thể hiểu nó giống như các record dữ liệu trong MySQL, tuy

nhiên nó có sự khác biệt là các cặp (key => value) có thể không giống nhau ở mỗi document.

### **2.3.2.3 Ưu điểm của MongoDB so với RDBMS**

- Schema less: MongoDB là cơ sở dữ liệu tài liệu trong đó một collection sẽ nắm giữ các tài liệu khác nhau. Số lượng các trường, nội dung, kích thước có thể khác nhau với document khác.
- Dữ liệu được caching (ghi đệm) lên RAM, hạn chế truy cập vào ổ cứng nên tốc độ truy xuất dữ liệu cao.
- Dễ dàng mở rộng lưu trữ.
- Mô hình dữ liệu và truy vấn linh hoạt.
- Không phức tạp khi nhập dữ liệu vào.
- Cấu trúc single object rõ ràng.

### **2.3.2.4 Tại sao nên sử dụng MongoDB**

- Cơ sở dữ liệu lưu trữ theo hướng tài liệu: Dữ liệu được lưu trữ dưới dạng JSON.
- Bất kỳ thuộc tính nào cũng đều có tính chỉ mục.
- Replication và High Availability: Do chấp nhận sự trùng lặp trong lưu trữ nên nếu một node nào đó bị chết cũng không ảnh hưởng tới toàn bộ hệ thống.
- Auto-Sharding: Là cơ chế tự động của MongoDB dùng để chia tách một dữ liệu kích thước lớn cho rất nhiều server (thường gọi là cluster).
- Rich Queries: Cung cấp dồi dào các câu truy vấn.
- Fast In-Place Updates: Cập nhật nhanh chóng.

### **2.3.2.5 Một số điểm trong việc thiết kế sơ đồ trong MongoDB**

- Thiết kế sơ đồ theo yêu cầu của người dùng.
- Kết hợp các đối tượng vào trong một document nếu cần sẽ sử dụng chúng với nhau. Nếu không muốn thì hãy tách chúng ra.

- Dữ liệu được phép trùng lặp (nhưng có giới hạn).
- Tối ưu hóa sơ đồ để sử dụng nó thường xuyên.
- Ví dụ về cấu trúc trong MongoDB

```
{
  _id: POST_ID
  title: TITLE_OF_POST,
  description: POST_DESCRIPTION,
  by: POST_BY,
  url: URL_OF_POST,
  tags: [TAG1, TAG2, TAG3],
  likes: TOTAL_LIKES,
  comments: [
    {
      user:'COMMENT_BY',
      message: TEXT,
      dateCreated: DATE_TIME,
      like: LIKES
    },
    {
      user:'COMMENT_BY',
      message: TEXT,
      dateCreated: DATE_TIME,
      like: LIKES
    }
  ]
}
```

*Hình 2.15 Cấu trúc trong MongoDB*

### 2.3.3 Amazon Web Services

#### 2.3.3.1 Giới thiệu

Amazon Web Services[1] (AWS) là dịch vụ điện toán đám mây được cung cấp bởi amazon.com. Dịch vụ được chính thức đưa ra thị trường vào năm 2006, nhanh chóng đạt được mốc 180.000 nhà phát triển đăng ký sử dụng vào 6/2007. Cùng với sự phát triển của điện toán đám mây trong những năm gần

đây, AWS đạt được mức tăng trưởng ấn tượng và là nhà cung cấp điện toán đám mây có doanh thu lớn nhất thế giới hiện nay (ước tính 3.8 tỉ \$ năm 2013). Hai dịch vụ trung tâm nền tảng mà AWS cung cấp là EC2 (Computing) và S3 (Storage).

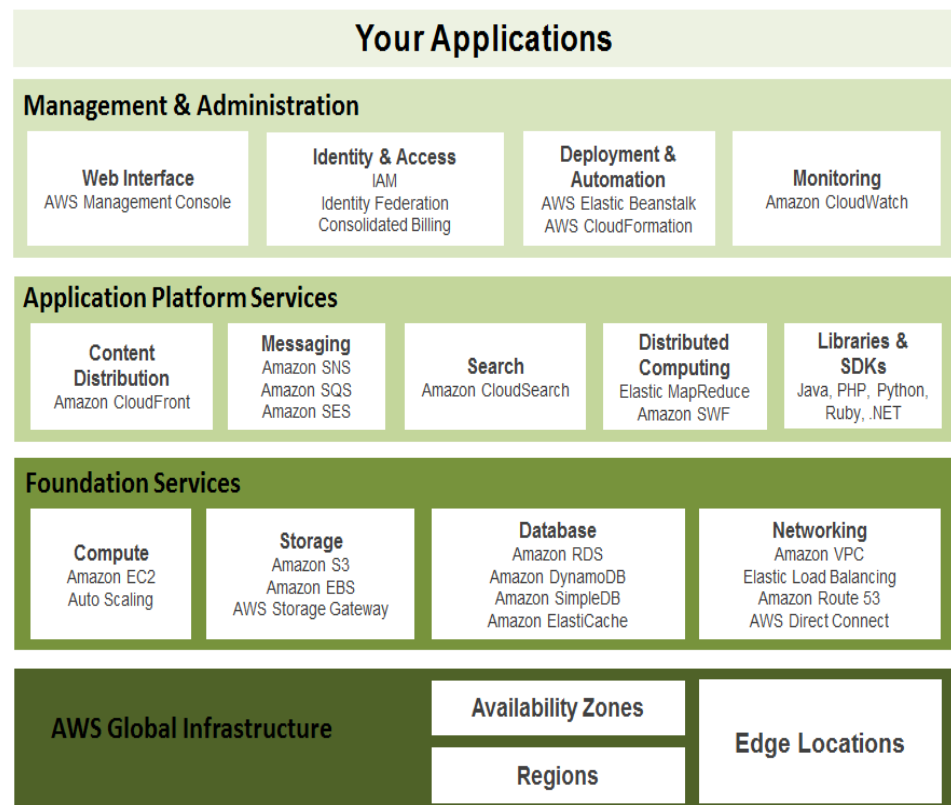
Amazon Elastic Compute Cloud (EC2) là nền tảng cơ sở cho môi trường điện toán đám mây do Amazone cung cấp. EC2 cho phép người dùng có thể thuê máy tính ảo để chạy các ứng dụng riêng của họ. EC2 cho phép khả năng mở rộng triển khai các ứng dụng bằng cách cung cấp một dịch vụ web, thông qua đó người dùng có thể khởi động một Amazon Machine Image để tạo một máy ảo mà Amazon gọi nó là một “instance” và chứa bất kỳ phần mềm mong muốn. Người dùng có thể tạo, khởi động, và chấm dứt máy chủ trong trường hợp cần thiết và trả tiền cho các máy chủ đang hoạt động, chính vì thế nó được gọi là “elastic”. EC2 cho phép người dùng có quyền kiểm soát vị trí địa lý trong trường hợp mà nó cho phép tối ưu hóa độ trễ và sự dư thừa ở cấp độ cao.

#### **2.3.3.2 Kiến trúc**

AWS có 9 trung tâm dữ liệu server tương ứng với 9 khu vực địa lý khác nhau tương ứng với các Availability Zones:

Một số dịch vụ của AWS được cung cấp để chạy cùng lúc trên nhiều khu vực khác nhau (S3, DynamoDB). Một số dịch vụ khác có thể được cấu hình để sao chép thành nhiều bản sao ở nhiều khu vực khác nhau để tăng cường tốc độ truy xuất và khả năng dự phòng.

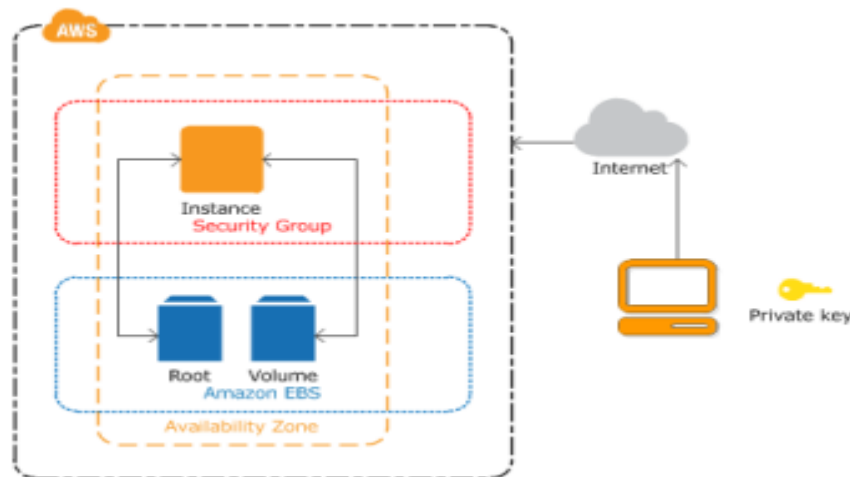
Người sử dụng dịch vụ truy cập qua 3 hình thức: HTTP, SOAP, REST.



Hình 2.16 Cấu trúc các tầng Amazon Web Services.

### Amazon Web Services gồm 3 tầng dịch vụ chính:

- Tầng dịch vụ nền tảng: bao gồm các dịch vụ nền tảng của điện toán đám mây như dịch vụ điện toán (Amazon EC2), phần cứng lưu trữ, cơ sở dữ liệu và mạng (Amazon VPC, AWS Direct Connect).
- Tầng dịch vụ ứng dụng nền tảng: bao gồm các dịch vụ như truyền tin, phân phối nội dung (Amazon CloudFront), các thư viện và SDKs cho Java, PHP, Python...
- Tầng quản trị và quản lý: bao gồm các dịch vụ quản lý (Amazon Cloud Watch), dịch vụ triển khai và tự động hóa (Amazon Elastic Beanstalk)



Hình 2.17 Sơ đồ hoạt động Amazon EC2.

Nhằm đảm bảo tính bảo mật của hệ thống máy tính cá nhân sử dụng private key mà Amazon cung cấp cho riêng instance đó kết nối và quản lý instance đó thông qua mạng Internet. Amazon sẽ cung cấp một giao diện quản lý trên nền web giúp người sử dụng có thể dễ dàng quản trị các instance.

Quản trị instance như là có thể tạo ra các instance mới đồng thời quản lý thời gian chạy của các instance, tùy vào nhu cầu sử dụng người dùng có thể bật tắt các instance này.

### 2.3.3.3 Tại sao lại sử dụng Amazon Web Services

Miễn phí sử dụng AWS trong vòng một năm.

Người dùng có thể trả phí theo hàng tháng hay hàng năm hoặc tùy theo thời lượng giờ sử dụng mà AWS sẽ thu phí tài khoản. Tạo nên sự linh hoạt trong việc thanh toán.

Các lợi ích chung của điện toán đám mây: Giá rẻ hơn, triển khai dịch vụ nhanh hơn, doanh nghiệp tập trung vào phát triển chuyên ngành của mình hơn, phần hệ thống back-end không mất nhiều chi phí và công sức do đã được thuê ngoài, ...

### 2.3.3.4 Một số nét đặc trưng của Amazon EC2

- Các môi trường của máy tính ảo, được biết như là instances

- Cấu hình lại các mẫu instances, được biết như là Amazon Machine Images (AMIs), trong đó các gói bit sẽ được dùng cho server (bao gồm hệ điều hành và phần mềm bổ sung).
- Các cấu hình khác nhau của CPU, bộ nhớ, lưu trữ và networking capacity của instances, được gọi là instance types.
- Thông tin đăng nhập an toàn của instances đều sử dụng key pairs (AWS chứa các public key, và nên được lưu trữ các private key ở một nơi nào đó an toàn).
- Dữ liệu tạm thời sẽ bị xóa khi tài khoản ngừng hoặc chấm dứt instance, được gọi là instance store volumes.
- Lưu trữ dữ liệu liên tục bằng việc sử dụng Amazon Elastic Block Store (Amazon EBS), được gọi là Amazon EBS volumes.
- Nhiều vị trí cho các tài nguyên, chẳng hạn như các instance và Amazon EBS volumes, được gọi là regions và Availability Zones.
- Một tường lửa sẽ cho phép chỉ định các giao thức, cổng, và các source IP range có thể tiếp cận đến các instance bằng cách sử dụng security groups.
- Địa chỉ IP tĩnh dùng cho dynamic cloud computing, được gọi là Elastic IP addresses.
- Metadata, được gọi là các thẻ, mà có thể được tạo và gán đến các tài nguyên Amazon EC2.
- Các Virtual network được tạo ra sẽ được cách li với phần còn lại của AWS cloud, và hoàn toàn có thể tùy chọn kết nối với mạng riêng, được gọi là virtual private clouds (VPCs).

#### **2.3.3.5 Ưu – Nhược điểm của Amazon Web Services**

Dịch vụ đa dạng, có nhiều dịch vụ kèm theo về deployment và management do có ưu thế về kinh nghiệm.

Giá dịch vụ tương đối cạnh tranh so với các nhà cung cấp khác.

Có 9 trung tâm máy chủ, nằm trải rộng trên các khu vực địa lý khác nhau, phù hợp với nhiều thị trường.

Có số người sử dụng lớn nhất trong các nhà cung cấp dịch vụ, document chi tiết, các hướng dẫn và khắc phục sự cố có thể dễ dàng tìm được trên internet.

Hệ thống do nhà cung cấp dịch vụ có thể không an toàn về mặt bảo mật nếu như lưu trữ các thông tin nhạy cảm.

## **2.3.4 Hệ điều hành Android**

### **2.3.4.1 Giới thiệu**

Android[2] là một hệ điều hành do Google phát triển dựa trên nền tảng Linux, và được thiết kế dành riêng cho các thiết bị di động có màn hình cảm ứng như điện thoại thông minh và máy tính bảng. Khác với các hệ điều hành trên thiết bị di động khác như iOS và Window Phone, Android là hệ điều hành mã nguồn mở. Hiện tại, Android đã phát triển tới phiên bản 5.0 và đang là hệ điều hành phổ biến nhất dành cho thiết bị di động.

### **2.3.4.2 Cấu trúc ứng dụng**

Ứng dụng Android được phát triển dựa trên bộ công cụ Android SDK và ngôn ngữ lập trình Java, cấu tạo gồm bốn thành phần chính:

- Activity: là một màn hình giao diện chịu trách nhiệm tương tác với người dùng. Một ứng dụng Android thường bao gồm một hay nhiều Activity tùy theo thiết kế của người phát triển. Những Activity của cùng một ứng dụng có cấu trúc độc lập với nhau, và có thể được khởi động từ ứng dụng khác nếu được cho phép.
- Service: là thành phần đảm nhiệm công việc thực hiện những tác vụ ngầm của ứng dụng và không cung cấp giao diện tương tác với người dùng. Việc sử dụng kết hợp Activity và Service đảm bảo cho ứng dụng có khả năng thực hiện những tác vụ mất nhiều thời gian



n như tính toán hay thao tác trên dữ liệu mà không ảnh hưởng đến tương tác trên giao diện ứng dụng.

- Content provider: là thành phần quản lý việc chia sẻ dữ liệu giữa các ứng dụng. Thông qua Content provider, một ứng dụng khác có thể truy suất và sửa đổi dữ liệu của ứng dụng khác nếu được ứng dụng đó cho phép.
- Broadcast receivers: là thành phần đảm nhận nhiệm vụ thu nhận và xử lý những thông báo từ hệ thống. Trong quá trình hoạt động, hệ thống có thể tạo những thông báo và truyền đi đến ứng dụng. Ngược lại, ứng dụng cũng có thể tạo thông báo và truyền đến các ứng dụng khác thông qua hệ thống.

Đặc trưng của ứng dụng Android so với ứng dụng trên các hệ điều hành khác là một ứng dụng có thể khởi động bất kỳ thành phần nào của ứng dụng khác nếu được ứng dụng đó cho phép. Lấy ví dụ nếu người phát triển muốn trang bị khả năng chụp ảnh cho ứng dụng, thay vì tự hiện thực có thể sử dụng tính năng chụp ảnh của ứng dụng khác có sẵn trong hệ thống. Như vậy, người phát triển có thể tập trung vào hiện thực những tính năng đặc trưng trên ứng dụng của mình.

#### **2.3.4.3 Tính tương thích**

Android là một hệ điều hành được thiết kế để có thể sử dụng trên nhiều loại thiết bị khác nhau như điện thoại, máy tính bảng, tivi... Mỗi loại thiết bị có những thông số khác nhau về mặt kỹ thuật như kích thước màn hình, các phần cứng được cung cấp, phiên bản hệ điều hành, ... đòi hỏi người phát triển phải nắm rõ yêu cầu và mục tiêu để ứng dụng có thể hoạt động mượt mà và ổn định.

Về cơ bản, tính tương thích có thể chia thành hai loại:

- Tương thích thiết bị: với đặc tính là một hệ điều hành mã nguồn mở, những nhà phát triển thiết bị thường tiến hành việc tùy chỉnh hệ điều hành Android sao cho phù hợp và tạo được đặc trưng cho dòng sản

phẩm của mình. Tuy nhiên, để có thể sử dụng được các dịch vụ của Google như Google Play Store, mỗi thiết bị phải vượt qua bài kiểm tra tương thích (Compatibility Test Suit) để đảm bảo tính tương thích với các ứng dụng đã và đang được phát triển.

- Tương thích ứng dụng: khác với tính tương thích thiết bị, vốn được giải quyết bởi nhà sản xuất, người phát triển phải đảm bảo cho ứng dụng của mình tương thích với những đặc tính của nhiều thiết bị khác nhau, bao gồm những phần cứng thiết bị cung cấp, kích cỡ màn hình và phiên bản hệ điều hành. Việc này đòi hỏi người phát triển phải xác định rõ ràng mục tiêu và phạm vi của ứng dụng, cũng như nhóm thiết bị trọng tâm để có thể đưa ra các giải pháp thiết kế phù hợp nhằm tối đa hóa trải nghiệm người dùng.

#### **2.3.4.4 Tài nguyên của ứng dụng**

Một ứng dụng Android còn bao gồm nhiều loại tài nguyên khác nhau như hình ảnh, âm thanh, và các loại tài nguyên liên quan trực tiếp đến hiển thị hình ảnh của ứng dụng. Những loại tài nguyên này bao gồm các file hình ảnh, định dạng hiệu ứng, màu sắc, bố cục giao diện. Việc quản lý tài nguyên riêng biệt với phần hiển thị thực có những lợi ích như sau:

- Dễ dàng cập nhật, thay đổi các loại tài nguyên mà không cần thay đổi mã nguồn hiển thị ứng dụng.
- Cho phép cung cấp nhiều loại tài nguyên khác nhau tùy theo đặc tính của từng thiết bị và thiết lập của người dùng. Khi khởi động, ứng dụng sẽ dựa vào thông số và thiết lập trên từng thiết bị như kích cỡ màn hình, phiên bản hệ điều hành, lựa chọn ngôn ngữ mà truy xuất những tài nguyên tương ứng. Điều này cho phép ứng dụng đảm bảo hiệu năng hoạt động cũng như tối đa hóa trải nghiệm người dùng.

#### **2.3.5 Google Map**

Google Map[3] là một dịch vụ bản đồ do Google phát triển trên cả hai nền tảng máy tính để bàn và thiết bị di động. Dịch vụ cung cấp cho người dùng

những tính năng cơ bản như bản đồ vệ tinh, bản đồ tuyến đường, cùng với những tính năng nâng cao như dẫn đường, xác định vị trí thông qua GPS, tìm đường đi ngắn nhất giữa hai địa điểm.

Song song với cung cấp dịch vụ bản đồ, Google cũng phát triển bộ giao thức lập trình (API) cho phép những nhà phát triển ứng dụng tích hợp và sử dụng những dịch vụ bản đồ của Google Map. Những tính năng mà bộ giao thức này cho phép bao gồm:

- Cung cấp nền tảng bản đồ như bản đồ vệ tinh, bản đồ tuyến đường.
- Hỗ trợ định vị dựa trên hệ thống định vị toàn cầu GPS.
- Tích hợp những thư viện đồ họa và hiệu ứng cho phép tùy chỉnh trên bản đồ.
- Cung cấp những dịch vụ nâng cao thông qua Web service như tìm đường đi ngắn nhất giữa hai địa điểm, dẫn đường, phân giải địa điểm dựa trên tên địa điểm hoặc địa chỉ GPS...

So với các nền tảng bản đồ của các nhà cung cấp khác như Bing Map của Microsoft, HERE Map của Nokia, hay nền tảng bản đồ mở OpenStreetMap, Google Map vượt trội hơn về độ chi tiết của dữ liệu cũng như sự đa dạng của các dịch vụ hỗ trợ. Tuy nhiên, một hạn chế của những dịch vụ bản đồ này là phụ thuộc vào kết nối Internet để có thể hoạt động.

## **2.3.6 Spring Framework**

### **2.3.6.1 Giới thiệu**

Spring là một cấu trúc dùng để xây dựng chương trình ứng dụng mã nguồn mở dành cho ngôn ngữ lập trình Java. Hàng triệu nhà phát triển trên khắp thế giới sử dụng Spring Framework để dễ dàng tái sử dụng, kiểm thử đoạn code và tăng tốc độ cho hệ thống.

Spring Framework được viết bởi Rod Johnson và được ra mắt vào tháng 6 năm 2003 dưới giấy phép Apache – phiên bản 2.0. Spring là một framework nhỏ gọn, bản cài đặt của nó chỉ khoảng 2MB.

Các đặc trưng chính của Spring Framework có thể được sử dụng trong phát triển bất kỳ ứng dụng Java nào, ngoài ra còn có các phần mở rộng khác dành cho việc xây dựng ứng dụng web trên nền Java Enterprise Edition. Mục tiêu của Spring Framework nhằm làm việc phát triển Java 2 Enterprise Edition dễ dàng hơn trong sử dụng và cải tiến khả năng lập trình thông qua áp dụng mô hình lập trình trên POJO.

### **2.3.6.2 Lợi ích khi sử dụng Spring Framework**

Spring cho phép nhà phát triển xây dựng các ứng dụng enterprise-class sử dụng POJOs. Lợi ích của việc chỉ dùng POJOs là không cần một sản phẩm Enterprise Java Bean container như một ứng dụng server và có thêm sự lựa chọn sử dụng servlet container như Tomcat hay vài sản phẩm thương mại khác.

Spring tổ chức theo kiểu module. Cho dù có một lượng lớn các gói hay lớp, nhưng sẽ chỉ cần quan tâm đến một vài cái cần thiết và bỏ qua phần còn lại.

Spring không làm việc nghiên cứu lại những thứ đã có, nó sử dụng các công nghệ sẵn có, ví dụ như các logging framework, Quartz cũng như các công nghệ khác.

Việc kiểm thử ứng dụng viết bởi Spring rất dễ dàng vì phần code phụ thuộc môi trường đã có trong framework. Cho đến nay, bằng sử dụng JavaBean-style POJOs, mọi việc trở nên dễ dàng hơn khi sử dụng dependency injection để đưa vào các dữ liệu kiểm thử.

Web framework của Spring là một web MVC framework được thiết kế hoàn hảo, cho phép một sự thay thế tuyệt vời cho các web framework khác như Struts hay các web framework phức tạp hoặc kém phổ biến khác.

Spring cung cấp một hệ thống API thuận tiện cho việc diễn dịch exception từ các công nghệ riêng biệt khác nhau (như Hibernate ) thành hệ thống unchecked exceptions thống nhất.

Lightweight Inverse of control containers có xu hướng rất nhẹ, đặc biệt là khi đem so sánh với EJB containers. Điều này vô cùng có lợi cho việc phát triển và triển khai các ứng dụng trên nền tảng máy tính bị giới hạn memory hay CPU resources.

Spring cung cấp một interface quản lý transaction thống nhất, cho phép giới hạn xuống từng local transaction (như sử dụng một CSDL duy nhất) và giới hạn lên thành global transactions.

### **2.3.6.3 Dependency Injection**

Công nghệ đặc thù nhất trong Spring chính là DI kế thừa từ IoC. IoC là một khái niệm tổng quát được thực hiện bằng nhiều cách khác nhau và DI là một trong các thực thể hóa của nó.

Khi viết một ứng dụng Java phức tạp, các lớp nên được viết độc lập nhất có thể với các lớp khác nhằm tăng khả năng tái sử dụng cũng như kiểm thử độc lập trong unit testing. DI sẽ giúp nối kết các class lại với nhau mà đồng thời vẫn giữ chúng độc lập.

Vậy rốt cục DI nó chính xác là cái gì? Hãy nhìn theo từng từ riêng biệt. Ở đây phần “dependency” được hiểu là một mối liên kết giữa hai class riêng biệt. Ví dụ, lớp A liên kết “dependent” đến lớp B. Bây giờ để ý đến phần thứ hai, injection. Chúng có nghĩa là class B sẽ được đưa – “inject” vào class A bằng IoC.

DI có thể thực hiện được bằng cách truyền biến tham số vào constructor hoặc dùng setter() trong post-construction. Và do DI là trái tim của Spring Framework, khái niệm này sẽ được giải thích trong một chương trình riêng biệt với một ví dụ thích hợp.

### **2.3.6.4 Aspect Oriented Programming**

Một trong những component quan trọng nhất của Spring chính là Aspect oriented programming (AOP) framework. Chức năng bắc cầu đa điểm trong một ứng dụng được gọi là “cross-cutting concerns”, theo khái niệm

chúng phân chia business logic của ứng dụng. Vài ví dụ phổ biến nhất của “aspects” là: logging, declarative transactions, security và caching...

Đơn vị chính của module trong OOP là “class”, trong khi với AOP thì nó là “aspect”. Trong khi DI giúp tách các object trong ứng dụng của mình ra riêng biệt, AOP sẽ giúp tách các “cross-cutting concerns” ra khỏi các objects mà chúng tác dụng lên.

Module AOP của Spring Framework cung cấp một phương pháp lập trình hướng “aspect” cho phép định nghĩa ra các “method-interceptors” và “pointcuts” để dễ dàng tách bóc các phần code thực hiện chức năng riêng biệt.

## **2.4 Xây dựng hệ thống**

### **2.4.1 Tương thích đa dạng thiết bị**

#### **Vấn đề**

Với đặc điểm là hệ điều hành mã nguồn mở, cùng với việc được sử dụng cho nhiều loại thiết bị có thông số kỹ thuật khác nhau như kích thước màn hình, mật độ điểm ảnh, phiên bản hệ điều hành, việc hiện thực ứng dụng phải đảm bảo tương thích với đa dạng thiết bị nhằm mở rộng phạm vi sử dụng đến nhiều đối tượng người dùng.

#### **Hướng giải quyết**

Trong phạm vi của đề tài, ứng dụng được hiện thực có những đặc tính tương thích như sau:

- Hỗ trợ phiên bản hệ điều hành Android 4.0 trở lên, hiện đang chiếm hơn 90% trong các phiên bản đang được sử dụng.
- Thiết kế các icon phù hợp với các thiết bị có mật độ điểm ảnh khác nhau.
- Giao diện ứng dụng được thiết kế tương thích với kích cỡ của điện thoại di động, là nhóm thiết bị mục tiêu mà ứng dụng đang hướng tới.

### **2.4.2 Xử lý và hiển thị hình ảnh**

#### **Vấn đề**

Hình ảnh là thành phần tốn nhiều dung lượng lưu trữ của hệ thống, cũng như thời gian truyền tải giữa client và server. Với các thiết bị di động có bộ nhớ giới hạn, việc lưu trữ và hiển thị hình ảnh thô chụp từ camera chưa qua xử lý trong ứng dụng có thể dẫn tới tràn bộ nhớ, nhất là trong những màn hình thể hiện nhiều hình ảnh cùng một lúc như trong danh sách kết quả tìm kiếm hay tin tức cập nhật.

### **Hướng giải quyết**

Đề tài sử dụng chiến thuật xử lý và lưu trữ hình ảnh như sau:

Hình ảnh của đề tài được lấy trên ứng dụng Foody và được lưu thành định dạng JPG, là định dạng sử dụng giải thuật nén nhằm giảm dung lượng bộ nhớ nhưng vẫn giữ được chất lượng hình ảnh.

### **2.4.3 Hiện thực hệ thống**

Trong phần này, báo cáo sẽ trình bày các tính năng của ứng dụng, một số thông tin trong quá trình hiện thực và miêu tả hình ảnh. Đây là những tính năng được hiện thực dựa trên cơ sở tính năng, cũng như kết quả của quá trình phân tích và thiết kế hệ thống như đã trình bày ở những nội dung trước.

Ứng dụng lựa chọn cách thiết kế sử dụng Fragment trong Android để tạo ra màn hình bố cục chung cho ứng dụng. Sau đó với từng màn hình cụ thể chỉ cần chỉnh sửa lại Fragment đó để phục vụ mục đích về tính năng.

### **Màn hình chính**

Màn hình chính của ứng dụng được thiết kế theo cấu trúc tab như đã trình bày ở phần phân tích. Các tab theo thứ tự bao gồm:

Tab News: quản lý những thông tin tin tức từ các địa điểm ăn uống như khuyến mãi, giảm giá.

Ảnh chi tiết



Hình 2.18 Giao diện tab News.

Mô tả chi tiết

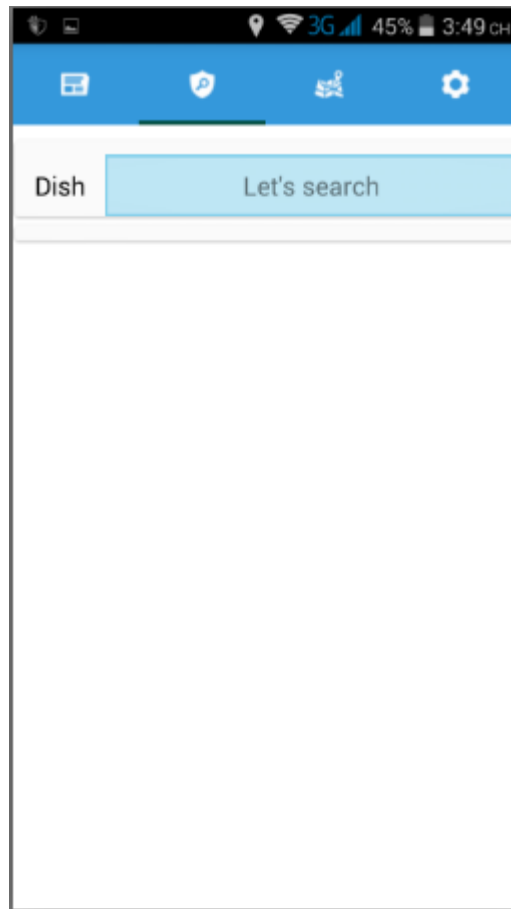
Bảng 2.5 Bảng mô tả tab News.

Tên	Loại	Ý nghĩa
Name	TextView	Tên địa điểm ăn uống
timeStamp	TextView	Thời gian diễn ra sự kiện
statusMsg	TextView	Dòng trạng thái của sự kiện
City	TextView	Tên thành phố
newsImageView	ImageView	Hình ảnh hiển thị của sự kiện
btnDetails	Button	Chuyển đổi qua trang xem chi tiết sự kiện

Tab Search: người dùng có thể chuyển qua màn hình tìm kiếm để tìm kiếm thông tin về những đối tượng trong ứng dụng bao gồm món ăn, nhà hàng.



Ảnh chi tiết



Hình 2.19 Giao diện tab Search.

Mô tả chi tiết

Bảng 2.6 Bảng mô tả tab Search.

Tên	Loại	Ý nghĩa
tgType	ToggleButton	Button chuyển đổi tính năng tìm kiếm giữa Dish và Shop
etSearch	EditText	Nơi nhập chữ để tìm kiếm
Adapter	SearchListAdapter	Nơi hiển thị danh sách tìm kiếm

Tab Map: nơi hiển thị vị trí các địa điểm ăn uống trên bản đồ. Ở đây người dùng có thể biết được vị trí hiện tại của mình và các địa điểm ăn uống

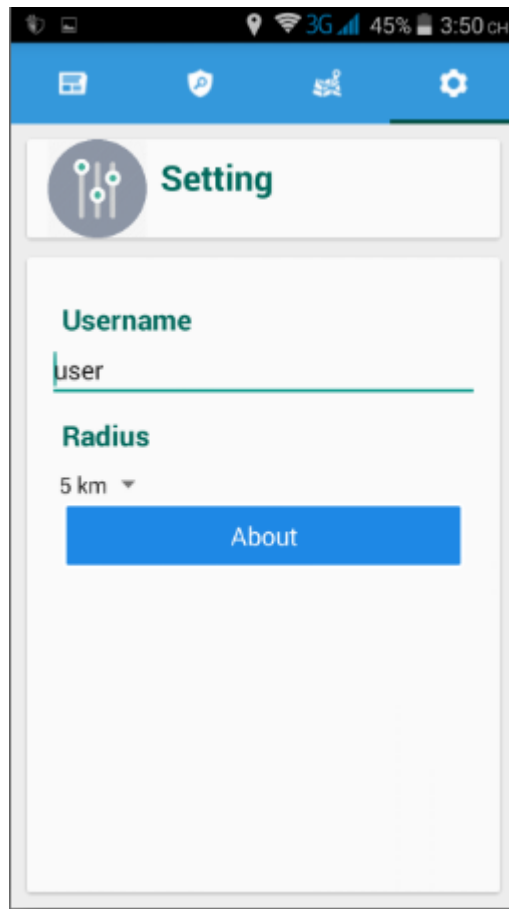
xung quanh bán kính, ngoài ra còn có chức năng chỉ đường đến địa điểm ăn uống.

Ảnh chi tiết



Hình 2.20 Giao diện tab Map.

Tab Setting: nơi hiển thị thông tin về ứng dụng và cũng là nơi để cài  
Ảnh chi tiết



Hình 2.21 Giao diện tab Setting.

Mô tả chi tiết

Bảng 2.7 Bảng mô tả tab Setting.

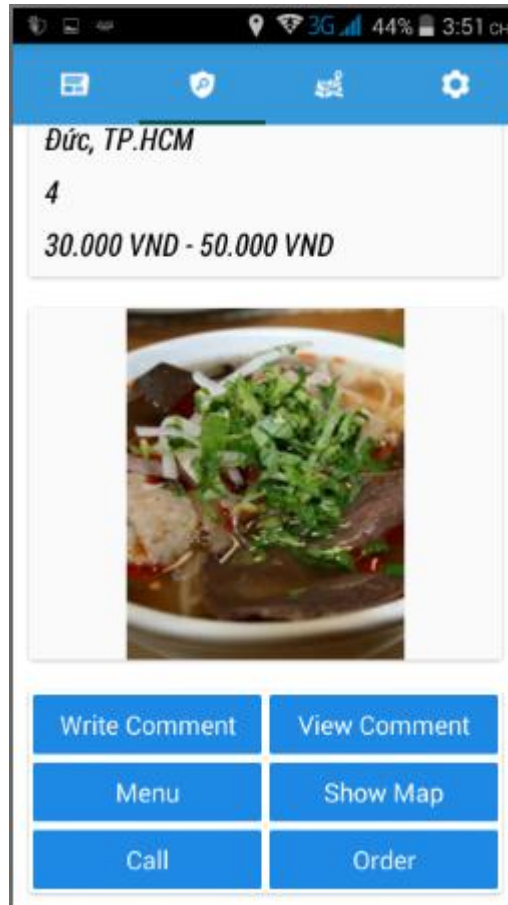
Tên	Loại	Ý nghĩa
etUsername	EditText	Nơi để nhập tên người dùng
spnRadius	Spinner	Thanh sổ xuống chứa phạm vi bán kính
bAbout	Button	Chuyển qua xem thông tin của ứng dụng

### Màn hình thông tin địa điểm

Nơi người dùng có thể xem thông tin chi tiết về địa điểm ăn uống bao gồm những thông tin như: tên địa điểm, địa chỉ, mức đánh giá, giá tiền, hình ảnh.

Ngoài ra ở màn hình này còn có các tính năng khác như: viết bình luận, xem bình luận, xem thực đơn, đặt hàng.

Ảnh chi tiết



Hình 2.22 Giao diện màn hình thông tin địa điểm.

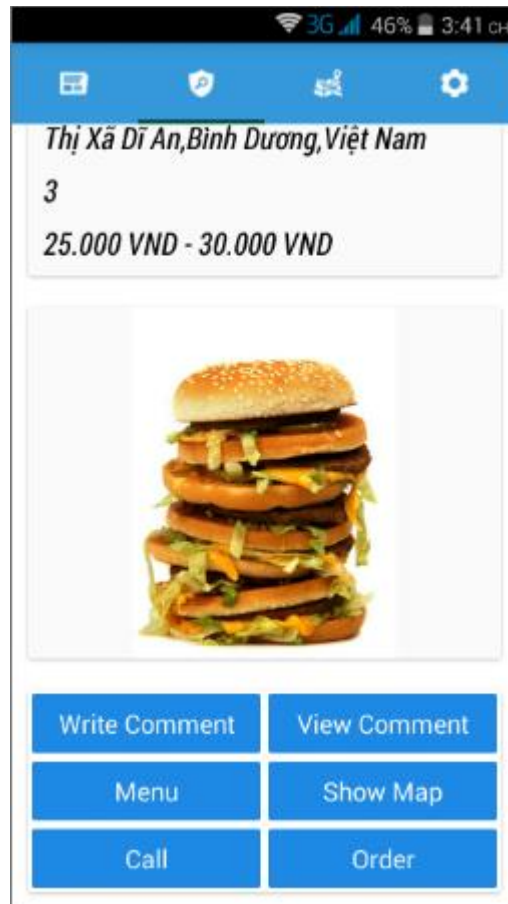
Mô tả chi tiết

*Bảng 2.8 Bảng mô tả chi tiết thông tin địa điểm.*

Tên	Loại	Ý nghĩa
tvShopName	TextView	Tên địa điểm ăn uống
tvShopAddress	TextView	Địa chỉ của địa điểm ăn uống
tvShopOCTime	TextView	Giờ mở cửa
tvShopRating	TextView	Đánh giá
tvShopPrice	TextView	Giá cả dao động của địa điểm
shopImage	Image	Hình ảnh của địa điểm
bCall	Button	Gọi đặt hàng qua điện thoại
bViewComment	Button	Xem nội dung bình luận
bViewDish	Button	Xem thực đơn của địa điểm
bOrder	Button	Đặt hàng qua website
bWriteComment	Button	Ghi bình luận

### Tính năng xem thông tin tin tức

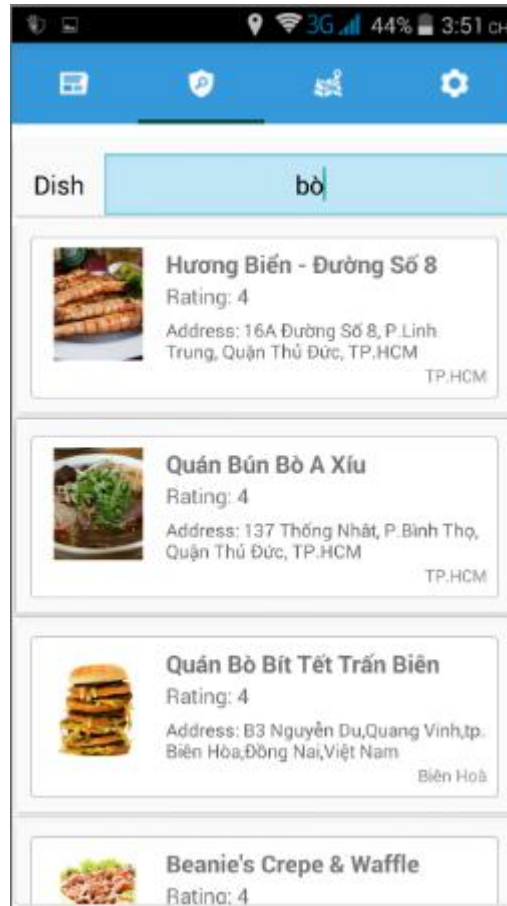
Người dùng có thể thực hiện tính năng này thông qua tab News của ứng dụng. Tính năng này giúp người dùng có thể xem thông tin chi tiết về một tin tức của một địa điểm ăn uống nào đó.



Hình 2.23 Giao diện thông tin tin tức.

### Tính năng tìm kiếm

Thông qua Tab Search ở màn hình chính, người dùng có thể tìm kiếm thông tin về những đối tượng bao gồm món ăn, nhà hàng, địa điểm ăn uống.

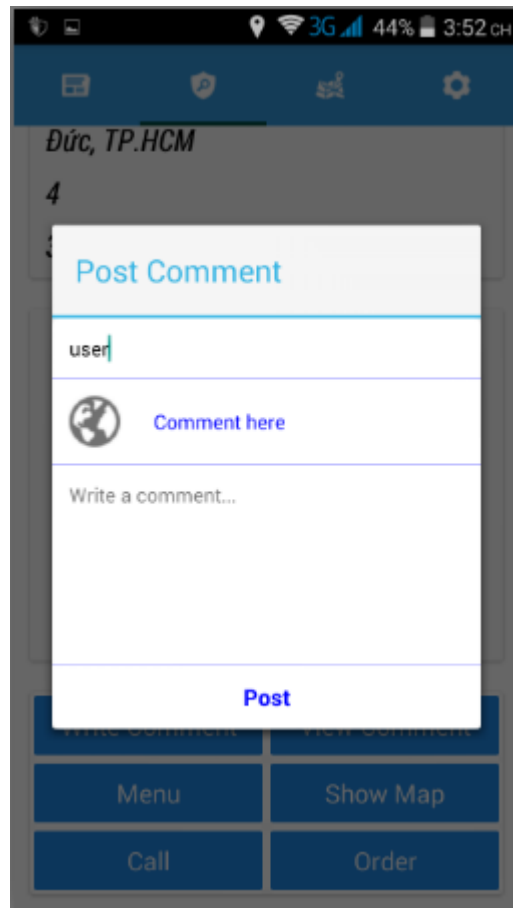


Hình 2.24 Giao diện tìm kiếm.

### Tính năng viết bình luận

Thông qua tính năng này người dùng có thể bình luận về một địa điểm ăn uống.

Ảnh chi tiết



Hình 2.25 Giao diện viết bình luận.

Mô tả chi tiết

Bảng 2.9 Bảng mô tả tính năng viết bình luận.

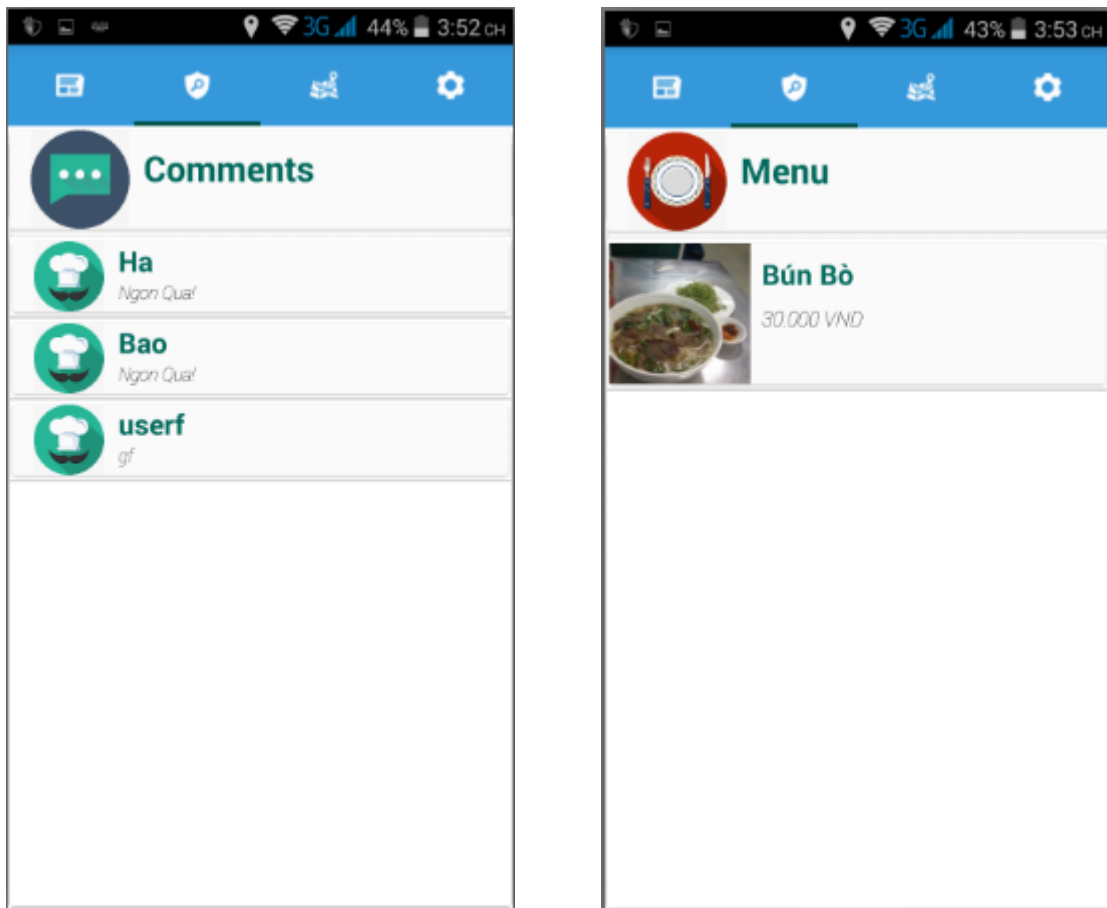
Tên	Loại	Ý nghĩa
tvComment	TextView	Tiêu đề của tính năng
editUser	EditText	Nơi nhập username
editContent	EditText	Nơi nhập nội dung bình luận
bPost	Button	Nơi post nội dung bình luận



### Tính năng xem bình luận, thực đơn của nhà hàng

Đây là những tính năng giúp người dùng có thể xem bình luận của những người dùng khác. Ngoài ra, người dùng còn có thể xem thực đơn của nhà hàng thông qua tính năng xem thực đơn.

Ảnh chi tiết



Hình 2.26 Giao diện xem bình luận và xem thực đơn.

Mô tả chi tiết

Bảng 2.10 Bảng mô tả chi tiết viết bình luận.

Tên	Loại	Ý nghĩa
tvComment	TextView	Tiêu đề xem bình luận
listView	ListView	Nơi chứa nội dung bình luận của các người dùng

*Bảng 2.11 Bảng mô tả chi tiết xem thực đơn.*

Tên	Loại	Ý nghĩa
tvMenu	TextView	Tiêu đề xem bình luận
listView	ListView	Nơi hiển thị danh sách các thực đơn của địa điểm

### Tính năng đặt hàng

Đây là tính năng giúp người dùng có thể đặt hàng thông qua điện thoại, khi tính năng này được thực thi thì màn hình sẽ chuyển qua chế độ gọi điện thoại cho nhà hàng đó. Bên cạnh đó, ứng dụng còn cho phép đặt hàng thông qua một form có sẵn, người dùng bắt buộc sẽ điền vào form rồi sau đó gửi đơn đặt hàng đến hệ thống.

Ảnh chi tiết

The screenshot shows a mobile application interface for ordering. At the top, there is a status bar with icons for signal, battery, and time (3:54 CH). Below the status bar is a blue header with four icons: a wallet, a shield, a person, and a gear. The main content area is white and contains the following elements:

- Total:** 30000 (in red text)
- Phone:** 01693141548 (in a text input field)
- Menu:** Salad Bò Tái (with a dropdown arrow)
- Quantity:** 1 (in a text input field)
- Buttons:** 'Add' and 'Order' (both in blue)
- Preview:** A small image of the 'Salad Bò Tái' dish, followed by the text 'Salad Bò Tái', '30.000 VND', and 'Quantity:1'.

*Hình 2.27 Giao diện đặt hàng.*

### Mô tả chi tiết

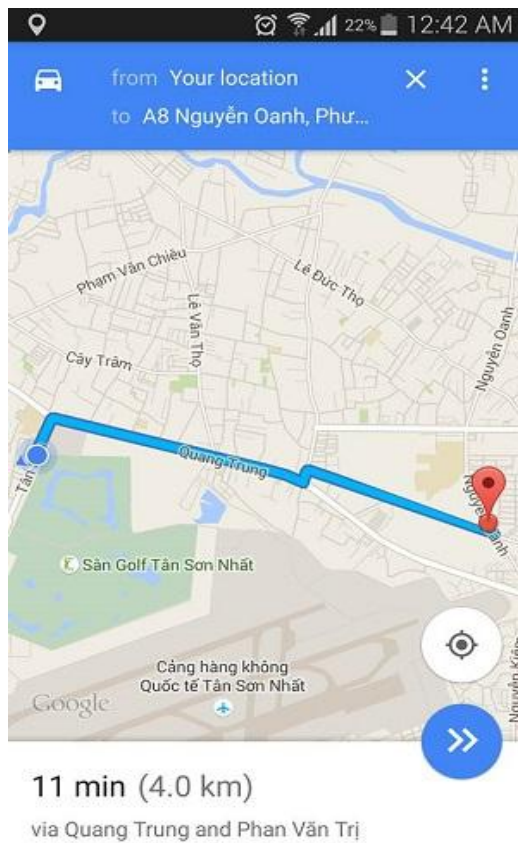
*Bảng 2.12 Bảng mô tả chi tiết giao diện đặt hàng.*

Tên	Loại	Ý nghĩa
tvOrder	TextView	Tiêu đề tính năng đặt hàng
tvTotal	TextView	Tổng chi phí của đơn đặt hàng
editPhone	EditText	Nơi nhập số điện thoại
spnDish	Spinner	Danh sách thực đơn trong nhà hàng
editQuantity	EditText	Nơi nhập số lượng
bAdd	Button	Thêm đơn đặt hàng vào để tính tiền
bOrder	Button	Gửi đơn đặt hàng

### Quản lý các địa điểm ăn uống

Cung cấp những công cụ quản lý địa điểm ăn uống bao gồm:

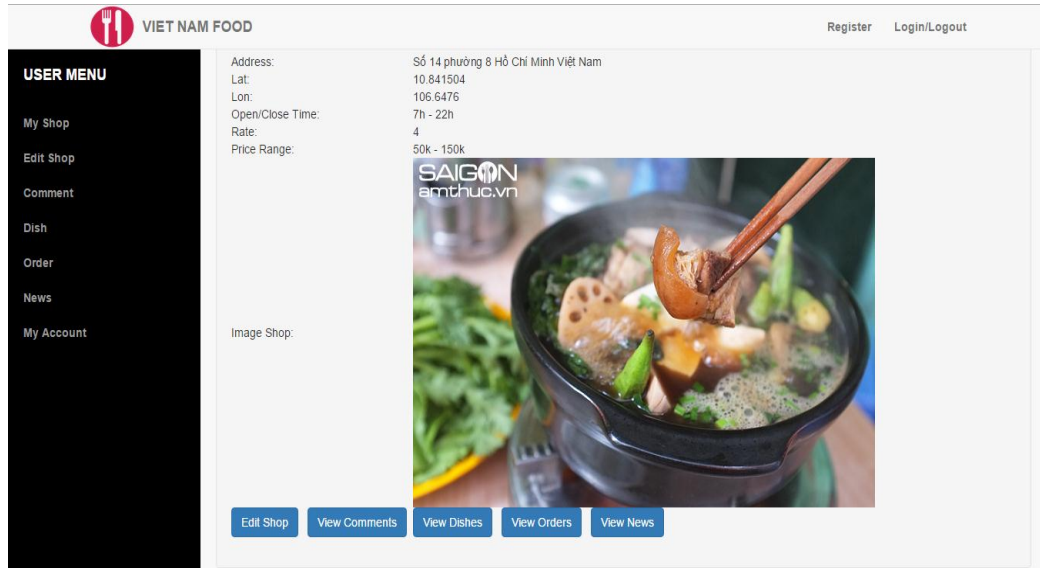
- Bản đồ dẫn đường đến một địa điểm
- Hiện thị địa điểm xung quanh vị trí hiện tại



Hình 2.28 Giao diện dẫn đường đến một địa điểm ăn uống và hiển thị các địa điểm.

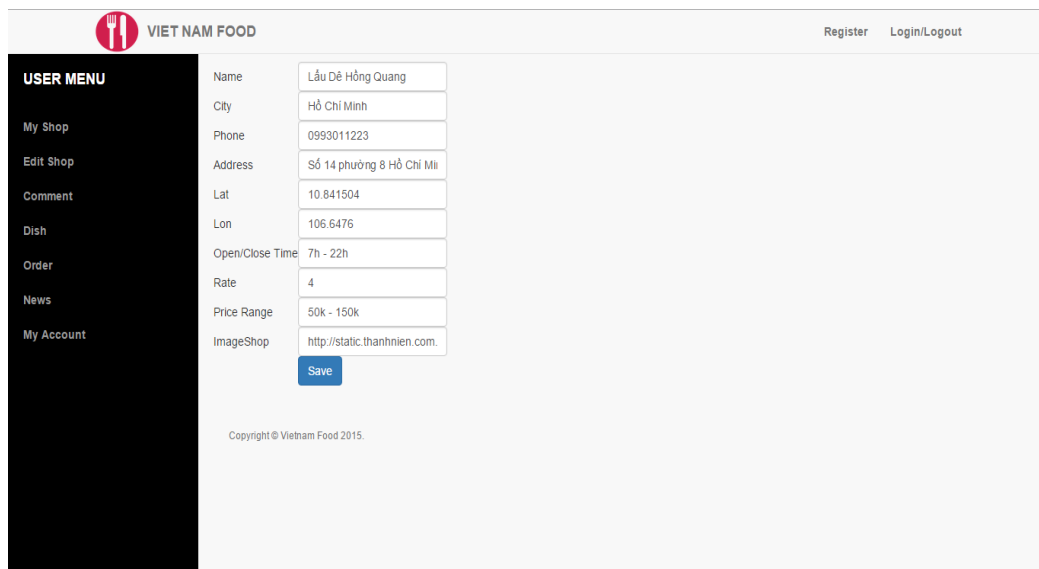
## Trang quản trị hệ thống

Đây là nơi để admin quản lý hệ thống bao gồm món ăn, tin tức, bình luận và các đơn đặt hàng.



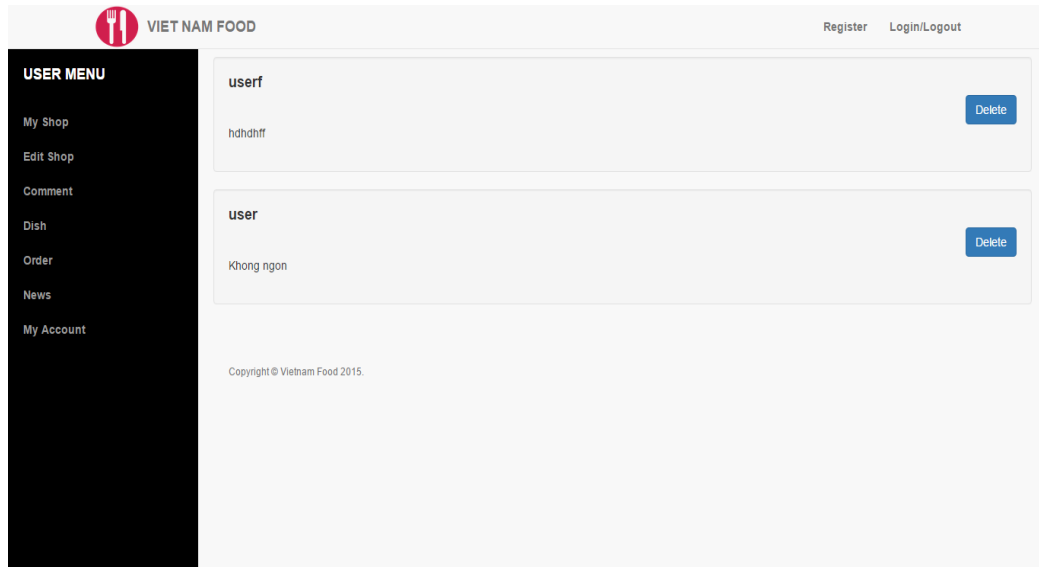
Hình 2.29 Giao diện quản lý địa điểm.

Là nơi để người quản trị chỉnh sửa thông tin về địa điểm.



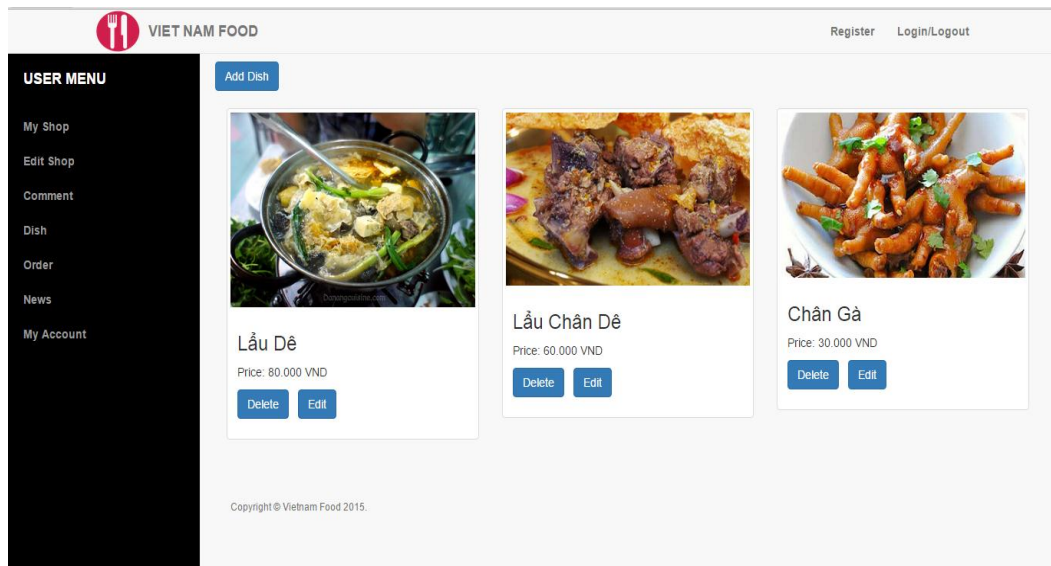
Hình 2.30 Giao diện Edit Shop.

Là nơi để quản lý các bình luận của người dùng.



Hình 2.31 Giao diện Comment.

Là nơi quản lý món ăn của địa điểm



Hình 2.32 Giao diện Dish.

Là nơi để quản lý các đơn đặt hàng của người dùng.

VIET NAM FOOD

Register Login/Logout

**USER MENU**

- My Shop
- Edit Shop
- Comment
- Dish
- Order
- News
- My Account

Id: 55d1f62e4656ed2135e5b566  
 Time: Mon Aug 17 2015 14:56:46 GMT+0000 (UTC)  
 Shop Id 55d1e99896222ea155f95f21  
 Phone 0993011223  
 Status no  
 Total: 170000  
 Name Lẩu Dê  
 Quantity 1  
 Name Chăn Gà  
 Quantity 3  
 Edit

Copyright © Vietnam Food 2015.

Hình 2.33 Giao diện Order.

Là nơi để quản lý các tin tức liên quan đến khuyến mãi, giảm giá của địa điểm

VIET NAM FOOD

Register Login/Logout

**USER MENU**

- My Shop
- Edit Shop
- Comment
- Dish
- Order
- News
- My Account

Add News

**Lẩu Dê Nhật Trường**  
 Status: Giảm giá 50%  
 Time stamp:  
 City: Biên Hòa  
 Edit Delete

**Lẩu Dê Hồng Quang**  
 Status: Giảm giá 50%  
 Time stamp:  
 City: Hồ Chí Minh  
 Edit Delete

Copyright © Vietnam Food 2015.

Hình 2.34 Giao diện News.

Là nơi để người quản trị thay đổi password đăng nhập hệ thống.

Hình 2.35 Giao diện My Account.

## 2.5 Triển khai server

Trước đây để có thể triển khai một ứng dụng (ví dụ một trang Web), cần phải đi mua/thuê một hay nhiều máy chủ (server), sau đó đặt máy chủ tại các trung tâm dữ liệu (data center) thì nay điện toán đám mây cho phép giảm lược quá trình mua/thuê đi. Chỉ cần nêu ra yêu cầu, hệ thống sẽ tự động gom nhặt các tài nguyên rồi (free) để đáp ứng yêu cầu đó.

Có rất nhiều nhà cung cấp dịch vụ điện toán đám mây, ví dụ Amazon, Microsoft, Google,... Tuy nhiên có lẽ Amazon là tên tuổi nổi trội nhất và mới đây nó có nhiều chính sách về việc dùng thử những dịch vụ đó trong vòng một năm cho nên đó cũng chính là lí do đề tài chọn việc triển khai hệ thống trên AWS.

Dịch vụ điện toán đám mây của Amazon cung cấp khá đa dạng bao gồm EC2, EBS, S3,... Do đó để việc sử dụng thuận tiện và đơn giản thì đề tài sẽ chọn EC2. EC2 cung cấp khả năng tính toán có thể mở rộng trong AWS cloud. Dùng



Amazon EC2 giúp loại bỏ nhu cầu đầu tư vào phần cứng trước tiên, vì vậy có thể tập trung phát triển và triển khai ứng dụng nhanh hơn.

**Các bước thực hiện:**

Bước 1: Đăng kí tài khoản AWS.

Bước 2: Đăng kí sử dụng EC2 để triển khai server.

Bước 3: Đăng kí Elastic Beanstalk để triển khai website.

Bước 4: Tạo Amazon machine image trên EC2 nhằm tạo ra các instance.

Bước 5: Tạo instance và đặt tên Linux NodeJS.

Bước 6: Tạo key pair để thiết lập kết nối server thông qua putty.

Bước 7: Tạo cổng giao tiếp HTTP, HTTPS, SSH.

Bước 8: Dùng putty kết nối đến server để cài đặt MongoDB.

Bước 9: Cài đặt NodeJS và các thư viện của nó.

Bước 10: Triển khai NodeJS từ từ localhost lên server.

Bước 11: Tạo file war từ eclipse để triển khai lên server.

Bước 12: Dùng Elastic Beanstalk khởi tạo môi trường Tomcat và Java.

Bước 13: Triển khai file war lên server và sử dụng URL mặc định.

### **CHƯƠNG 3: KẾT LUẬN VÀ ĐỀ NGHỊ**

#### **3.1 Kết quả đạt được và nhận xét**

Sau quá trình thực hiện, đề tài đã đạt được nhiều kết quả khả quan, trong đó nổi bật nhất là:

Tiến hành khảo sát những ứng dụng tìm kiếm địa điểm ăn uống hiện có và nhu cầu sử dụng của người dùng. Từ đó, rút ra được những tính năng cần thiết để xây dựng và phát triển một ứng dụng tìm kiếm địa điểm ăn uống ở Việt Nam.

Tìm hiểu những nền tảng công nghệ để xây dựng một ứng dụng Android. Từ đó phân tích và thiết kế mô hình phù hợp với yêu cầu của người dùng.

Xác định được những vấn đề mà ứng dụng cần phải giải quyết ở từng tính năng, từ đó dựa trên nền tảng công nghệ đã tìm hiểu mà so sánh, đưa ra giải pháp thích hợp.

Hiện thực thành công những tính năng cơ bản của ứng dụng (tìm kiếm, tra cứu thông tin về một địa điểm, ...) và những tính năng nâng cao khác như đặt hàng, liệt kê các địa điểm ở gần mình trong phạm vi bán kính 5km. Tuy những tính năng còn ở mức cơ bản, nhưng cũng góp phần định hướng phát triển ứng dụng trong tương lai.

Quá trình thực hiện đề tài đã giúp bản thân sinh viên học hỏi được những công nghệ, đồng thời phát triển khả năng tư duy, phân tích và giải quyết vấn đề, thiết kế hệ thống, so sánh và lựa chọn các phương án hiệu quả.

So sánh với các phần mềm khác trên thị trường thì chắc chắn là chưa thể nào có thể có một sự so sánh một cách toàn diện về hiệu năng và sự hoàn thiện được vì đây còn là một ứng dụng còn khá non trẻ và cách phát triển mang còn mang tính chủ quan. Tuy nhiên ứng dụng cũng hoàn thành các mục tiêu đề ra khi bắt đầu xây dựng ứng dụng là tạo được một ứng dụng có giao diện đơn giản, thân thiện và tích hợp một số tính năng mà các ứng dụng khác còn khiếm khuyết như tính năng đặt hàng qua ứng dụng, tìm kiếm theo tên món ăn.

#### **3.2 Hạn chế**

Với hạn chế về thời gian tìm hiểu công nghệ cũng như khả năng hiện thực, đề tài cũng có những mặt hạn chế như sau:

Hiệu năng của ứng dụng chưa được đánh giá toàn diện. Việc đánh giá một ứng dụng trên thiết bị di động phụ thuộc vào rất nhiều yếu tố và thiết bị thử nghiệm, đòi hỏi phải có những công cụ kiểm tra và đánh giá, từ đó xác định vấn đề và nâng cao hiệu năng của ứng dụng.

Những tính năng bản đồ ứng dụng sử dụng còn khá cơ bản.

Thiết kế giao diện chưa thực sự đảm bảo. Vấn đề này phụ thuộc phần lớn vào kinh nghiệm cũng như khả năng sáng tạo của người phát triển.

Dữ liệu được ứng dụng cung cấp chỉ nhằm mục đích thử nghiệm. Nếu ứng dụng tiếp tục được phát triển, đòi hỏi phải thu thập nhiều dữ liệu hơn để đáp ứng nhu cầu người dùng.

### **3.3 Hướng phát triển**

Với nền tảng đã được xây dựng, đề tài có nhiều hướng đi để tiếp tục phát triển và hoàn thiện hơn, cụ thể:

Giải quyết các vấn đề về hiệu năng và bảo mật. Nếu tiếp tục được phát triển, hệ thống phải có khả năng đáp ứng cho nhiều người dùng cũng như thao tác trên lượng lớn dữ liệu, đòi hỏi phải có những công cụ đánh giá phù hợp cũng như tìm hiểu các công nghệ khác nhằm nâng cao hiệu năng của ứng dụng.

Phát triển trên các nền tảng khác, điển hình nhất là các nền tảng thiết bị di động phổ biến như iOS, Window Phone hay kể cả nền tảng truyền thống như Web. Với thiết kế đã xây dựng, việc phát triển trên các nền tảng khác là khá dễ dàng. Việc này nhằm mở rộng đối tượng sử dụng, giúp ứng dụng trở nên phổ biến.

Hoàn thiện tính năng bản đồ. Với một ứng dụng tìm kiếm địa điểm ăn uống, bản đồ là yếu tố quan trọng hàng đầu, không những hỗ trợ người dùng trong quá trình tìm kiếm, mà còn giúp người dùng có thể dễ dàng di chuyển trên đường đi hơn.

Nghiên cứu phát triển các tính năng thông minh dựa trên dữ liệu người dùng. Trên cơ sở những dữ liệu thu được trong quá trình sử dụng (địa điểm yêu

thích, thêm địa điểm mới, ...), hệ thống sẽ tiến hành phân tích, xác định hành vi di chuyển của người dùng mà đưa ra các gợi ý phù hợp, giúp người dùng có thể lựa chọn, nhằm giảm thiểu công sức và nâng cao trải nghiệm với ứng dụng.

## TÀI LIỆU THAM KHẢO

[1] Amazon Web Services (10/4/2015)

<<https://aws.amazon.com/ec2/>>.

[2] Android Operating System (12/5/2015)

<[http://en.wikipedia.org/wiki/Android\\_\(operating\\_system\)](http://en.wikipedia.org/wiki/Android_(operating_system))>.

[3] Google Map API (14/6/2015)

<<https://developers.google.com/maps/>>.

[4] MongoDB. (21/9/2015)

<<https://docs.mongodb.org/manual/>>.

[5] Node JS (22/8/2015)

<<https://nodejs.org/api/>>.