

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN  
KHOA MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG**

**NGUYỄN CHÍ HIẾU – 10520490**

**KHÓA LUẬN TỐT NGHIỆP  
XÂY DỰNG HỆ THỐNG ĐẶT HÀNG SỬ DỤNG  
ĐIỆN THOẠI DI ĐỘNG**

**KỸ SƯ NGÀNH MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG**

**GIẢNG VIÊN HƯỚNG DẪN  
TS. ĐÀM QUANG HỒNG HẢI**

**TP. HỒ CHÍ MINH, NĂM 2015**

# NHẬN XÉT KHÓA LUẬN TỐT NGHIỆP

## (CỦA CÁN BỘ HƯỚNG DẪN)

**Tên khóa luận:**

**XÂY DỰNG HỆ THỐNG ĐẶT HÀNG QUA ĐIỆN THOẠI DI ĐỘNG**

**Nhóm SV thực hiện:**

Nguyễn Chí Hiếu

10520490

**Cán bộ phản biện:**

TS. Đàm Quang Hồng Hải

**Đánh giá Khóa luận**

1. Về cuốn báo cáo:

Số trang	80 _____	Số chương	5 _____
Số bảng số liệu	14 _____	Số hình vẽ	34 _____
Số tài liệu tham khảo	09 _____	Sản phẩm	1 _____

Một số nhận xét về hình thức cuốn báo cáo:

Định dạng cuốn báo cáo phù hợp với yêu cầu, cách thức viết báo cáo tốt, việc phân bố nội dung, chương mục hợp lý.

2. Về nội dung nghiên cứu:

Trong xã hội ngày nay, nhu cầu liên lạc và cập nhập thông tin ở mọi lúc mọi nơi đã trở thành nhu cầu tất yếu của mọi người. Hiện nay, các ứng dụng Mạng truyền thông mang lại giá trị gia tăng cho người sử dụng đang nhận được quan tâm to lớn do nhu cầu cung cấp thông tin ngày càng cao. Với công nghệ GPS, người dùng hiện nay có thể sử dụng các bản đồ thông minh với những chỉ dẫn đường đi trên máy tính và các thiết bị di động, đặc biệt khi các ứng dụng sử dụng Google Maps API đang trở nên rất thông dụng.

Đề tài “Xây dựng hệ thống đặt hàng qua dụng điện thoại di động ” của sinh viên Nguyễn Chí Hiếu mang tính thực tế cao khi nhu cầu sử dụng các ứng dụng xử lý thông tin bản đồ thông minh đang phát triển cùng với môi trường mạng viễn thông ở Việt Nam đang cải thiện nhanh chóng. Trong khóa luận này, sinh viên Nguyễn Chí Hiếu đã nghiên cứu bài toán xây dựng hệ thống đăng ký mua hàng trực tuyến và hệ thống phân phối giao hàng với công nghệ Google Maps API và xây dựng một ứng dụng thực tế chạy trên Web và thiết bị di động.

3. Về chương trình ứng dụng:

Chương trình ứng dụng đã được thực hiện tốt trên cả giao diện Web và trên thiết bị di động, các phần mềm đã tích hợp được tiếng Việt với giao diện khá phù hợp cho người sử dụng. Các kết quả của sinh viên Nguyễn Chí Hiếu trong khóa luận có thể tiếp tục phát triển trong thực tế.

4. Về thái độ làm việc của sinh viên:

Sinh viên Nguyễn Chí Hiếu đã có tinh thần làm việc chăm chỉ, thái độ nghiêm túc và cầu thị. Việc sinh viên một mình thực hiện hoàn chỉnh một hệ thống đặt hàng qua điện thoại di động cho thấy sinh viên Nguyễn Chí Hiếu đã có nhiều cố gắng trong việc nghiên cứu và thực hiện đề tài của mình.

**Đánh giá chung:** Khóa luận đạt yêu cầu của một khóa luận tốt nghiệp kỹ sư, xếp loại Giỏi.

**Điểm phản biện:**

Nguyễn Chí Hiếu: **9.0/10**

**Người nhận xét**

Đàm Quan Hồng Hải

# NHẬN XÉT KHÓA LUẬN TỐT NGHIỆP

## (CỦA CÁN BỘ PHẢN BIỆN)

**Tên khóa luận:**

### XÂY DỰNG HỆ THỐNG ĐẶT HÀNG QUA ĐIỆN THOẠI DI ĐỘNG

#### ORDER - SYSTEM

**Nhóm SV thực hiện:**

Nguyễn Chí Hiếu

10520490

**Cán bộ phản biện:**

ThS. Tô Nguyễn Nhật Quang

**Đánh giá Khóa luận**

1. Về cuốn báo cáo:

Số trang	80 _____	Số chương	5 _____
Số bảng số liệu	_____	Số hình vẽ	_____
Số tài liệu tham khảo	09 _____	Sản phẩm	1 _____

Một số nhận xét về hình thức cuốn báo cáo:

- Báo cáo còn rất nhiều lỗi chính tả, lỗi font chữ và lỗi trình bày. Đề nghị sinh viên phải chỉnh sửa tất cả những lỗi này trên báo cáo chính thức.
- Phân bố nội dung của báo cáo chưa hợp lý, các chương 2 và chương 3 trình bày về phần công nghệ khá dài.

2. Về nội dung nghiên cứu:

Khoá luận đưa ra một giải pháp xây dựng hệ thống đặt hàng thức ăn nhanh qua điện thoại di động trên nền tảng hệ điều hành Android giúp cho việc đặt hàng trực tuyến, giao hàng và quản lý điều hành việc kinh doanh trên máy tính hoặc smartphone được thực hiện một cách dễ dàng và hiệu quả.

- Ưu điểm: Phương pháp nghiên cứu dựa trên việc kết hợp giữa xây dựng web với PHP, MySQL, Java Script và Google Maps API trên nền tảng Android là một cách tiếp cận hợp lý.
- Khuyết điểm: Phần thiết kế cơ sở dữ liệu chưa hợp lý.

3. Về chương trình ứng dụng:

- Ưu điểm: giao diện trên web và trên smartphone thân thiện, dễ sử dụng với người dùng khách hàng, người quản lý, người giao hàng.

- Khuyết điểm: Ứng dụng chạy chưa được ổn định và số chức năng chưa phong phú.

4. Về thái độ làm việc của sinh viên:

- Sinh viên có thái độ lịch sự, lễ phép, nghiêm túc khi làm việc.

**Đánh giá chung:** Khóa luận đạt yêu cầu của một khóa luận tốt nghiệp kỹ sư ngành Truyền thông và Mạng máy tính.

**Điểm phản biện:**

**Người nhận xét**

Tô Nguyễn Nhật Quang

## **DANH SÁCH HỘI ĐỒNG BẢO VỆ KHÓA LUẬN**

Hội đồng chấm khóa luận tốt nghiệp, thành lập theo Quyết định số ..... ngày ..... của Hiệu trưởng Trường Đại học Công nghệ Thông tin.

1. .... – Chủ tịch.
2. .... – Thư ký.
3. .... – Ủy viên.
4. .... – Ủy viên.

## MỞ ĐẦU

Ngày nay với sự tiến bộ vượt bậc của khoa học công nghệ, cuộc sống của chúng ta ngày càng được thoải mái và thuận tiện hơn. Một trong những công nghệ phổ biến hiện nay đó là công nghệ định vị thông qua hệ thống định vị toàn cầu. Hiện nay có rất nhiều tổ chức quốc gia phát triển hệ thống này mà tiêu biểu như: Hệ thống Beidou của Cộng Hòa Dân Chủ Nhân Dân Trung Hoa, GLONASS của Nga, QZSS của Nhật, Galileo của liên minh châu Âu EU, nhưng đặc biệt nhất là GPS (Global Positioning System – Hệ thống định vị toàn cầu) do bộ quốc phòng Mỹ thiết kế, xây dựng và được triển khai rộng rãi vào năm 1980. Hệ thống GPS đang được sử dụng rất phổ biến trên toàn thế giới nhờ khả năng định vị chính xác và được cung cấp miễn phí hoàn toàn. Ở Việt Nam, hệ thống GPS không còn quá xa lạ. Nó đang được sử dụng để phục vụ trong một số lĩnh vực như: quản lý điều hành xe, tìm đường, du lịch, thám hiểm, thiết lập bản đồ ...

Kể từ khi ra đời màn hình cảm ứng đầu tiên – IBM Simon được ra mắt bởi Bell South vào năm 1993, nhưng thời đại của smartphone tại Mỹ thực sự bắt đầu vào năm 2002 khi công nghệ CPA<sup>1</sup> giúp thực hiện cuộc gọi trên smartphone xuất hiện. Năm 2006, smartphone chỉ chiếm 6% thị phần của Mỹ. Tại thời điểm này, so với tốc độ phát triển của PC, tốc độ "phủ sóng" của smartphone ngày càng rộng lớn hơn. Và cho đến nay, Nielsen cho biết smartphone đã chiếm tới 2/3 số lượng điện thoại di động bán ra, dù đang diễn ra cuộc khủng hoảng kinh tế toàn cầu<sup>1</sup>. Với sự phát triển mạnh mẽ như vậy, chip xử lý trong smartphone ngày càng được thu nhỏ, rẻ và có hiệu suất cao giúp giảm giá thành tạo điều kiện cho smartphone được sử dụng rộng rãi hơn nữa ở các nước đang phát triển như Việt Nam.

Tận dụng khả năng định vị của một số thiết bị GPS. Chúng ta có thể lấy thông tin tức thời và chính xác của thiết bị về vị trí, tốc độ di chuyển. Nhờ vào khả năng này, nó có thể được sử dụng vào trong việc đặt và giao hàng. Chẳng hạn như đề án “Giải pháp đặt hàng, giám sát qua công nghệ GPS” giành giải thưởng Nhân tài Đất Việt 2013 ứng dụng GPS để giúp các doanh nghiệp có thể giám sát toàn bộ quá trình INBOUND - OUTBOUND trong chuỗi cung ứng..

Từ thực tế trên, em đã đưa ra giải pháp là tận dụng khả năng định vị GPS được tích hợp trên hầu hết những smartphone vào việc đặt và giao hàng dễ. Nội dung chính của giải pháp này là kết nối smartphone đến máy chủ website đặt hàng để đặt hàng bằng ứng dụng đồng thời truyền tải thông tin về vị trí dùng cho việc giao hàng. Về phía cửa hàng việc sử dụng giải pháp này để giúp cho việc giao hàng trở nên tiện .

Hệ thống đặt hàng qua điện thoại di động được thực hiện trên nền tảng Android, một hệ điều hành dành cho các thiết bị di động được phát triển bởi Google, bên cạnh

---

<sup>1</sup> Call Progress Analysis

đó cũng xây dựng một Web Application thực hiện chức năng điều phối, quản lý đơn hàng và web đặt hàng bằng ngôn ngữ PHP.

Luận văn được cấu trúc như sau. Chương 2 nêu khái quát các kiến thức có liên quan được dùng để phục vụ đề tài, đồng thời khảo sát các công việc có liên quan đến đề tài. Chương 3 giới thiệu cách tiếp cận của chúng em để giải quyết vấn đề. Phân hiện thực hệ thống đặt hàng qua điện thoại di động cũng như đánh giá được trình bày trong chương 4. Trong chương 5, em tổng kết đề tài và nêu ra một số hướng phát triển tiếp theo của đề tài.

TP. Hồ Chí Minh, ngày 17 tháng 06 năm 2015

Sinh viên

Nguyễn Chí Hiếu



## **LỜI CẢM ƠN**

Em xin bày tỏ lòng biết ơn chân thành và sâu sắc nhất đến TS. Đàm Quang Hồng Hải, người đã tận tình hướng dẫn và tạo mọi điều kiện tốt nhất cho em trong quá trình định hướng, nghiên cứu và xây dựng đề tài này.

Em xin chân thành cảm ơn các thầy cô Khoa Mạng Máy Tính Và Truyền Thông cũng như tất cả các thầy cô, cán bộ nhân viên trường Đại Học Công Nghệ Thông Tin – Đại Học Quốc Gia Thành Phố Hồ Chí Minh đã tận tình giảng dạy, tạo điều kiện tốt nhất cho chúng em trong suốt quá trình học tập tại trường, giúp chúng em học hỏi và tiếp thu được kiến thức một cách vững vàng.

Cuối cùng chúng em xin bày tỏ lòng biết ơn chân thành đến gia đình, người thân và bạn bè đã giúp đỡ, động viên, khích lệ chúng em trong quá trình hoàn thành đề tài này.

TP. Hồ Chí Minh, ngày 17 tháng 06 năm 2015

Sinh viên

Nguyễn Chí Hiếu

## **LỜI CAM ĐOAN**

Em xin cam đoan tất cả nội dung trình bày trong khóa luận tốt nghiệp này có được dựa trên việc nghiên cứu, tổng hợp kiến thức lý thuyết và làm việc thực tế của em. Mọi thông tin trích dẫn đều được chú thích và liệt kê rõ ràng trong danh mục các tài liệu tham khảo.

Em xác nhận khóa luận tốt nghiệp này là sản phẩm của em xây dựng dưới sự hướng dẫn của TS. Đàm Quang Hồng Hải.

TP. Hồ Chí Minh, ngày 17 tháng 06 năm 2015

Sinh viên

Nguyễn Chí Hiếu

## ĐỀ CƯƠNG CHI TIẾT

<b>TÊN ĐỀ TÀI: XÂY DỰNG HỆ THỐNG ĐẶT HÀNG SỬ DỤNG ĐIỆN THOẠI DI ĐỘNG</b>
<b>Cán bộ hướng dẫn:</b> TS. Đàm Quang Hồng Hải
<b>Thời gian thực hiện:</b> Từ ngày 19/01/2015 đến ngày 22/06/2015
<b>Sinh viên thực hiện:</b> Nguyễn Chí Hiếu - 10520490
<b>Nội dung đề tài:</b>  <b>1. Mục tiêu đề tài :</b>  Hiện nay, các cửa hàng ngày càng phát triển và yêu cầu đặt hàng của khách ngày càng cao nhưng lại gặp phải một số bất cập: <ul style="list-style-type: none"><li>○ Đặt hàng theo cách truyền thống hiệu suất không cao.</li><li>○ Phức tạp đối với khách hàng.</li><li>○ Khó khăn trong việc quản lý của cửa hàng.</li></ul> Xây dựng hệ thống nhằm tạo sự thuận tiện cho cả quản lý, người giao hàng và hành khách. Quản lý dễ dàng quản lý được thông tin đơn hàng. Người giao hàng không còn vất vả phải nhớ từng địa chỉ khách hàng. Khách hàng sẽ tiện lợi hơn trong việc đặt hàng.  <b>2. Ý nghĩa đề tài</b>  Với những mục tiêu cụ thể đã đặt ra như trên nhóm em hi vọng với hệ thống này sẽ giúp cho việc hàng của khách hàng trở nên dễ dàng hơn.  <b>3. Phạm vi nghiên cứu</b>  Phạm vi nghiên cứu của đề tài được tập trung vào những phần sau: <ul style="list-style-type: none"><li>- Tập trung vào kỹ thuật phát triển ứng dụng trên Google Maps API trên nền tảng Android kết hợp với định vị người dùng bằng hệ thống GPS.</li><li>- Phát triển Web Application với PHP, MYSQL và JAVASCRIPT bằng Yii Framework.</li><li>- Ứng dụng Google Map JavaScript API trong việc hiển thị dữ liệu đơn hàng dùng trong phân phối.</li></ul>

#### 4. Phương pháp thực hiện

Đề tài được thực hiện dựa trên phương pháp khảo sát, phân tích, ứng dụng các đối tượng nghiên cứu để xây dựng hệ thống.

#### 5. Kết quả mong đợi

Xây dựng thành công hệ thống đặt hàng sử dụng điện thoại di động đáp ứng những mục tiêu đã đề ra:

- Web Application cung cấp các chức năng quản lý và điều hành hệ thống sử dụng thông qua trình duyệt web nên rất tiện dụng và nhanh chóng.
- Lưu trữ dữ liệu đơn hàng có thể được sử dụng bằng các phần mềm kế toán tự xây dựng.
- Tiếp nhận, xử lý thông tin yêu cầu phục vụ của khách hàng, tiến hành phân phối nhân viên giao hàng.
- Cho phép thêm các sản phẩm để người dùng có thể đặt hàng.

Ứng dụng client android application dùng cho nhân viên giao hàng:

- Hiện thị vị trí các đơn hàng được phân phối, chức năng tìm đường dựa trên Google Map.
- Cung cấp việc thông tin với khách hàng qua sms của điện thoại.

Ứng dụng client android application và web site dùng cho khách hàng:

- Cho phép khách hàng dễ dàng sử dụng.

#### Kế hoạch thực hiện:

##### PHÂN CÔNG CÔNG VIỆC

TÊN SINH VIÊN	CÔNG VIỆC
Nguyễn Chí Hiếu	Tìm hiểu và xây dựng hệ thống.
<b>Xác nhận của CBHD</b> (Ký tên và ghi rõ họ tên)	<b>TP. HCM, ngày....tháng .....năm 2015</b> <b>Sinh viên</b> (Ký tên và ghi rõ họ tên)

## **DANH TỪ VIẾT TẮT**

PHP	Hypertext Preprocessor
JSON	JavaScript Object Notation
CMS	Content Management System
API	Application Programming Interface
DFD	Data Flow Diagram
ERD	Entri Relationship Diagram

## **DANH MỤC BẢNG**

Bảng 2-1	Yêu cầu cấu hình tối thiểu cho Android
Bảng 2-2	Các Class trong GooleMap
Bảng 3-1	Các usecase của hệ thống
Bảng 3-2	Các kho dữ liệu trong mô hình DFD tổng quát
Bảng 3-3	Bảng Staff
Bảng 3-4	Bảng Delivery_staff
Bảng 3-5	Bảng content_categories
Bảng 3-6	Bảng content_product
Bảng 3-7	Bảng combo_relation
Bảng 3-8	bảng content_saleoff
Bảng 3-9	Bảng saleoff_relation
Bảng 3-10	Bảng content_order
Bảng 3-11	Bảng order_relation
Bảng 3-12	Bảng delivery_token
Bảng 4-1	API hỗ trợ truy vấn dữ liệu cho android application đặt hàng
Bảng 4-1	API hỗ trợ truy vấn dữ liệu cho android application giao hàng

## DANH MỤC HÌNH ẢNH

Hình 2-1	Thống kê tỉ lệ sử dụng các Map API
Hình 2-2	Thống kê thị phần các hệ điều hành di động
Hình 2-3	Các tầng trong một hệ thống Android
Hình 2-4	Linux Kernel 2.6
Hình 2-5	Android Framework
Hình 2-6	Application Framework
Hình 2-7	Application
Hình 2-8	GPS
Hình 2-9	3 phân đoạn chính của GPS
Hình 2-10	Phân đoạn điều khiển của GPS
Hình 2-11	Mô hình MVC tổng quát.
Hình 2-12	Hoạt động của mô hình MVC
Hình 2-13	Cấu trúc tingx của ứng dụng Yii
Hình 2-14	Quy trình làm việc của ứng dụng Yii
Hình 2-15	Kiến trúc Apache
Hình 2-16	Kiến trúc chi tiết phân lõi của Apache
Hình 2-17	Biểu diễn json của một đối tượng
Hình 2-18	Biểu diễn json của một mảng
Hình 1-1	Mô hình usecase của hệ thống
Hình 5-2	Sơ đồ phân rã chức năng của hệ thống
Hình 5-3	Mô hình mức ngữ cảnh của hệ thống
Hình 5-4	Mô hình DFD tổng quát
Hình 5-5	Sơ đồ DFD đối với khách hàng
Hình 5-6	Sơ đồ DFD đối với nhân viên giao hàng
Hình 5-7	Sơ đồ DFD đối với quản lý
Hình 5-8	Mô hình thực thể của hệ thống
Hình 5-9	Mô hình cơ sở dữ liệu MYSQL
Hình 5-10	Mô tả thiết kế hệ thống
Hình 5-11	Word Geodetic System 84[7]
Hình 4-1	Cách thức hoạt động của ứng dụng
Hình 4-2	Ứng dụng CMS web quản lý hệ thống
Hình 4-3	Thành phần điều phối đơn hàng.
Hình 4-4	Hiện thị danh sách đơn hàng.
Hình 4-5	Ứng dụng web đặt hàng và android application đặt hàng
Hình 4-6	Chức năng cho ứng dụng android application giao hàng.

Hình 4-7	Màn hình đăng nhập web quản lý
Hình 4-8	Màn hình chính web quản lý
Hình 4-9	Màn hình chính web bán hàng
Hình 4-10	Màn hình giỏ hàng web bán hàng
Hình 4-11	Màn hình thanh toán web bán hàng
Hình 4-12	Màn hình chính android application đặt hàng
Hình 4-13	Màn hình sản phẩm và thanh toán android application đặt hàng
Hình 4-14	Màn hình đăng nhập và màn hình chính android application giao hàng
Hình 4-15	Màn hình chi tiết đơn hàng và bản đồ android application giao hàng



## DANH MỤC CÁC THUẬT NGỮ ANH VIỆT

3G	Là công nghệ truyền thông thế hệ thứ ba.
Activity	Cửa sổ chứa giao diện ứng dụng trong Android.
Block	Vệ tinh khối
Framework	Chương trình khung.
IOS	Hệ điều hành được phát triển bởi Apple
Hexa	Hệ thập lục phân.
Latitude	Vĩ độ.
Longitude	Kinh độ.
Mashup	Là thuật ngữ dùng để chỉ các ứng dụng và web sites có sự kết hợp dữ liệu hoặc chức năng từ hai hay nhiều nguồn khác nhau.
Package	Gói chương trình.
Smartphone	Điện thoại thông minh.
Web Application	Là một ứng dụng được người dùng truy cập thông qua mạng.
Widget	Ô điều khiển trên màn hình.
Window Phone	Là một hệ điều hành dành cho các thiết bị di động được phát triển bởi Microsoft.

# MỤC LỤC

<b>CHƯƠNG 1. GIỚI THIỆU .....</b>	<b>1</b>
<b>1.1 Tên đề tài .....</b>	<b>1</b>
<b>1.2 Nội dung và giới hạn của đề tài .....</b>	<b>1</b>
1.2.1 Nội dung.....	1
1.2.2 Giới hạn.....	2
<b>1.3 Đối tượng, phạm vi và phương pháp nghiên cứu của đề tài.....</b>	<b>2</b>
1.3.1 Đối tượng nghiên cứu .....	2
1.3.2 Phạm vi nghiên cứu .....	2
1.3.3 Phương pháp nghiên cứu .....	2
<b>1.4 Thời gian nghiên cứu .....</b>	<b>2</b>
<b>1.5 Cấu trúc luận văn.....</b>	<b>3</b>
<b>CHƯƠNG 2. CÔNG NGHỆ GPS, HỆ ĐIỀU HÀNH ANDROID VÀ NHỮNG CÔNG NGHỆ SỬ DỤNG TRONG QUÁ TRÌNH XÂY DỰNG HỆ THỐNG .....</b>	<b>4</b>
<b>2.1 Map API .....</b>	<b>4</b>
2.1.1 Google Map API.....	5
2.1.2 Geolocation (geoplugin API) .....	7
<b>2.2 Hệ điều hành Android.....</b>	<b>8</b>
2.2.1 Lịch sử phát triển Android .....	8
2.2.2 Sự phát triển của Android .....	9
2.2.3 Cấu trúc của Android.....	11
2.2.4 Các thành phần chính của Android .....	14
2.2.5 Google Map API Android.....	16
2.2.6 Google Place API Android .....	26
2.2.7 Google Direction API Android .....	28
<b>2.3 Công nghệ GPS .....</b>	<b>29</b>
2.3.1 Tổng quan về GPS.....	30
2.3.2 Những phân đoạn GPS .....	30
2.3.3 Những thế hệ của vệ tinh GPS .....	31
2.3.4 Những vị trí điều khiển.....	32

2.3.5	GPS-một vài khái niệm cơ bản .....	33
2.3.6	Nguyên lý hoạt động của GPS .....	34
<b>2.4</b>	<b>PHP, MySQL, APACHE .....</b>	<b>35</b>
2.4.1	PHP và Yii Framework.....	35
2.4.2	MYSQL.....	41
2.4.3	Apache HTTP server .....	42
<b>2.5</b>	<b>JSON và thư viện mã nguồn mở GSON.....</b>	<b>45</b>
2.5.1	Định nghĩa JSON.....	45
2.5.2	Thư viện mã nguồn mở GSON .....	46
<b>2.6</b>	<b>Kết luận .....</b>	<b>47</b>
<b>CHƯƠNG 3.</b>	<b>PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG ĐẶT HÀNG SỬ D DỤNG ĐIỆN THOẠI DI ĐỘNG .....</b>	<b>48</b>
<b>3.1</b>	<b>Xác định yêu cầu và các đối tượng trong hệ thống.....</b>	<b>48</b>
3.1.1	Yêu cầu đối với hệ thống .....	48
3.1.2	Các đối tượng trong hệ thống.....	48
<b>3.2</b>	<b>Phân tích hệ thống.....</b>	<b>50</b>
3.2.1	Sơ đồ use case của thống .....	50
3.2.2	Sơ đồ phân rã chức năng:.....	51
3.2.3	Sơ đồ DFD .....	52
3.2.4	Thiết kế cơ sở dữ liệu .....	55
<b>3.3</b>	<b>Thiết kế hệ thống.....</b>	<b>62</b>
<b>3.4</b>	<b>Giải pháp hiển thị bản đồ .....</b>	<b>64</b>
<b>3.5</b>	<b>Tổng kết.....</b>	<b>66</b>
<b>CHƯƠNG 4.</b>	<b>HIỆN THỰC HỆ THỐNG ĐẶT HÀNG BẰNG ĐIỆN THOẠI DI ĐỘNG .....</b>	<b>67</b>
<b>4.1</b>	<b>Xây dựng API Web Application.....</b>	<b>67</b>
4.1.1	Truy vấn dùng cho android application đặt hàng:.....	67
4.1.2	Truy vấn dùng cho android application giao hàng hàng: .....	68
<b>4.2</b>	<b>Xây dựng ứng dụng web quản lý hệ thống.....</b>	<b>69</b>
4.2.1	Cấu trúc ứng dụng dùng chung cho web quản lí và web đặt hàng .....	69
4.2.2	Xây dựng các chức năng cho các đối tượng sử dụng hệ thống .....	69
4.2.3	Xây dựng chức năng điều phối .....	70

<b>4.3</b>	<b>Xây dựng các chức năng cho ứng dụng web đặt hàng và android application đặt hàng.....</b>	<b>72</b>
<b>4.4</b>	<b>Xây dựng ứng dụng android application giao hàng .....</b>	<b>72</b>
4.4.1	Các xử lý về thông tin GPS.....	72
4.4.2	Các xử lý và triển khai Google Direction API.....	74
4.4.3	Các xử lý cơ bản trên Google Map API v2 .....	75
4.4.4	Xây dựng các chức năng cho ứng dụng android application giao hàng...	77
<b>4.5</b>	<b>Giao diện hệ thống .....</b>	<b>77</b>
4.5.1	Giao diện web quản lý hệ thống.....	77
4.5.2	Giao diện web bán hàng.....	78
4.5.3	Giao diện android application đặt hàng.....	80
4.5.4	Giao diện android application giao hàng.....	81
<b>4.6</b>	<b>Tổng kết.....</b>	<b>82</b>
<b>CHƯƠNG 5.</b>	<b>KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN .....</b>	<b>83</b>
<b>5.1</b>	<b>Tổng kết đề tài.....</b>	<b>83</b>
<b>5.2</b>	<b>Những khó khăn và hạn chế.....</b>	<b>84</b>
<b>5.3</b>	<b>Hướng phát triển đề tài.....</b>	<b>84</b>
<b>DANH MỤC TÀI LIỆU THAM KHẢO .....</b>		<b>85</b>

# CHƯƠNG 1. GIỚI THIỆU

## 1.1 Tên đề tài

Xây dựng hệ thống đặt hàng qua điện thoại di động.

## 1.2 Nội dung và giới hạn của đề tài

### 1.2.1 Nội dung

Cơ sở hạ tầng và ở Việt Nam ngày càng phát triển. Trong đó hệ thống cửa hàng thức ăn nhanh phục vụ khách hàng đã trở nên phổ biến và tiện dụng. Ta có thể thấy một vài đặc điểm trong việc đặt hàng hiện nay như nhân viên điều phối phân phối đơn hàng với các nhân viên giao hàng bằng cách truyền thống và khó kiểm soát đơn hàng. Chính điều này đã gây ra một số hạn chế cho nhân viên trong việc điều phối cũng như việc thông báo, cung cấp thông tin của khách hàng cho nhân viên giao hàng. Để khắc phục những khó khăn này luận văn này đưa ra giải pháp xây dựng hệ thống đặt hàng qua điện thoại di động trong đó Smartphone nền tảng Android được sử dụng giúp cho nhân viên giao hàng có thể dễ dàng giao hàng và nhân viên quản lý có thể phân phối đơn hàng một cách dễ dàng và hiệu quả hơn. Với sự phát triển mạnh mẽ của Smartphone, cơ sở hạ tầng mạng viễn thông hiện nay sẽ là nền tảng giúp cho hệ thống vận hành và đáp ứng tốt những yêu cầu của người sử dụng.

Đề tài này sẽ thực hiện những công việc sau:

Phía quản lý cửa hàng:

Cho phép khách hàng sử dụng android application hoặc web đặt hàng các sản phẩm của mình:

- Sản phẩm và các thông số (như tên, giá cả, miêu tả) của các sản phẩm được bán.
- Cho phép khách hàng đặt hàng các sản phẩm.
- Quản lý, điều hành việc giao hàng dựa trên nền tảng Web.

Phía người giao hàng:

- Smartphone chạy hệ điều hành Android có cài đặt ứng dụng “Post Order” của hệ thống với các chức năng như sau:
  - Giao tiếp với web server quản lý điều hành thông qua Web application.
  - Cập nhật danh sách của các đơn hàng cần giao.
  - Cập nhật bản đồ chứa vị trí của các đơn hàng.
  - Nhận thông tin đường đi để đơn hàng cần giao.
  - Thực hiện xác nhận giao hàng.
  - Thông báo có sự cố xảy ra với khách hàng trực tiếp thông qua tin nhắn

Phía người giao hàng:

- Đặt hàng thông qua web hoặc application đặt hàng trên smartphone chạy hệ điều hành android.
- Xác nhận các thông tin cần thiết để đặt hàng (tên, số điện thoại địa chỉ giao hàng).

### 1.2.2 Giới hạn

Lĩnh vực kinh doanh, đặc biệt là kinh doanh dịch vụ muốn vận hành được thì hệ thống cần phải có sự kết hợp chặt chẽ của nhiều thành phần khác nhau. Ở đây em chỉ tập trung vào việc xây dựng hệ thống đặt hàng qua điện thoại đối với cửa hàng thức ăn nhanh. Các vấn đề khác như quản lý hành chính, quản lý nhân sự, quản lý tài sản, quản lý kinh doanh... nằm ngoài tầm vóc của luận văn này.

## 1.3 Đối tượng, phạm vi và phương pháp nghiên cứu của đề tài

### 1.3.1 Đối tượng nghiên cứu

- Các yêu cầu kỹ thuật và chức năng của một hệ thống đặt hàng qua điện thoại.
- Tìm hiểu kỹ thuật lập trình trên thiết bị di động nền tảng Android.
- Kỹ thuật định vị người dùng.
- Tìm hiểu về kỹ thuật xây dựng Web Application với PHP, MYSQL và JAVASCRIPT bằng Yii Framework.
- Nghiên cứu và ứng dụng Google Maps API.

### 1.3.2 Phạm vi nghiên cứu

Phạm vi nghiên cứu của đề tài được tập trung vào những phần sau:

- Tập trung vào kỹ thuật phát triển ứng dụng trên Google Maps API trên nền tảng Android kết hợp với định vị người dùng bằng hệ thống GPS.
- Phát triển Web Application với PHP, MYSQL và JAVASCRIPT bằng Yii Framework.
- Ứng dụng Google Map JavaScript API trong việc hiển thị dữ liệu đơn hàng dùng trong phân phối.

### 1.3.3 Phương pháp nghiên cứu

Đề tài được thực hiện dựa trên phương pháp khảo sát, phân tích, ứng dụng các đối tượng nghiên cứu để xây dựng hệ thống.

## 1.4 Thời gian nghiên cứu

Thời gian bắt đầu: 15/04/2012

Thời gian hoàn thành: 17/06/2015

## **1.5 Cấu trúc luận văn**

Luận văn được cấu trúc như sau:

- Chương 1: Giới thiệu.
- Chương 2: Công nghệ GPS, Hệ điều hành Android và những công nghệ sử dụng trong việc xây dựng hệ thống.
- Chương 3: Phân tích và thiết kế hệ thống đặt hàng sử dụng điện thoại di động.
- Chương 4: Hiện thực hệ thống đặt hàng sử dụng điện thoại di động.
- Chương 5: Kết luận và hướng phát triển.

## CHƯƠNG 2.

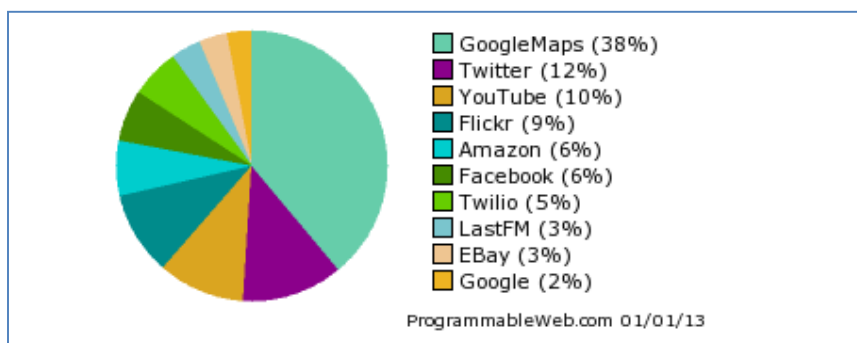
### CÔNG NGHỆ GPS, HỆ ĐIỀU HÀNH ANDROID VÀ NHỮNG CÔNG NGHỆ SỬ DỤNG TRONG QUÁ TRÌNH XÂY DỰNG HỆ THỐNG

Chương này nhóm em sẽ trình bày những nền tảng lý thuyết chính được sử dụng để làm cơ sở xây dựng hệ thống giao hàng qua điện thoại di động bao gồm: Map API, Hệ điều hành Android, Công nghệ GPS, PHP, MySQL, JSON và thư viện mã nguồn mở GSON.

#### 2.1 Map API

Ngày nay trên các hệ thống websites các bản đồ là một thành phần xuất hiện khá phổ biến. Các bản đồ này được sử dụng để giúp hiển thị vị trí một địa điểm cụ thể lên bản đồ, tìm kiếm một địa chỉ, tìm hướng dẫn chỉ đường đến một địa chỉ và còn nhiều tiện ích khác nữa được các bản đồ hỗ trợ. Một sự vật bất kì chỉ cần nó có thông tin về vị trí thì đều có thể được hiển thị lên bản đồ.

Hiện nay có rất nhiều nhà cung cấp giải pháp về bản đồ như Yahoo! Maps, Bing Maps ... nhưng được sử dụng phổ biến nhất vẫn là Google Maps. Theo thống kê ngày 1/1/2013 của website [www.programmableweb.com/apis](http://www.programmableweb.com/apis) Google Maps chiếm tới 38% các nội dung được sử dụng trong các Mashup<sup>2</sup>. Mashup là thuật ngữ dùng để chỉ các ứng dụng và web site có sự kết hợp dữ liệu hoặc chức năng từ hai hay nhiều nguồn khác nhau, cách thức này đang ngày càng trở nên phổ biến trong các ứng dụng web.



**Hình 2-1: Thống kê tỷ lệ sử dụng các Map API**

<sup>2</sup> Là thuật ngữ dùng để chỉ các ứng dụng và web sites có sự kết hợp dữ liệu hoặc chức năng từ hai hay nhiều nguồn khác nhau.



### 2.1.1 Google Map API

Google Map API tập hợp các API dành cho các nhà phát triển ứng dụng có thể tương tác, cũng như sử dụng các dịch vụ được Google và được cung cấp một cách miễn phí và có phí. Nền tảng của các API này là bản đồ của Google ở những dạng khác nhau giúp dễ dàng tích hợp nó vào trong các ứng dụng khác nhau như Website, smartphone... Ngoài ra, với Google Map cũng như các dịch vụ khác, chúng ta còn có thể tạo ra một bản đồ chỉ hiển thị một khu vực được định sẵn của riêng mình.

#### 2.1.1.1 Google Maps Javascript API

Google Maps Javascript API được cung cấp dành riêng cho việc sử dụng Google Map trên các website. API này giúp hiển thị cũng như tương tác với bản đồ, kèm theo đó là các dịch vụ như tìm kiếm, chỉ đường... được cung cấp một cách miễn phí. Kết hợp với tài khoản Google, chúng ta có thể đánh dấu các địa điểm của riêng mình kèm theo hình ảnh và nội dung. Website sử dụng API này do Google tạo ra để cung cấp cho người sử dụng là <http://maps.google.com>.

#### 2.1.1.2 Google Maps API Flash

Có chức năng tương tự như Google Maps Javascript API, nhưng sử dụng Flash để hiển thị thay vì Javascript. Google Maps API Flash dùng để phát triển các ứng dụng sử dụng Flash.

#### 2.1.1.3 Google Earth API

Tương tự như Google Map Javascript API nhưng cung cấp thêm tính năng hiển thị hình ảnh 3D kỹ thuật số toàn cầu. Sử dụng API này chúng ta có thể đánh dấu, vẽ đường đi, hiển thị hình ảnh trên bề mặt bản đồ. Ngoài ra với API này, chúng ta còn có thể xây dựng ứng dụng bản đồ hiển thị 3D phức tạp để thêm vào website. Để sử dụng được Google Earth API thì yêu cầu trình duyệt phải được cài đặt thêm một số plugin.

#### 2.1.1.4 Google Maps Image API

Là API dùng để cung cấp hình ảnh về một địa điểm cụ thể cho người dùng. Tham số đưa vào là tọa độ của điểm cần hiển thị và kích thước của hình ảnh tạo ra. Google Map Image API gồm hai loại chính là StaticMap và StreetView:

- Static Map: cung cấp cho người dùng hình ảnh tĩnh của bản đồ để hiển thị vào trong website.
- StreetView: cung cấp cho người dùng hình ảnh đường phố tại một địa điểm bất kỳ. Nhưng dịch vụ này chỉ cung cấp trong một số nước nhất định.

Google Maps Image API được cung cấp hoàn toàn miễn phí và không giới hạn số lượt sử dụng.

### 2.1.1.5 Maps API Web Service

Map API Web Service là tập hợp các dịch vụ về bản đồ được Google cung cấp thông qua giao thức HTTP, người dùng sử dụng các định dạng URL cùng những tham số khác nhau để truy vấn thông tin từ Google. Kết quả trả về là thông tin được lưu theo định dạng Json hay XML. Maps API Web Service gồm các dịch vụ sau đây:

- Direction API: cho phép người dùng tìm kiếm đường đi ngắn nhất giữa hai điểm trên bản đồ tùy thuộc vào loại phương tiện di chuyển. Phương tiện di chuyển được cung cấp gồm có 3 loại là xe ô tô, xe buýt và đi bộ. Kết quả trả về là một tập tin Json lưu trữ thông tin hướng dẫn đường đi, khoảng cách, danh sách các điểm để có thể vẽ đường đi trên Google Map. Đây là một dịch vụ hoàn toàn miễn phí, nhưng ở một số nước trong đó có Việt Nam vẫn chưa cung cấp toàn bộ các dịch vụ này.

Định dạng truy vấn của Direction API là:

<http://maps.googleapis.com/maps/api/directions/output?parameters>

- Distance Matrix API: là một dịch vụ của google giúp tính toán khoảng cách và thời gian giữa hai hay nhiều điểm trên bản đồ khi di chuyển bằng các phương tiện khác nhau. Dịch vụ này chỉ cho phép tính toán tối đa 100 địa điểm trên một lần truy vấn, 100 địa điểm trong 10 giây, 2500 địa điểm trong một ngày. Đối với phiên bản thương mại thì được sử dụng: 625 địa điểm trong một yêu cầu, 1000 địa điểm trong 10 giây và 100000 địa điểm trong 1 ngày.

Định dạng truy vấn của Distance Matrix API là:

<http://maps.googleapis.com/maps/api/distancematrix/output?parameters>

- Elevation API: là một dịch vụ của Google dùng để truy vấn thông tin về độ cao, độ sâu của bất kỳ điểm nào trên bản đồ. Giá trị này được tính toán thông qua độ cao trung bình của 4 điểm gần đó nhất. Cũng như Distance Matrix API, Elevation API cũng chỉ cho phép thực hiện tối đa 512 địa điểm trong một truy vấn, 2500 truy vấn trong một ngày. Với phiên bản thương mại có thể thực hiện tối đa 100.000 truy vấn trong một ngày và mỗi truy vấn chứa tối đa 1.000.000 địa điểm.

Định dạng truy vấn của Distance Matrix API là:

<http://maps.googleapis.com/maps/api/elevation/outputFormat?parameters>

- Geocoding API: là một dịch vụ của Google cho phép chuyển đổi qua lại các địa chỉ cụ thể thực tế thành tọa độ vật lý. Nhờ đó chúng ta có thể tìm kiếm chính xác vị trí trên bản đồ thông qua địa chỉ. Nó được cung cấp miễn phí nhưng cũng rất hạn chế trong việc truy xuất. Mỗi ngày đối với một người dùng chỉ được sử dụng 2500 truy vấn và 10000 truy vấn đối với phiên bản thương mại.

Định dạng truy vấn của Geocoding API theo dạng:

<http://maps.googleapis.com/maps/api/geocode/output?parameters>

- Place API: là dịch vụ của Google cho phép truy vấn để tìm kiếm thông tin về địa điểm, thông tin được cung cấp bao gồm địa chỉ, phân loại, điểm đánh giá... Nhưng Place API chỉ cho phép người dùng thực hiện 1000 truy vấn trong một ngày.

Định dạng truy vấn của Geocoding API theo dạng:

<https://maps.googleapis.com/maps/api/place/search/output?parameters>

- Time Zone API: là dịch vụ của Google cho phép truy vấn múi giờ cho một vị trí trên trái đất. Tham số đầu vào là vị trí cần truy vấn thời gian, API sẽ trả về tên và id của múi giờ và thời gian lệch so với múi giờ gốc.

Định dạng truy vấn của Geocoding API theo dạng:

<https://maps.googleapis.com/maps/api/timezone/outputFormat?parameters>

### 2.1.2 Geolocation (geoplugin API)

Geolocation là công nghệ định vị trên web dựa trên IP nên còn được biết đến với tên IP Location. GeoPlugin là cách dễ nhất để bạn biết nơi truy cập của bạn là từ đó bạn có thể cung cấp cho họ với nội dung phù hợp hơn với vị trí địa lý của họ.

GeoPlugin độc đáo ở chỗ nó cho phép bạn làm điều này một đơn giản với nhiều ngôn ngữ lập trình khác nhau. Không cần cài đặt phần mềm cần thiết, và cho dù ngôn ngữ lập trình của bạn lựa chọn là javascript, php, xml hoặc bất cứ điều gì, geoPlugin có thể tìm ra vùng truy cập của bạn.

#### PHP web service

Để kết nối với dịch vụ đơn giản chỉ cần thực hiện truy vấn đến địa chỉ:

```
http://www.geoplugin.net/php.gp?ip=xx.xx.xx.xx
```

Ví dụ thực hiện truy vấn sau:

```
var_dump(unserialize(file_get_contents('http://www.geoplugin.net/php.gp?ip='.$_SERVER['REMOTE_ADDR'])))
```

Kết quả nhận được sẽ tương tự như sau:

```
array (
  'geoplugin_request' => '58.186.19.182',
  'geoplugin_status' => 200,
  'geoplugin_credit' => 'Some of the returned data includes GeoLite data created by MaxMind,
available from http://www.maxmind.com.',
  'geoplugin_city' => 'Ho Chi Minh City',
```

```

'geoplugin_region' => 'Hồ Chí Minh',
'geoplugin_areaCode' => '0',
'geoplugin_dmaCode' => '0',
'geoplugin_countryCode' => 'VN',
'geoplugin_countryName' => 'Vietnam',
'geoplugin_continentCode' => 'AS',
'geoplugin_latitude' => '10.8142',
'geoplugin_longitude' => '106.643799',
'geoplugin_regionCode' => '20',
'geoplugin_regionName' => 'Hồ Chí Minh',
'geoplugin_currencyCode' => 'VND',
'geoplugin_currencySymbol' => '₫',
'geoplugin_currencySymbol_UTF8' => '₫',
'geoplugin_currencyConverter' => '21805.495',
)

```

Trong hệ thống ORDER-SYSTEM sẽ dùng geoplugin để xác định vị trí khi khách hàng thực hiện đặt hàng trên web site.

## 2.2 Hệ điều hành Android

### 2.2.1 Lịch sử phát triển Android

Vào khoảng những năm đầu của thế kỷ 21, Andy Rubin<sup>3</sup> đã tạo ra Android như một hệ điều hành mở dành cho thiết bị di động. Năm 2005, Google đã mua sản phẩm này từ tay Andy Rubin và cử ông làm giám đốc phát triển nền tảng di động cho Google. Đây thực sự là một quyết định mang tính đột phá của Google, một mặt giúp Google khẳng định mạnh mẽ hơn vị trí của mình trong lĩnh vực mới được xem là Internet 2.0, nhưng mặt khác cũng tạo ra cuộc cạnh tranh khốc liệt giữa Google và đối thủ nặng ký Apple Iphone, đang là nhà cung cấp được ưu chuộng nhất thời gian này.

Tháng 11 năm 2007, Android chính thức ra mắt phiên bản 1.0 ra đời cùng với sự thành lập tổ chức OHA<sup>4</sup> gồm nhiều công ty phần cứng, phần mềm. Mục tiêu chính của tổ chức này là cùng nhau nghiên cứu phát triển tiêu chuẩn mở cho các thiết bị di động, để giảm giá thành sản phẩm và nâng cao chất lượng dịch vụ. Android là hệ điều hành mở dành cho thiết bị di động đầu tiên, ra đời theo mục tiêu này. Nó dựa trên nền tảng mở Linux và ngôn ngữ lập trình Java của Oracle. Nền tảng Android phải hoạt động như một máy tính chạy trên chip, nghĩa là đủ nhỏ để vừa với thiết bị và đủ mạnh để chạy các ứng dụng. Ngoài ra, Android còn cung cấp các công cụ, tài liệu, thư viện hỗ trợ những nhà phát triển phần mềm và phần cứng phát triển ứng dụng cho thiết bị di động một cách dễ dàng. Trải qua nhiều năm, Android đã phát triển và nâng cấp lên các phiên bản khác nhau và phiên bản mới nhất hiện nay là 4.2. Thiết bị cầm tay cài đặt Android trên thị trường là thiết bị T-Mobile G1 do HTC sản xuất và bán ra thị trường Mỹ. Ngày nay, rất nhiều nhà sản xuất điện thoại di động như HTC, Samsung,

<sup>3</sup> Andy Rubin: [http://en.wikipedia.org/wiki/Andy\\_Rubin](http://en.wikipedia.org/wiki/Andy_Rubin)

<sup>4</sup> Open Handset Alliance: <http://www.openhandsetalliance.com/>

LG Electronic, T-Mobile sử dụng Android làm hệ điều hành cho các dòng điện thoại mới của họ.

## 2.2.2 Sự phát triển của Android

Trong quý 3/2011, hệ điều hành Android có mặt trong 71 triệu điện thoại được phát hành và nắm trong tay khoảng 57,5% thị phần.

**Top Six Smartphone Mobile Operating Systems, Shipments, and Market Share, Q3 2012 (Preliminary) (Units in Millions)**

Operating System	3Q12 Shipment Volumes	3Q12 Market Share	3Q11 Shipment Volumes	3Q11 Market Share	Year-Over-Year Change
Android	136.0	75.0%	71.0	57.5%	91.5%
iOS	26.9	14.9%	17.1	13.8%	57.3%
BlackBerry	7.7	4.3%	11.8	9.5%	-34.7%
Symbian	4.1	2.3%	18.1	14.6%	-77.3%
Windows Phone 7/ Windows Mobile	3.6	2.0%	1.5	1.2%	140.0%
Linux	2.8	1.5%	4.1	3.3%	-31.7%
Others	0.0	0.0%	0.1	0.1%	-100.0%
Totals	181.1	100.0%	123.7	100.0%	46.4%

**Hình 2-2: Thống kê thị phần các hệ điều hành di động**

Bên cạnh sự phát triển chóng mặt của Android, có thêm hai hệ điều hành khác đang có đà tăng trưởng nhanh. Đầu tiên phải nhắc đến nền tảng iOS, một nền tảng từng thống trị thị trường smartphone. Theo IDC<sup>5</sup>, số lượng điện thoại iOS được xuất xưởng tăng tới 57,3%, trong quý 3 xuất xưởng trên 26.9 triệu chiếc iPhone chiếm 14,9% thị trường, tăng 17,1 triệu chiếc so với cùng kỳ năm ngoái. Tiếp tới là nền tảng Windows Phone của Microsoft. Hệ điều hành di động này dù ra đời sau nhưng cũng có mức tăng trưởng thật sự ấn tượng. Thống kê của IDC cho thấy đã có khoảng 3,6 triệu điện thoại Windows Phone được phát hành trong quý 3 nhưng con số này vẫn còn đứng sau 7,7 triệu điện thoại BlackBerry của RIM và 4,1 triệu điện thoại Nokia chạy trên nền tảng Symbian.

Sự phát triển của Android không chỉ thể hiện trên thị trường sản phẩm mà còn trên cả thị trường ứng dụng. Đến tháng 10 năm 2012, đã có khoảng 700.000 ứng dụng được viết cho Android, và số lượng tải về ứng dụng từ Google Play - cửa hàng ứng dụng chính của Android - được ước tính là 25 lượt<sup>6</sup>.

Trong lần đầu tiên ra mắt vào ngày 22 tháng 10 năm 2008, hệ điều hành Android bao gồm 12.000.000 dòng mã lệnh trong đó có 3.000.000 dòng XML,

<sup>5</sup> IDC : [http://en.wikipedia.org/wiki/International\\_Data\\_Corporation](http://en.wikipedia.org/wiki/International_Data_Corporation)

<sup>6</sup> Thống kê trên Google Market: <http://vi.wikipedia.org/wiki/Android>

2.800.000 dòng C, 2.100.000 dòng Java và 1.750.000 dòng C++. Từ lúc xuất hiện đến nay thì Android đã cho ra mắt những phiên bản<sup>7</sup>:

- Android 1.0 Donut (API level 1): Ngày 15 tháng 9 năm 2008, Android 1.6 SDK ra đời với tên gọi là dubbed Donut dựa trên nền tảng Linux 2.6.29 bao gồm rất nhiều những cập nhật mới. Thiết bị đầu tiên được sử dụng là HTC Dream.
- Android 1.1 (API level 2): Ngày 9 tháng 2 năm 2009, phiên bản cập nhật Android 1.1 ra đời. Thiết bị được sử dụng đầu tiên cũng là HTC Dream. Phiên bản này không có đột phá về mặt ý tưởng, mà chỉ là phiên bản sửa lỗi của phiên bản 1.0.
- Android 1.5 Cupcake (API level 3): Ra đời ngày 30 tháng 4 năm 2009, Android 1.5 được chính thức ra đời dựa trên nền tảng linux 2.6.27. Đây là phiên bản đầu tiên được đặt tên dựa theo những món tráng miệng. Phiên bản này cung cấp những cải tiến mới như Android 1.5 hỗ trợ hiển thị bàn phím trên màn hình, Widget, ghi âm và hình ảnh.
- Android 1.6 Donut (API level 4): Ra đời ngày 15 tháng 9 năm 2009, Android 1.6 SDK dubbed Donut được ra đời dựa trên nền tảng Linux 2.6.29 bao gồm những cập nhật mới như: chức năng tìm kiếm nhanh, đọc văn bản, xóa nhiều tập tin hình ảnh cùng lúc, hỗ trợ màn hình có độ phân giải WVGA...
- Android 2.0 Éclair (API level 5): được chính thức ra đời vào ngày 26 tháng 10 năm 2009, Android 2.0 với tên gọi là Eclair chạy trên nền tảng Linux 2.6.29, với những tính năng ngày càng hoàn thiện và tối ưu hơn chẳng hạn như: cho phép quản lý nhiều tài khoản Email, tìm kiếm tin nhắn, Google Map 3.1.2, hỗ trợ nhiều kích thước và độ phân giải của màn hình...
- Android 2.01 Éclair (API level 6): Ra đời ngày 3 tháng 12 năm 2009, đây chỉ là phiên bản sửa lỗi cho phiên bản trước đó là Android 2.0
- Android 2.1 Eclair (API level 7): Phiên bản ra đời ngày 12 tháng 1 năm 2010, không có nâng cấp đáng kể trong phiên bản này.
- Android 2.2 Proyo (API level 8): Vào ngày 20 tháng 5 năm 2010, Android 2.2 SDK được ra đời dựa trên nền tảng Linux 2.6.32, với những tính năng bổ sung như: tối ưu hóa tốc độ và sử dụng bộ nhớ, tích hợp Javascript version 8 của Chrome vào trong trình duyệt web, hỗ trợ dịch vụ C2DM, hỗ trợ Adobe Flash...
- Android 2.3 Gingerbread (API level 9): Ra đời vào ngày 6 tháng 12 năm 2010 dựa trên nền tảng Linux 2.6.35 với những thay đổi như: hỗ trợ thiết kế giao diện đơn giản và hiệu quả, hỗ trợ hiển thị trên màn hình có độ phân giải lớn, nhập văn bản thông minh và nhanh hơn, hỗ trợ chức năng sao chép và dán, hỗ trợ tính năng dọn rác để tăng khả năng xử lý cho hệ điều hành, .....
- Android 2.3.3 Gingerbread(API level 10): Ra đời vào ngày 9 tháng 2 năm 2011 được sử dụng để sửa lỗi cho phiên bản trước.
- Android 3.0 Honeycomb (API level 11): Ngày 22 tháng 2 năm 2011 phiên bản Android 3.0 SDK lần đầu tiên xuất hiện trên máy tính bảng chạy hệ điều hành Android dựa trên nền tảng Linux 2.6.36 với những thay đổi lớn như: thêm

<sup>7</sup> Android version history: [http://en.wikipedia.org/wiki/Android\\_version\\_history](http://en.wikipedia.org/wiki/Android_version_history)

System Bar và Action Bar, hỗ trợ đa tác vụ, bàn phím ảo được thiết kế để nhập văn bản nhanh hơn, trình duyệt với nhiều tab, hỗ trợ những bộ vi xử lý đa lõi, có khả năng mã hóa tất cả thông tin người dùng, Google Talk hỗ trợ chức năng Video Chat.

- Android 3.1 Honeycomb (API level 12): Ra đời ngày 10 tháng năm 2011 nhằm mục đích cung cấp thêm một số tính năng so với phiên bản trước như: hỗ trợ kết nối USB, hỗ trợ kết nối tay cầm chơi game, hỗ trợ HTTP proxy cho mỗi kết nối Wifi...
- Android 3.2 Honeycomb (API level 13): Ra đời ngày 15 tháng 7 năm 2011 nhưng không có sự thay đổi gì nổi bật ngoại trừ một số cải tiến như: khả năng truy cập vào các tập tin nằm trong thẻ nhớ.
- Android 4.0 Ice Cream Sandwich (API level 14): Được xây dựng trên nền tảng Linux 3.0.1 chính thức công bố vào ngày 19 tháng 10 năm 2011. Phiên bản này kết hợp thiết kế của cả 2 phiên bản là 3.x và 2.x vì vậy nó có thể chạy trên bất cứ thiết bị di động nào dù là điện thoại di động hay máy tính bảng, cùng với đó là những tính năng ưu việt hơn như: giao diện đẹp, Widget có thể thay đổi kích thước, cho phép khóa màn hình ...
- Android 4.1 Jelly Bean (API level 16): Google công bố Android 4.1 ở hội nghị Google I/O vào ngày 27 tháng 6 năm 2012. Dựa trên nền tảng linux 3.0.31, JellyBean được xem như là một bản cập nhật với mục đích chính là cải thiện chức năng và hiệu suất của giao diện, tăng khả năng tương tác với người dùng, máy tính bảng Nexus 7 là thiết bị đầu tiên được nâng cấp lên phiên bản này vào ngày 13 tháng 7 năm 2012.
- Android 4.2 Jelly Bean (API level 17): Google dự định công bố phiên bản Android 4.2 vào sự kiện ở thành phố New York nhưng bị hủy và dời vào ngày 13 tháng 12 năm 2012. LG Nexus 4 và Samsung's Nexus 10 là hai sản phẩm đầu tiên được cập nhật phiên bản này. Các tính năng mới trong phiên bản này như: thêm vào tính năng "Photo Sphere", bàn phím nhập hiệu quả hơn, màn hình khóa hỗ trợ thêm việc hiển thị Widget và chuyển trực tiếp sang camera, hỗ trợ đa người dùng trên máy tính bảng, hiển thị mạng wireless...

## 2.2.3 Cấu trúc của Android

### 2.2.3.1 Phần cứng

Android là một nền tảng phần mềm được chạy trên các thiết bị di động vì vậy những yêu cầu về phần cứng của Android tương đối thấp, sau đây là những yêu cầu về cấu hình tối thiểu mà Android yêu cầu:

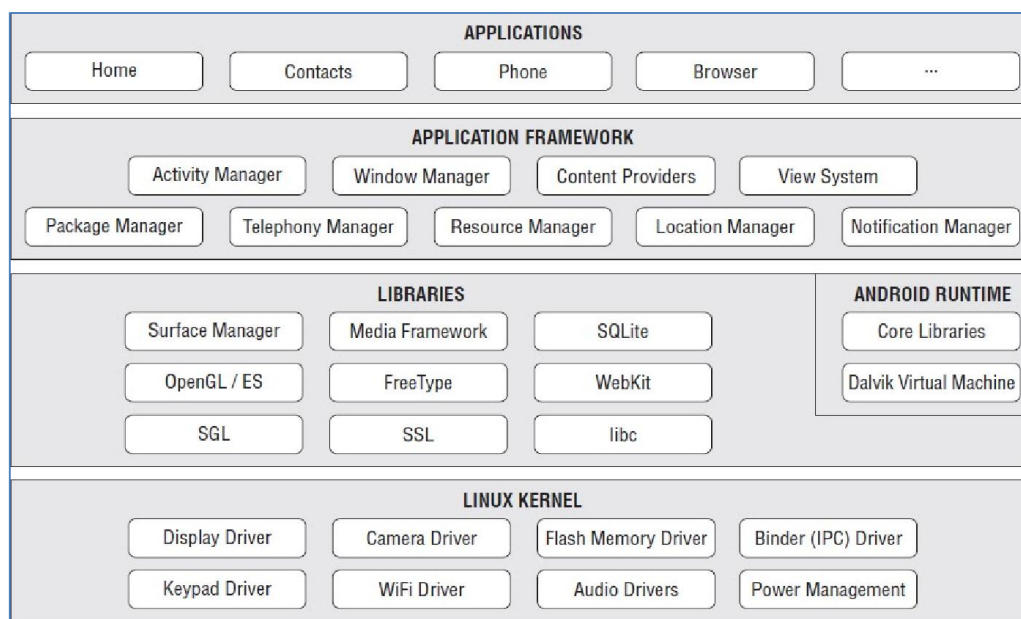
Phần cứng	Những yêu cầu tối thiểu
Chip set	ARM-based
Ram	128MB Ram. 256MB Flash External
Ổ cứng	Minu hay Micro SD
Màn hình chính	QVGA TFT LCD hoặc lớn hơn.
Navigation Key	2MP CMOS
Bluetooth	1.2 hoặc 2.0

Bảng 2-1: Yêu cầu cấu hình tối thiểu cho Android

### 2.2.3.2 Hệ điều hành Android

Có 5 tầng phân biệt trong một hệ thống Android:

- Tầng lõi ARM Linux: là tầng thấp nhất được xây dựng từ hai thành phần chính là Linux kernel và bộ vi xử lý ARM<sup>8</sup>. Bộ vi xử lý ARM hoạt động với hiệu suất cao nhưng tiêu thụ năng lượng rất ít. Còn Linux kernel phiên bản 2.6 được chứng minh là phiên bản có tính ổn định cao. Sự kết hợp giữa hai thành phần này, nhằm giải quyết phần nào giới hạn về nguồn năng lượng sử dụng và tối ưu việc sử dụng các tài nguyên của thiết bị.
- Tầng thư viện: chứa các mã nguồn cấp thấp được sử dụng cho các chức năng cơ bản như: mã hóa và giải mã âm thanh, hình ảnh kỹ thuật số, trình bày các giao diện đồ họa, bảo mật lưu lượng TCP/IP cũng như các thành phần cho trình duyệt Web, hỗ trợ xuất cơ sở dữ liệu (SQLite)...
- Tầng máy ảo Android: đây là một phần mềm dùng để chạy các ứng dụng trên thiết bị Android, bao gồm một tập các thư viện lõi cung cấp hầu hết các chức năng trong thư viện lõi của ngôn ngữ lập trình Java và máy ảo Dalvik<sup>9</sup>. Mỗi ứng dụng Android sẽ chạy trong một tiến trình riêng dành cho nó, với một máy ảo Dalvik riêng cũng được thiết lập dành riêng cho mỗi ứng dụng. Dalvik được viết để Android có thể chạy nhiều máy ảo cùng lúc nhằm tăng hiệu suất.
- Tầng khung ứng dụng Android: tầng này có chức năng định nghĩa các Android API, cung cấp bộ công cụ ở mức cao để lập trình viên Android có thể dễ dàng xây dựng ứng dụng.
- Tầng ứng dụng lõi Android: bao gồm các ứng dụng cơ bản hỗ trợ thêm cho người lập trình như: Webkit browser, Google Calendar, Gmail, Maps.
- Application...Chúng ta có thể viết thêm các ứng dụng cho tầng này.



**Hình 2-3: Các tầng trong một hệ thống Android**

<sup>8</sup> [https://vi.wikipedia.org/wiki/C%E1%BA%A5u\\_tr%C3%BAC\\_ARM](https://vi.wikipedia.org/wiki/C%E1%BA%A5u_tr%C3%BAC_ARM)

<sup>9</sup> Dalvik: [http://en.wikipedia.org/wiki/Dalvik\\_\(software\)](http://en.wikipedia.org/wiki/Dalvik_(software))

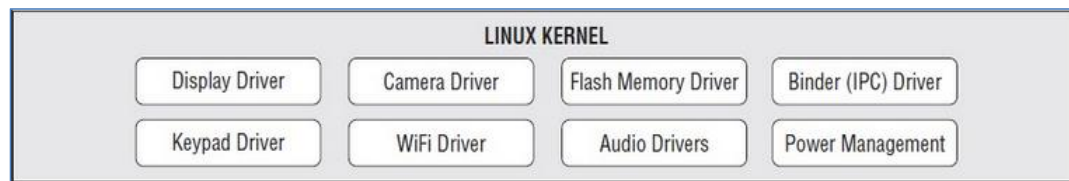


## Linux Kernel 2.6

Android được xây dựng trên nền tảng Linux kernel 2.6 mã nguồn mở, chứa những trình điều khiển dùng để điều khiển phần cứng như Keypad, Wifi, Camera, Audio, màn hình... được xem là trung tâm chỉ huy của nền tảng Android. Các giải thuật quản lý tài nguyên, chuyển đổi qua lại giữa các tác vụ qua lại, phân chia quyền, giải quyết tranh chấp được cài đặt trong linux kernel đảm nhận trách nhiệm tương tác với các thiết bị ngoại vi. Tuy nhiên người lập trình không thể lập trình trực tiếp trên tầng này.

Những tính năng nổi bật của Linux Kernel:

- Mô hình bảo mật tốt.
- Quản lý bộ nhớ tối ưu.
- Quản lý tiến trình hiệu quả.
- Quản lý các kênh kết nối mạng tốt.

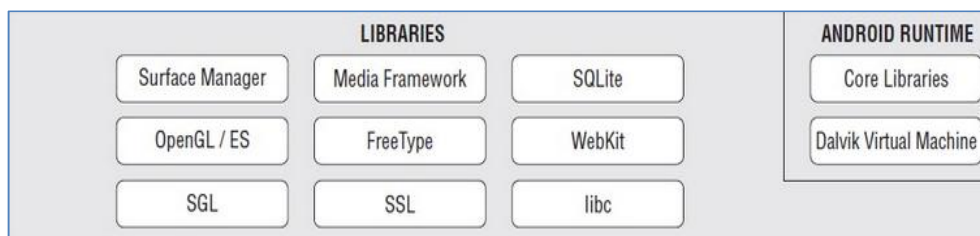


Hình 2-4: Linux Kernel 2.6

## Android Framework

Được phát triển trên nền của Linux kernel, tầng này được cấu thành từ hai hay nhiều tầng con, cung cấp những tính năng cho Android:

- Android native libraries: đây là tập thư viện C/C++ và bạn sẽ gọi thư viện này thông qua giao diện Java. Phần này gồm thư viện:
  - Surface Manager dùng để tạo ra các cửa sổ giao diện.
  - OpenGL dùng để hỗ trợ xây dựng ứng dụng đồ họa 2D và 3D.
  - Media Framework dùng để xây dựng các ứng dụng âm thanh và hình ảnh.
  - SSL cung cấp giao thức truyền bảo mật.
  - SQLite cung cấp khả năng quản lý cơ sở dữ liệu.
- Android Runtime: gồm các thư viện lõi của Java và máy ảo Dalvik. Các ứng dụng Android được viết bằng ngôn ngữ Java, nên khi biên dịch sẽ thành các tập tin \*.class. Máy ảo Dalvik sẽ làm chức năng đóng gói các tập tin này thành các tập tin nhỏ gọn \*.dex có thể dùng để chạy trên thiết bị, giúp tối ưu bộ nhớ và nguồn năng lượng giới hạn.

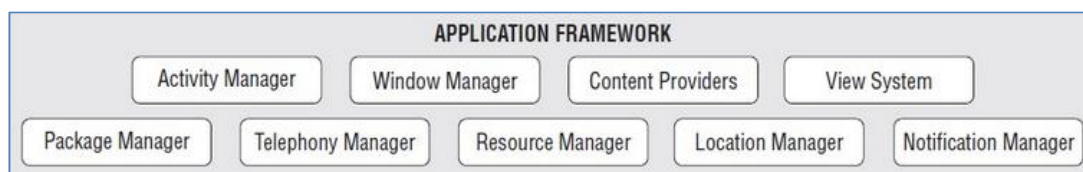


**Hình 2-5: Android Framework**

### Application Framework

Tầng này chứa các thư viện Java hỗ trợ người dùng giao tiếp với tầng Android framework. Một phần của tầng này là do Google cung cấp sẵn cho chúng ta, một phần là do chúng ta tạo ra. Trong tầng này thành phần quan trọng nhất là Activity Manager dùng để quản lý chu trình sống của Activity.

- Activity Manager: quản lý chu trình sống của Activity trong ứng dụng Android.
- Telephony Manager: cung cấp thư viện để truy xuất đến các dịch vụ điện thoại cũng như thuê bao.
- View System: xử lý giao diện trong Android của bạn.
- Location Manager: cung cấp thư viện hỗ trợ xác định vị trí của chúng ta.



**Hình 2-6: Application Framework**

### Application

Đây là tầng cao nhất và ứng dụng của bạn sẽ nằm trong tầng này. Tuy nhiên, để tiện lợi cho việc truy cập một số thông tin ở các tầng dưới, Google đã viết một số thư viện tiện ích trong tầng này giúp chúng ta có thể truy xuất thông tin các tầng ở dưới linh hoạt và hiệu quả nhất.



**Hình 2-7: Application**

#### 2.2.4 Các thành phần chính của Android

Những ứng dụng Android được viết bằng ngôn ngữ Java, sử dụng bộ công cụ Android SDK để biên dịch toàn bộ chương trình thành tập tin \*.apk để có thể cài đặt trên thiết bị chạy Android.

Mỗi ứng dụng sau khi được cài đặt trên thiết bị sẽ có những đặc điểm sau:

- Mỗi ứng dụng sẽ được Android gán cho một User Id phân biệt với nhau.
- Hệ thống sẽ thiết lập quyền hạn riêng cho mỗi ứng dụng trong việc truy nhập vào tập tin của hệ thống.
- Mỗi tiểu trình của ứng dụng sẽ chạy trên các máy ảo khác nhau vì vậy những ứng dụng này sẽ hoạt động độc lập với nhau.

Dù khác nhau ở cách thức hoạt động nhưng bất kỳ chương trình Android nào cũng bao gồm 3 thành phần chính là:

- Application Components.
- Tập tin Manifest.
- Application Resources.

#### 2.2.4.1 *Application Components*

Application Components đóng vai trò là thành phần giúp giao tiếp giữa ứng dụng của chúng ta và hệ thống. Có 4 loại Application Component chính như sau:

- **Activities:** Đại diện cho một cửa sổ chứa giao diện của chương trình giao tiếp với người dùng. Activities chứa giao diện và các hàm xử lý của chương trình. Mỗi Activity được tạo ra trong ứng dụng sẽ kế thừa một lớp Activity của Android Platform. Có 3 dạng Activity chính là:
  - **Floating Activity:** là loại Activity nổi, có kích thước không phủ lấp toàn bộ màn hình.
  - **Activity Group:** là một nhóm các Activity nhúng vào trong Activity khác.
  - **Full Screen Activity:** là dạng Activity phủ kín toàn bộ màn hình.
- Một vòng đời của Activity gồm 3 trạng thái chính là:
  - **Resumed:** là trạng thái chạy của Activity, khi Activity chuyển sang trạng thái này, giao diện của nó sẽ được hiển thị trên màn hình.
  - **Paused:** là khi một Activity đang chạy mà có một Activity khác được khởi động thì trạng thái của Activity cũ sẽ chuyển sang dạng Paused. Khi ở trạng thái này thì Activity sẽ không còn được hiển thị trên màn hình.
  - **Stopped:** Khi ở trạng thái này thì Activity sẽ bị hủy nếu có một ứng dụng khác yêu cầu thêm bộ nhớ từ hệ thống.
- Android cũng cấp một số sự kiện để quản lý 3 trạng thái này của Activity:
  - **onCreate():** Sự kiện này được gọi khi Activity lần đầu tiên được tạo. Trong sự kiện này chúng ta phải thực hiện một số công việc như: tạo giao diện cho Activity, tải dữ liệu vào danh sách... Sau sự kiện này thì sự kiện onStart() sẽ được gọi.
  - **onStart():** Sự kiện này được gọi khi giao diện của ứng dụng được hiển thị lên màn hình. Lúc này tương tác của người dùng vẫn chưa được thiết lập.
  - **onRestart():** Sự kiện này được gọi khi một Activity đang ở trạng thái Stopped nhưng muốn hiển thị và dành quyền điều khiển.

- onResume(): Sự kiện này được gọi khi Activity bắt đầu cho phép tương tác với người dùng.
- onPause(): Sự kiện này được gọi khi một Activity khác muốn dành quyền hiển thị và điều khiển của Activity hiện hành.
- onStop(): Sự kiện này được gọi khi một Activity bị che khuất hoàn toàn.
- onDestroy(): Sự kiện này là sự kiện cuối cùng trước khi Activity bị hủy.
- Services: đây là Application Component chạy nền để thực hiện những tác vụ ẩn bên dưới mà không cần phải hiển thị hay chiếm quyền điều khiển. Một Services được tạo ra trong ứng dụng sẽ là một lớp con trong lớp Service trong Android Platform và chúng ta có thể dùng Activity để khởi động Service này.
- Content Providers: được sử dụng để quản lý một tập các dữ liệu chia sẻ ở những định dạng khác nhau như dưới dạng tập tin, cơ sở dữ liệu SQLite, trên Website hay bất cứ nơi nào mà chúng ta có thể truy xuất được. Thông qua Content Provider, những ứng dụng khác có thể truy xuất dữ liệu từ ứng dụng của chúng ta nếu được cho phép.
- Broadcast Receivers: là loại Application Component trong ứng dụng dùng để lắng nghe các thông điệp được gửi đi từ hệ thống. Bản thân những ứng dụng cũng có thể gửi đi những Broadcast Receivers để những ứng dụng khác biết. Một đối tượng BroadcastReceivers được tạo ra từ lớp cha BroadcastReceiver trong Android Platform.

#### 2.2.4.2 Tập tin Manifest

Được sử dụng để khai báo các Application Component, tập tin Manifest sẽ bao gồm những nội dung sau:

- Quyền hạn của ứng dụng:
- Khai báo các API được sử dụng trong chương trình.
- Khai báo những tính năng phần cứng mà ứng dụng có sử dụng.
- Các thư viện.
- Danh sách các Application Component.

#### 2.2.4.3 Application Resources

Là những tài nguyên được sử dụng cho việc thiết kế giao diện cho chương trình bao gồm hình ảnh, âm thanh, giao diện... Tất cả những Application Resources này được lưu trữ trong thư mục res/. Mỗi thành phần trong Application Resources đều được hệ điều hành Android gán cho một ID tương ứng và lưu trữ trong tập tin R.java trong thư mục gen/ của ứng dụng.

### 2.2.5 Google Map API Android

Google Maps API là thư viện lập trình miễn phí được cung cấp cho người phát triển ứng dụng cùng với đó là các dịch vụ đi kèm như tìm kiếm địa điểm, tìm đường... Mặc dù miễn phí nhưng để có thể sử dụng các dịch vụ này chúng ta cần phải đăng ký và chứng thực với Google.

Trước khi làm việc với Google Map API, chúng ta cần tải API này và chắc chắn rằng mình đã có một Google Map Android API v2 key. Cả API và key đều được cung cấp miễn phí.

Để tạo một ứng dụng android mới sử dụng Google Map Android API v2 chúng ta phải thực hiện theo một số bước khác nhau. Nhiều bước chúng ta chỉ thực hiện một lần nhưng thông tin thu được sẽ hữu ích cho các ứng dụng tương lai sau này. Toàn bộ quá trình thêm một bản đồ được cung cấp bởi Google vào trong một ứng dụng Android được thực hiện như sau:

1. Tải và cấu hình Google Play service SDK<sup>10</sup>. Google Map Android API là một thành phần của SDK này.
2. Lấy được một API key: Để thực hiện được điều này chúng ta cần phải đăng ký một project trong Google APIs Console và nhận một chữ ký chứng thực cho chương trình của chúng ta:
3. Ghi rõ các cài đặt trong Application Manifest.
4. Thêm một Google Map tới một Android project mới hay đã tồn tại.
5. Xuất bản ứng dụng của chúng ta để có thể được sử dụng rộng rãi.

Google Map Android API v2 là một bộ phận của Google Play Service SDK, chúng ta có thể tải nó từ Android SDK manager. Để sử dụng google Maps android apiv2 trong chương trình, chúng ta phải cài đặt Google Play service SDK trước.

#### 2.2.5.1 Google Map API Android v2 Key

Google maps android api v2 sử dụng một hệ thống quản lý Key mới. Những Key từ phiên bản Google Map Android v1 thông thường được biết với tên gọi là MapView sẽ không hoạt động trên phiên bản v2.

Để truy cập vào dịch vụ Google Maps với Maps API, chúng ta phải thêm một Maps API Key vào chương trình. Key này được cung cấp miễn phí cho tất cả mọi người. Chúng ta có thể dùng nó cho bất kỳ chương trình nào gọi tới Maps API và nó hỗ trợ không giới hạn số người dùng. Chúng ta thu được một Maps API Key từ Google APIs Console bằng việc cung cấp chữ ký chứng thực của chương trình chúng ta và tên của package. Một khi đã có được Key, chúng ta thêm nó vào ứng dụng của chúng ta bằng cách khai báo thêm một thành phần trong file AndroidManifest.xml của chương trình.

Để hiểu được quá trình đăng ký chương trình của chúng ta và thu được một Key yêu cầu một vài kiến thức về quá trình xuất bản một ứng dụng Android và những yêu cầu của nó. Tóm lại, tất cả các ứng dụng android phải đăng ký một chữ ký số mà bạn giữ Key riêng. Bởi vì chúng thực số chỉ có duy nhất, họ cung cấp một cách đơn giản để nhận biết chương trình của chúng ta. Nó rất hiệu quả trong việc tìm kiếm chương trình của chúng ta trong hệ thống cũng như Google Play Store, cũng như dò

<sup>10</sup> Google Play service SDK: <http://developer.android.com/google/play-services/index.html>

tìm những tài nguyên mà chương trình của chúng ta sử dụng như Google Maps servers.

Maps API Key được liên kết rõ ràng với cặp chứng thực/package, thay vì cho người sử dụng hay chương trình. Chúng ta chỉ cần duy nhất một Key cho mỗi chứng thực, bất kể bao nhiêu người dùng ứng dụng. Những ứng dụng mà sử dụng chung chứng thực có thể sử dụng chung API Key. Tuy nhiên, nên đăng ký cho mỗi chương trình một chứng thực khác nhau và nhận một khóa khác nhau cho mỗi chương trình.

Để nhận một Key cho một chương trình chúng ta cần thực hiện qua một số bước, những bước thực hiện được nêu ở đây và mô tả cụ thể trong chi tiết của mỗi phần:

1. Lấy lại thông tin về chứng thực của chương trình.
2. Đăng ký một project trong Google APIs Console<sup>11</sup> và thêm vào Maps API như là một dịch vụ của project.
3. Một khi chúng ta có một project, chúng ta có thể yêu cầu một hay nhiều Key.
4. Cuối cùng, chúng ta có thể thêm một hay nhiều Key vào chương trình.

#### 2.2.5.2 *Hiển thị thông tin chứng thực*

Maps API Key dựa trên một hình thức ngắn của chứng thực số, được biết đến với tên gọi SHA-1 fingerprint. Fingerprint là một chuỗi ký tự duy nhất được tạo ra từ thuật toán Hash<sup>12</sup> của SHA-1<sup>13</sup>. Bởi vì fingerprint là một thuật toán đặc biệt nên Google Map sử dụng nó như một cách để chứng thực cho chương trình.

Để hiển thị SHA-1 fingerprint cho chứng thực của chúng ta thì trước tiên chúng ta phải đảm bảo chúng ta đã có chứng thực của mình. Chúng ta có hai cách để nhận chứng thực này:

- Debug certificate: bộ công cụ android SDK tạo ra chứng thực này một cách tự động khi chúng ta thực hiện debug chương trình từ dòng lệnh hay khi chúng ta build hay run chương trình từ eclipse mà không xuất bản chương trình. Chứng thực chỉ được dùng cho một chương trình mà chúng ta đang thử nghiệm. Chúng ta không thể xuất bản chương trình với debug certificate. Chúng ta có thể tạo ra một API Key từ chứng thực này nhưng Key này chỉ được sử dụng cho chương trình thử nghiệm.
- Release certificate: Bộ công cụ Android SDK tạo ra chứng thực này khi chúng ta sử dụng chức năng “release” của IDE. Chúng ta cũng có thể tạo ra chứng thực này thông qua chương trình keytool. Chứng thực này có thể được sử dụng cho một chương trình mà bạn muốn xuất bản. Một khi chúng ta có chứng thực

<sup>11</sup> Google APIs Console: <https://code.google.com/apis/console/>

<sup>12</sup> Hash: [http://en.wikipedia.org/wiki/Hash\\_function](http://en.wikipedia.org/wiki/Hash_function)

<sup>13</sup> SHA-1: <http://en.wikipedia.org/wiki/SHA-1>

chính xác mà chúng ta cần, chúng ta có thể hiển thị SHA-1 fingerprint sử dụng chương trình keytool<sup>14</sup>.

Để có thể nhận được chính xác loại chứng thực cần thiết thì chúng ta phải thực hiện đầy đủ một số bước sau đây:

### Đối với Debug certificate

1. Xác định chính xác vị trí của tập tin “debug.keystore”, mặc định nó nằm chung thư mục với thư mục Android Virtual Device của bạn:
  - OS X and Linux: ~/.android/
  - Windows Vista và Windows 7: C:\Users\your\_username\.android\
 Nếu chúng ta sử dụng eclipse với ADT và chúng ta không chắc tập tin này nằm ở đâu thì bạn có thể kiểm tra đường dẫn tại: Windows>Prefs>Android>Build.
2. Hiển thị thông tin SHA-1 fingerprint
  - Đối với Linux hay OS X, mở giao diện dòng lệnh và nhập vào dòng lệnh sau:  
keytool -list -v -keystore ~/.android/debug.keystore -alias androiddebugkey -storepass android -keypass android
  - Đối với Vista hay Win 7, chạy dòng lệnh sau:  
keytool -list -v -keystore "C:\Users\your\_user\_name\.android\debug.keystore" -alias androiddebugkey -storepass android -keypass android
3. Chúng ta sẽ nhìn thấy kết quả như bên dưới đây:

```
Alias name: androiddebugkey
Creation date: June 01, 2015
Entry type: PrivateKeyEntry
Certificate chain length: 1
Certificate[1]:
Owner: CN=Android Debug, O=Android, C=US
Issuer: CN=Android Debug, O=Android, C=US
Serial number: 4aa9b300
Valid from: Mon Jan 01 08:04:04 UTC 2013 until: Mon Jan 01 18:04:04 PST
2033
Certificate fingerprints:
MD5: AE:9F:95:D0:A6:86:89:BC:A8:70:BA:34:FF:6A:AC:F9
SHA1: BB:0D:AC:74:D3:21:E1:43:07:71:9B:62:90:AF:A1:66:6E:44:5D:75
Signature algorithm name: SHA1withRSA
Version: 3
```

Dòng bắt đầu bằng chuỗi SHA1 sẽ chứa mã chứng thực SHA-1 fingerprint. Chuỗi chứng thực fingerprint chứa 20 ký tự hexa<sup>15</sup> và được ngăn cách với nhau bởi dấu hai chấm.

<sup>14</sup> Keytool: <http://docs.oracle.com/javase/1.4.2/docs/tooldocs/windows/keytool.html>

Đối với release fingerprint

1. Không có vị trí mặc định cho tập tin release keystore, nếu bạn không chỉ định nó khi xuất bản chương trình thì nó sẽ cho ra một tập tin apk không được đăng ký. Bạn phải đăng ký nó trước khi đưa ra cộng đồng sử dụng. Đối với release certificate, chúng ta cần biết về tên, mật khẩu, của keystore và chứng thực của nó. Có thể hiển thị tất cả các tên của khóa trong keystore<sup>16</sup> bằng dòng lệnh:

```
keytool -list -keystore your_keystore_name
```

2. Thay thế “*your\_keystore\_name*” bằng đường dẫn và tên của keystore của chúng ta bao gồm cả đuôi .keystore, cần cung cấp mật khẩu của keystore để có thể hiển thị được những thông tin này.  
Nhập vào dòng lệnh sau trong giao diện dòng lệnh:

```
keytool -list -v -keystore your_keystore_name -alias  
your_alias_name
```

3. Thay thế “*your\_keystore\_name*” bằng đường dẫn và tên của keystore của chúng ta bao gồm cả đuôi .keystore. Thay thế “*your\_alias\_name*” bằng tên mà chúng ta đã đăng ký khi khởi tạo.  
Kết quả hiển thị ra sẽ tương tự như bên dưới:

```
Alias name: <alias_name>
Creation date: Jun 02, 2015
Entry type: PrivateKeyEntry
Certificate chain length: 1
Certificate[1]:
Owner: CN=Android Debug, O=Android, C=US
Issuer: CN=Android Debug, O=Android, C=US
Serial number: 4cc9b300
Valid from: Mon Feb 02 08:01:04 UTC 2013 until: Mon Feb 02
18:05:04 PST 2033
Certificate fingerprints:
MD5: AE:9F:95:D0:A6:86:89:BC:A8:70:BA:34:FF:6B:AC:F9
SHA1:
BB:0D:AC:74:D3:21:E1:43:67:71:9B:62:90:AF:A1:66:6E:44:5D:75
Signature algorithm name: SHA1withRSA
Version: 3
```

<sup>15</sup> Hexa: <http://en.wikipedia.org/wiki/Hexadecimal>

<sup>16</sup> Keystore: <http://docs.oracle.com/javase/1.4.2/docs/api/java/security/KeyStore.html>



Dòng bắt đầu bằng chuỗi SHA1 sẽ chứa mã chứng thực SHA-1 fingerprint. Chuỗi chứng thực fingerprint chứa 20 ký tự hexa và được ngăn cách với nhau bởi dấu hai chấm.

#### 2.2.5.3 Tạo một API project

Một khi chúng ta đã đăng ký thành công chứng thực fingerprint, tạo và chỉnh sửa một project cho chương trình của chúng ta trên Google APIs Console và đăng ký Maps API.

Để nhận được một project và đăng ký với API chúng ta cần thực hiện một số bước như sau:

1. Trong trình duyệt, chúng ta chuyển hướng tới trang <https://code.google.com/apis/console/>
  - Nếu chúng ta chưa từng sử dụng Google API Console trước đây thì cần phải tạo mới một project thông qua nhấp chuột vào nút “Create Project”. Console sẽ tạo cho chúng ta một project có tên là API Project. Ở trang tiếp theo, tên này sẽ được hiện ra phía bên trái, tại đây chúng ta sẽ có thể đổi tên cũng như quản lý Project.
  - Nếu trước đây chúng ta đã sử dụng Google API Console thì sẽ thấy ngay danh sách các project và các dịch vụ. Lúc này chúng ta có thể tạo một project mới hoặc đổi tên project cũ lại để sử dụng
2. Chúng ta sẽ thấy danh sách các dịch vụ nằm ở trung tâm của trang. Nếu vẫn chưa thấy, chọn “service” bên thanh định hướng phía bên trái.
3. Trong danh sách các định vụ này, kéo xuống cho đến khi bạn thấy Google Maps Android API v2. Bật dịch vụ này.
4. Màn hình tiếp theo hiển thị các điều khoản của dịch vụ. Nếu đồng ý, nhấp vào “Accept” để đồng ý. Sau đó nó sẽ trả lại màn hình chính.

Bây giờ chúng ta đã có thể nhận khóa Maps API.

#### 2.2.5.4 Nhận một API Key

Nếu chương trình của chúng ta đã đăng ký sử dụng dịch vụ Google Android Map v2, thì chúng ta có thể yêu cầu nhận một API Key, có thể đăng ký nhiều khóa trên một project.

Để nhận được Key chúng ta phải thực hiện các bước sau đây:

1. Trên thanh định hướng phía bên trái, nhấp chuột vào API Access.
2. Trên trang kết quả, nhấp chuột vào Create New Android Key.
3. Sau đó nhập vào SHA-1 fingerprint và tên của Packet của bạn. ví dụ như:

BB:0D:AC:74:D3:21:E1:43:67:71:9B:62:91:AF:A1:66:6E:44:5D:75;com.example.android.mapexample

4. Google API console trả lại chuỗi gồm 40 ký tự đó là Key của chương trình Android.

**AIzaSyBdVI**-cTICSwYKrZ95SuvNw7dbMuDt1KG0

5. Lưu Key này để sử dụng cho các chương trình sử dụng dịch vụ Google map API v2.

### 2.2.5.5 Một số class cơ bản dùng để hiển thị và quản lý trên android

Ta có thể tóm tắt các class trong GooleMap như sau:

Tên Class	Chức năng
BitmapDescriptor	Dùng để tạo ra một bức hình, đối với marker có thể dùng để tạo ra marker icon, đối với Ground Overlay nó có thể thiết lập một hình ảnh tại một địa điểm trên bề mặt bản đồ. Để có được một Bitmapdescriptor ta sử dụng class BitmapDescriptorFactory.
BitmapDescriptorFactory	Dùng để tạo ra BitmapDescriptor
CameraPosition	Một class dùng để tập hợp tất cả các thuộc tính của việc hiển thị một điểm trên bản đồ.
CameraPosition.Builder	Dùng để xây dựng một camera position.
GroundOverlay	Dùng để hiển thị một hình ảnh trên bề mặt bản đồ tại một vị trí xác định.
GroundOverlayOptions	Dùng để tập hợp các thuộc tính của GroundOverlay với các thuộc tính cơ bản như: <ul style="list-style-type: none"> <li>- Anchor(float u, float v): Xác định điểm neo của hình.</li> <li>- Position(LatLng location, float width, float height): Chỉ ra vị trí, chiều dài và chiều rộng của khung hình.</li> <li>- Image(BitmapDescriptor image): thiết lập hình ảnh hiển thị trong GroundOverlay</li> <li>- Bearing(float bearing): thiết lập góc nhìn nghiêng với tham số là độ lớn của góc, mặc định bằng 0</li> </ul>
LatLng	Là class đại diện cho cặp tọa độ latitude/longitude dùng để xác định vị trí của một điểm bất kỳ trên bản đồ trước khi thể hiện nó. Tham số đầu vào là cặp giá trị kiểu double đại diện cho latitude/longitude.
LatLngBounds	Là class đại diện cho những cặp tọa độ LatLng để tạo thành hình chữ nhật.
Marker	Là class dùng để đánh dấu một điểm trên bản đồ bằng một icon. Marker được vẽ theo định hướng màn hình của thiết bị hơn là so với bề mặt bản đồ. Một Marker gồm có bốn thuộc tính cơ bản là: <ul style="list-style-type: none"> <li>- Anchor: là điểm trên hình mà sẽ đặt vào vị trí trên bản đồ. Mặc định là 50% về phía bên trái của hình và bên cạnh trên của hình.</li> </ul>

	<ul style="list-style-type: none"> <li>- Position: là giá trị LatLng cho vị trí của marker trên bản đồ. Chúng ta có thể thay đổi vị trí này bất kể khi nào chúng ta muốn di chuyển marker.</li> <li>- Title: Là chuỗi ký tự sẽ hiển thị trên cửa sổ khi chúng ta nhấp vào marker.</li> <li>- Snippet: là chuỗi ký tự hiển thị bên dưới Title.</li> <li>- Icon: là ảnh sẽ hiển thị đại diện cho marker. Nếu không được thiết lập thì marker sẽ sử dụng giá trị mặc định. Chúng ta không thể thay đổi thuộc tính này của marker một khi chúng ta đã tạo ra nó.</li> <li>- Drag status: Nếu chúng ta muốn kéo thả Marker thì thiết lập giá trị này là true.</li> </ul> <p>Visibility: Mặc định giá trị này sẽ thiết lập là true, nếu muốn ẩn marker thì chuyển thành false.</p>
MarkerOptions	Là class tập hợp cho các thuộc tính của Marker.
Polygon	<p>Là class dùng để vẽ một hình nhiều cạnh trên bề mặt trái đất. Hình này có thể lõm hoặc lõm gồm các thuộc tính sau đây:</p> <ul style="list-style-type: none"> <li>- Outline: là danh sách các đỉnh của hình sắp xếp theo chiều ngược của kim đồng hồ.</li> <li>- Holes: là vùng nằm bên trong Polygon không được lấp kín. Holes được xác định tương tự theo cách của Outline. Một Outline có thể chứa nhiều holes nhưng các holes này không được chồng lên nhau.</li> <li>- Stroke Width: độ dày của đường nối giữa các đỉnh của polygon.</li> <li>- Stroke color: màu của đường nối giữa các đỉnh của polygon.</li> <li>- Fill color: màu nền của polygon.</li> <li>- Visibility: dùng để thiết lập thuộc tính ẩn hiện cho polygon.</li> <li>- Geodesis status: xác định đường nối các đỉnh của polygon có vẽ theo địa hình hay không.</li> </ul>
PolygonOptions	Là class tập hợp các thuộc tính của polygon
Polyline	<p>Là class đại diện cho danh sách các điểm được nối với nhau bằng các đường thẳng. Các thuộc tính cơ bản của polyline bao gồm:</p> <ul style="list-style-type: none"> <li>- Points: là danh sách các điểm được nối liên tiếp với nhau. Mặc định polyline không khép kín trừ khi điểm đầu trùng với điểm cuối.</li> <li>- Width: thể hiện chiều rộng của đường nối các điểm.</li> <li>- Color: thể hiện màu sắc của các đường nối các điểm.</li> <li>- Visibility: dùng để thiết lập thuộc tính ẩn hiện cho polyline.</li> <li>- Geodesis status: xác định đường nối các đỉnh của polyline có vẽ theo địa hình hay không.</li> </ul>

PolylineOptions	Là class thể hiện các thuộc tính của polyline.
Tile	Là class thể hiện các thuộc tính của polyline.
TileOverlay	Là class dùng để hiển thị hình ảnh trên đỉnh của bản đồ. Những hình ảnh này có thể trong suốt nên cho phép chúng ta có thể thêm những tính năng vào trong bản đồ.
TileOverlayOptions	Là class tập hợp các thuộc tính của TileOverlay.
TileProvider	Là class dùng để dùng cho việc hiển thị hình ảnh trong TileOverlay.
UriTileProvider	Là một bộ phận của TileProvider nhưng nó chỉ yêu cầu một URL trỏ đến hình ảnh được cung cấp.
VisibleRegion	Là class dùng để vẽ một polyline bằng việc xác định bốn điểm.
CameraUpdate	Xác định sự di chuyển của camera giúp thay đổi góc nhìn của người dùng bằng cách sử dụng các phương thức như animateCamera(CameraUpdate), animateCamera(CameraUpdate, GoogleMap.CancelableCallback) hay moveCamera(CameraUpdate).
CameraUpdateFactory	Là class dùng để chứa các phương thức để khởi tạo một đối tượng CameraUpdate. Để thay đổi đối tượng góc nhìn của người sử dụng trong bản đồ chúng ta dùng các phương thức như: animateCamera(CameraUpdate), animateCamera(CameraUpdate, GoogleMap.CancelableCallback) hay moveCamera(CameraUpdate)
GoogleMap	Là một interface mà một đối tượng nào đó phải kế thừa khi muốn hiển thị và quản lý bản đồ.
GoogleMapOptions	Là class chứa các thiết lập để khởi tạo một đối tượng GoogleMap.
MapView	Dùng để hiển thị bản đồ.
Projection	Được sử dụng để chuyển đổi giữa vị trí trên màn hình thành tọa độ trên bản đồ. Class này chỉ gồm hai phương thức chính là: <ul style="list-style-type: none"> <li>- public LatLng fromScreenLocation (Point point): giúp chuyển một điểm trên màn hình về tọa độ vật lý trên bản đồ.</li> <li>- public Point toScreenLocation (LatLng location): giúp chuyển một tọa độ vật lý trên bản đồ thành tọa độ điểm ảnh trên màn hình.</li> </ul>
UiSettings	Là class chứa các thiết lập về việc hiển thị các thành phần hỗ trợ trên bản đồ.

**Bảng 2-2: Các Class trong GooleMap**

## 2.2.6 Google Place API Android

Google Places API cho phép chúng ta thực hiện các truy vấn thông tin về địa điểm bao gồm các địa điểm lân cận cũng như chính xác các địa điểm. Google Place API trả về danh sách các địa điểm cùng thông tin tóm gọn về mỗi địa điểm, còn thông tin chi tiết sẽ được cung cấp thông qua truy vấn Place Details. Theo [2] ta có thể tóm tắt các loại truy vấn của Google Place API Android như sau:

### 2.2.6.1 Truy vấn thông tin gần chúng ta

Cung cấp cho chúng ta thông tin về địa điểm bên trong một khu vực được chỉ định sẵn. Chúng ta có thể chọn lọc kết quả thông qua việc cung cấp thêm các từ khóa hay loại của địa điểm mà chúng ta đang muốn tìm kiếm.

HTTP URL theo dạng:

<https://maps.googleapis.com/maps/api/place/nearbysearch/output?parameters>

Giá trị tham số output là định dạng văn bản của Google sẽ trả về khi chúng ta gửi yêu cầu lên server bao gồm:

- Json: chỉ ra giá trị trả về là JavaScript Object Notation(Json)
- Xml: chỉ ra giá trị trả về là XML.

Chúng ta cần chắc chắn cung cấp đầy đủ giá trị của các tham số quan trọng trong yêu cầu để có thể nhận được chính xác thông tin về địa điểm cần tìm, giống như URLs chuẩn thì các tham số sẽ được cách nhau bằng dấu “&”.

Những tham số yêu cầu cơ bản bao gồm:

- Key: giá trị của tham số là API Key mà chúng ta được cung cấp thông qua việc đăng ký với google.
- Language: giá trị của tham số là ngôn ngữ mà Google sẽ trả về.
- Radius: giá trị của tham số là bán kính tính từ tâm của điểm mà chúng ta muốn truy vấn thông tin.
- Sensor: giá trị của tham số này là true hay false dùng để xác định có hay không sử dụng những bộ cảm biến như GPS ... để xác định vị trí.

Maps API sử dụng cho khách hàng là doanh nghiệp không bao gồm tham số chữ ký trong các truy vấn.

Ví dụ cần truy vấn các địa điểm về ẩm thực tại Sydney trong bán kính 500m kể từ vị trí hiện tại chúng ta sẽ sử dụng câu truy vấn sau:

<https://maps.googleapis.com/maps/api/place/nearbysearch/json?location=33.8670522,151.1957362&radius=500&types=food&name=harbour&sensor=false&key=AddYourOwnKeyHere>

### 2.2.6.2 Text Search Request

Google Places API Text Search Service là một dịch vụ web dùng để trả lại thông tin về thông tin của một địa điểm cụ thể. Dịch vụ sẽ trả lại danh sách địa điểm phù hợp với truy vấn. Chúng ta có thể sử dụng Google Detail để có thêm được những thông tin cụ thể về các địa điểm được trả về.

- Truy vấn của Place Text Search là một HTTP URL theo dạng:

```
https://maps.googleapis.com/maps/api/place/textsearch/output?parameters
```

- Giá trị của tham số output xác định kiểu của kết quả trả về là Json hay xml.
- Các tham số yêu cầu cơ bản bao gồm:
  - Query: là một chuỗi truy vấn theo kiểu String dùng để tìm kiếm địa điểm.
  - Key và sensor: là những tham số tương tự với Nearby Search Request.
- Ví dụ chúng ta muốn tìm khách sạn tại Sydney thì chuỗi truy vấn sẽ như sau:

```
https://maps.googleapis.com/maps/api/place/textsearch/xml?query=restaurants+in+Sydney&sensor=true&key=AddYourOwnKeyHere
```

### 2.2.6.3 Radar Search Request

Google Places API Radar Search Service cho phép chúng ta tìm trên 200 địa điểm một lần, tuy nhiên thông tin về những địa điểm này sẽ ít hơn bình thường so với sử dụng Text Search hay Nearby Search. Với Radar Search giúp người sử dụng có thể xác định khu vực yêu thích của mình trên bản đồ. Để có thêm thông tin cụ thể về các địa điểm trong danh sách trả về chúng ta có thể sử dụng Google Detail.

- Truy vấn của Radar Search Request là một HTTP URL theo dạng:

```
https://maps.googleapis.com/maps/api/place/radarsearch/output?parameters
```

- Giá trị của tham số output xác định kiểu của kết quả trả về là Json hay xml.
- Các tham số cơ bản của Radar Search tương tự như Nearby Search.
- Ví dụ chúng ta muốn tìm kiếm những quán cà phê có tên là vegetarian trong khu vực có bán kính 5000m thì chuỗi truy vấn sẽ như sau:

<https://maps.googleapis.com/maps/api/place/radarsearch/json?location=48.859294,2.347589&radius=5000&types=food|cafe&sensor=false&keyword=vegetarian&key=AddYourOwnKeyHere>

#### 2.2.6.4 Kết quả truy vấn được trả về từ Google Search

- Định dạng của kết quả trả về được chỉ định thông qua tham số output trong truy vấn đến Google Search.
- Kết quả trả về của TextSearch tương tự như NearbySearch ngoại trừ thuộc tính formatted\_address thay thế bằng vicinity.

Một Json trả về thì gồm 4 thành phần chính là:

- “status” chứa metadata của yêu cầu.
- “results” chứa mảng cách địa điểm tra về với thông tin cụ thể của từng địa điểm.
- “http\_attributions” chứa các thiết lập về thẩm quyền của danh sách các địa điểm đang được hiển thị.
- “next\_page\_token” chứa token dùng để hiển thị cho 20 kết quả tiếp theo. Để hiển thị thêm các kết quả này thì chúng ta chỉ cần truyền giá trị này vào tham số pagetoken. Tối đa cho một kết quả trả về từ Google là 60 địa điểm và được chia thành 3 trang khác nhau.

### 2.2.7 Google Direction API Android

Google Directions API là một dịch vụ dùng để tính toán đường đi giữa hai điểm trên bản đồ sử dụng các truy vấn HTTP. Chúng ta có thể tìm kiếm đường đi theo các chế độ khác nhau như: xe buýt, đi bộ, xe ô tô. Đường đi được xác định thông qua tọa độ của điểm đầu và điểm cuối được nhập vào thông qua một chuỗi ký tự hay tọa độ latitude/longitude. Google Direction API có thể trả lại nhiều đường đi khác nhau dựa trên danh sách các điểm mốc. Dù là một dịch vụ miễn phí nhưng đối với một người dùng thông thường thì chỉ có thể sử dụng 2500 truy vấn trong một ngày, còn đối với người dùng trong kinh doanh thì có thể sử dụng 100000 truy vấn trong một ngày.

Cấu trúc truy vấn của Google Directions API theo dạng:

<http://maps.googleapis.com/maps/api/directions/output?parameters>

Giá trị của biến output là Json hay XML.

Các tham số yêu cầu bao gồm:



- Origin: Vị trí điểm đầu của đường đi. Giá trị tham số có thể là tọa độ latitude/longitude hoặc địa chỉ. Chúng ta cần chú ý rằng trong chuỗi giá trị của tham số này không được sử dụng khoảng trắng.
- Destination: Vị trí điểm cuối của đường đi. Giá trị tham số có thể là tọa độ latitude/longitude hoặc địa chỉ. Chúng ta cần chú ý rằng trong chuỗi giá trị của tham số này không được sử dụng khoảng trắng.
- Sensor: Xác định có hay không sử dụng bộ cảm biến cho việc tính toán đường đi.

Các tham số tùy chọn cơ bản bao gồm:

- Mode: dùng để xác định loại của tuyến đường cần tìm. Google cung cấp 4 loại chính bao gồm:
  - Driving: loại dùng để tìm đường cho xe ô tô hay các phương tiện lớn.
  - Walking: loại dùng để tìm đường cho người đi bộ.
  - Bicycling: loại dùng để tìm đường cho xe đạp.
  - Transit: loại dùng để tìm đường cho các phương tiện công cộng.
- Waypoints: là danh sách các điểm cố định mà tuyến đường cần phải đi qua.
- Avoid: tham số đưa vào khi tìm đường sẽ tránh bao gồm 2 giá trị là:
  - Tolls: tránh những chỗ có thu phí.
  - Highways: tránh đường cao tốc.
- Alternatives: tham số này xác định có hay không việc hiển thị nhiều hơn một tuyến đường cho người dùng lựa chọn.
- Language: xác định ngôn ngữ của hướng dẫn chỉ đường được trả về. Mặc định sẽ là tiếng Anh.

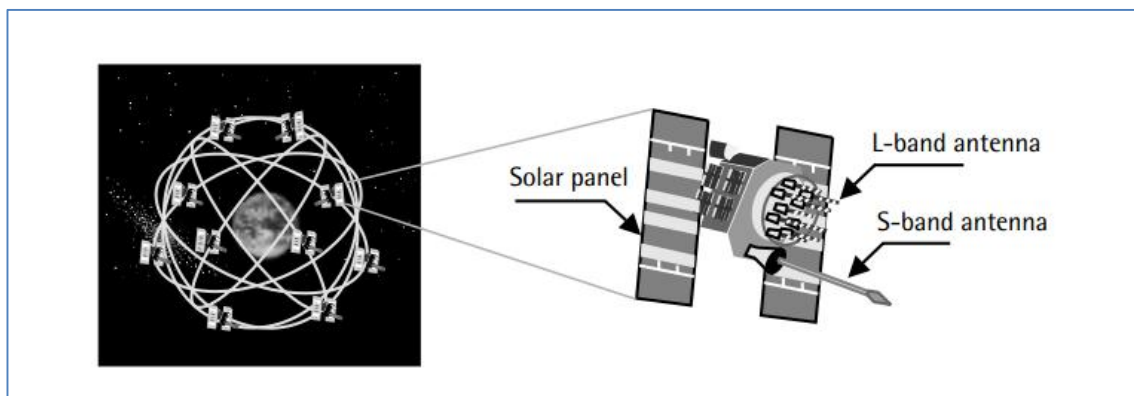
## 2.3 Công nghệ GPS

Hệ thống định vị toàn cầu (GPS) là một hệ thống vệ tinh điều hướng được phát triển bởi bộ quốc phòng mỹ trong những năm đầu của thập niên 70. Ban đầu, GPS được phát triển như là một hệ thống được sử dụng trong mục đích quân sự. Tuy nhiên, thời gian sau nó được sử dụng thêm vào các mục đích dân sự.

GPS cung cấp liên tục thông tin vị trí và thời gian bất kỳ đâu trên thế giới trong bất kỳ điều kiện thời tiết nào. Nó phục vụ một không hạn chế người dùng cũng như được sử dụng cho các lý do an ninh. GPS là hệ thống một chiều thụ động có nghĩa là những người dùng chỉ có thể nhận tín hiệu từ vệ tinh mà không thể gửi trả. Để sử dụng GPS người dùng chỉ cần có thiết bị thu sóng và không phải trả bất kỳ chi phí nào khác. Sau khi tổng hợp từ [3] chúng ta có thể đưa ra một số khái niệm cơ bản về GPS như dưới đây.

### 2.3.1 Tổng quan về GPS

GPS bao gồm một 24 vệ tinh được liên kết với nhau. Hệ thống này được hoàn thành vào tháng 7 năm 1993 và công bố chính thức vào ngày 8 tháng 12 năm 1993. Nó được xem là cơ sở quan trọng của hệ thống GPS với tên gọi “Năng lực hoạt động ban đầu” gọi tắt là IOC(initial operational capability). Để đảm bảo hệ thống vệ tinh này bao phủ toàn bộ trái đất một cách liên tục thì 24 vệ tinh được sắp xếp sao cho cứ mỗi 4 vệ tinh được đặt trong 6 mặt phẳng quỹ đạo (Hình 2-8). Với sự bố trí này sẽ có trên 4 vệ tinh luôn hiện hữu ở bất kỳ vị trí nào trên trái đất, nếu góc ngẩng là  $10^\circ$ . Để định vị tại một vị trí ta cần tối thiểu 4 vệ tinh.



Hình 2-8: GPS

Quỹ đạo của các vệ tinh GPS gần như là hình tròn (Một hình elipse với tâm sai cực đại khoảng 0.01) với độ nghiêng khoảng  $55^\circ$  so với xích đạo. Nửa trục lớn của quỹ đạo GPS có độ dài khoảng 26560 km, tức là vệ tinh có độ cao khoảng 20200km bên trên bề mặt trái đất. Chu kỳ quỹ đạo vệ tinh tương ứng khoảng 12 giờ thiên văn.

Hệ thống GPS đạt được khả năng hoạt động đầy đủ gọi tắt là FOC(full operational capability) vào ngày 17 tháng 7 năm 1995 đảm bảo khả năng hoạt động thực tế tối thiểu 24 vệ tinh GPS. Thực tế thì khi GPS đạt FOC thì trên quỹ đạo hoạt động thường có nhiều hơn 24 vệ tinh.

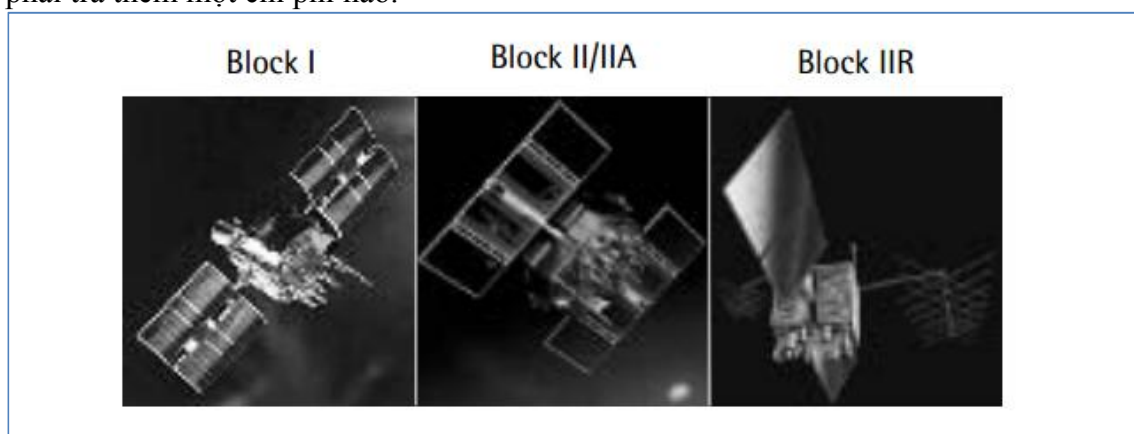
### 2.3.2 Những phân đoạn GPS

GPS bao gồm 3 phân đoạn chính là: phân đoạn không gian, phân đoạn điều khiển, phân đoạn người dùng.

- Phân đoạn không gian chính là nhóm gồm 24 vệ tinh, mỗi vệ tinh GPS phát đi tín hiệu bao gồm những thành phần sau: hai sóng sin là thành phần sóng mang, hai chuỗi dữ liệu số, và một thông điệp điều hướng, dữ liệu số và thông điệp

điều hướng. Sóng mang và chuỗi dữ liệu số chủ yếu dùng để xác định khoảng cách từ máy thu của người sử dụng đến những vệ tinh GPS. Thông điệp điều hướng bao gồm tọa độ vệ tinh được biểu diễn dưới hàm biến đổi theo thời gian và các thông tin cần thiết khác. Những tín hiệu này được phát đi bởi các đồng hồ nguyên tử có độ chính xác cao được gắn trên những vệ tinh.

- Phân đoạn điều khiển hệ thống GPS bao gồm một mạng lưới rộng khắp những trạm theo dõi và một trạm điều khiển đặt tại bang Colorado của Mỹ. Nhiệm vụ của phân đoạn này là theo dõi, quản lý các vệ tinh GPS bao gồm vị trí của vệ tinh, tình trạng hệ thống, hoạt động của các đồng hồ nguyên tử.. và một số vấn đề khác. Những thông tin này được truyền tải trong thông tin về vị trí của vệ tinh và được chứa trong máy thu, khi vệ tinh di chuyển thì các thông tin này liên tục được cập nhật lại. Những thông tin này sau đó được đóng gói lại và gửi lên vệ tinh.
- Phân đoạn người dùng bao gồm tất cả người dân và quân đội. Với một bộ thu được kết nối với anten GPS, một người dùng có thể nhận tín hiệu GPS, tín hiệu này dùng để xác định vị trí mà không cần quan tâm vị trí của người đó đang ở đâu. GPS là một dịch vụ được cung cấp cho tất cả người sử dụng mà không phải trả thêm một chi phí nào.



**Hình 2-9: 3 phân đoạn chính của GPS**

### 2.3.3 Những thế hệ của vệ tinh GPS

Trong hệ thống ban đầu, nhóm vệ tinh GPS được xây dựng với chuỗi gồm 11 vệ tinh, gọi là những Block I. Vệ tinh đầu tiên trong chuỗi này bắt đầu hoạt động đầu tiên vào ngày 22 tháng 2 năm 1978, cái cuối cùng hoạt động vào ngày 9 tháng 10 năm 1985. Block I chủ yếu được sử dụng với mục đích chủ yếu là thí nghiệm. Góc nghiêng mặt phẳng quỹ đạo của vệ tinh này so với đường xích đạo là 630, sau đó được sửa đổi ở các thế hệ sau. Block I cuối cùng được rút ra vào ngày 18 tháng 11 năm 1995.

Thế hệ thứ 2 của các vệ tinh GPS được biết tới với tên gọi là Block II/IIA. Block IIA là phiên bản nâng cấp của khối II, với việc tăng khả năng lưu trữ dữ liệu thông điệp điều hướng từ 14 ngày đối với khối II đến 180 ngày đối với khối IIA, điều

này giúp tăng khả năng hoạt động độc lập của vệ tinh mà không cần sự hỗ trợ từ mặt đất. Có tổng cộng 28 Block II/IIA đã hoạt động trong khoảng thời gian từ tháng 2 năm 1989 đến tháng 11 năm 1997. Mặt phẳng nghiêng của Block II/IIA so với đường xích đạo là 550, thời gian hoạt động trong không trung là 7.5 năm nhưng thường chúng tồn tại trên không gian lâu hơn thế rất nhiều. Block II/IIA được tăng cường khả năng bảo mật hơn so với các phiên bản trước bằng việc thêm vào khả năng lựa chọn người dùng và chống lừa đảo trong cấu trúc tín hiệu truyền về máy thu.

Bản nâng cấp tiếp theo của Block IIA là Block IIR, nó có khả năng tương thích với các Block thế hệ trước đó. Block IIR bao gồm 21 vệ tinh với thời gian sống được thiết kế là 10 năm. Thêm vào đó là độ chính xác cao hơn trước, những Block IIR có khả năng hoạt động độc lập cao nhờ giảm được phương sai của độ chính xác theo thời gian. Khả năng định vị của Block này được nâng cao nhờ chúng có khả năng tự sắp xếp lẫn nhau. Vào tháng 7 năm 2001, 6 vệ tinh thuộc Block IIR đã hoạt động thành công.

Theo sau Block IIR là Block IIF bao gồm 33 vệ tinh, khoảng thời gian sống của vệ tinh là 15 năm. Những vệ tinh thuộc khối này được tăng cường tính chính xác trong định vị.

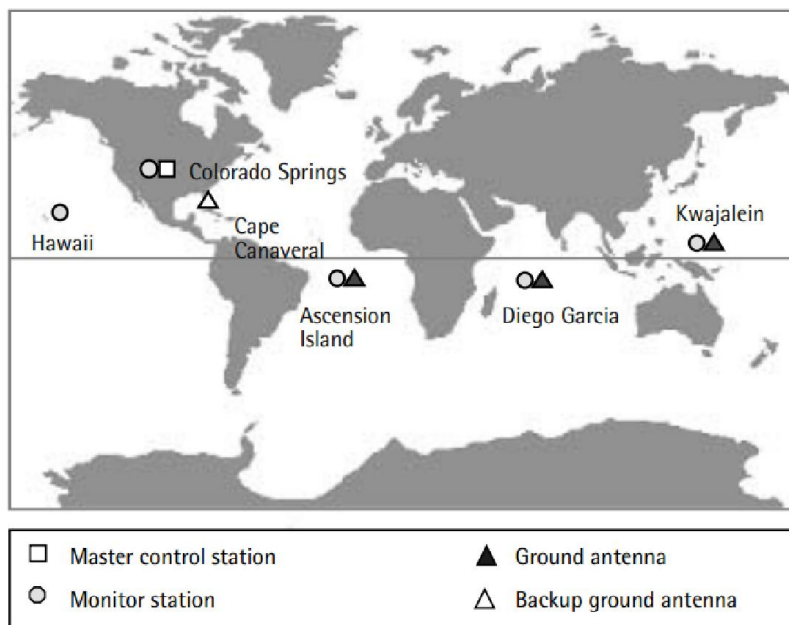
Hệ thống Block III là hệ thống mới nhất được thiết kế và phát triển bởi tập đoàn Lockheed Martin<sup>17</sup> theo đơn đặt hàng của bộ quốc phòng mỹ, vòng đời của vệ tinh thuộc khối này là 15 năm. Hiện tại đang có 4 vệ tinh đang được phát triển và đến năm 2014 sẽ nâng lên thành 35 vệ tinh. Các vệ tinh này được thiết kế với trọng lượng nhẹ, độ bền cao bằng vật liệu composite<sup>18</sup>. Sự thay đổi đầu tiên là trong việc truyền tín hiệu GPS dành cho dân sự sẽ không sử dụng đường truyền có tần số L1 mà thay vào đó bằng tần số L2 (1227,6 MHz), nó được sử dụng cho các Block IIR và các Block sau này. Tín hiệu truyền trên tần số L2 tăng thêm độ chính xác về vị trí cũng như tăng khả năng dẫn đường nhờ vào hệ thống tín hiệu dự phòng giảm khả năng bị nhiễu và mất sóng.

#### **2.3.4 Những vị trí điều khiển**

Phân đoạn điều khiển của GPS bao gồm một trạm điều khiển chính và rộng khắp những trạm giám sát và những trạm điều khiển mặt đất.

<sup>17</sup> Lockheed Martin: [http://vi.wikipedia.org/wiki/Lockheed\\_Martin](http://vi.wikipedia.org/wiki/Lockheed_Martin)

<sup>18</sup> Composite: [http://vi.wikipedia.org/wiki/V%E1%BA%ADt\\_li%E1%BB%87u\\_composite](http://vi.wikipedia.org/wiki/V%E1%BA%ADt_li%E1%BB%87u_composite)



**Hình 2-10: Phân đoạn điều khiển của GPS**

Có 5 trạm giám sát và định vị tại Colorado Springs, Hawaii, Kwajalein, Diego Garcia, và đảo Ascension. Vị trí hoặc tọa độ của những trạm giám sát này được xác định chính xác. Những trạm này có mục đích theo dấu vết liên tục tất cả những vệ tinh GPS trong tầm nhìn, upload thông tin, điều chỉnh quản lý hoạt động của các vệ tinh.

### 2.3.5 GPS-một vài khái niệm cơ bản

Mỗi vệ tinh GPS phát liên tục một tín hiệu vô tuyến tạo thành từ tổng thể 2 sóng mang<sup>19</sup>, 2 mã và một thông điệp điều hướng, bộ thu GPS phân tích thông tin này thu được bao gồm các khoảng cách tới những vệ tinh GPS và tọa độ của vệ tinh xác định qua thông điệp điều hướng.

Theo lý thuyết chỉ duy nhất cần 3 khoảng cách đến 3 vệ tinh mà được theo dấu vết một cách đồng thời. Trong trường hợp này, bộ thu sẽ được định vị tại chỗ giao nhau của ba hình cầu, mỗi hình cầu này có một bán kính tương ứng với khoảng cách vệ tinh – bộ thu và tâm vệ tinh. Tuy nhiên cần đến 4 vệ tinh để mô tả độ lệch giữa đồng hồ bộ thu và đồng hồ vệ tinh. Độ chính xác của phương pháp này là giới hạn trong 100m theo chiều ngang, 156m theo chiều dọc, và 340ns theo thành phần thời gian. Độ chính xác này tương đối thấp vì vậy phương pháp này không còn được sử dụng và được thay thế bằng phương pháp vi sai, trong đó sử dụng hai bộ thu theo dấu

<sup>19</sup> Sóng mang trong điều chế tín hiệu:

[http://vi.wikipedia.org/wiki/%C4%90i%E1%BB%81u\\_ch%E1%BA%BF\\_t%C3%ADn\\_hi%E1%BB%87u](http://vi.wikipedia.org/wiki/%C4%90i%E1%BB%81u_ch%E1%BA%BF_t%C3%ADn_hi%E1%BB%87u)

vết đồng thời cùng một vệ tinh. Trong trường hợp này, có thể đạt được độ chính xác từ vài centimet đến vài mét.

Lợi ích khác của GPS là khả năng mô tả vận tốc của người sử dụng thông qua một vài phương pháp. Phương pháp được sử dụng rộng rãi nhất là đánh giá tần số Doppler<sup>20</sup> của tín hiệu GPS thu được. Độ dịch Doppler được xem như là chuyển động tương đối giữa bộ thu và vệ tinh.

## 2.3.6 Nguyên lý hoạt động của GPS

### 2.3.6.1 Hoạt động cơ bản

Các vệ tinh GPS bay vòng quanh Trái Đất hai lần trong một ngày theo một quỹ đạo rất chính xác và phát tín hiệu có thông tin xuống Trái Đất. Các máy thu GPS nhận thông tin này và bằng phép tính lượng giác tính được chính xác vị trí của người dùng. Về bản chất máy thu GPS so sánh thời gian tín hiệu được phát đi từ vệ tinh với thời gian nhận được chúng. Sai lệch về thời gian cho biết máy thu GPS ở cách vệ tinh bao xa. Rồi với nhiều quãng cách đo được tới nhiều vệ tinh máy thu có thể tính được vị trí của người dùng và hiển thị lên bản đồ điện tử của máy. Máy thu phải nhận được tín hiệu của ít nhất ba vệ tinh để tính ra vị trí hai chiều (kinh độ và vĩ độ) và để theo dõi được chuyển động. Khi nhận được tín hiệu của ít nhất 4 vệ tinh thì máy thu có thể tính được vị trí ba chiều (kinh độ, vĩ độ và độ cao). Một khi vị trí người dùng đã tính được thì máy thu GPS có thể tính các thông tin khác, như tốc độ, hướng chuyển động, bám sát di chuyển, khoảng hành trình, quãng cách tới điểm đến, thời gian Mặt Trời mọc, lặn và nhiều thứ khác nữa.

### 2.3.6.2 Độ chính xác của GPS

Độ chính xác của các thiết bị GPS ngày nay được tăng lên đáng kể, nhờ vào thiết kế hoạt động song song nhiều kênh của nó. Các máy thu 12 kênh song song nhanh chóng bám sát các quả vệ tinh khi vừa mới phát tín hiệu và chúng duy trì chắc chắn quan hệ này dù là trong ở rừng rậm rạp hay trong các tòa nhà cao tầng. Độ chính xác của các máy GPS trung bình khoảng 15 mét, các yếu tố ảnh hưởng đến độ chính xác này là trạng thái khí quyển và các yếu tố gây nhiễu trong không khí khác. Các máy thu phát sóng được trang bị khả năng WAAS (Wide Area Augmentation System) có thể tăng độ chính xác trung bình dưới 3m. WAAS là một dịch vụ sửa lỗi để tăng độ chính xác của các thiết bị GPS. Hệ thống bao gồm một mạng các đài thu tín hiệu GPS và phát tín hiệu đã sửa lỗi bằng các máy phát hiệu chỉnh. Để thu được tín hiệu đã sửa lỗi, người dùng phải có máy thu tín hiệu vi sai bao gồm cả ăng-ten để dùng cùng với máy thu GPS của họ.

---

<sup>20</sup> Doppler: [http://vi.wikipedia.org/wiki/Hi%E1%BB%87u\\_%E1%BB%A9ng\\_Doppler](http://vi.wikipedia.org/wiki/Hi%E1%BB%87u_%E1%BB%A9ng_Doppler)

## 2.4 PHP, MySQL, APACHE

### 2.4.1 PHP và Yii Framework

#### 2.4.1.1 PHP và mô hình MVC

##### PHP

PHP (PHP: Hypertext Preprocessor) là một ngôn ngữ lập trình kịch bản hay một loại mã lệnh chủ yếu được dùng để phát triển các ứng dụng viết cho máy chủ, mã nguồn mở, dùng cho mục đích tổng quát. Nó rất thích hợp với web do được tối ưu hóa cho các ứng dụng web, tốc độ nhanh, nhỏ gọn, cú pháp giống C và Java, dễ học và thời gian xây dựng sản phẩm tương đối ngắn hơn so với các ngôn ngữ khác nên PHP đã nhanh chóng trở thành một ngôn ngữ lập trình web phổ biến nhất thế giới.

PHP được phát triển từ một sản phẩm có tên là PHP/FI. PHP/FI do Rasmus Lerdorf tạo ra năm 1995, ban đầu được xem như là một tập con đơn giản của các mã kịch bản Perl để theo dõi tình hình truy cập đến bản sơ yếu lý lịch của ông trên mạng. Ông đã đặt tên cho bộ mã kịch bản này là ‘Personal Home Page Tools’.

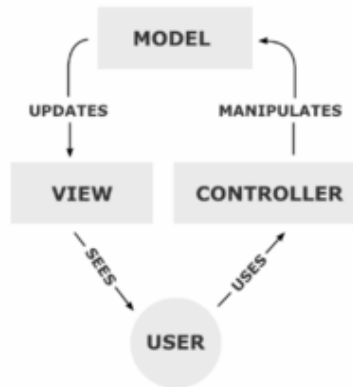
##### Mô hình MVC

MVC được khai sinh với framework được phát triển bởi Tryve Reenskaug cho nền tảng Smalltalk vào cuối những năm 70. Từ đó trở đi nó đóng vai trò là một qui tắc quan trọng trong hầu hết các framework giao diện người dùng và trong các ý tưởng thiết kế giao diện người dùng. Một số thông tin cơ bản về MVC:

- Tương tác giữa các thành phần.

Ngoài việc chia ứng dụng thành ba thành phần, thiết kế MVC xác định các tương tác giữa chúng.

- Một Controller có thể gửi các lệnh đến các view liên kết đến nó để thay đổi cách hiển thị của view. Nó có thể gửi các lệnh đến model để cập nhật trạng thái của model.
- Một model thông báo cho các view và các controller liên quan với nó khi có sự thay đổi trạng thái. Việc thông báo cho phép các view xử lý các output đã thay đổi, và các controller thay đổi các lệnh có thể thực thi. Một sự thực thi MVC thụ động bỏ qua các thông báo này, bởi vì ứng dụng không yêu cầu chúng hoặc nền tảng phần mềm không hỗ trợ.
- Một view thu thập thông tin từ model để tạo ra một giao diện đầu ra.



**Hình 2-11: Mô hình MVC tổng quát.**

- Cách sử dụng MVC trong các ứng dụng web.

MVC được xem như nhắm tới để trở thành một kiến trúc cho các ứng dụng World Wide Web. Một vài framework ứng dụng thương mại và phi thương mại đã được tạo ra với việc sử dụng MVC là bắt buộc. Theo cách tiếp cận này, các client gửi các request hyperlink hoặc form input đến controller và sau đó nhận về một trang web đầy đủ và cập nhật (Hoặc một tài liệu khác) từ view; thành phần model tồn tại hoàn toàn trên server. Khi các công nghệ client đã phát triển, các framework như JavaScriptMVC và Backbone đã được tạo ra cho phép các thành phần MVC được thực thi một phần trên client.

- MVC làm việc như thế nào.

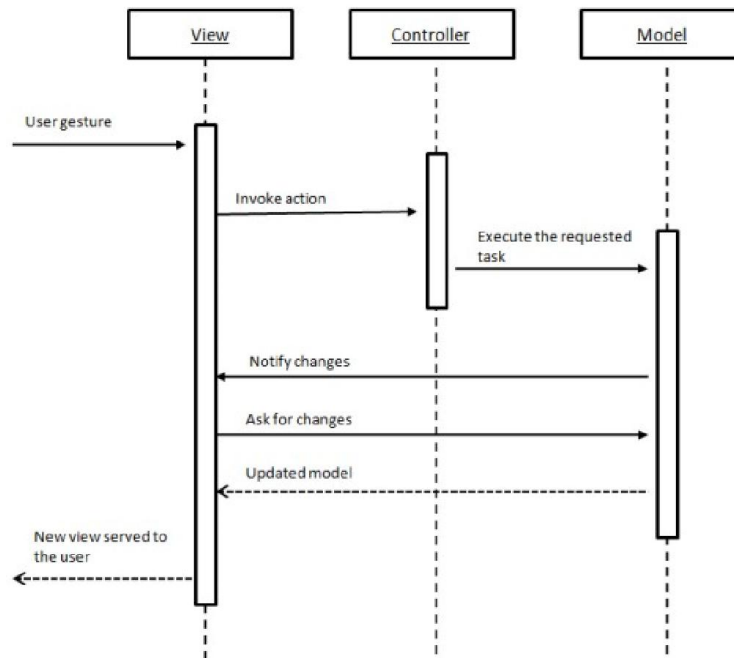
Nhìn lại sơ đồ phía trên, ta thấy có mũi tên nét liền và những mũi tên nét đứt. Những mũi tên nét đứt được hình thành trên quan điểm của người dùng mà không phải là của những nhà thiết kế phần mềm thực sự. Do đó chúng ta chỉ quan tâm đến những mũi tên còn lại.

Đây là một cách đơn giản để mô tả lại luồng sự kiện được xử lý trong MVC:

- User tương tác với View, bằng cách click vào button, user gửi yêu cầu đi.
- Controller nhận và điều hướng chúng đến đúng phương thức xử lý ở Model.
- Model nhận thông tin và thực thi các yêu cầu.



- Khi Model hoàn tất việc xử lý, View sẽ nhận kết quả từ Model và hiển thị lại cho người dùng.



**Hình 2-12: Hoạt động của mô hình MVC**

## MVC và PHP

Mô hình MVC là một cách chia nhỏ ứng dụng, thậm chí là một phần của giao diện của ứng dụng thành 3 phần: model, view và controller. MVC được phát triển ban đầu để ánh xạ các quá trình truyền thống như: nhập, xử lý, xuất thành các lĩnh vực GUI:

Input → Processing → Output

Controller → Model → View.

### - Model.

Một model là một đối tượng đại diện cho dữ liệu hoặc một hành động như: một bảng trong CSDL.

Model quản lý hành vi hoặc dữ liệu của miền ứng dụng, phản hồi lại các yêu cầu truy xuất thông tin về trạng thái của nó và phản hồi lại các chỉ thị để thay đổi trạng thái.

Model là một phần đại diện cho trạng thái và các hành vi mức thấp của component. Nó quản lý trạng thái và thực hiện các sự chuyển đổi trạng thái. Model không có tri thức cụ thể về controller hoặc view của nó. View là một phần quản lý sự hiển thị hình ảnh của trạng thái được đưa ra bởi model. Một model có thể có nhiều hơn 1 view.

Chú ý rằng model không cần thiết có một database cố định. Nó có thể truy cập qua một đối tượng truy cập dữ liệu (DAO) riêng biệt.

#### - **View.**

Một view là một số form hiển thị trạng thái của model.

View quản lý việc xuất ra các dữ liệu đồ họa hay text

View chỉ ra nội dung của một model. Nó truy cập vào CSDL qua model và xác định cách dữ liệu được đưa ra.

#### - **Controller.**

Một controller cung cấp các cơ sở để thay đổi trạng thái của model. Controller dịch các input từ chuột và bàn phím của người dùng, truyền lệnh đến model hay view để thay đổi cho phù hợp.

Một controller là các phương thức mà người dùng tương tác với ứng dụng. Một controller cho phép nhập thông tin từ người dùng và chỉ thị cho model và view thực hiện các hành động dựa trên đầu vào. Trong thực tế, controller có thể phản hồi cho việc ánh xạ các action của người cuối đến các phản hồi của ứng dụng.

Controller dịch các tương tác với view thành các hành động để thực thi bởi model. Trong một GUI client đơn lẻ, các tương tác người dùng có thể là các click vào button và nhiều sự lựa chọn trong khi một ứng dụng web xuất hiện như một request HTTP GET hoặc POST. Các action được thực hiện bởi model bao gồm việc kích hoạt các quá trình nghiệp vụ hoặc thay đổi trạng thái của model. Dựa trên các tương tác người dùng và đầu ra của các tương tác với model, controller phản hồi bằng việc xuất ra một view phù hợp.

Controller là thành phần quản lý tương tác người dùng với model. Nó cung cấp các cơ chế để các thay đổi trạng thái của model được tạo ra.

#### 2.4.1.2 *Yii framework*

Yii là một PHP MVC Framework mã nguồn mở hoàn toàn miễn phí, có hiệu năng cao, giúp bạn phát triển tốt các ứng dụng Web. Yii cho phép tái sử dụng tối đa các thành phần (components) của ứng dụng để tăng tốc độ viết ứng dụng.

Yii là từ viết tắt của từ *Yes, it is!* Yii được phát âm là \i:\, vì thế nó cũng được hiểu là viết ngắn gọn 3 chữ cái đầu của Easy (dễ dàng), Efficient (hiệu suất) và Extensible (khả năng mở rộng).

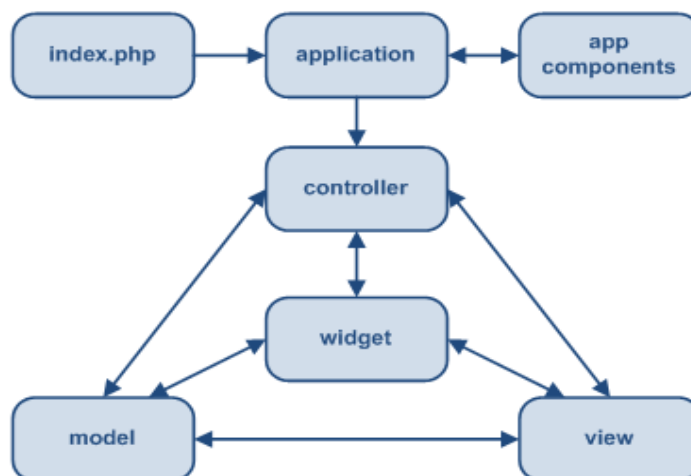
Ưu điểm của Yii là tính đơn giản trong cách viết kết hợp với thế mạnh “scripting” của PHP và cơ chế mảng Index (Associative Array).

Yii là một MVC framework nhưng có kiến trúc OOP rất tốt và định hướng theo component nên mức độ tái sử dụng giữa các project rất cao. Yii đặc biệt chạy nhanh hơn Zend và Symfony.

Tiền thân của Yii Framework chính là PRADO Framework. Tác giả của Yii Framework là Qiang Xue, 1 người Trung Quốc. Qiang Xue bắt đầu xây dựng Yii Framework vào khoảng 01/01/2008. Có nghĩa là tính tới nay đã được 6 năm. Qiang Xue cũng chính là người đã từng phát triển và maintain PRADO, tuy nhiên, nhận thấy PRADO có nhiều hạn chế và bản thân Qiang Xue muốn phát triển 1 framework mang những đặc điểm như dễ sử dụng, dễ tiếp cận, bảo mật cao và chuyên nghiệp hơn nên QX đã sử dụng kinh nghiệm của mình trong thời gian phát triển PRADO để viết ra Yii. Sau gần 1 năm phát triển, tới 08/12/2008, bản Yii Framework đầu tiên đã được public và nhanh chóng được mọi người đón nhận.

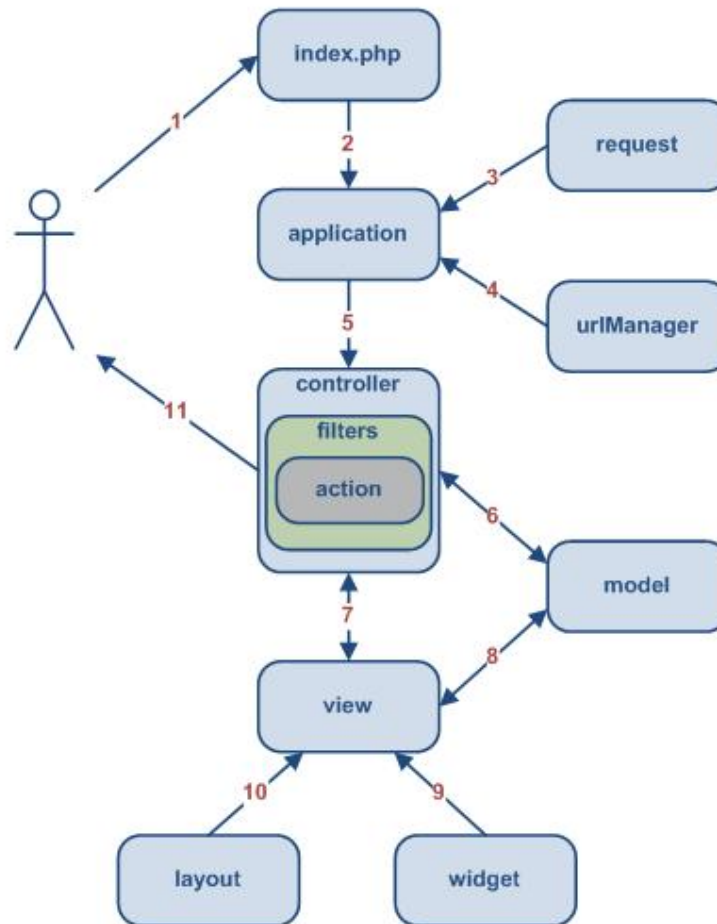
Đặc điểm của yii:

- Yii triển khai trên mô hình thiết kế MVC, được thực hiện rộng rãi trong lập trình web. MVC giúp nhà phát triển có thể dễ dàng thay đổi các phần mà không thay đổi phần khác. Trong MVC, model đưa ra thông tin và các luật, view chứa các yếu tố của giao diện người dùng, và controller quản lý tương tác giữa model và view.
- Bên cạnh việc triển khai MVC, Yii cũng giới thiệu 1 front-controller được gọi là Application, nó đóng gói ngữ cảnh thực thi việc xử lý các yêu cầu. Application thu thập một vài thông tin về một yêu cầu người dùng và sau đó gửi nó vào 1 controller phù hợp cho việc trao đổi sau này.
- Sơ đồ sau cho thấy cấu trúc tĩnh của 1 ứng dụng Yii



Hình 2-13: Cấu trúc tĩnh của ứng dụng Yii

- Quy trình làm việc cụ thể:



Hình 2-14: Quy trình làm việc của ứng dụng Yii

1. Một người sử tạo một request với URL: <http://example.com/index.php?r=post/show&id=1> và web server trao đổi yêu cầu bằng việc thực thi bootstrap script index.php
2. Bootstrap script tạo ra một thực thể Application và chạy chúng.
3. Application lấy thông tin chi tiết về yêu cầu người dùng từ một thành phần của application được đặt tên là request.
4. Application xác controller được request và action với sự giúp đỡ của một thành phần application tên là urlManager. Trường hợp này controller là post sẽ được thay thế bằng lớp PostController và action là show, phương thức thực tế của nó sẽ được xác định bởi controller.

5. Application tạo ra một thực thể của controller được request để trao đổi request của người dùng sau này. Controller xác định action show tương ứng phương thức actionShow trong lớp controller. Nó sau đó sẽ tạo ra và thực thi bộ lọc liên kết với action này. Action được thực thi nếu nó được cho phép bởi bộ lọc.
6. Action đọc một model có id=1 trong cơ sở dữ liệu.
7. Action trả lại một view có tên show với model post.
8. View đọc và hiển thị các thuộc tính của model post.
9. View thực thi một vài widget (thành phần được nhúng vào một view script để tạo ra giao diện người dùng phức tạp).
10. View trả lại kết quả được nhúng vào trong một layout (Một view đặc biệt được sử dụng để trang trí các view, thường là 1 phần của giao diện người dùng nằm giữa các view).
11. Action hoàn tất việc trả lại view và hiển thị kết quả tới người dùng.

### 2.4.2 MYSQL

MySQL là hệ quản trị cơ sở dữ liệu mã nguồn mở phổ biến nhất trên thế giới. Facebook, Twitter, Wikipedia, YouTube hiện đang sử dụng hệ quản trị cơ sở dữ liệu này. Hầu hết các cơ sở dữ liệu quan hệ một vài thập kỷ trở lại đây đều được thiết kế hướng đến việc xây dựng các ứng dụng ERP phức tạp thì MySQL được thiết kế và tối ưu hóa hỗ trợ cho việc xây dựng những ứng dụng Web. Với hiệu suất khả năng mở rộng, độ tin cậy cao cũng như dễ dàng trong sử dụng, tiết kiệm được chi phí mà MySQL đã trở thành hệ quản trị cơ sở dữ liệu mặc định với các ứng dụng Web-based đối với đa số các lập trình viên.

Một điểm mạnh nữa của MySQL là phần mềm mã nguồn mở miễn phí, rất dễ dàng triển khai trên các hệ điều hành khác nhau như Windows, Linux, Mac OS X, Unix ...

Hiện tại MySQL được triển khai với những cải thiện đáng kể như: tăng hiệu suất, khả năng mở rộng, tăng độ khả dụng với người dùng, kiến trúc đa xử lý về phần cứng, phần mềm ... MySQL là sự kết hợp giữa cơ sở dữ liệu MySQL và InnoDB storage engine (từ phiên bản MYSQL 5.5) cung cấp cho chúng ta một giải pháp quản lý dữ liệu mạnh mẽ với những ưu điểm sau:

- Tăng hiệu suất và khả năng mở rộng trên Windows.

- Tương thích, tận dụng được tối đa hiệu suất với các hệ thống kiến trúc đa xử lý trên các nền tảng khác nhau.
- Tăng khả năng đáp ứng, hiệu quả, khả năng quản lý, dễ dàng hơn cho người sử dụng...

Với InnoDB mặc định trong MySQL, những bảng được tạo mới sẽ có Engine Storage mặc định là InnoDB giúp cho phép lập trình các ứng dụng Transaction mà không cần phải thay đổi lại cấu hình. Ngoài ra InnoDB đã được tái cấu trúc giúp tối ưu hóa hiệu suất bằng cách tận dụng tối đa công suất các hệ thống phần cứng đa lõi, đa luồng. Sau đây là các kết quả có được khi so sánh giữa MySQL 5.5 và MySQL 5.1:

- Windows: tăng 1500% hiệu suất khi đọc và ghi dữ liệu, 500% khi chỉ đọc.
- Linux: tăng 360% hiệu suất khi đọc và ghi dữ liệu, 200% khi chỉ đọc.

Chính vì những ưu điểm đó mà chúng em quyết định chọn MySQL để triển cơ sở dữ liệu cho hệ thống này.

### 2.4.3 Apache HTTP server

**Apache** hay là **chương trình máy chủ HTTP** là một chương trình dành cho máy chủ đối thoại qua giao thức HTTP. Apache chạy trên các hệ điều hành tương tự như Unix, Microsoft Windows, Novell Netware và các hệ điều hành khác. Apache đóng một vai trò quan trọng trong quá trình phát triển của mạng World Wide Web.

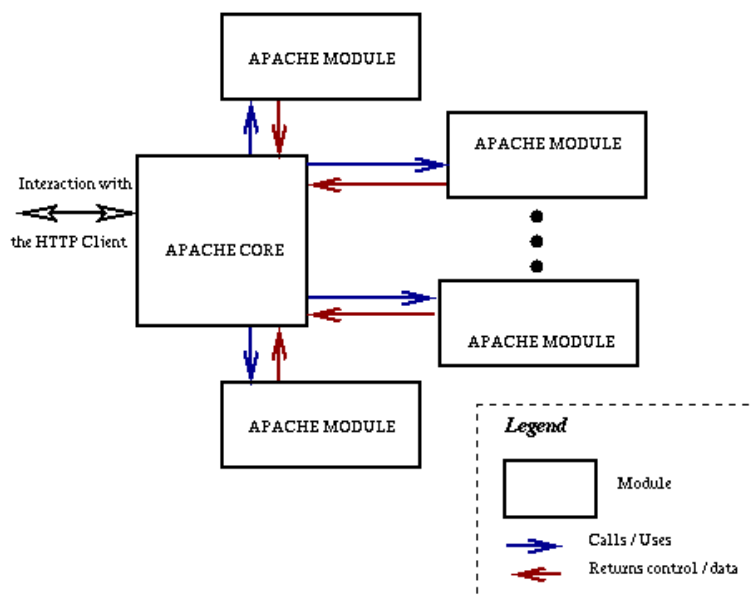
Khi được phát hành lần đầu, Apache là chương trình máy chủ mã nguồn mở duy nhất có khả năng cạnh tranh với chương trình máy chủ tương tự của Netscape Communications Corporation mà ngày nay được biết đến qua tên thương mại Sun Java System Web Server. Từ đó trở đi, Apache đã không ngừng tiến triển và trở thành một phần mềm có sức cạnh tranh mạnh so với các chương trình máy chủ khác về mặt hiệu suất và tính năng phong phú. Từ tháng 4 năm 1996, Apache trở thành một chương trình máy chủ HTTP thông dụng nhất. Hơn nữa, Apache thường được dùng để so sánh với các phần mềm khác có chức năng tương tự. Tính đến tháng 1 năm 2007 thì Apache chiếm đến 60% thị trường các chương trình phân phối trang web (xem bài tiếng Anh)

Apache được phát triển và duy trì bởi một cộng đồng mã nguồn mở dưới sự bảo trợ của Apache Software Foundation. Apache được phát hành với giấy phép Apache License và là một phần mềm tự do và miễn phí.

#### 2.4.3.1 Kiến trúc Apache

Apache là một Web Server được xây dựng theo cấu trúc module, tức là Apache là một tập hợp nhiều thành phần nhỏ được gắn kết với nhau. Các thành phần nhỏ này được phân ra thành 2 loại chính là Apache Core (thành phần lõi) và Apache Modules

Tổng quát, cấu trúc của Apache được trình bày trong sơ đồ dưới đây:



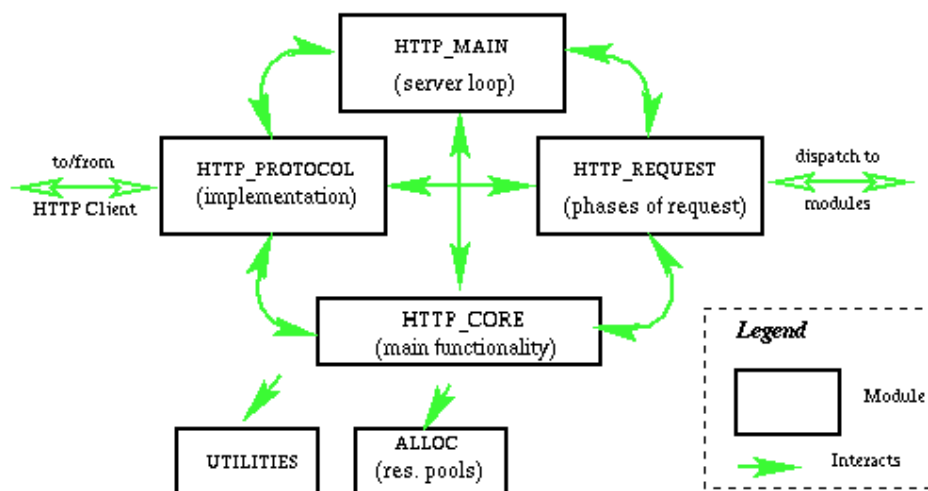
Hình 2-15: Kiến trúc Apache

#### 2.4.3.2 Thành phần lõi của apache

Tất cả các chức năng cơ bản nhất của Apache Web Server được cài đặt trong thành phần này. Chính thành phần này sẽ cung cấp các tài nguyên phù hợp cho từng request mà Web Server nhận được. Lõi của Apache được cấu thành từ các thành phần nhỏ sau:

- **http\_protocol.c:** chứa các thủ tục làm nhiệm vụ giao tiếp trực tiếp với các client thông qua socket bằng giao thức HTTP. Tất cả các công việc trao đổi dữ liệu với client đều do thành phần này đảm trách.
- **http\_main.c:** thành phần làm nhiệm vụ khởi động server và tạo vòng lặp chính để đợi và chấp nhận các kết nối. Đồng thời cũng làm nhiệm vụ quản lý các bộ thời gian timeout.
- **http\_request.c:** thành phần này làm nhiệm vụ quản lý các tiến trình xử lý bản tin request, đảm bảo chuyển các bản tin điều khiển tới các module phù hợp theo đúng thứ tự. Ngoài ra, nó còn đảm nhận vai trò quản lý các lỗi xảy ra trên server.
- **http\_core.c:** thành phần này sẽ triển khai các chức năng cơ bản nhất của Apache.
- **Alloc.c:** thành phần kiểm soát việc phân chia tài nguyên và lưu trữ các thông tin về sự phân chia đó.
- **http\_config.c:** đảm nhận vai trò xử lý file cấu hình và hỗ trợ cho các virtual host. Một trong những chức năng quan trọng của thành phần này là liệt kê những module được sử dụng trong Apache.

Kiến trúc chi tiết của lõi Apache được thể hiện trong sơ đồ sau:



**Hình 2-16: Kiến trúc chi tiết phần lõi của Apache**

#### 2.4.3.3 Các bước xử lý một gói tin request của Apache

Khi một request được gửi đến Web Server, thì nhiệm vụ xử lý gói Request đó sẽ được nhiều modules khác nhau đảm nhận. Mỗi một phản hồi của Web Server là kết quả của một quá trình xử lý trong Apache. Quá trình xử lý này được phân chia thành nhiều công đoạn nhỏ hơn, mỗi công đoạn do một module đảm nhận. Mỗi module chỉ liên lạc trực tiếp với core để nhận lệnh và xử lý, sau đó gửi trả kết quả lại cho core, và quá trình đó lại lặp lại đối với các modules khác. Lưu ý là, các module không biết tới các module khác.

Quá trình xử lý một gói tin Request của Apache được thông qua các bước sau:

1. Phân giải: Chuyển đổi từ một URI thành một đường dẫn chứa file tương ứng trên server
2. Kiểm tra truy nhập: dựa vào thông tin hostname và các thông tin khác.
3. Kiểm tra User ID từ gói tin HTTP Request và kiểm tra tính hợp lệ của user.
4. Cho user quyền để truy cập đến các tài nguyên cụ thể.
5. Xác định MIME của đối tượng bị truy vấn. Tức là thông tin về kiểu định dạng của tài nguyên trên server được gọi tên trong gói HTTP Request (danh sách các MIME content type: <http://www.iana.org/assignments/media-types/index.html>)
6. Chỉnh sửa lại một số thông tin, ví dụ thay đổi alias thành một đường dẫn thực (định nghĩa alias sẽ được trình bày trong phần về mod\_alias dưới đây)



7. Phản hồi dữ liệu lại cho client.

8. Ghi lại log.

Tiến trình xử lý này sẽ được điều khiển bởi thành phần `http_request.c` trong core của Apache.

## 2.5 JSON và thư viện mã nguồn mở GSON

### 2.5.1 Định nghĩa JSON<sup>21</sup>

Json là một ngôn ngữ định dạng dữ liệu hoàn toàn độc lập, dùng để lưu trữ và trao đổi các thông tin dạng văn bản. Json giống với Xml nhưng cấu trúc đơn giản hơn, giúp cho việc phân tích, xử lý dữ liệu một cách nhanh hơn và dễ dàng hơn. Json được dựa trên cơ sở ngôn ngữ lập trình JavaScript, chuẩn ECMA-262 3rd tháng 12, 1999. Một thể hiện của json như sau:

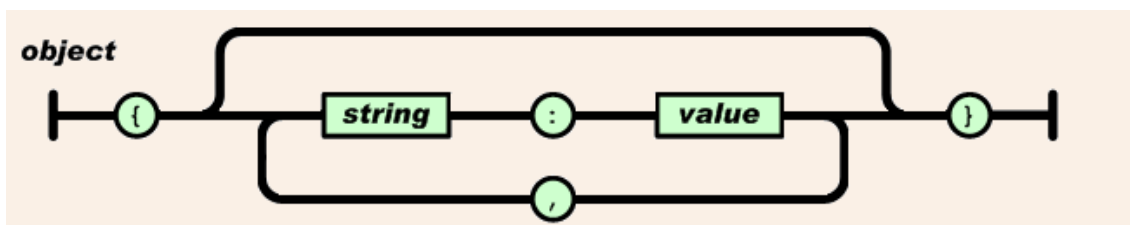
```
{ "employees": [ { "firstName": "John", "lastName": "Doe" }, { "firstName": "Anna", "lastName": "Smith" }, { "firstName": "Peter", "lastName": "Jones" } ] }
```

Json được xây dựng trên hai cấu trúc

- Là một tập hợp các cặp tên – giá trị tương tự như các đối tượng trong các ngôn ngữ lập trình khác.
- Một dãy các giá trị được sắp xếp tương tự như các array, vector, list hay sequence.

Hai hình thức biểu diễn của Json được sử dụng trong hệ thống này là:

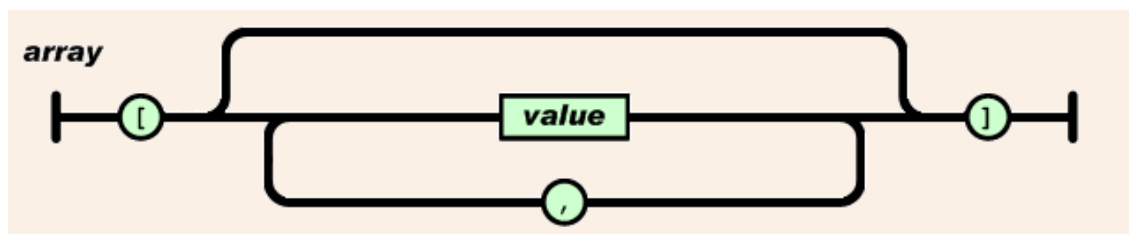
- Object (Đối tượng): Là một dãy các cặp Tên/ Giá trị. Một Object được bắt đầu bằng dấu ngoặc đơn trái ( { ) và kết thúc bằng dấu ngoặc đơn phải ( } ). Tên và giá trị ngăn cách bởi dấu hai chấm ( : ). Mỗi cặp Tên/ Giá trị được ngăn cách bởi dấu phẩy ( , ).



Hình 2-16: Biểu diễn json của một đối tượng

- Array(mảng): Array là một dãy các cặp giá trị, Array được bắt đầu bằng dấu ngoặc vuông trái ( [ ) và kết thúc bằng dấu ngoặc vuông phải ( ] ). Các giá trị được ngăn cách bởi dấu phẩy ( , ).

<sup>21</sup> <http://www.json.org/index.html>



Hình 2-17: Biểu diễn json của một mảng

Trong hệ thống ORSER-SYSTEM này định dạng dữ liệu sẽ sử dụng cho việc truyền thông giữa server API và android application chính là Json.

### 2.5.2 Thư viện mã nguồn mở GSON<sup>22</sup>

Gson là một thư viện Java được sử dụng trong việc chuyển đổi một đối tượng Java bất kỳ sang định dạng Json. Gson cũng hỗ trợ chuyển đổi từ một chuỗi định dạng Json sang đối tượng Java tương ứng. Đây là một thư viện mã nguồn mở được lưu trữ tại <http://code.google.com/p/google-gson>.

Những ưu điểm của Gson:

- Cung cấp các phương thức giúp việc chuyển đổi qua lại giữa đối tượng Java và Json trở nên đơn giản.
- Hỗ trợ chuyển đổi qua lại các đối tượng bất kỳ.
- Cho phép tùy chỉnh các đối tượng.
- Hỗ trợ các đối tượng phức tạp.
- Tạo ra đối tượng Json gọn nhẹ, dễ phân tích.

Lớp chính trong thư viện này là Gson, được khởi tạo đơn giản bằng cách gọi `new Gson()`. Ngoài ra còn có một lớp `GsonBuilder` sử dụng để tạo ra một đối tượng Gson với tùy chỉnh thích hợp. Các đối tượng Gson sẽ không duy trì trạng thái khi chuyển đổi qua lại giữa đối tượng Java và Json vì thế ta có thể sử dụng lại đối tượng Gson này trong các lần chuyển đổi tiếp theo. Dưới đây là ví dụ đơn giản cho thấy cách thức sử dụng đối tượng Gson:

- Chuyển đổi từ đối tượng Java sang Json
 

```
Gson gson = new Gson();

gson.toJson(1);

gson.toJson("abcd");

gson.toJson(new Long(10));

int[] values = { 1 };

gson.toJson(values);
```

<sup>22</sup> <https://sites.google.com/site/gson/gson-user-guide>

- Chuyển đổi từ Json sang đối tượng Java
 

```
int one = gson.fromJson("1", int.class);

Integer one = gson.fromJson("1", Integer.class);

Long one = gson.fromJson("1", Long.class);

Boolean false = gson.fromJson("false", Boolean.class);

String str = gson.fromJson("\"abc\"", String.class);

String anotherStr = gson.fromJson("[\"abc\"]", String.class);
```

Trong hệ thống này Gson sẽ được sử dụng để chuyển đổi định dạng dữ liệu truy vấn từ cơ sở dữ liệu sang định dạng là Json trong android application.

Phiên bản Google Gson Version 2.3.1 được phép tải miễn phí tại:

<http://search.maven.org/#artifactdetails|com.google.code.gson|gson|2.3.1|jar>

## 2.6 Kết luận

Qua chương này chúng ta đã nắm vững được những công nghệ để xây dựng hệ thống một cách chi tiết và rõ ràng. Bằng việc nắm vững những kiến thức, công nghệ nền tảng như đã trình bày ở trên, em tiến hành chuyển sang việc phân tích thiết kế hệ thống ORDER-SYSTEM, phần này sẽ được trình bày trong chương tiếp theo.

### **CHƯƠNG 3.**

## **PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG ĐẶT HÀNG SỬ DỤNG ĐIỆN THOẠI DI ĐỘNG**

Nội dung chương này sẽ làm rõ những yêu cầu cho hệ thống ORDER-SYSTEM trong thực tế để từ đó có thể tiến hành phân tích những yêu cầu cần phải đáp ứng khi xây dựng hệ thống để từ đó đưa ra phương pháp giải quyết phù hợp trong việc xây dựng hệ thống này.

### **3.1 Xác định yêu cầu và các đối tượng trong hệ thống**

Sau đây chúng ta sẽ tiến hành xác định các đối tượng và yêu cầu cần có với một hệ thống đặt hàng.

#### **3.1.1 Yêu cầu đối với hệ thống**

Đối với một hệ thống đặt hàng luôn có những yêu cầu bắt buộc hệ thống phải đáp ứng được nếu muốn được đưa vào sử dụng trong thực tế.

Hệ thống của em có khả năng đáp ứng được những yêu cầu sau đây:

- Cho phép người dùng đầu cuối có thể xem và đặt mua sản phẩm thông qua web hoặc android application.
- Cho phép người quản lý cửa hàng có thể phân phối đơn hàng cho người giao hàng thông qua bản đồ Google Map.
- Cho phép người giao hàng nhìn thấy các đơn hàng cần giao thông qua android application.
- Cho phép người giao có thể tìm đường, báo trễ hoặc xác nhận hoàn thành việc giao hàng bằng android application.

Đối với việc xây dựng client chuyên dụng để phục vụ người dùng đầu cuối và người giao hàng trong hệ thống sẽ có đáp ứng những yêu cầu sau:

- Là Smartphone chạy hệ điều hành Android có cài đặt ứng dụng android application của hệ thống với các chức năng như sau:
- Giao tiếp với database quản lý điều hành thông qua Web API server.
- Cập nhật trạng thái vị trí của mình.

#### **3.1.2 Các đối tượng trong hệ thống**

Sau khi nắm bắt được các yêu cầu của hệ thống, em sẽ phân tích những đối tượng hoạt động chủ yếu trong hệ thống này là khách hàng, người giao hàng và người quản lý hệ thống.

### 3.1.2.1 *Người quản lý hệ thống*

Người quản lý hệ thống là người trực tiếp làm việc với hệ thống này thông qua trình duyệt web để thực hiện việc quản lý và điều phối các nhân viên giao hàng trong hệ thống. Người quản lý có một tài khoản để đăng nhập vào hệ thống thực hiện việc điều phối nhân viên giao hàng thích hợp để giao các đơn hàng. Quá trình diễn ra như sau:

- Người quản lý sẽ kiểm tra các đơn hàng của khách cần giao.
- Thực hiện việc điều phối bằng cách: thông qua bản đồ trạng thái của hệ thống sẽ hiển thị vị trí của các đơn hàng, người quản lý tiến hành tìm các đơn hàng gần nhau và giao cho nhân viên giao hàng thích hợp.
- Ngoài ra, nhân viên quản lý còn có thể thực hiện các thao tác thiết lập cho hệ thống như thêm đơn sản phẩm, tạo ra một sự kiện khuyến mãi và thống kê lại các đơn hàng đã giao.

### 3.1.2.2 *Nhân viên giao hàng*

Là người liên hệ trực tiếp phục vụ khách hàng trong hệ thống. Các nhân viên giao hàng sẽ phải đăng nhập vào hệ thống thông qua ứng dụng client được cài đặt trên Smartphone chạy hệ điều hành Android. Khi nhân viên giao hàng đăng nhập vào hệ thống, client này sẽ có nhiệm vụ tự động cập nhật thông tin về đơn hàng và vị trí các đơn hàng và hiển thị chúng trên bản đồ.

Trong bản đồ này sẽ hiển thị đường đi đến vị trí đơn hàng cần giao mà nhân viên giao hàng đã chọn. Đồng thời cũng gửi tin nhắn đến khách hàng khi sắp đến vị trí cần giao.

### 3.1.2.3 *Khách hàng*

Là đối tượng phục vụ chính. Khách hàng thông qua web site hoặc android application để thực hiện đặt hàng các sản phẩm của hệ thống thống:

Khi đặt hàng khách hàng phải cung cấp các thông tin sau đây:

- Tên khách hàng.
- Địa chỉ email.
- Số điện thoại.
- Vị trí của đơn hàng và địa chỉ cần giao.

Sau khi đơn hàng đã được tạo khách hàng sẽ chờ nhân viên giao hàng liên lạc thông báo việc nhận hàng.

## 3.2 Phân tích hệ thống

### 3.2.1 Sơ đồ use case của thống

Sau khi đã nắm bắt được những đối tượng và yêu cầu cần có đối với hệ thống này, em đưa ra sơ đồ use case của hệ thống như sau:



**Hình 3-1: Mô hình usecase của hệ thống**

Trong đó các use case được mô tả như sau:

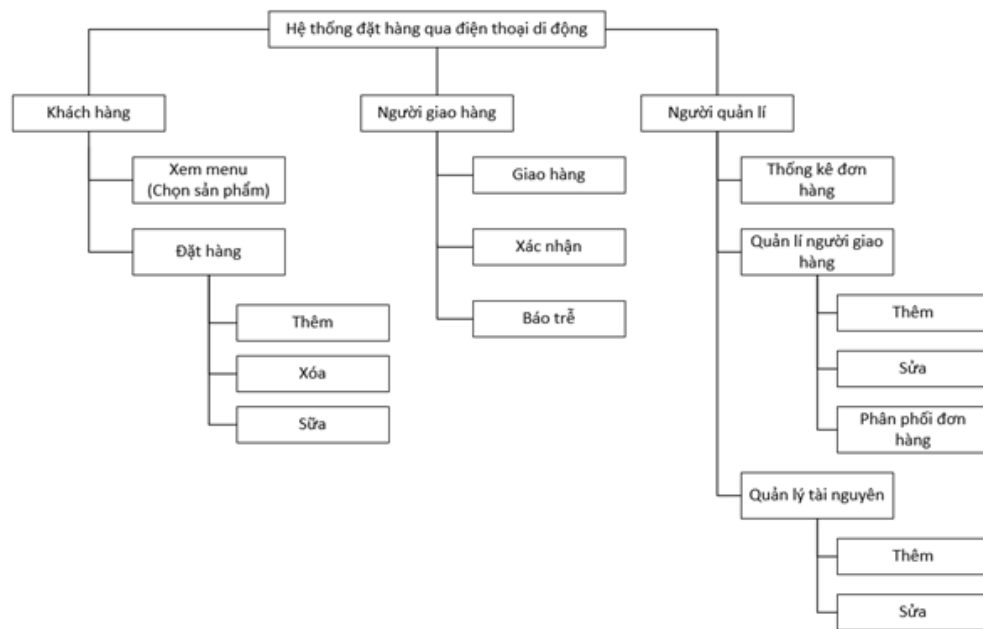
Stt	Tên Thuộc tính	Diễn giải
1	Thống kê đơn hàng	Nhân viên quản lý sử dụng CMS để thống kê đơn hàng
2	Quản lý người giao hàng	Nhân viên quản lý sử dụng CMS để quản lý các nhân viên giao hàng
3	Quản lý tài nguyên	Nhân viên quản lý sử dụng CMS để quản lý các tài nguyên của cửa hàng (sản phẩm,)
4	Đặt hàng	Khách hàng sử dụng android application hoặc web application để đặt hàng sản phẩm
5	Xem menu	Khách hàng chọn các sản phẩm từ menu

6	Thao tác đơn hàng	Khách hàng tạo(thêm, xóa, sửa) đơn hàng đã tạo
7	Nhận hàng	Khách hàng nhận hàng từ nhân viên giao hàng
8	Giao hàng	Nhân viên giao hàng giao hàng cho khách
9	Nhận thanh toán	Nhân viên giao hàng nhận thanh toán trực tiếp từ khách
10	Thông báo trễ	Nhân viên giao hàng thông báo giao hàng trễ với khách nếu xảy ra vấn đề.

**Bảng 3-1: Các usecase của hệ thống**

### 3.2.2 Sơ đồ phân rã chức năng:

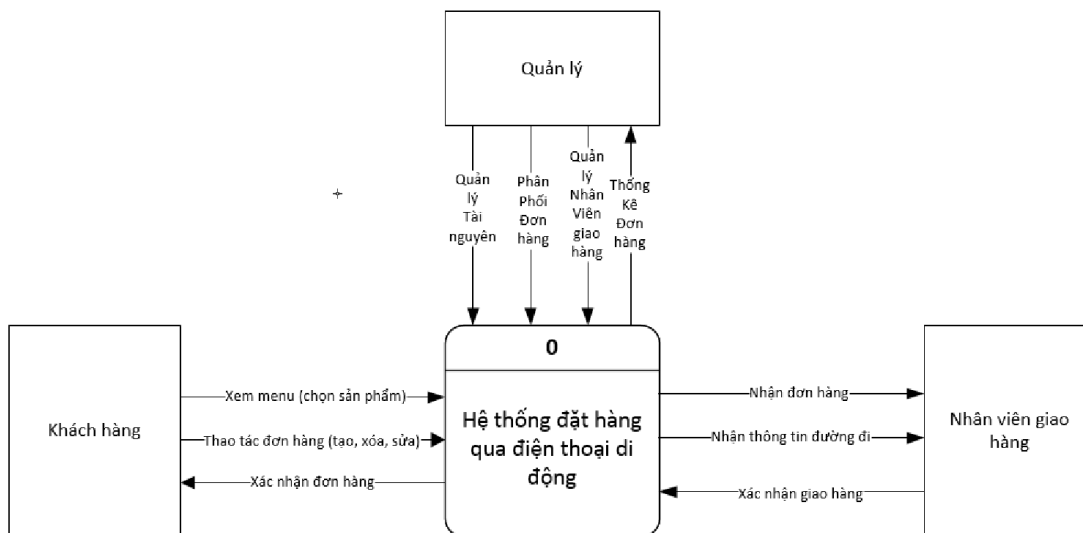
Sau khi đã có lược đồ use case em tiếp tục xem xét và đưa ra sơ đồ phân rã chức năng như sau:



**Hình 3-2: Sơ đồ phân rã chức năng của hệ thống**

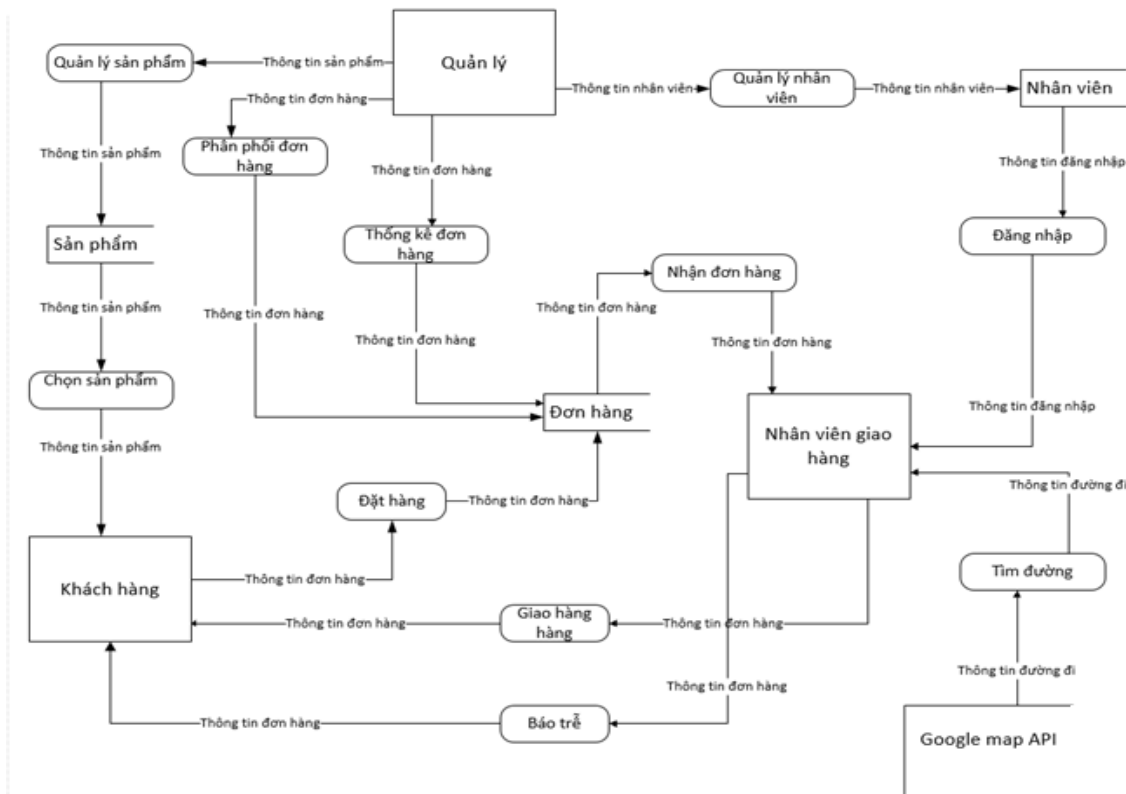
### 3.2.3 Sơ đồ DFD

#### 3.2.3.1 Mô hình mức ngữ cảnh



**Hình 3-3: Mô hình mức ngữ cảnh của hệ thống**

#### 3.2.3.2 Mô hình DFD tổng quát



**Hình 3-4: Mô hình DFD tổng quát**



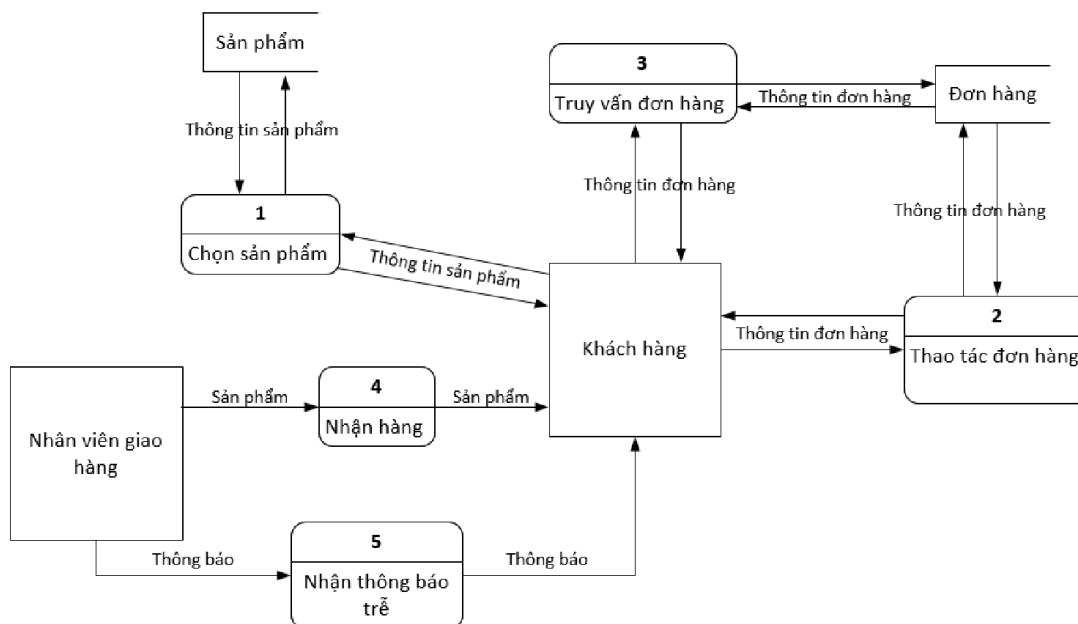
Trong đó các cơ sở dữ liệu được miêu tả như sau:

Stt	Tên Thuộc tính	Diễn giải
1	Sản phẩm	Bao gồm các dữ liệu về sản phẩm ( category, sản phẩm).
2	Đơn hàng	Bao gồm các dữ liệu về đơn hàng(đơn hàng, nội dung đơn hàng, các sự kiện khuyến mãi).
3	Nhân viên	Bao gồm các dữ liệu về nhân viên.
4	Google Api map	Kho dữ liệu của google về bản đồ, tọa độ và đường đi.

**Bảng 3-2: Các kho dữ liệu trong mô hình DFD tổng quát**

Sau khi đã đưa ra mô hình DFD tổng quát, em tiếp tục xây dựng sơ đồ DFD cho từng đối tượng như sau:

### 3.2.3.3 Sơ đồ DFD đối với khách hàng:



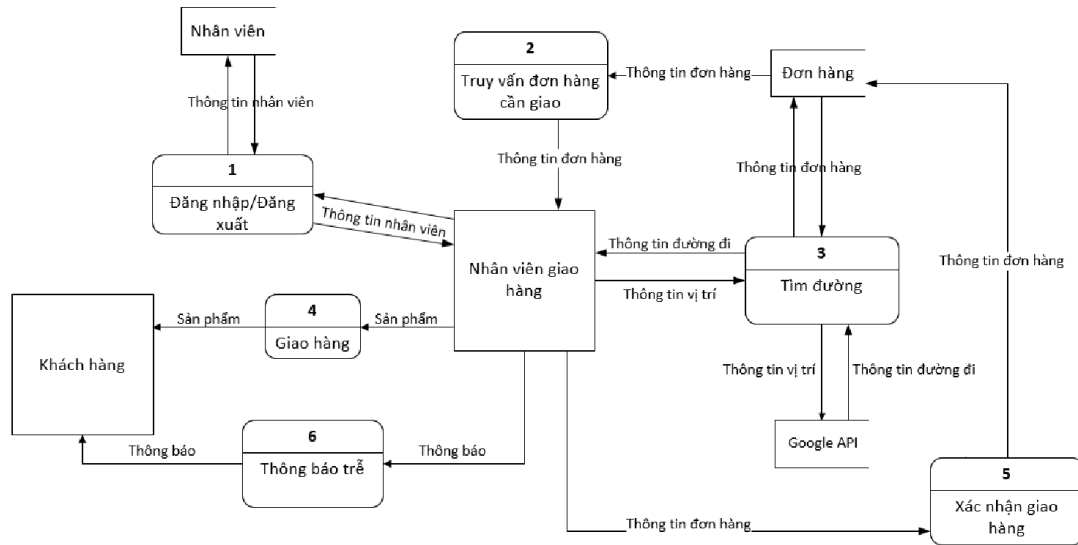
**Hình 3-5: Sơ đồ DFD đối với khách hàng**

Đặc tả mô hình ngữ cảnh đối với khách hàng.

- Ô xử lý 1: chọn sản phẩm.
  - o Khách hàng truy cập vào website hoặc android application.
  - o Hệ thống sẽ gửi các sản phẩm hiện đang được bán cho khách hàng.
- Ô xử lý 2: Thao tác đơn hàng.
  - o Khách hàng chọn các sản phẩm thêm vào giỏ hàng và gửi lên server để tạo một đơn ha (thêm, xóa, sửa đơn hàng).
  - o Server sẽ trả về cho khách hàng mã đơn hàng hoặc thông báo lỗi nếu thất bại.

- Ô xử lý 3: Truy vấn đơn hàng.
  - o Sau khi có mã đơn hàng khách hàng có thể truy vấn với server để biết được thông tin về đơn hàng của mình.
- Ô xử lý 4: Nhận hàng: Hàng sẽ được giao bởi nhân viên của cửa hàng, khách hàng sẽ nhận hàng trực tiếp từ nhân viên và thanh toán hóa đơn.
- Ô xử lý 5: Nhận thông báo trễ. Nếu có vấn đề về đơn hàng, nhân viên giao hàng sẽ thông báo với khách hàng bằng sms để khách hàng được biết.

#### 3.2.3.4 Sơ đồ DFD đối với nhân viên giao hàng

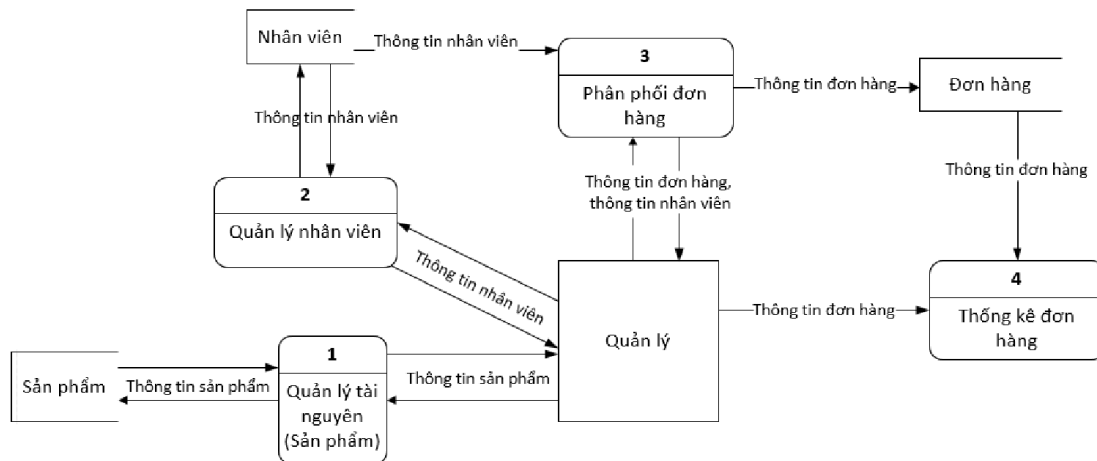


**Hình 3-6: Sơ đồ DFD đối với nhân viên giao hàng**

Đặc tả mô hình ngữ cảnh đối với nhân viên khách.

- Ô xử lý 1: đăng nhập hệ thống.
  - o Nhân viên giao hàng cần phải đăng nhập hệ thống để có thể sử dụng ứng dụng.
- Ô xử lý 2: Truy vấn đơn hàng.
  - o Sau khi đăng nhập thành công nhân viên giao hàng sẽ có thể truy vấn các đơn hàng chưa giao được phân công.
- Ô xử lý 3: Truy vấn đường đi.
  - o Với mỗi đơn hàng nhân viên giao hàng sẽ được phép truy vấn đường đi từ google maps API.
- Ô xử lý 4: Giao hàng: Hàng sẽ được giao bởi nhân viên của cửa hàng tới khách hàng.
- Ô xử lý 5: Xác nhận giao hàng. Sau khi đã giao hàng cho khách nhân viên giao hàng cần thông báo với server là đã giao hàng thành công.
- Ô xử lý 6: Thông báo trễ. Nếu có vấn đề về đơn hàng, nhân viên giao hàng sẽ thông báo với khách hàng bằng sms để khách hàng được biết.

### 3.2.3.5 Sơ đồ DFD đối với nhân viên quản lý

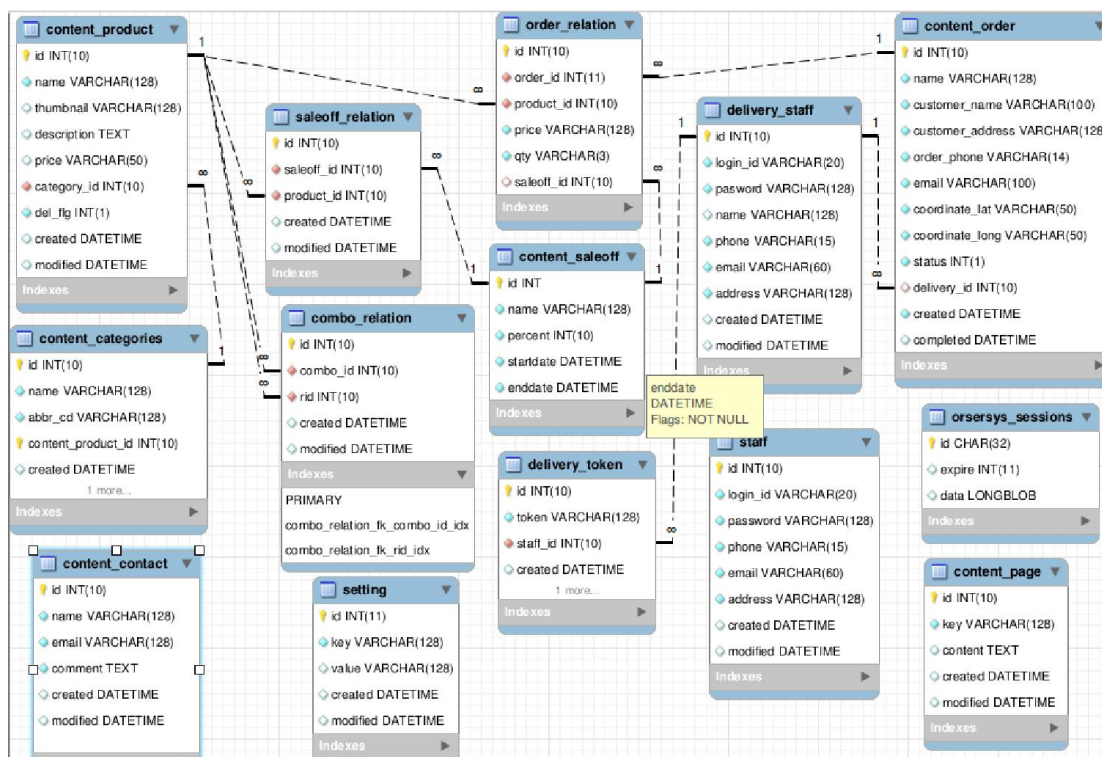


**Hình 3-7: Sơ đồ DFD đối với quản lý**

Đặc tả mô hình ngữ cảnh đối với quản lý.

- Ô xử lý 1: Quản lý tài nguyên.
  - o Quản lý các tài nguyên của hệ thống (sản phẩm).
- Ô xử lý 2: Quản lý nhân viên.
  - o Nhân viên giao hàng được quản lý qua các thao tác thêm, chỉnh sửa.
- Ô xử lý 3: Phân phối đơn hàng.
  - o Đơn hàng sẽ được quản lý phân phối cho nhân viên giao hàng thông qua google maps API.
- Ô xử lý 4: Thống kê đơn hàng: Các đơn hàng đã giao xong có thể được quản lý thống kê.





### Hình 3-9: Mô hình cơ sở dữ liệu MYSQL

#### 3.2.4.3 Bảng STAFF

Ý nghĩa: lưu giữ thông tin người quản lý.

Tên tính	Thuộc	Diễn giải	Kiểu dữ liệu	Giá trị	Ghi chú
<u>id</u>		Id của tài khoản người quản lý	int(10)	notNull	
login_id		Tài khoản đăng nhập hệ thống quản lý	varchar(20)	notNull	
password		Mật khẩu của tài khoản quản lý	varchar(128)	notNull	

### Bảng 3-3: Bảng Staff

#### 3.2.4.4 Bảng *DELIVERY\_STAFF*

Ý nghĩa: lưu trữ thông tin các nhân viên giao hàng.

Tên Thuộc tính	Diễn giải	Kiểu dữ liệu	Giá trị	Ghi chú
<b><u>Id</u></b>	Id tài khoản của nhân viên giao hàng	int(10)	notNull	
login_id	Tài khoản của nhân viên giao hàng	varchar(20)	notNull	
password	Mật khách hàng của nhân viên giao hàng	varchar(128)	notNull	

**Bảng 3-4: Bảng Delivery\_staff**

#### 3.2.4.5 Bảng *CONTENT\_CATEGORIES*

Ý nghĩa: lưu trữ thông tin về các category của cửa hàng.

Tên Thuộc tính	Diễn giải	Kiểu dữ liệu	Giá trị	Ghi chú
<b><u>Id</u></b>	Id của category	Int(10)	notNull	
Name	Tên của category	varchar(128)	notNull	
Content_product_id	Cờ cho biết category này đang được mở cho khách hàng.	Int(10)	notNull	Giá trị là 1 sẽ hiển thị trên website và android application

**Bảng 3-5: Bảng content\_categories**

#### 3.2.4.6 Bảng *CONTENT\_PRODUCT*

Ý nghĩa: lưu trữ danh sách sản phẩm của cửa hàng.

Tên Thuộc tính	Diễn giải	Kiểu dữ liệu	Giá trị	Ghi chú
<b><u>Id</u></b>	Id của sản phẩm	int(10)	notNull	
Name	Tên sản phẩm	varchar(123)	notNull	
Thumbnail	Hình ảnh của sản phẩm	varchar(128)	notNull	Là link tới hình ảnh của sản

				phẩm
Description	Mô tả về sản phẩm	Text	notNull	
Price	Giá của sản phẩm	varchar(50)	notNull	
Category_id	Id của category chứa sản phẩm	Int(100)	notNull	
Del_flg	Cờ hiển thị sản phẩm	Int(1)	notNull	Nếu là 1 thì sẽ hiển thị (sản phẩm không thể xóa để đảm bảo tính toàn vẹn dữ liệu)
Is_combo	Cờ để biết sản phẩm là một combo gồm nhiều sản phẩm	Int(1)	notNull	Nếu là 1 thì sản phẩm là 1 combo các sản phẩm

**Bảng 3-6: Bảng content\_product**

### 3.2.4.7 Bảng COMBO\_RELATION

Ý nghĩa: Lưu trữ thông tin về combo sản phẩm.

Tên Thuộc tính	Diễn giải	Kiểu dữ liệu	Giá trị	Ghi chú
<b><u>Id</u></b>	Id của bảng	Int(10)	notNull	
Combo_id	Là id của sản phẩm combo	Int(10)	notNull	
Rid	Là id sản phẩm nằm trong combo có id là combo_id	time	notNull	

**Bảng 3-7: Bảng combo\_relation**

### 3.2.4.8 Bảng CONTENT\_SALEOFF

Ý nghĩa: Lưu trữ thông tin khuyến mãi của các sản phẩm.

Tên Thuộc tính	Diễn giải	Kiểu dữ liệu	Giá trị	Ghi chú
<b><u>Id</u></b>	Id của chương trình khuyến mãi	Int(10)	notNull	
Name	Tên của chương trình khuyến mãi	Varchar(128)	notNull	
Percent	Phần trăm giảm giá	Int(10)	notNull	
Startdate	Ngày bắt đầu của chương trình	Datetime	notNull	
Enddate	Ngày kết thúc của chương trình	Datetime	notNull	

**Bảng 3-8: bảng content\_saleoff**

### 3.2.4.9 Bảng SALEOFF\_RELATION

Ý nghĩa: lưu trữ thông tin các sản phẩm nằm trong chương trình khuyến mãi.

Tên Thuộc tính	Diễn giải	Kiểu dữ liệu	Giá trị	Ghi chú
<b><u>Id</u></b>	Id xác định trong bảng	Int(10)	notNull	
Saleoff_id	Id của chương trình khuyến mãi	Int(10)	notNull	
Product_id	Id của sản phẩm được khuyến mãi	Int(10)	notNull	

**Bảng 3-9: Bảng saleoff\_relation**



3.2.4.10 *Bảng CONTENT\_ORDER*

Ý nghĩa: lưu trữ thông tin các đơn hàng.

Tên Thuộc tính	Diễn giải	Kiểu dữ liệu	Giá trị	Ghi chú
<b><u>Id</u></b>	Id của đơn hàng	Int(10)	notNull	
Name	Tên đơn hàng	Varchar(128)	notNull	Sẽ có dạng ORDER_d dmmYYY HHmms Đây cũng chính là mã số đơn hàng.
Customer_name	Tên khách hàng	Varchar(128)	notNull	
Customer_address	Địa chỉ của khách hàng	Varchar(128)	notNull	
Order_phone	Số điện thoại liên lạc	Varchar(14)	notNull	
Email	Email của khách hàng	Varchar(100)	notNull	
Coordinate_lat	Kinh độ của điểm giao hàng	Varchar(50)	notNull	
Coordinae_long	Vĩ độ của điểm giao hàng	Varchar(50)	notNull	
Status	Status hiện tại của đơn hàng	Int(1)	notNull	0: đơn hàng vừa mới được đặt. 1: đơn hàng đang được giao 2: đơn hàng đã được giao thành công
Delivery_id	Id của nhân viên giao hàng chịu trách nhiệm giao đơn hàng	Int(10)	notNull	
Created	Ngày tạo đơn hàng	Datetime	notNull	
Completed	Ngày đơn hàng	Datetime	notNull	

	được giao			
--	-----------	--	--	--

**Bảng 3-10: Bảng content\_order**3.2.4.11 *Bảng ORDER\_RELATION*

Tên Thuộc tính	Diễn giải	Kiểu dữ liệu	Giá trị	Ghi chú
<b><u>Id</u></b>	Id của thông tin trong bảng	Int(10)	notNull	
Order_id	Id của đơn hàng	Int(10)	notNull	
Product_id	Id của sản phẩm trong đơn hàng	Int(10)	notNull	
Price	Giá của sản phẩm tại thời điểm mua hàng	Varchar(128)	notNull	
Qty	Số lượng sản phẩm được mua	Varchar(3)	notNull	
Saleoff_id	Id của chương trình khuyến mãi	Int(10)	Default: Null	Nếu sản phẩm nằm trong chương trình khuyến mãi tại thời điểm mua thì đây là id của chương trình

**Bảng 3-11: Bảng order\_relation**

#### 3.2.4.12 Bảng *DELIVERY\_TOKEN*

Ý nghĩa: lưu trữ thông tin token đăng nhập của các nhân viên giao hàng khi đăng nhập vào hệ thống.

Tên Thuộc tính	Diễn giải	Kiểu dữ liệu	Giá trị	Ghi chú
<b><u>Id</u></b>	Id của token	Int(10)	notNull	
Token	Token xác thực	Varchar(128)	notNull	
Staff_id	Id của nhân viên giao hàng	Int(10)	notNull	

**Bảng 3-12: Bảng delivery\_token**

#### 3.2.4.13 Các bảng khác:

Các bảng còn lại của cơ sở dữ liệu được dùng để lưu các thông tin thiết lập cho việc vận hành hệ thống.

- Bảng CONTENT\_CONTACT: lưu các phản hồi của khách hàng đối với hệ thống.
- Bảng SETTING: lưu các cài đặt hệ thống như: tọa độ khách hàng, địa chỉ, số điện thoại liên lạc, email, ...
- Bảng CONTENT\_PAGE: lưu các page tĩnh của website.
- Bảng ORDERSYS\_SESSIONS: đây là bảng được tạo ra bởi yii framework dùng để lưu sessions của người dùng.

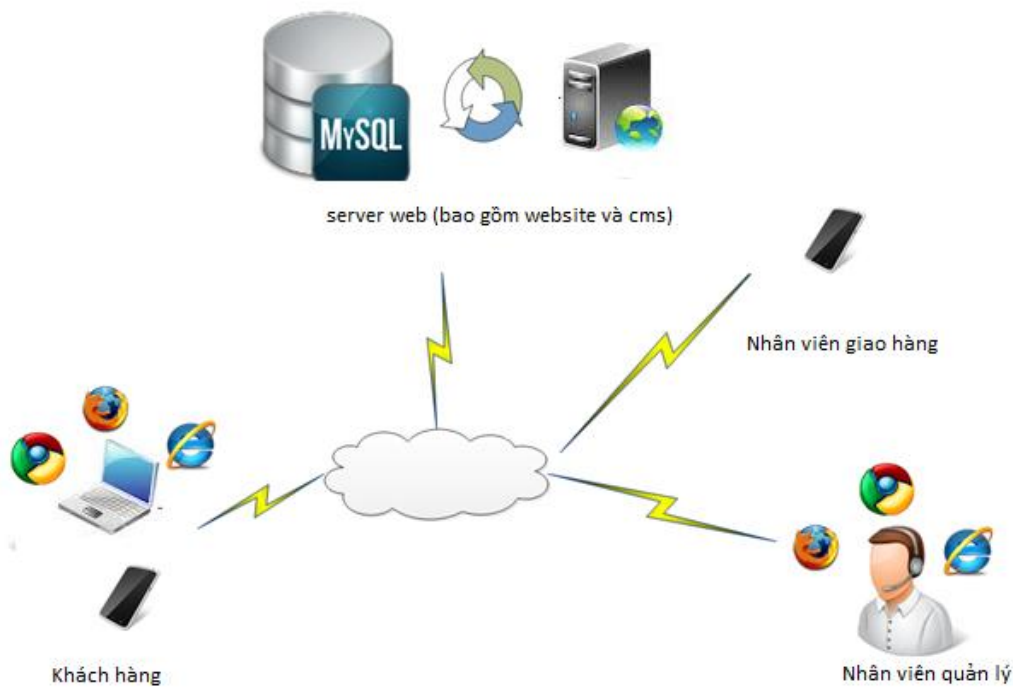
### 3.3 Thiết kế hệ thống

Hệ thống đặt hàng sử dụng điện thoại di động là hệ thống hoạt động theo mô hình Client – Server bao gồm năm thành phần chính sau đây:

- CMS Web Application là một ứng dụng web cung cấp các chức năng:
  - o Web Application cung cấp các chức năng quản lý và điều hành hệ thống sử dụng thông qua trình duyệt web nên rất tiện dụng và nhanh chóng.
  - o Cung cấp cho nhân viên quản lý hệ thống có cái nhìn tổng thể về hệ thống với việc hiển thị vị trí, trạng thái của các đơn hàng của khách hàng lên bản đồ số giúp tăng hiệu quả trong việc phân phối đơn hàng.
  - o Lưu trữ lịch sử đơn hàng.
  - o Thêm mới các sản phẩm để khách đặt hàng.
- Web Application trực tuyến :
  - o Cho phép khách hàng đặt hàng các sản phẩm.
  - o Xác định vị trí tương đối của khách hàng thông qua ip internet.
- API Web Application trực tuyến :
  - o Cho phép các android application giao tiếp với cơ sở dữ liệu.
  - o Là một web application chạy trên nền PHP.

- Android application đặt hàng là ứng dụng được cài đặt trên Smartphone chạy hệ điều hành android với các chức năng:
  - Hiện thị các sản phẩm đang bán của cửa hàng.
  - Cho phép khách hàng đặt hàng thông qua application.
  - Cho phép khách hàng xem các đơn hàng đã đặt của mình.
- Android application giao hàng là ứng dụng được cài đặt trên Smartphone chạy hệ điều hành android dùng cho nhân viên giao hàng với các chức năng:
  - Hiện thị các các đơn hàng cần giao trên google map.
  - Cho phép nhân viên giao hàng xác nhận giao hàng và báo trễ với khách hàng.
  - Cho phép gửi sms tự động khi đến địa điểm giao hàng.

Thiết kế hệ thống được mô tả như hình sau:



**Hình 3-10: Mô tả thiết kế hệ thống**

### 3.4 Giải pháp hiển thị bản đồ

Giải pháp về bản đồ là một trong những thành phần quan trọng trong hệ thống này. Google Maps API có khả năng hỗ trợ lập trình viên khai thác sức mạnh của Google Maps để sử dụng trong các ứng dụng riêng của mình. Cụ thể đây là công cụ giúp hiển thị dữ liệu cụ thể của từng ứng kết hợp với bản đồ rất hiệu quả. Và trong hệ thống này, chúng em sử dụng Google Maps API để hiển thị dữ liệu về vị trí và trạng thái của các taxi lên trên bản đồ giúp cho người điều hành thuận tiện hơn trong việc quản lý, giám sát và điều hành các xe taxi trong hệ thống.

Google Maps được giới thiệu trên Blog post của google trong tháng 2 năm 2005. Ưu điểm tại thời điểm này của Google Maps là cho phép người dùng kéo chuột để di chuyển bản đồ. Chi phí cho các giải pháp về bản đồ lúc này rất tốn kém và đòi hỏi phải có một máy chủ bản đồ đặc biệt và hầu như chỉ được tạo ra nhằm đáp ứng các yêu cầu riêng của Google do đó nó chứa rất nhiều tính năng không cần thiết làm cho API trên nên cồng kềnh hơn. Từ năm 2005, sự bùng nổ của các thiết bị di động khiến API cũ trở nên chậm chạp và không còn phù hợp cho những thiết bị này nữa. Chính vì vậy các nhà phát triển giải pháp bản đồ đã quyết định xây dựng mới lại API này. Trong quá trình phát triển Google đã cung cấp các Api giúp các lập trình viên dễ dàng tiếp cận và tích hợp bản đồ vào trong ứng dụng của mình. Giúp cho Google Maps trở thành giải pháp bản đồ được sử dụng phổ biến như ngày nay.

Trong hệ thống chúng em sử dụng hai Api cụ thể là Google Maps JavaScript API v3 và Google Maps Android API v2.

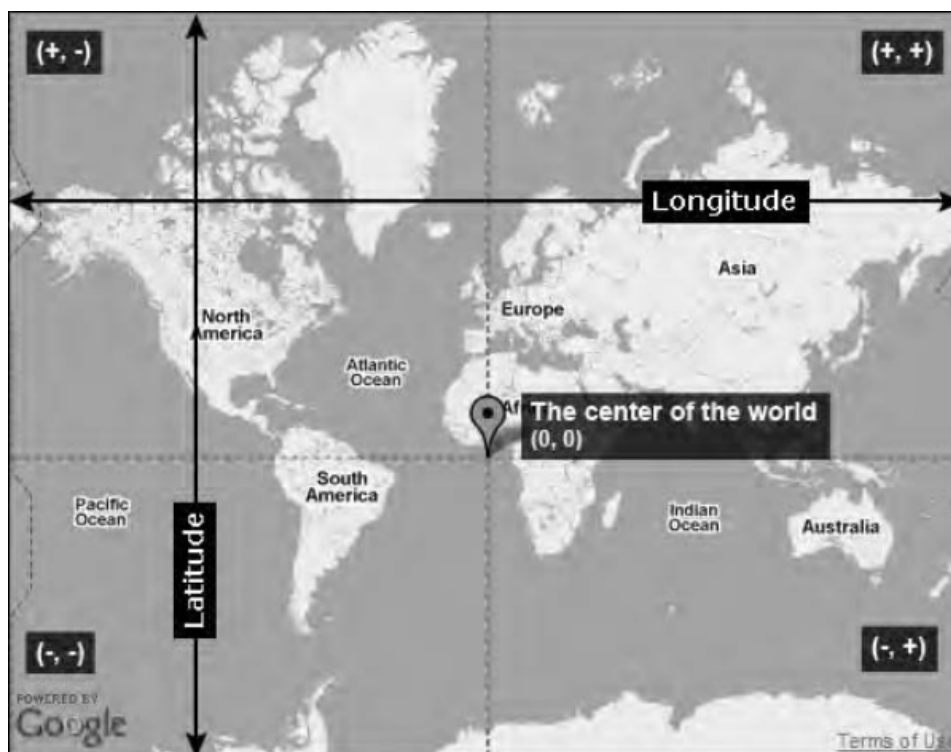
Phiên bản 3 của Google Maps JavaScript API là phiên bản được làm mới hoàn toàn so với các phiên bản trước của Google. Api này cho phép nhúng trực tiếp Google Maps vào trang web riêng của chúng ta, nó được thiết kế đặc biệt để hoạt động nhanh và nhiều tính năng hơn cho các thiết bị di động và trình duyệt web. Api này cung cấp các tiện ích giúp cho việc thao tác bản đồ (Giống như trang <http://maps.google.com>), việc xử lý nội dung trên bản đồ được thực hiện thông qua các Web Service do Google cung cấp giúp chúng ta có thể tạo ra các ứng dụng bản đồ một cách dễ dàng và mạnh mẽ hơn.

Google Maps JavaScript thực chất là sự kết hợp của HTML, CSS và JavaScript. Bản đồ được chia thành từng ô nhỏ là những hình ảnh của bản đồ. Những ảnh này được tải về bằng Ajax sau đó chèn vào một thẻ Div trong trang HTML. Khi người dùng thao tác với bản đồ, API sẽ gửi tọa độ mới và mức độ zoom của bản đồ để gọi Ajax để lấy về những hình ảnh phù hợp với thao tác trên bản đồ.

API này bao gồm các tập tin JavaScript có chứa các lớp với các thuộc tính và phương thức để chúng ta có thể sử dụng khi thao tác với bản đồ. API này được thiết kế chú trọng đến việc tăng hiệu suất cả khi sử dụng trên thiết bị di động và trên các

máy tính. Để thực hiện điều này, với phiên bản cũ trình duyệt phải tải và thực thi rất nhiều Script trước khi có thể hiển thị được bản đồ thì phiên bản mới được module hóa sẽ tải về những code cần thiết trước, hiển thị bản đồ sau đó tải tiếp những thứ còn lại sau khi đã hiển thị bản đồ.

Hệ thống tọa độ được sử dụng trong Google Maps là Word Geodetic System 84 (WGS 84), cùng chung hệ thống với Global Positioning System. Vì vậy sẽ rất tiện lợi khi kết hợp bản đồ Google maps với GPS trên Smartphone vì chúng ta không cần tính toán chuyển đổi tọa độ. Để xác định một điểm trong hệ tọa độ này ta sẽ sử dụng vĩ độ (Latitude) và kinh độ (Longitude), hai giá trị này thể hiện bằng số thập phân. Các giá trị vĩ độ luôn đứng trước kinh độ (vĩ độ, kinh độ). Ví dụ như (40.714,-47.005)



Hình 3-11: Word Geodetic System 84[7]

### 3.5 Tổng kết

Qua chương này chúng ta đã thấy được chi tiết hơn những yêu cầu và đối tượng trong hệ thống. Bước đầu xác định các thành phần, các xử lý chính của hệ thống. Chúng ta sẽ đi vào phần thiết kế chi tiết của hệ thống ở chương tiếp theo.

## CHƯƠNG 4.

### HIỆN THỰC HỆ THỐNG ĐẶT HÀNG BẰNG ĐIỆN THOẠI DI ĐỘNG

Sau khi đã nắm vững cơ sở lý thuyết và tiến hành phân tích yêu cầu, đưa ra phương pháp giải quyết bài toán mà đề tài đặt ra, chương này sẽ trình bày chi tiết việc thiết kế, xây dựng và mô hình triển khai hệ thống.

Do hệ thống này gồm có năm phần: web quản lý, web đặt hàng, API web application, android application đặt hàng và android application dùng cho việc giao hàng. Các phần này gần như tương đồng với nhau nên trong mỗi phần chỉ trình bày những phần chính và không trình bày source code.

#### 4.1 Xây dựng API Web Application

Việc hiển thị dữ liệu của hệ thống đòi hỏi việc phải truy vấn dữ liệu được lưu trữ trong cơ sở dữ liệu MySQL. Để thực hiện việc truy vấn dữ liệu hệ thống xây dựng API web application để thực hiện việc truy vấn dữ liệu từ MySQL theo yêu cầu của từng chức năng trong hệ thống. Dữ liệu truy dưới dạng json sẽ được xử lý và kết quả sẽ được chuyển thành định dạng Json với sự hỗ trợ của PHP và được phản hồi lại theo từng yêu cầu.

##### 4.1.1 Truy vấn dùng cho android application đặt hàng:

Phương thức	URI	Chức năng
POST	{domain_API}/category/index	Lấy thông tin của tất cả các categories
POST	{domain_API}/product/index	Lấy thông tin của tất cả các sản phẩm thuộc category hoặc của cả cửa hàng
POST	{domain_API}/product/gethot	Lấy 5 sản phẩm có số lượng đặt hàng nhiều nhất
POST	{domain_API}/cart/getcart	Lấy các sản phẩm hiện tại trong giỏ
POST	{domain_API}/cart/add	Thêm một sản phẩm vào giỏ
POST	{domain_API}/cart/edit	Chỉnh sửa hoặc xóa sản phẩm

		trong giỏ hiện tại
<b>POST</b>	{domain_API}/cart/destroy	Xóa giỏ hàng
<b>POST</b>	{domain_API}/cart/checkout	Thực hiện đặt một đơn hàng.
<b>POST</b>	{domain_API}/order/getorder	Tìm tất cả các đơn hàng được đặt bởi khách hàng xác định.
<b>POST</b>	{domain_API}/order/getorderdetail	Lấy thông tin về một đơn hàng xác định
<b>POST</b>	{domain_API}/order/cancelorder	Hủy một đơn hàng chưa được giao

**Bảng 4-1: API hỗ trợ truy vấn dữ liệu cho android application đặt hàng**

Trong quá trình truy vấn các truy vấn điều phải kèm theo một secret key được quy định trong code để xác định truy vấn là hợp lệ. Nếu truy vấn xảy ra lỗi thì sẽ trả về lỗi được quy định trong file đính kèm.

#### 4.1.2 Truy vấn dùng cho android application giao hàng hàng:

Phương thức	URI	Chức năng
<b>POST</b>	{domain_API}/delivery/login	Nhân viên giao hàng đăng nhập vào hệ thống
<b>POST</b>	{domain_API}/delivery/checktoken	Kiểm tra tính hợp lệ của token
<b>POST</b>	{domain_API}/delivery/getdeliveryorders	Lấy các đơn hàng được phân công
<b>POST</b>	{domain_API}/delivery/deliveryorderdetail	Lấy thông tin chi tiết của đơn hàng
<b>POST</b>	{domain_API}/delivery/completedeliveryorder	Thông báo hoàn thành giao một đơn hàng

**Bảng 4-1: API hỗ trợ truy vấn dữ liệu cho android application giao hàng**

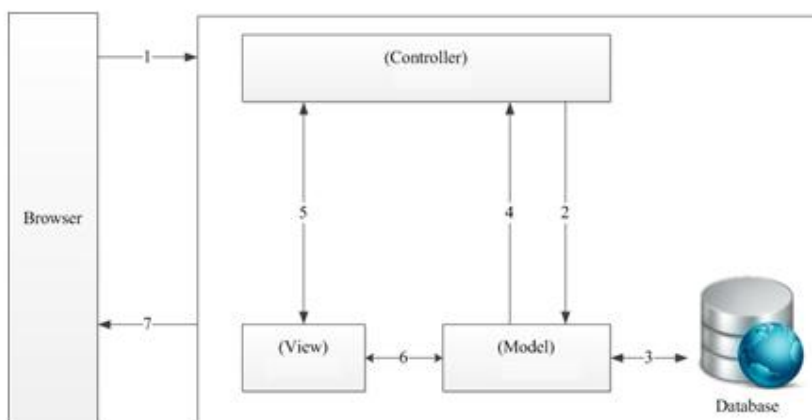


## 4.2 Xây dựng ứng dụng web quản lý hệ thống

Đây là nơi nhân viên quản lý điều hành hệ thống và phân phối giao đơn hàng.

### 4.2.1 Cấu trúc ứng dụng dùng chung cho web quản lý và web đặt hàng

Ứng dụng hỗ trợ là một PHP Web Application xây dựng bằng yiiframework với cơ sở dữ liệu là MySQL được xây dựng theo mô hình MVC(Model – View – Controller).



**Hình 4-1: Cách thức hoạt động của ứng dụng**

Controller: được hỗ trợ bởi yii framework dưới dạng các action truy vấn giúp phân chia nhỏ các truy vấn và chuyên môn hóa.

Model: mỗi model là ánh xạ của 1 bảng trong cơ sở dữ liệu được viết dưới dạng đối tượng giúp truy vấn nhanh chóng và chính xác thông tin.

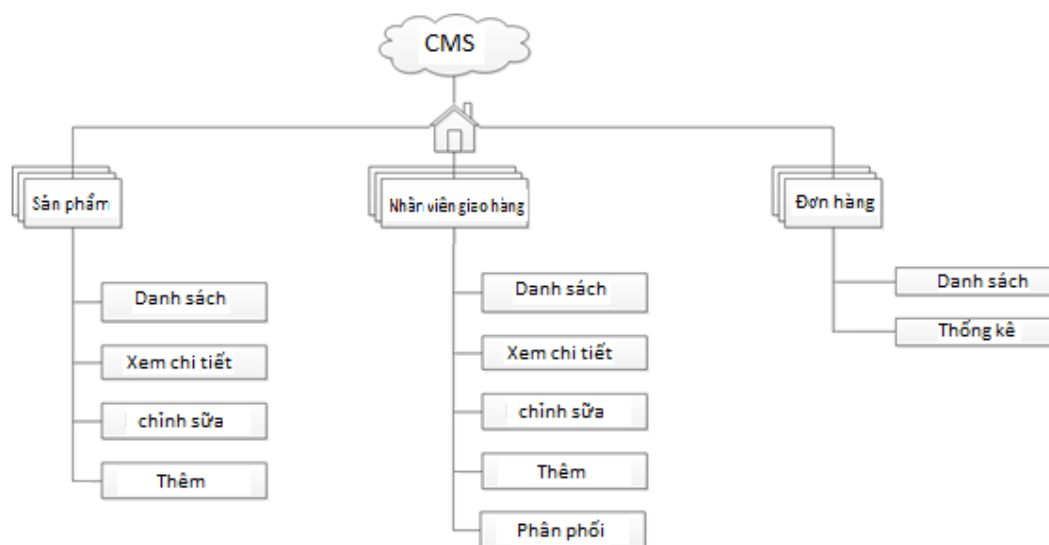
View: là tập tin html dùng hiển thị các thông tin sau khi xử lý và tương tác với người dùng.

Các bước hoạt động của ứng dụng:

- Nhân viên quản lý thực hiện truy vấn đến CMS web của hệ thống.
- Các (Controller): có nhiệm vụ nhận các yêu cầu, phân tích các giá trị tham số trong yêu cầu để tiến hành xử lý, lựa chọn các Model để thực hiện.
- Model thực hiện các yêu cầu kết nối, truy vấn dữ liệu từ cơ sở dữ liệu MySQL và trả về kết quả cho Controller
- Controller nhận kết quả trả về từ Model thực hiện lựa chọn View để hiển thị kết quả nhận được.

### 4.2.2 Xây dựng các chức năng cho các đối tượng sử dụng hệ thống

Ứng dụng web quản lý có nhiều chức năng nhưng chủ yếu là các chức năng phục vụ ba đối tượng của hệ thống được mô tả như hình sau:



**Hình 4-2: Ứng dụng CMS web quản lý hệ thống**

Mỗi đối tượng sẽ có tác vụ cụ thể như sau:

- Sản phẩm: Là thành phần quan trọng nhất của hệ thống, được người quản lý thao tác trực tiếp thông qua CMS có thể tạo mới xem chi tiết, chỉnh sửa thông tin đây chính là đối tượng phục vụ của hệ thống.
- Nhân viên giao hàng: Là nhân viên của cửa hàng được cấp tài khoản dùng cho việc phân phối và giao các đơn hàng được nhân viên quản lý quản lý trực tiếp thông qua CMS.
- Đơn hàng: đối tượng này được nhân viên quản lý phân phối cho từng nhân viên giao hàng.

#### 4.2.3 Xây dựng chức năng điều phối

Đây là thành phần chiếm vai trò rất quan trọng trong hệ thống. Nhân viên quản lý sẽ sử dụng các control trên form để phân phối đơn hàng cho từng nhân viên được chọn.

Hình dưới đây là giao diện sử dụng của thành phần điều phối đơn hàng:

Login id

delivery\_01

Tên nhân viên

HieuNG

Chọn đơn hàng

Đơn hàng được chọn

ORDER\_08092214052015

Xóa

Save

Cancel

Hình 4-3: Thành phần điều phối đơn hàng.

Từ đó ta thực hiện việc điều phối như sau:

- Đơn hàng: Các biểu tượng đỏ trên bản đồ là các đơn hàng cần giao.
- Với mỗi nhân viên được chọn: ta click tương ứng vào các đơn hàng.
- Danh sách các đơn hàng được chọn: danh sách các đơn hàng dự định phân phối cho nhân viên.

Sau khi đã lựa chọn được đơn các đơn hàng phù hợp, nhân viên quản lý sẽ nhấn chọn nút “Save” để thực hiện việc điều phối đơn hàng.

Sau khi hoàn thành thao tác trong phần nhân viên sẽ xuất hiện danh sách đơn hàng được phân phối:

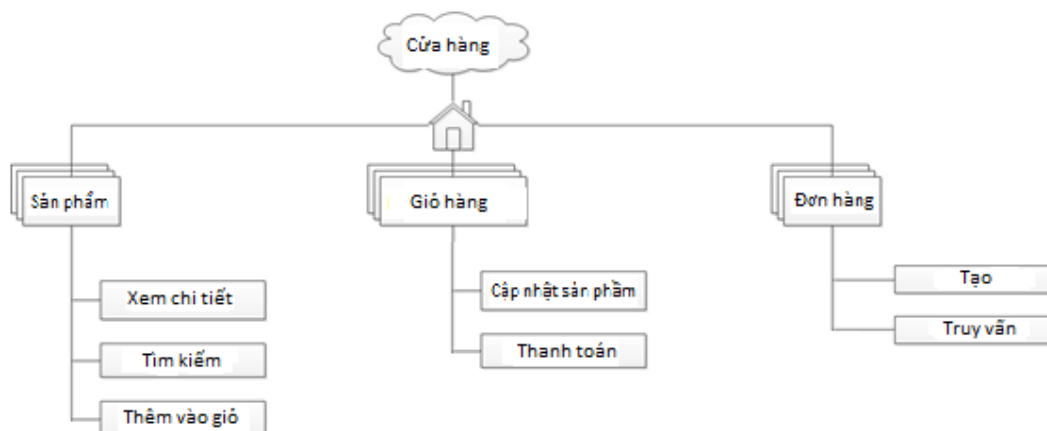
Tên đơn hàng	Tên khách hàng	Số điện thoại	Địa chỉ
ORDER_09033117042015	HieuNG	01997815475	123
ORDER_08092214052015	10520490	0773500757	364 Cộng Hòa, Phường 13, Quận Tân Bình, TP.Hồ Chí Minh

▲ PAGE TOP

Hình 4-4: Hiện thị danh sách đơn hàng.

### 4.3 Xây dựng các chức năng cho ứng dụng web đặt hàng và android application đặt hàng

Ứng dụng web đặt hàng và android application đặt hàng có nhiều chức năng nhưng chủ yếu là các chức năng phục vụ ba đối tượng của hệ thống được mô tả như hình sau:



Hình 4-5: Ứng dụng web đặt hàng và android application đặt hàng

Mỗi đối tượng sẽ có tác vụ cụ thể như sau:

- Sản phẩm: Là thành phần chủ thể được khách hàng đặt hàng, có thể được tìm kiếm xem chi tiết và thêm vào giỏ hàng.
- Giỏ hàng: đối tượng tương tác với khách hàng, có thể được cập nhật lại, từ các sản phẩm được chọn và thực hiện thanh toán.
- Đơn hàng: đối tượng này được khách hàng tạo ra để gửi cho hệ thống, và có thể được truy vấn bởi khách hàng.

### 4.4 Xây dựng ứng dụng android application giao hàng

#### 4.4.1 Các xử lý về thông tin GPS

Đa số các điện thoại sử dụng hệ điều hành android đều sử dụng chip xử lý có khả năng nhận tín hiệu GPS từ vệ tinh. Vì vậy android cung cấp một bộ thư viện dùng để giao tiếp và xử lý các thông tin lấy từ GPS nằm trong package android.location. Trong package này gồm có các thành phần như LocationManager, Location... Nhưng quan trọng nhất là lớp location, lớp này đại diện cho tập hợp các thông tin thu từ vệ tinh bao gồm: kinh độ, vĩ độ, các tùy chọn về cao độ, tốc độ, độ chính xác của vị trí....

Cách thức việc lấy thông tin GPS được thực hiện thông qua các nhà cung cấp. Có hai nhà cung cấp chính ở đây là GPS\_PROVIDER và NETWORK\_PROVIDER được đặt trung bởi lớp provider. Lớp này được khởi tạo thông qua hàm `getBestProvider()`, Android sẽ tự động chọn nhà cung cấp có độ chính xác cao nhất tại thời điểm hiện tại cho chúng ta. Một số hàm cơ bản để lấy thông tin về Location cơ bản là:

```
Location mLocation = null;

mLocation.getAccuracy();    // Trả lại thông tin là độ chính xác của GPS.
mLocation.getAltitude();    // Trả lại thông tin là cao độ của vị trí hiện tại.
mLocation.getLatitude();    // Trả lại thông tin là Latitude;
mLocation.getLongitude();   // Trả lại thông tin là Longitude;
mLocation.getSpeed();       // Trả lại thông tin là tốc độ hiện tại;
```

Để có thể nhận biết sự thay đổi về vị trí khi người dùng di chuyển, chúng ta cần phải thực hiện implement một interface là `LocationListener` để có được những hàm lắng nghe sự thay đổi về Location. Interface này gồm có 4 hàm chính với chức năng khác nhau nhưng quan trọng nhất là hàm `onLocationChanged(Location arg0)` dùng để xử lý khi có sự thay đổi về tọa độ.

```
@Override
public void onLocationChanged(Location location) {
    // TODO Auto-generated method stub
}
@Override
public void onProviderDisabled(String provider) {
    // TODO Auto-generated method stub
}
@Override
public void onProviderEnabled(String provider) {
    // TODO Auto-generated method stub
}
@Override
public void onStatusChanged(String provider, int status, Bundle extras) {
    // TODO Auto-generated method stub
}
```

Những quyền cần thiết để có thể truy xuất thông tin về GPS tối thiểu cần có bao gồm:

- `Android.permission.ACCESS_FINE_LOCATION`: quyền này giúp chúng ta có thể sử dụng `LocationManager.GPS_PROVIDER` để xác định vị trí thông qua GPS.
- `Android.permission.ACCESS_FINE_LOCATION` hay `android.permission.ACCESS_COARSE_LOCATION`: những quyền này

giúp chúng ta có thể sử dụng `LocationManager.NETWORK_PROVIDER` để xác định vị trí hiện tại.

#### 4.4.2 Các xử lý và triển khai Google Direction API

Chức năng tìm đường là một trong các dịch vụ mà google cung cấp cho người sử dụng. Đối với việc phát triển ứng dụng thì google cung cấp Direct API để thực hiện điều này. Hệ thống tìm đường này cung cấp tìm đường cho ba loại phương tiện chính bao gồm: xe hơi, xe buýt và đi bộ. Tuy nhiên tùy thuộc vào quốc gia mà dịch vụ này có được cung cấp đầy đủ và chính xác không. Ở Việt Nam thì chủ yếu sử dụng dịch vụ tìm đường cho xe hơi nhưng độ chính xác vẫn chưa được cao. Direct API đã được giới thiệu cụ thể ở chương trước, vì vậy ở phần này chúng ta chủ yếu tập trung vào các bước thực hiện. Như đã biết, để có thể sử dụng được dịch vụ này chúng ta cần gửi các yêu cầu lên server của Google. Cấu trúc URL như sau:

<http://maps.google.com/maps/api/directions/json?origin=lat1,lon1&destination=lat2,lon2&language=vi&sensor=true&mode=driving>

Các tham số cần chú ý là:

- Origin= latitude và longitude của điểm bắt đầu.
- Destination= latitude và longitude của điểm kết thúc.
- Language= ngôn ngữ sử dụng cho các chỉ dẫn, google chỉ hỗ trợ một số ngôn ngữ trong đó có tiếng việt.
- Sensor = giá trị này xác định có sử dụng hay không các cảm biến trong việc xác định vị trí và tìm đường.
- Mode= giá trị này chỉ ra chế độ tìm đường đang được sử dụng.

Đối với phiên bản hiện tại mặc định giá trị trả về từ các dịch vụ của Google là tập tin theo định dạng Json. Phân tích thông tin từ tập tin này chúng ta sẽ có được các chỉ dẫn về tìm đường và danh sách các điểm để có thể vẽ đường trên bản đồ. Các thành phần chính từ tập tin trả về bao gồm nhiều thẻ khác nhau, mỗi thẻ chứa một thông tin riêng.

- Thẻ “routes”: là thẻ chính lớn nhất của tập tin Json. Nó chứa toàn bộ thông tin mà chúng ta cần có.
- Thẻ “bounds”: là thẻ chứa tọa độ của hai điểm mút thuộc hướng đông bắc và tây nam.
- Thẻ “copyrights”: là thẻ chứa thông tin về bản quyền thông tin này.
- Thẻ “legs”: là thẻ chứa thông tin về việc tìm đường. Thẻ “legs” bao gồm nhiều thẻ con bên trong:

- Thẻ “distance”: chứa thông tin về khoảng cách giữa hai điểm.
- Thẻ “duration”: chứa thông tin về thời gian duy chuyển giữa hai điểm.
- Thẻ “end\_address”, “end\_location”, “start\_address”, “start\_location”: chứa thông tin về điểm bắt đầu và điểm kết thúc.
- Thẻ “steps”: chứa thông tin về những đoạn đường nhỏ khác nhau, mỗi đoạn này chứa các thông tin như: khoảng cách, thời gian, điểm bắt đầu, điểm kết thúc, chỉ dẫn, và danh sách các tọa độ dùng để vẽ đoạn đường này trên bản đồ.
- Thẻ “overview\_polyline”: chứa thông tin về tập hợp tất cả tọa độ của quãng đường, nó được chuyển thành mã ascii. Vì vậy muốn sử dụng chúng ta phải giải mã để thu được danh sách các điểm này.

Xử lý các thông tin này chúng ta có thể vẽ đoạn đường này trên bản đồ và hiển thị những thông tin cần thiết.

#### 4.4.3 Các xử lý cơ bản trên Google Map API v2

##### 4.4.3.1 Đánh dấu vị trí trên bản đồ bằng Marker

Marker đại diện cho một vị trí trên bản đồ. Chúng ta có thể thay đổi các tính chất cơ bản của Marker như màu sắc, vị trí, hình đại diện... Marker được hướng dựa vào màn hình của thiết bị hơn là bề mặt bản đồ vì vậy khi ta thực hiện các hành động như xoay, phóng to thu nhỏ, nghiêng thì sẽ không làm thay đổi sự định hướng của Marker. Chúng ta có thể tương tác với Marker thông qua một số hành động được cung cấp bởi Google Map API bao gồm click và kéo thả.

Để thêm một marker vào trong bản đồ chúng ta cần phải sử dụng hàm `addMarker()` của đối tượng `GoogleMap` với tham số là `MarkerOption`.

```
private GoogleMap mMap;
mMap = ((MapFragment)
getFragmentManager().findFragmentById(R.id.map)).getMap();
mMap.addMarker(new MarkerOptions()
    .position(new LatLng(0, 0))
    .title("Marker Name"));
```

Marker Option là class dùng để chứa toàn bộ thông tin về các thuộc tính của Marker, các thuộc tính này bao gồm:

- Position: là thuộc tính cần phải thiết lập đối với bất kỳ marker nào, dùng để xác định vị trí gắn marker.

- Title: chứa thông tin sẽ hiển thị trong tiêu đề của InfoWindow.
- Snippet: chứa thông tin sẽ hiển thị trong nội dung của InfoWindow.
- Draggable: chứa thuộc tính có hay không cho người dùng kéo thả Marker.
- Visible: chứa thuộc tính có hay không hiển thị marker.
- Anchor: chứa điểm trên ảnh được dùng để gắn vào bản đồ, mặc định là giữa ảnh.
- Icon: hình ảnh sẽ được hiển thị tại vị trí đặt marker, chúng ta không thể thay đổi thuộc tính này đối với các marker đã được gắn vào bản đồ.

Map API cho phép chúng ta bắt các sự kiện click vào marker thông qua phương thức `OnMarkerClickListener`, để thiết lập phương thức này trên bản đồ ta gọi hàm `GoogleMap.setOnMarkerClickListener(OnMarkerClickListener)`. Khi người dùng click vào một marker thì mặc định InfoWindow sẽ hiện ra và di chuyển camera bản đồ vào vị trí của marker vừa được click.

#### 4.4.3.2 *Vẽ đường thẳng trên bản đồ bằng Polyline*

Polyline là một đối tượng mới trong Google Map API v2 được sử dụng để vẽ hình trên bản đồ bằng các đường thẳng nối với nhau. Chúng ta có thể thay đổi các thuộc tính của Polyline như màu sắc, chiều rộng của nét vẽ, thuộc tính ẩn. Đối tượng Polyline bao gồm danh sách các điểm trên bản đồ, sau đó các điểm này sẽ được nối với nhau bằng những đường thẳng.

Để tạo ra một đối tượng Polyline trước tiên chúng ta cần khởi tạo một đối tượng `PolylineOptions` dùng để chứa các thuộc tính của Polyline và thêm danh sách các điểm vào nó, những điểm này được xác định trên bản đồ vì vậy chúng ta khởi tạo chúng như những đối tượng `LatLng`. Đường thẳng nối các điểm thực hiện theo thứ tự của cách điểm trong danh sách lưu trong đối tượng `PolylineOptions`. Để vào `PolylineOption` chúng ta sử dụng phương thức `PolylineOption.add()`.

*// Khởi tạo đối tượng PolylineOption và thêm danh sách các điểm vào đối tượng vừa khởi tạo.*

```
PolylineOptions rectOptions = new PolylineOptions()
    .add(new LatLng(37.35, -122.0))
    .add(new LatLng(37.45, -122.0))
    .add(new LatLng(37.45, -122.2))
    .add(new LatLng(37.35, -122.2))
    .add(new LatLng(37.35, -122.0));
```

*// thiết lập màu của đường thẳng nối các điểm*

```
rectOptions.color(Color.RED);
```

*// Thêm Polyline vào trong bản đồ*

```
Polyline polyline = myMap.addPolyline(rectOptions);
```

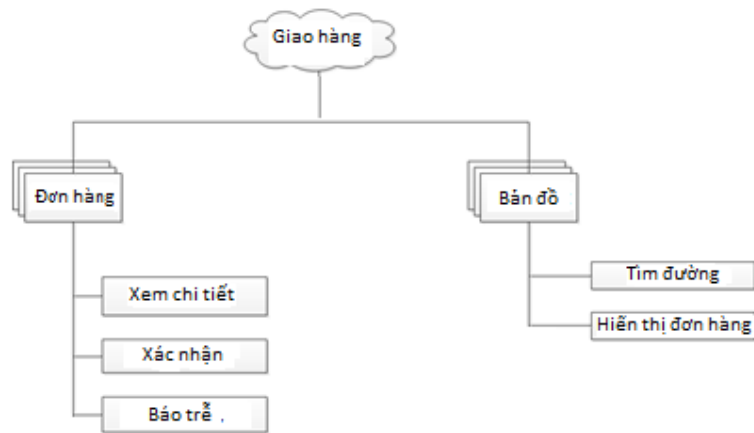
Các thuộc tính cơ bản của Polyline bao gồm:



- Color: dùng để thiết lập màu của đường nối các điểm, giá trị này được thiết lập thông qua phương thức PolylineOptions.color(), chúng ta cũng có thể thay đổi được thuộc tính này sau khi đã thêm Polyline vào bản đồ bằng phương thức Polyline.setcolor().
- Width: dùng để thiết lập bề ngang của đường nối các điểm tính bằng pixel(px) thông qua phương thức PolylineOptions.width().
- Visibility: là giá trị boolea dùng để xác định có hay không việc hiển thị Polyne bằng phương thức PolylineOptions.visible().

#### 4.4.4 Xây dựng các chức năng cho ứng dụng android application giao hàng

Các chức năng của ứng dụng được thể hiện trong sơ đồ sau:



**Hình 4-6: Chức năng cho ứng dụng android application giao hàng.**

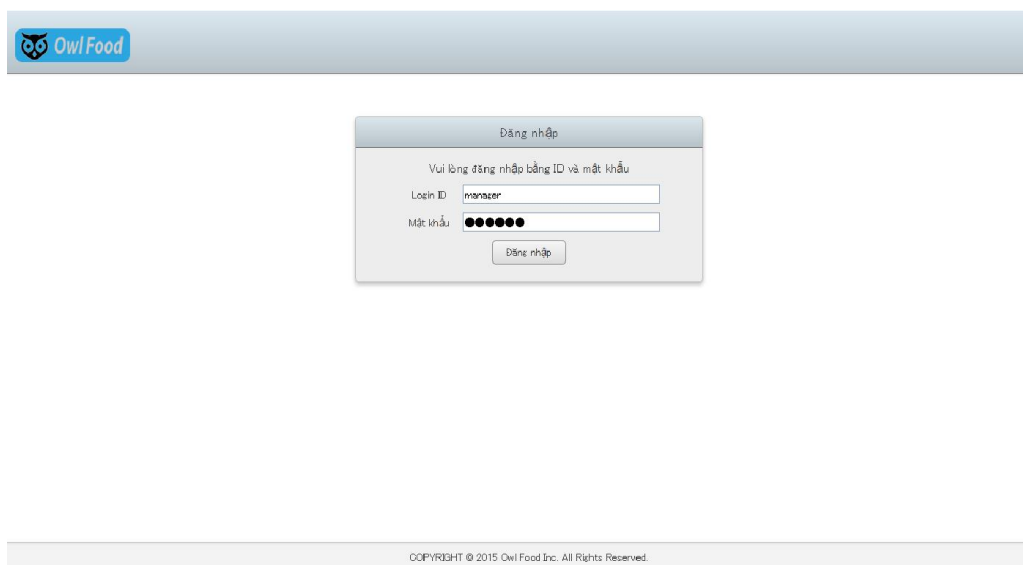
Ứng dụng có hai đối tượng chính mỗi đối tượng sẽ có tác vụ cụ thể như sau:

- Đơn hàng: Là đối tượng chính của ứng dụng giao hàng, có thể được xem chi tiết, được xác nhận hoàn thành với hệ thống và được nhân viên giao hàng thông tin báo trễ đến khách hàng.
- Bản đồ: đối tượng tương tác với nhân viên giao hàng giúp cho việc tìm đường để địa điểm giao hàng đồng thời giúp hiển thị các đơn hàng cần giao.

### 4.5 Giao diện hệ thống

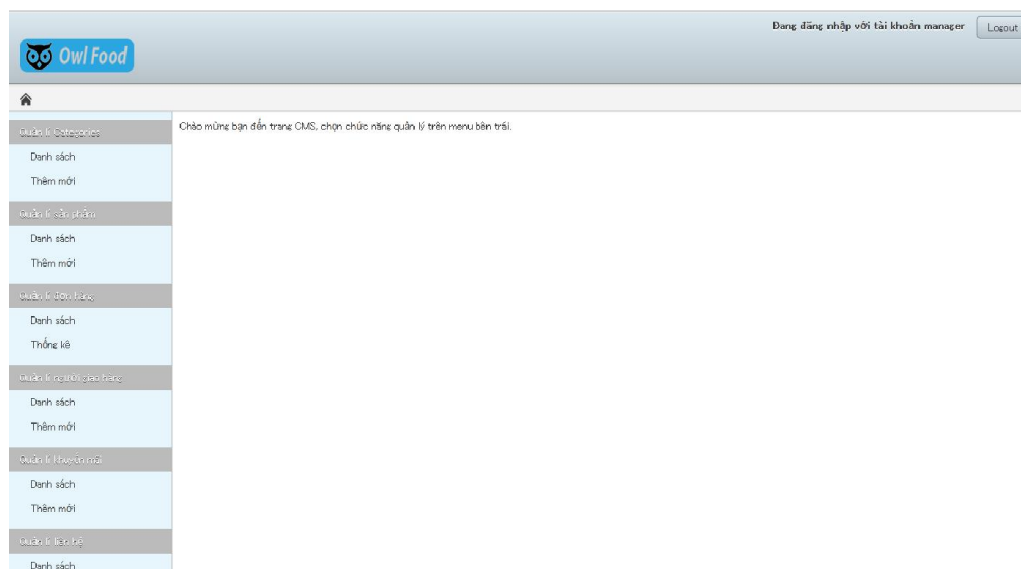
#### 4.5.1 Giao diện web quản lý hệ thống

Màn hình đăng nhập: được sử dụng để nhân viên quản lý nhập tài khoản, sau khi chứng thực thành công trên server thì nhân viên quản lý mới có thể sử dụng các chức năng.



**Hình 4-7: Màn hình đăng nhập web quản lý**

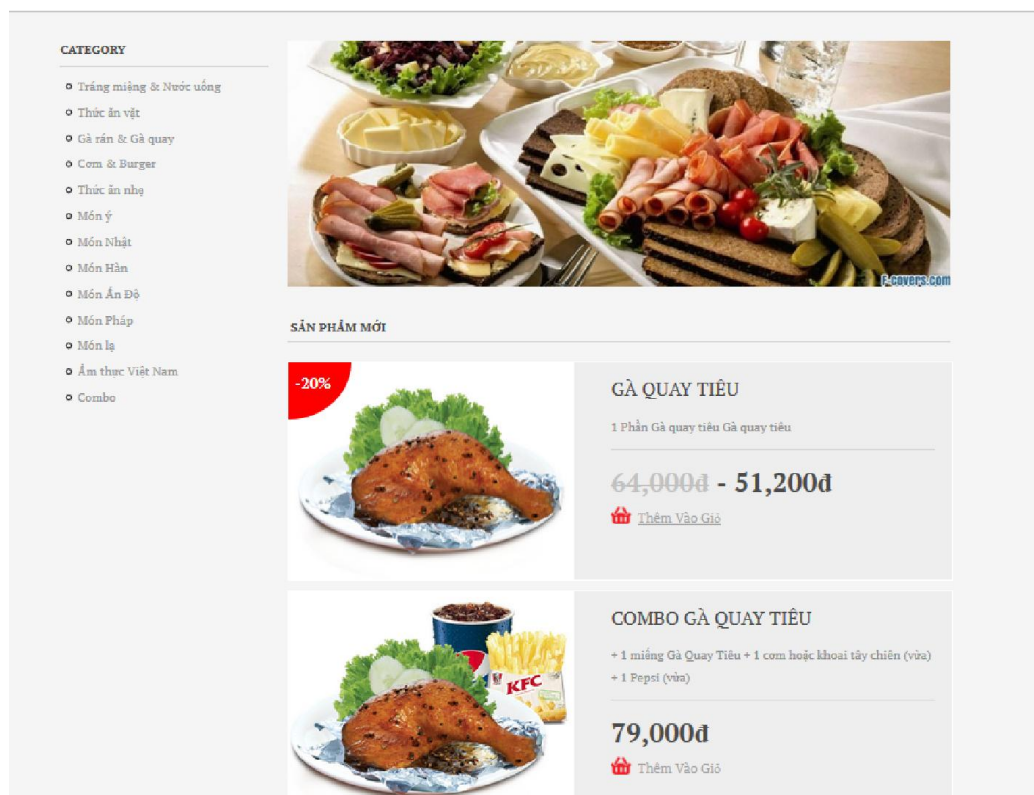
Màn hình chính: là giao diện web quản lý, bao gồm menu bên trái và phần nội dung bên phải với mỗi menu lựa chọn sẽ hiển thị nội dung tương ứng.



**Hình 4-8: Màn hình chính web quản lý**

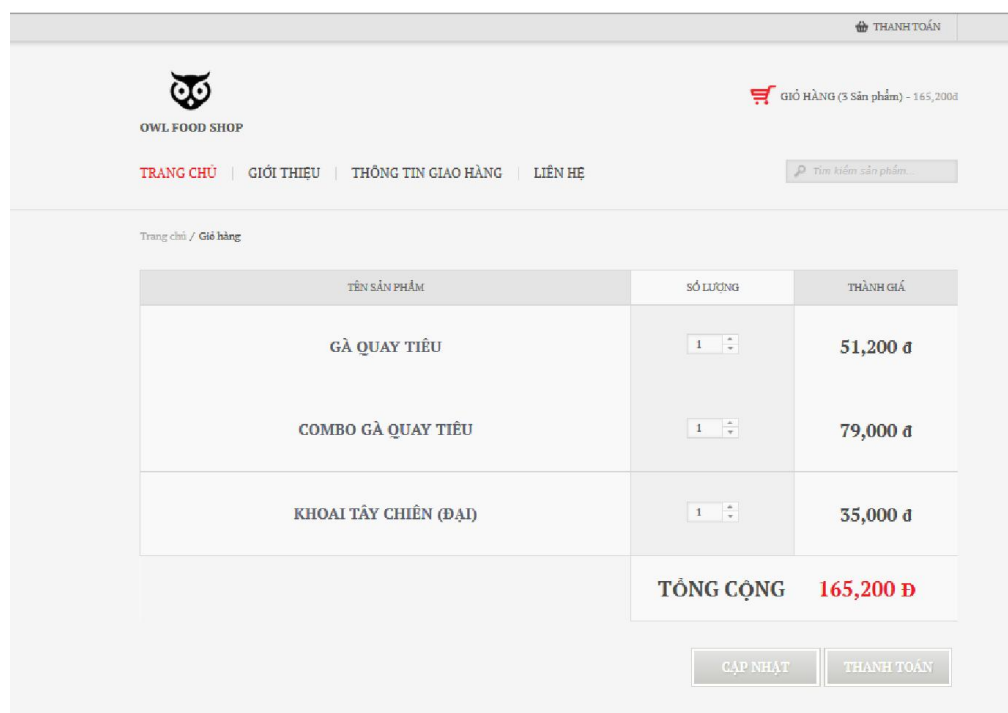
#### 4.5.2 Giao diện web bán hàng

Màn hình chính: nơi hiển thị thông tin về sản phẩm và các tùy chọn.



Hình 4-9: Màn hình chính web bán hàng

Màn hình giỏ hàng: nơi hiển thị các sản phẩm khách hàng lựa chọn.



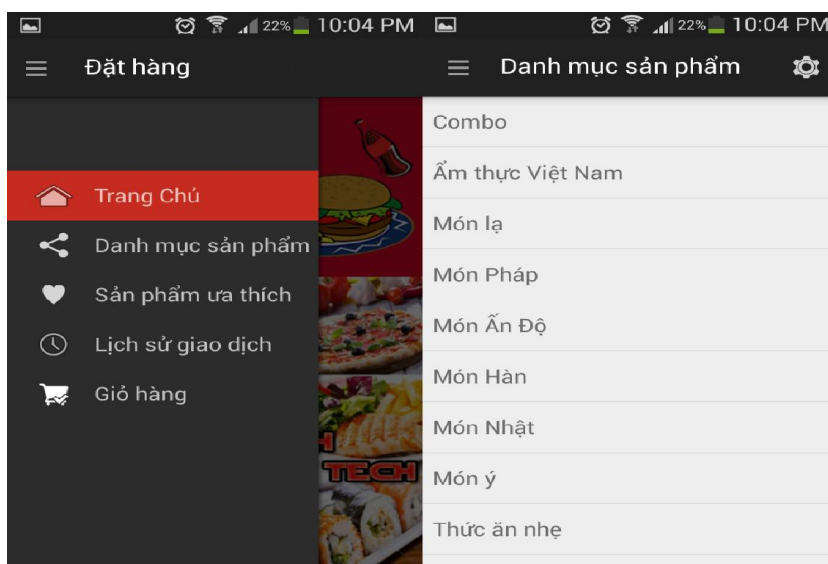
Hình 4-10: Màn hình giỏ hàng web bán hàng

Màn hình thanh toán: nơi thực hiện việc tạo đơn hàng và cung cấp thông tin từ khách hàng

Hình 4-11: Màn hình thanh toán web bán hàng

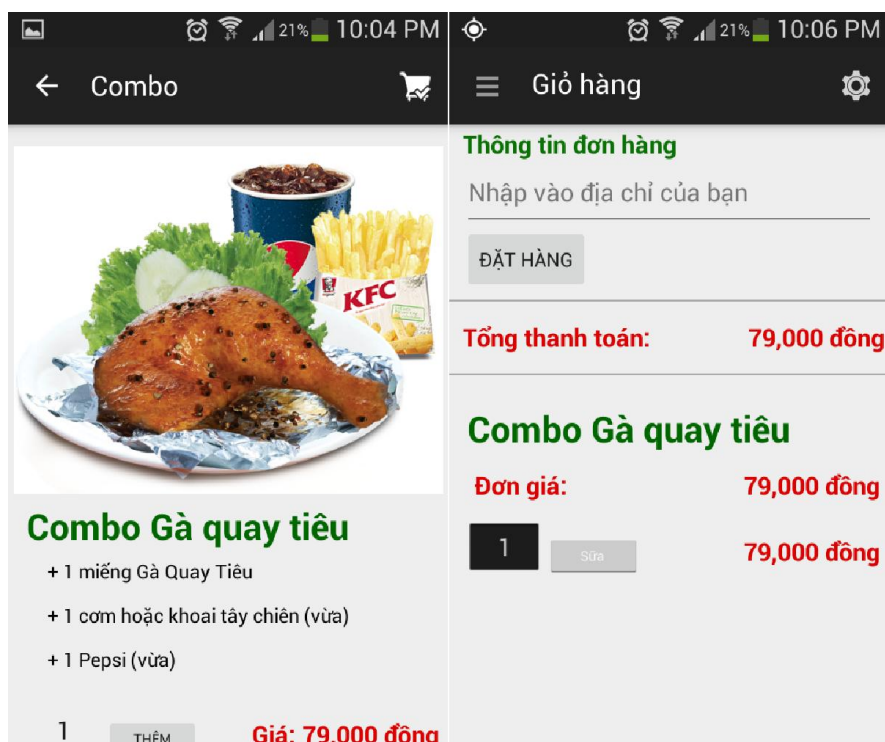
#### 4.5.3 Giao diện android application đặt hàng

Màn hình chính: bao gồm hình ảnh quảng cáo và menu chính



Hình 4-12: Màn hình chính android application đặt hàng

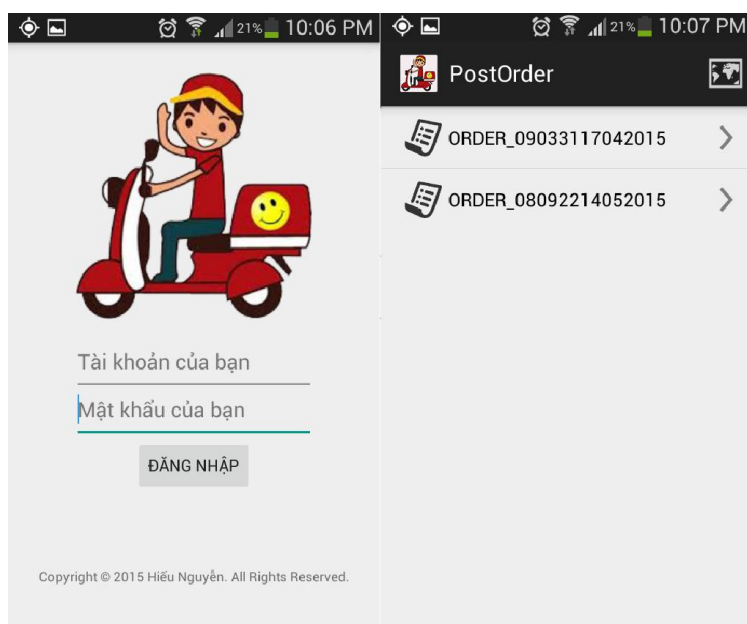
Màn hình sản phẩm và thanh toán: nơi thực hiện chọn sản phẩm và tạo đơn hàng.



Hình 4-13: Màn hình sản phẩm và thanh toán android application đặt hàng

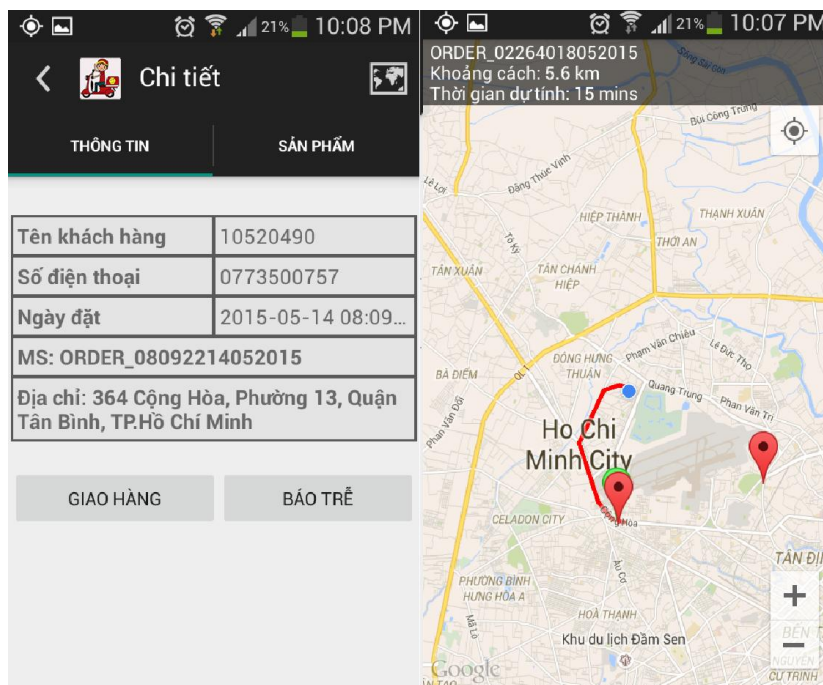
#### 4.5.4 Giao diện android application giao hàng

Màn hình đăng nhập và màn hình chính: đăng nhập và màn hình danh sách các đơn hàng:



Hình 4-14: Màn hình đăng nhập và màn hình chính android application giao hàng

Màn hình chi tiết đơn hàng và bản đồ: chứa thông tin của đơn hàng và bản đồ hiển thị.



Hình 4-15: Màn hình chi tiết đơn hàng và bản đồ android application giao hàng

#### 4.6 Tổng kết

Sau các thao tác trên em đã đi vào thiết kế chi tiết từng thành phần trong hệ thống đảm bảo hệ thống có thể đáp ứng đầy đủ những yêu cầu đã đề ra và có khả năng triển khai hoạt động trong thực tế. Hiện tại hệ thống đã được đưa vào sử dụng tại địa chỉ website: <http://order-sys.tk/>

## **CHƯƠNG 5.**

### **KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

#### **5.1 Tổng kết đề tài**

Khóa luận đã tổng kết và trình bày một cách hệ thống và cụ thể về kiến thức nền tảng, kiến thức về công nghệ để xây dựng hệ thống ORDER-SYSTEM dựa trên nền tảng Google Map Service và một số công nghệ khác. Đề tài đã xây dựng được một ứng dụng hoàn chỉnh trên cả client và server bao gồm các chức năng chính như sau:

Trung tâm điều hành với các chức năng:

- Web Application cung cấp các chức năng quản lý và điều hành hệ thống sử dụng thông qua trình duyệt web nên rất tiện dụng và nhanh chóng.
- Lưu trữ dữ liệu đơn hàng có thể được sử dụng bằng các phần mềm kế toán tự xây dựng.
- Tiếp nhận, xử lý thông tin yêu cầu phục vụ của khách hàng, tiến hành phân phối cho nhân viên giao hàng.
- Cho phép thêm các sản phẩm để người dùng có thể đặt hàng.

Ứng dụng client android application dùng cho nhân viên giao hàng:

- Hiện thị vị trí các đơn hàng được phân phối, chức năng tìm đường dựa trên Google Map.
- Cung cấp việc thông tin với khách hàng qua sms của điện thoại.

Tóm lại, đề tài “XÂY DỰNG HỆ THỐNG ĐẶT HÀNG BẰNG ĐIỆN THOẠI DI ĐỘNG” đã hoàn thành các yêu cầu đặt ra và có khả năng triển khai hoạt động trong thực tế. Việc quản lý nhân viên giao hàng cũng được thực hiện dễ dàng hơn. Hệ thống được triển khai giúp có thêm một hướng tiếp cận mới trong việc quản lý và điều hành việc giao hàng trong bối cảnh hiện nay bằng cách tận dụng sự phát triển mạnh mẽ của các công nghệ và thiết bị di động những năm gần đây mà điển hình trong hệ thống này là Smartphone Android. Đề tài được hoàn thành dựa trên nền tảng nghiên cứu lý thuyết, khảo sát yêu cầu đối với một hệ thống giao hàng trên thực tế giúp khắc phục những hạn chế của hệ thống truyền thống, tăng hiệu suất trong quá trình quản lý, điều hành hệ thống tại trung tâm điều khiển.

## 5.2 Những khó khăn và hạn chế

Trong quá trình thực hiện nhóm tác giả đã gặp một số khó khăn khác nhau về công nghệ, thiết bị và việc thử nghiệm đề tài trong thực tế. Cụ thể những khó khăn hạn chế gặp phải như sau:

- Những công nghệ áp dụng đòi hỏi tính thực tế cao chưa có tài liệu tiếng Việt nên tốn thời gian đầu tư nghiên cứu nhiều, đặc biệt là Google Map API v2.
- Những hạn chế về mặt dịch vụ cũng như độ chính xác của Google Map ở Việt Nam gây ra một số sai lệch trong việc tìm kiếm và phân giải địa chỉ.
- Trong quá trình thực hiện nhiều lần tuyến cáp quang AAG xảy ra sự cố ảnh hưởng đến quá trình thực hiện.
- Những hạn chế về mặt kinh phí và thiết bị khiến cho đề tài còn giới hạn.

## 5.3 Hướng phát triển đề tài

Đề tài bước đầu đã xây dựng được hệ thống đặt hàng sử dụng điện thoại di động thông qua thiết bị di động nền tảng Android. Nếu có thêm thời gian và kinh phí đầu tư thêm từ nhà trường và các công ty thì tương lai có thể phát triển hệ thống một cách hoàn chỉnh hơn, không chỉ có chức năng quản lý mà còn hỗ trợ tối đa cho khách hàng và nhân viên giao hàng. Một số hướng phát triển như:

- Mở rộng hệ thống để sử dụng cho chuỗi cửa hàng.
- Thêm khả năng quản lý thông qua việc sử dụng tổng đài VOICE IP để thông tin với khách hàng và nhân viên giao hàng.
- Xây dựng một bản đồ riêng cho khu vực Hồ Chí Minh để tăng tốc độ xử lý và giảm sai sót trong quá trình phân giải địa chỉ thành tọa độ trên bản đồ.
- Áp dụng các thuật toán để tính tự động hóa việc phân phối đơn hàng cho nhân viên giao hàng đồng thời tìm ra đường đi tối ưu cho nhân viên giao hàng.
- Áp dụng bản đồ lưu lượng giao thông để tránh các tình huống trễ do giao thông tắc nghẽn.
- Viết thêm ứng dụng cho di động trên nhiều nền tảng như IOS hay WINDOWS PHONE giúp hệ thống có thể được sử dụng rộng rãi.



## DANH MỤC TÀI LIỆU THAM KHẢO

- [1] Android Map API. [Online].  
<https://developers.google.com/android/reference/com/google/android/gms/maps/package-summary>
- [2] Google Place API. [Online].  
<https://developers.google.com/places/>
- [3] A. El-Rabbany, "Chaper 1: Introduction to GPS," in *Introduction to GPS - The Global Positioning System*. Boston, London: Artech House, 2002, p. 1.
- [4] G. Svennerberg, *Beginning Google Maps API 3*. Apress, 2010.
- [5] Android Developer. [Online].  
<http://www.developer.android.com>
- [6] Google Map API v2. [Online].  
<https://developers.google.com/maps/documentation/android/>
- [7] Google Map API v2. [Online].  
<http://www.google.com>
- [8] StackOverFlow. [Online].  
<http://stackoverflow.com>
- [9] Wikipedia. [Online].  
[https://en.wikipedia.org/wiki/Main\\_Page](https://en.wikipedia.org/wiki/Main_Page)  
[https://vi.wikipedia.org/wiki/Trang\\_Ch%C3%ADnh](https://vi.wikipedia.org/wiki/Trang_Ch%C3%ADnh)
- [10] Yiiframework document. [Online].  
<http://www.yiiframework.com/doc/>

