

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



ĐỒ GIA BẢO

**NGHIÊN CỨU VÀ XÂY DỰNG
HỆ THỐNG HỖ TRỢ GIAO HÀNG TẬN NHÀ
SỬ DỤNG CÔNG NGHỆ GPS**

LUẬN VĂN THẠC SĨ KỸ THUẬT

HỒ CHÍ MINH – 2014

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



ĐỖ GIA BẢO

**NGHIÊN CỨU VÀ XÂY DỰNG
HỆ THỐNG HỖ TRỢ GIAO HÀNG TẬN NHÀ
SỬ DỤNG CÔNG NGHỆ GPS**

Chuyên ngành: Hệ thống thông tin

Mã số: 60.48.01.04

LUẬN VĂN THẠC SĨ KỸ THUẬT

NGƯỜI HƯỚNG DẪN KHOA HỌC : TS. ĐÀM QUANG HỒNG HẢI

HỒ CHÍ MINH – 2014

LỜI CAM ĐOAN

Tôi cam đoan đây là công trình nghiên cứu của riêng tôi.

Các số liệu, kết quả nêu trong luận văn là trung thực và chưa từng được ai công bố trong bất kỳ công trình nào khác.

Tác giả luận văn ký và ghi rõ họ tên

Đỗ Gia Bảo

LỜI CẢM ƠN

Người đầu tiên em xin dành sự cảm ơn và kính trọng đặc biệt là TS. Đàm Quang Hồng Hải. Thầy đã tạo nhiều cảm hứng và làm em ngày một say mê hơn trong công việc nghiên cứu của mình. Thầy cũng chính là người đã tin tưởng, trực tiếp hướng dẫn, chia sẻ kiến thức và tạo điều kiện tốt nhất để em có thể hoàn thành Luận văn này.

Em xin gửi lời cảm ơn đến Quý Thầy Cô Học viện Công nghệ Bưu chính Viễn thông đã tận tình giảng dạy em trong suốt quá trình học tập tại trường.

Em cũng xin cảm ơn đến Quý Thầy Cô trường Đại học Sài Gòn, và bạn bè đã động viên tinh thần và chia sẻ để em có thể vượt qua mọi khó khăn trong cuộc sống.

Những người cuối cùng mà em xin dành lời cảm ơn, là gia đình em. Họ luôn là điểm tựa đầu tiên và mãi mãi của em.

Vì thời gian để thực hiện luận văn này hạn hẹp nên em không thể tránh khỏi những thiếu sót. Em kính mong nhận được sự thông cảm và chỉ bảo từ Quý Thầy Cô.

Cuối cùng, em xin chúc Quý Thầy Cô, gia đình và các bạn một lời chúc sức khỏe và đạt nhiều thành công trong cuộc sống.

TP. Hồ Chí Minh, Ngày 31 tháng 5 năm 2014

MỤC LỤC

LỜI CAM ĐOAN	i
LỜI CẢM ƠN	ii
MỤC LỤC.....	iii
DANH MỤC CÁC THUẬT NGỮ, CHỮ VIẾT TẮT.....	vi
DANH SÁCH BẢNG	viii
DANH SÁCH HÌNH VẼ	ix
MỞ ĐẦU.....	1
CHƯƠNG 1. TỔNG QUAN NGHIÊN CỨU.....	4
1.1. Tổng quan về hệ thống hỗ trợ giao hàng	4
1.2. Giới hạn của đề tài	5
1.3. Đối tượng, phạm vi và phương pháp nghiên cứu của đề tài.....	6
1.3.1. Đối tượng nghiên cứu.....	6
1.3.2. Phạm vi nghiên cứu.....	6
1.3.3. Phương pháp nghiên cứu.....	7
1.4. Kết luận.....	7
CHƯƠNG 2. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG HODESS	8
2.1. Khảo sát	8
2.1.1. GPS Vehicle Tracking Systems - The Shadow Tracker® Line.....	8
2.1.2. OpenGTST TM - Open GPS Tracking System.....	9
2.1.3. Ưu điểm của đề tài so với những hệ thống đã có.....	10
2.2. Xác định yêu cầu và các đối tượng trong hệ thống hỗ trợ giao hàng	11
2.2.1. Yêu cầu đối với hệ thống	11
2.2.2. Các đối tượng trong hệ thống.....	12
2.3. Thiết kế hệ thống hỗ trợ giao hàng	15
2.3.1. Server - ứng dụng Web	15
2.3.2. Client - ứng dụng cài đặt trên Smartphone	15
2.4. Mô hình dữ liệu.....	16

2.5. Thiết kế Web Service.....	18
2.6. Mô tả ứng dụng Client trên Android	19
2.7. Giải pháp tìm hành trình giao hàng ngắn nhất.....	22
2.8. Giải pháp VoIP	24
2.9. Kết luận.....	25
CHƯƠNG 3. BÀI TOÁN TRAVELING SALESMAN PROBLEM.....	26
3.1. Nguồn gốc bài toán	26
3.2. Định nghĩa.....	26
3.3. Độ phức tạp.....	27
3.4. Xấp xỉ.....	27
3.5. Xây dựng hành trình	27
3.6. Cải thiện hành trình.....	28
3.7. Một số giải thuật tiêu biểu	30
3.7.1. Lin-Kernighan	30
3.7.2. Tabu-Search	30
3.7.3. Simulated Annealing	30
3.7.4. Genetic Algorithm.....	31
3.7.5. Branch and Bound.....	32
3.7.6. Ant colony optimization.....	33
3.8. Xây dựng chức năng tìm hành trình giao hàng ngắn nhất	34
CHƯƠNG 4. HIỆN THỰC HỆ THỐNG HODESS.....	36
4.1. Những công nghệ sử dụng trong quá trình xây dựng hệ thống hỗ trợ giao hàng	36
4.1.1. Google Map.....	36
4.1.2. Công nghệ GPS	36
4.1.3. Java Server Page và Java Servlet	37
4.1.4. RESTful Web Service	38
4.1.5. MySQL 5.5 và Apache Tomcat 7	38
4.1.6. JSON và thư viện mã nguồn mở GSON	40

4.1.7. Hệ thống tổng đài Asterisk.....	41
4.1.8. Tổng quan về SIP	43
4.2. Xây dựng cơ sở dữ liệu	44
4.2.1. Mô hình dữ liệu	44
4.2.2. Chi tiết cơ sở dữ liệu	45
4.3. Xây dựng Web Service	49
4.3.1. Chức năng đăng nhập.....	49
4.3.2. Chức năng đăng xuất.....	50
4.3.3. Chức năng hiển thị các thời điểm giao hàng trong ngày.....	51
4.3.4. Chức năng cập nhật trạng thái và hiển thị hành trình giao hàng ngắn nhất	52
4.3.5. Chức năng hiển thị thông tin chi tiết khách hàng	53
4.3.6. Chức năng cập nhật trạng thái địa chỉ khách hàng trong một thời điểm giao hàng	54
4.3.7. Chức năng cập nhật thông tin phương tiện giao hàng.....	55
4.4. Xây dựng Web Application điều hành giao hàng.....	56
4.4.1. Xây dựng chức năng điều phối và thiết lập hành trình	57
4.4.2. Xây dựng hệ thống bản đồ	59
4.5. Xây dựng ứng dụng Client.....	63
4.6. Triển khai tổng đài VoIP	66
4.7. Kết luận.....	67
KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN ĐỀ TÀI.....	68
TÀI LIỆU THAM KHẢO.....	70

DANH MỤC CÁC THUẬT NGỮ, CHỮ VIẾT TẮT

Viết tắt	Tiếng Anh	Tiếng Việt
3G	Third-generation technology	Công nghệ truyền thông thế hệ thứ ba
ACO	Ant Colony Optimization	Thuật toán đàn kiến
AJAX	Asynchronous JavaScript and XML	JavaScript và XML không đồng bộ
AP	Assignment Problem	Bài toán phân công
ATSP	Asymmetric Traveling Salesman Problem	TSP bất đối xứng
API	Application Programming Interface	Giao diện lập trình ứng dụng
BnB	Branch and Bound	Thuật toán nhánh cận
CSDL	Database	Cơ sở dữ liệu
DFBnB	Depth-First Branch & Bound	Thuật toán nhánh cận theo chiều sâu
ERP	Enterprise resource planning	Phần mềm hoạch định khai thác nguồn tài nguyên doanh nghiệp
GA	Genetic Algorithm	Giải thuật di truyền
GPS	Global Positioning System	Hệ thống định vị toàn cầu
GUI	Graphical user interface	Giao diện người dùng đồ họa
HODESS	Home Delivery Support System	Hệ thống hỗ trợ giao hàng tận nhà
HTML	HyperText Markup Language	Ngôn ngữ hiển thị siêu văn bản
HTTP	Hypertext Transfer Protocol	Giao thức truyền tải siêu văn bản
Json	JavaScript Object Notation	Cú pháp để lưu trữ và trao đổi thông tin văn bản
JSP	Java Server Page	Bộ tiền xử lý văn lệnh Java
PBX	Private Branch Exchange	Tổng đài nội bộ

Viết tắt	Tiếng Anh	Tiếng Việt
REST	Representational State Transfer	Chuyển giao trạng thái đại diện
SA	Simulated Annealing	Giải thuật mô phỏng luyện kim
SCTP	Stream Control Transmission Protocol	Giao thức điều khiển truyền vận dòng
SIP	Session Initiation Protocol	Giao thức VoIP
SMTP	Simple Mail Transfer Protocol	Giao thức truyền tải thư tín đơn giản
STSP	Symmetric Traveling Salesman Problem	TSP đối xứng
SOAP	Simple Object Access Protocol	Giao thức truy cập đối tượng đơn giản
TCP	Transmission Control Protocol	Giao thức điều khiển truyền vận
TSP	Traveling Salesman Problem	Bài toán người đi du lịch
UDP	User Datagram Protocol	Giao thức gói dữ liệu người dùng
VoIP	Voice over Internet Protocol	Thoại trên giao thức Internet
WSDL	Web Services Description Language	Ngôn ngữ mô tả dịch vụ Web
XML	Extensible Markup Language	Ngôn ngữ đánh dấu mở rộng

DANH SÁCH BẢNG

Bảng 2.1.	Số khả năng và thời gian tính toán dựa trên số lượng thành phố	23
Bảng 4.1.	Các phiên bản Tomcat Server	40
Bảng 4.2.	Quan hệ ADMINISTRATOR	45
Bảng 4.3.	Quan hệ REGION	45
Bảng 4.4.	Quan hệ TEAM	45
Bảng 4.5.	Quan hệ DRIVER	46
Bảng 4.6.	Quan hệ DRIVERDETAIL	46
Bảng 4.7.	Quan hệ OPERATOR	47
Bảng 4.8.	Quan hệ TRANSPORT	47
Bảng 4.9.	Quan hệ CUSTOMER	48
Bảng 4.10.	Quan hệ CALENDAR	48
Bảng 4.11.	Trạng thái địa chỉ giao hàng	58
Bảng 4.12.	Trạng thái xe trong hệ thống hỗ trợ giao hàng	60
Bảng 4.13.	Danh sách Marker trên bản đồ Client	64

DANH SÁCH HÌNH VẼ

Hình 2.1.	Shadow Tracker® Micro Vision™	8
Hình 2.2.	Tracker® Vision™ II plus	9
Hình 2.3.	Shadow Tracker® Live	9
Hình 2.4.	OpenGTS™ - Open GPS Tracking System	10
Hình 2.5.	Mô tả thiết kế hệ thống hỗ trợ giao hàng	16
Hình 2.6.	Mô hình thực thể kết hợp	17
Hình 2.7.	Mô hình xử lý Web service của hệ thống hỗ trợ giao hàng	19
Hình 2.8.	Quy trình giao tiếp giữa ứng dụng và Google Map Service	20
Hình 2.9.	Quy trình giao tiếp giữa ứng dụng và HODESS Web Service	21
Hình 2.10.	Mô tả bài toán TSP dạng đồ thị	23
Hình 2.11.	Sự bùng nổ nhanh chóng của bài toán tổ hợp TSP	24
Hình 3.1.	Phương pháp chuyển đổi 2-opt và 3-opt	28
Hình 3.2.	Phương pháp chuyển đổi double-bridge	29
Hình 3.3.	Mô tả giải thuật di truyền	32
Hình 3.4.	Kết quả giữa DFBNB và tìm kiếm cục bộ với 300 thành phố ATSP ngẫu nhiên	33
Hình 3.5.	Kết quả tìm được hành trình ngắn nhất sử dụng thuật toán BnB của TSP	34
Hình 3.6.	Quy trình xử lý tìm hành trình ngắn nhất dựa trên Google API và thuật toán BnB của bài toán TSP	35
Hình 4.1.	Quy trình thực thi một trang JSP	37
Hình 4.2.	Servlet	38
Hình 4.3.	Kiến trúc tổ chức của Asterisk	42
Hình 4.4.	Cơ sở dữ liệu	44
Hình 4.5.	Chức năng đăng nhập	49
Hình 4.6.	Chức năng đăng xuất	50
Hình 4.7.	Chức năng hiển thị các danh sách thời điểm giao hàng trong	51

	ngày	
Hình 4.8.	Chức năng cập nhật trạng thái địa chỉ và hiển thị hành trình giao hàng gần nhất	52
Hình 4.9.	Chức năng hiển thị thông tin chi tiết khách hàng	53
Hình 4.10.	Chức năng cập nhật trạng thái địa chỉ trong một thời điểm giao hàng	54
Hình 4.11.	Chức năng cập nhật thông tin phương tiện giao hàng	55
Hình 4.12.	Ứng dụng web quản lý hỗ trợ giao hàng	56
Hình 4.13.	Thành phần điều phối địa chỉ giao hàng cho một tài xế	57
Hình 4.14.	Danh sách tài xế có khả năng giao trong việc lập lịch giao hàng cho khách	59
Hình 4.15.	Danh sách khách hàng trong ngày	59
Hình 4.16.	Hệ thống bản đồ giám sát hành trình giao hàng	61
Hình 4.17.	Hệ thống giám sát trạng thái các xe	61
Hình 4.18.	Số lượng các xe trong hệ thống	62
Hình 4.19.	Hiển thị theo phân loại trạng thái xe	62
Hình 4.20.	Giám sát hành trình của một xe	63
Hình 4.21.	Theo dõi lịch giao hàng của một xe	63
Hình 4.22.	Màn hình chính Client	65
Hình 4.23.	Danh sách các thời điểm giao hàng kèm thông tin khách hàng	65
Hình 4.24.	Mô hình triển khai tổng đài	66
Hình 4.25.	X-Lite	67

MỞ ĐẦU

Trong nhịp sống hiện nay, quỹ thời gian dành cho gia đình của cư dân thành phố dường như bị co hẹp lại. Vì việc mua sắm xưa nay vốn chiếm nhiều thời gian và nhu cầu mua sắm được đẩy mạnh, nên nhiều người đã đưa ra nhiều cách thức lựa chọn đặt mua hàng và yêu cầu việc giao hàng đến tận nhà.

Dịch vụ giao hàng được đẩy mạnh, song vẫn chưa giải quyết được những vấn đề bất cập hiện nay như điều phối chi nhánh giao, người giao hàng, lập lịch ứng với từng địa điểm cần giao, và quan trọng nhất là thiết lập hành trình giao hàng ngắn nhất qua từng địa chỉ phù hợp với khoảng thời gian cho phép. Ngày nay, hoạt động giao hàng vẫn còn thủ công dẫn đến xảy ra nhiều rủi ro không đáng có như giao nhầm địa chỉ, giao hàng không đúng giờ, hành trình giao hàng hầu như đều do người giao tự sắp xếp, nhân viên điều phối không thể giám sát được quá trình giao hàng của người giao, kèm theo đó là việc xử lý sự cố trong quá trình giao hàng cũng gặp nhiều khó khăn hơn.

Từ khi công nghệ trên thiết bị di động ngày càng phát triển, làm cho số lượng người sử dụng di động ngày một tăng theo, và một trong những công nghệ phổ biến nhất 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, điển hình như: Beidou của Trung Quốc, GLONASS của Nga, QZSS của Nhật Bản, Galileo của liên minh châu Âu EU, nhưng phổ biến nhất vẫn là GPS (Global Positioning System – Hệ thống định vị toàn cầu) do Bộ Quốc phòng Mỹ xây dựng, triển khai rộng rãi vào năm 1980. Hệ thống GPS đang được sử dụng rộng rãi trên toàn thế giới, và ở Việt Nam thì không còn xa lạ với công nghệ này nhờ khả năng định vị chính xác và được cung cấp miễn phí. Nó được sử dụng trong một số lĩnh vực như: quản lý điều hành xe, du lịch, thiết lập bản đồ, khảo sát tuyến đường... Đối với quản lý điều phối xe, thông thường các xe được trang bị thiết bị GPS chuyên dụng để định vị trí, tuy nhiên việc trang bị thiết bị này tại Việt Nam phải tốn chi phí cao, triển khai phức tạp, và chưa phổ biến.

Nếu hệ thống hỗ trợ giao hàng không trang bị thiết bị GPS chuyên dụng thì nhân viên điều phối không thể giám sát trạng thái và vị trí của người giao hàng.

Từ thực tế trên, em đã đưa ra giải pháp tận dụng khả năng định vị GPS được tích hợp trên hầu hết các Smartphone sử dụng hệ điều hành Android vào việc quản lý hệ thống hỗ trợ giao hàng. Vấn đề chính của giải pháp này là kết nối Smartphone đến trung tâm điều phối để truyền tải thông tin về vị trí, trạng thái hoạt động của xe, và thông tin giao hàng.

Đối với vấn đề tìm hành trình giao hàng ngắn nhất, thông thường ta nghĩ đến việc sử dụng chức năng tìm đường của Google Maps cung cấp. Hiện nay, chức năng này của Google Maps được sử dụng rộng rãi không chỉ được áp dụng trên Website mà còn được tích hợp trên hầu hết các Smartphone. Ngoài khả năng tìm đường ra, Google còn cung cấp API tính khoảng cách giữa hai tọa độ địa chỉ, khoảng cách này là khoảng cách đường bộ trên thực tế. Tuy nhiên, với chức năng tìm đường chỉ có thể xử lý được với hai tọa độ địa chỉ, trong khi số lượng tọa độ địa chỉ trong hệ thống hỗ trợ giao hàng nhiều hơn, và bài toán Traveling Salesman Problem được nhắc đến như là giải pháp để khắc phục cho trường hợp này. Bài toán Traveling Salesman Problem (TSP) cổ điển (hay còn gọi là bài toán người đi du lịch), là một bài toán tối ưu tổ hợp thuộc lớp NP-khó lần đầu đưa ra như một vấn đề toán học vào năm 1930. Bài toán được đưa ra để tìm lời giải về hành trình đường đi ngắn nhất qua tất cả các thành phố, và đã có nhiều giải thuật để giải bài toán này với độ chính xác cao với lượng vừa đủ các thành phố.

Nắm bắt được nhu cầu trên, dựa vào danh sách khoảng cách được thiết lập từ Google Maps và áp dụng giải thuật của bài toán TSP, ta xây dựng được chức năng thiết lập hành trình giao hàng ngắn nhất trong khoảng thời gian cho trước đối với mỗi người giao hàng. Người giao hàng chỉ việc sử dụng Smartphone của mình để cập nhật mọi thông tin về hành trình giao, thời điểm giao, thông tin khách hàng... từ trung tâm điều phối. Bên cạnh đó, em còn xây dựng một Web Application để tích hợp chức năng thiết lập hành trình giao hàng này, các chức năng quản lý khác như phương tiện chuyên chở, đội xe, tài xế, khách hàng, lập lịch giao hàng... Ngoài ra,

ta còn thực hiện được việc liên lạc thông qua tổng đài VoIP của trung tâm trên mạng 3G.

Tổng hợp vấn đề trên, luận văn có cấu trúc như sau:

Chương 1: Tổng quan vấn đề nghiên cứu

Em sẽ giới thiệu tổng quan về hệ thống cần nghiên cứu, quan trọng là nghiên cứu về chức năng tìm hành trình ngắn nhất dựa trên API của Google Maps để tìm khoảng cách và áp dụng giải thuật giải bài toán TSP.

Chương 2: Phân tích – thiết kế hệ thống HODESS

Chương này em sẽ làm rõ những yêu cầu, khó khăn đặt ra cho bài toán quản lý và điều phối giao hàng trong thực tế, bài toán về tìm hành trình đường đi ngắn nhất trong phạm vi cụ thể, để từ đó tiến hành phân tích những yêu cầu cần thiết cần phải đáp ứng khi xây dựng hệ thống hỗ trợ giao hàng này.

Chương 3: Bài toán Traveling Salesman Problem

Trong chương này, em sẽ giới thiệu lịch sử và một số heuristic của bài toán TSP cổ điển. Sau đó, phân tích và xây dựng chức năng tìm hành trình giao hàng ngắn nhất trong hệ thống hỗ trợ giao hàng dựa trên bài toán này.

Chương 4: Hiện thực hệ thống HODESS

Chương này em sẽ giới thiệu những công nghệ sẽ áp dụng. Đồng thời, em sẽ làm nổi bật thuật toán TSP được áp dụng cho hệ thống hỗ trợ giao hàng trong việc điều phối hành trình giao hàng cho mỗi tài xế. Cuối cùng, em sẽ trình bày thiết kế cơ sở dữ liệu, web service và mô tả hoạt động của hệ thống quản lý và ứng dụng được cài trên Smartphone.

CHƯƠNG 1. TỔNG QUAN NGHIÊN CỨU

1.1. Tổng quan về hệ thống hỗ trợ giao hàng

Hệ thống hỗ trợ giao hàng được xây dựng dựa trên các nghiên cứu như: GPS trong việc giám sát vị trí, cập nhật thông tin trạng thái và vị trí từ ứng dụng trên Smartphone đến trung tâm điều khiển thông qua Web Service, giải pháp tìm hành trình ngắn nhất dựa trên bài toán cổ điển TSP và hiển thị lên bản đồ của Google Maps, thực hiện việc liên lạc thông qua tổng đài VoIP trên mạng 3G.

Đối với hệ thống GPS, ta sẽ sử dụng thiết bị này được tích hợp trên Smartphone để xác định vị trí trên bản đồ giám sát. Ngoài ra, Smartphone sử dụng hệ điều hành Android có cài đặt ứng dụng Client để thực hiện việc cập nhật thông tin vị trí, trạng thái về trung tâm điều khiển thông qua Web Service.

Đối với việc xây dựng chức năng tìm hành trình giao hàng ngắn nhất trong hệ thống hỗ trợ giao hàng, ta dựa trên bài toán TSP cổ điển. Ban đầu, bài toán TSP chỉ tập trung vào việc tìm đường qua các thành phố với khoảng cách là đường thẳng nối trực tiếp giữa các thành phố trong tập danh sách cho trước, lời giải cho bài toán này là tìm được hành trình ngắn nhất đi qua tất cả các thành phố và trở về vị trí xuất phát sao cho quãng đường đi được là ngắn nhất. Google Maps chỉ cung cấp chức năng tìm đường đi ngắn nhất giữa hai địa chỉ, ta sử dụng API của nó để tìm đường, tính khoảng cách đường bộ giữa hai địa chỉ tùy vào việc ta chọn loại phương tiện lưu thông nào, chẳng hạn như người đi bộ, xe máy, xe ô tô, xe buýt. Trong hệ thống hỗ trợ giao hàng, ta chỉ sử dụng khoảng cách đường đi của loại phương tiện ô tô chuyên chở để tìm hành trình ngắn nhất trong phạm vi một thành phố mà Google Maps cung cấp, thay cho việc sử dụng khoảng cách là đường thẳng nối trực tiếp giữa các thành phố như trong bài toán TSP đã đề cập. Trên cơ sở đó, ta sẽ nghiên cứu giải pháp tìm đường cho hệ thống hỗ trợ giao hàng và hiển thị lên bản đồ như sau:

- Dựa trên API tính khoảng cách giữa hai tọa độ địa chỉ của Google Maps, ta lập ma trận trọng số khoảng cách ngắn nhất giữa các cặp tọa độ địa chỉ khách hàng đã điều phối cho tài xế.
- Sau đó, sử dụng thuật toán giải bài toán TSP để tìm hành trình đi qua tất cả các tọa độ địa chỉ và trở về vị trí xuất phát sao cho quãng đường đi được là ngắn nhất trong khoảng thời gian cho phép.
- Mỗi địa chỉ giao hàng sẽ được đánh số thứ tự sau khi tìm được hành trình và được lưu vào cơ sở dữ liệu.
- Để kiểm tra việc hiển thị lên bản đồ của Google, ta sẽ truy vấn danh sách địa chỉ đã được sắp xếp sẵn từ cơ sở dữ liệu, sử dụng API để vẽ đường đi cho từng cặp địa chỉ trên bản đồ, lần lượt theo thứ tự cho đến hết và quay trở về vị trí ban đầu.

Hệ thống hỗ trợ giao hàng sẽ áp dụng việc thiết lập hành trình giao hàng này cho mỗi tài xế tại một thời điểm nhất định. Ngoài ra, việc hiển thị hành trình trên bản đồ không chỉ áp dụng cho hệ thống giám sát tại trung tâm điều khiển mà còn áp dụng trên ứng dụng Client được cài đặt trên Smartphone của mỗi tài xế. Mọi thông tin được cập nhật như vị trí, trạng thái xe, thời điểm giao hàng, thông tin hành trình, thông tin khách hàng của ứng dụng trên Client đều được thực hiện thông qua Web Service.

Về phần liên lạc thông qua tổng đài VoIP, mỗi tài xế sẽ được cung cấp một đầu số để thực hiện việc liên lạc với nhân viên điều phối trên mạng 3G thay thế cho hệ thống liên lạc bằng bộ đàm truyền thống.

Như vậy, bài toán lập lịch giao hàng đưa ra sẽ khả thi hơn so với cách giao hàng truyền thống.

1.2. Giới hạn của đề tài

Trong hệ thống hỗ trợ giao hàng, tại một thời điểm mỗi hành trình được thiết lập có tối đa 20 địa chỉ, vì tính đặc thù của bài toán chỉ xét trong phạm vi một thành phố. Còn trong lĩnh vực kinh doanh bán hàng và giao hàng từ xa thường được kết hợp chặt chẽ với nhiều thành phần khác nhau, nên ta chỉ tập trung vào việc xây

dựng hệ thống hỗ trợ giao hàng tận nhà. Các vấn đề khác như quản lý hành chính, quản lý nhân sự, quản lý hàng hóa,... nằm ngoài phạm vi 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

- Tìm hiểu kỹ thuật định vị GPS.
- Nghiên cứu, ứng dụng Google Map API v2 vào hệ thống hỗ trợ giao hàng trên Android.
- Nghiên cứu và áp dụng các thuật toán tìm hành trình đường đi ngắn nhất qua các điểm dựa trên bài toán Traveling salesman problem (TSP).
- Tìm hiểu các phương thức, chức năng của hệ thống hỗ trợ giao hàng.
- Tìm hiểu Web Application trong xây dựng hệ thống quản lý điều phối, giám sát.
- Nghiên cứu RESTful Web Service.
- Xây dựng hệ thống tổng đài VoIP, giao tiếp và truyền dữ liệu thoại từ Smartphone Android với tổng đài thông qua mạng 3G.

1.3.2. Phạm vi nghiên cứu

- Nghiên cứu phát triển ứng dụng trên Google Map API v2 trên nền tảng Android kết hợp định vị GPS.
- Phát triển Java Web Application với Java EE 6 Web, Server Apache Tomcat 7, và MySQL 5.5.
- Ứng dụng Google Map Javascript API vào việc hiển thị dữ liệu trong hệ thống hỗ trợ giao hàng.
- Ứng dụng thuật toán Branch and Bound – một trong những thuật toán giải bài toán TSP để xây dựng chức năng định tuyến hành trình giao hàng ngắn nhất.
- Tạo Web Service để trao đổi dữ liệu giữa Client và Server.
- Xây dựng tổng đài dựa trên mã nguồn mở Asterisk, xây dựng Client sử dụng giao thức SIP giao tiếp với tổng đài.

1.3.3. Phương pháp nghiên cứu

Đề tài được nghiên cứu dựa trên phương pháp phân tích, kiểm thử, áp dụng các công nghệ, giải thuật tìm đường để xây dựng hệ thống hỗ trợ giao hàng.

1.4. Kết luận

Thông qua những vấn đề trình bày trên ta xây dựng được đề tài “Nghiên và xây dựng hệ thống hỗ trợ giao hàng tận nhà sử dụng công nghệ GPS”. Để hiểu được chi tiết hơn ta sẽ xem qua phân tích, thiết kế mô hình hoạt động của hệ thống hỗ trợ giao hàng được trình bày trong chương 2.

CHƯƠNG 2. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG HODESS

Chương này sẽ làm rõ những yêu cầu, khó khăn đặt ra cho bài toán quản lý và điều phối giao hàng trong thực tế, bài toán về tìm hành trình đường đi ngắn nhất trong phạm vi cụ thể, để từ đó tiến hành phân tích những yêu cầu cần thiết cần phải đáp ứng khi xây dựng hệ thống hỗ trợ giao hàng này.

2.1. Khảo sát

2.1.1. *GPS Vehicle Tracking Systems - The Shadow Tracker® Line [26]*

Đây là hệ thống kiểm soát các phương tiện vận tải. Bao gồm các sản phẩm truyền tải thông tin từ xa bằng USB, sóng radio, sóng di động thông qua một thiết bị chuyên dụng gọi là Shadow Tracker®, thiết bị này sẽ cập nhật thông tin về trung tâm sau 10 giây.

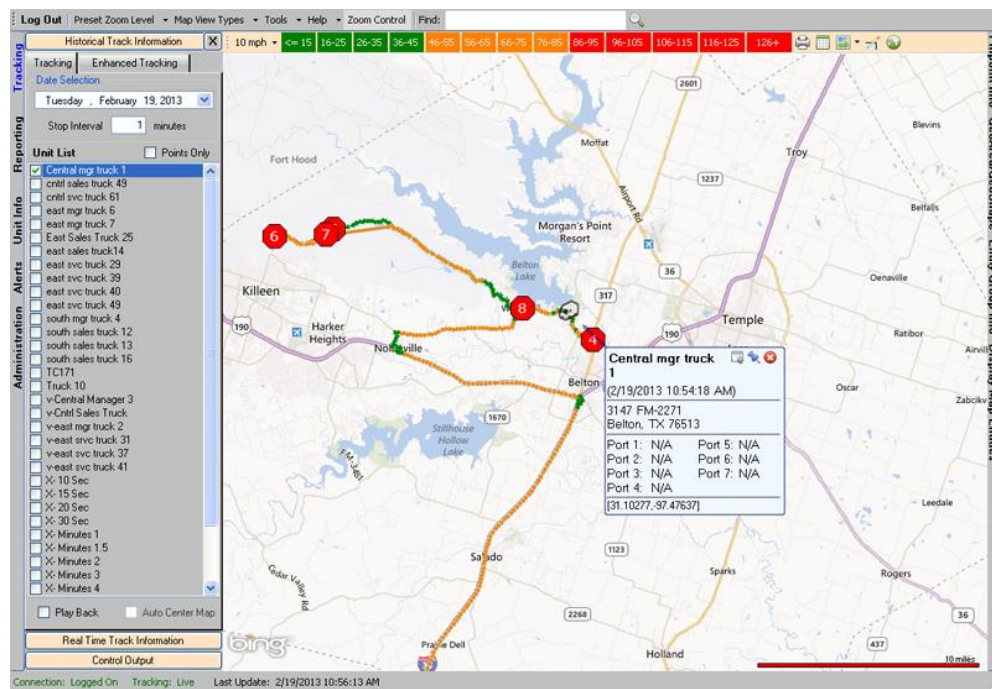


Hình 2.1: Shadow Tracker® Micro Vision™ [26]



Hình 2.2: Shadow Tracker® Vision™ II plus [26]

Ngoài ra hệ thống này còn trang bị thêm một phần mềm theo dõi theo thời gian thực bằng cách kết nối với Microsoft® Virtual Earth®. Nhà quản lý theo dõi vị trí và trạng thái các xe trong hệ thống.



Hình 2.3: Shadow Tracker® Live [26]

2.1.2. OpenGTS™ - Open GPS Tracking System [25]

Đây là dự án mã nguồn mở đầu tiên được thiết kế trong việc sử dụng GPS theo dõi các phương tiện giao thông, điện thoại di động trên trình duyệt web.



Hình 2.4: OpenGTSTTM - Open GPS Tracking System [25]

2.1.3. Ưu điểm của đề tài so với những hệ thống đã có

Thông qua việc tìm hiểu các hệ thống giám sát đã được triển khai thực tế, em đưa ra những ưu điểm của hệ thống hỗ trợ giao hàng tận nhà HODESS sử dụng Smartphone để giám sát so với hệ thống cũ sử dụng thiết bị định vị như sau:

- Dễ triển khai, nâng cấp phần mềm.
- Khả năng hiển thị chi tiết, tương tác tốt.
- Thiết lập hành trình đường đi ngắn nhất cho mỗi xe qua nhiều địa chỉ.
- Kết nối đến tổng đài trên mạng 3G.
- Tích hợp thêm nhiều tiện ích hỗ trợ lái xe.
- Tích hợp giao tiếp giữa trung tâm qua VoIP thay vì sử dụng bộ đàm.

Ưu điểm của hệ thống hỗ trợ giao hàng này vượt trội hơn với chi phí bỏ ra để trang bị một Smartphone thay cho một thiết bị định vị GPS chênh lệch không nhiều trong khoảng từ 2 đến 6 triệu.

Một Smartphone được tích hợp thêm GPS để thực hiện việc giám sát hành trình xe là một hướng tiếp cận mới của hệ thống hỗ trợ giao hàng HODESS. Ta sẽ phân tích những yêu cầu cần thiết đối với hệ thống này trong phần tiếp theo.

2.2. Xác định yêu cầu và các đối tượng trong hệ thống hỗ trợ giao hàng

Sau đây ta sẽ xác định các đối tượng và yêu cầu cần thiết với hệ thống hỗ trợ giao hàng HODESS.

2.2.1. Yêu cầu đối với hệ thống

Hệ thống quản lý của hỗ trợ giao hàng luôn có những yêu cầu cần phải đáp ứng sau:

- Cho phép người điều phối thấy được tổng thể của toàn bộ xe giao hàng đang hoạt động.
- Trạng thái hiện hành của mỗi xe: xe đang giao hàng, xe đang về vị trí trực, xe đang chờ nhận nhiệm vụ, xe gặp sự cố, xe chạy quá tốc độ, tài xế chưa đăng nhập chương trình.
- Vị trí và tốc độ xe.
- Trạng thái của mỗi điểm địa chỉ khách hàng trên bản đồ như: chưa giao, đang giao, đã giao.
- Tiếp nhận thông tin đặt hàng của khách, xác định vị trí tọa độ của khách để chọn ra chi nhánh giao hàng có vị trí gần nhất.
- Điều phối xe: mỗi tài xế sẽ được thiết lập một hành trình đường đi ngắn nhất qua tất cả các địa chỉ khách hàng mà người điều phối phân công giao, nơi xuất phát và kết thúc tại vị trí chi nhánh giao hàng trong một phạm vi nhất định.
- Việc liên lạc nội bộ giữa tài xế và trung tâm điều khiển, phòng hội thoại trên tổng đài VoIP được thực hiện thông qua mạng 3G.

- Quản lý, điều hành của hệ thống hỗ trợ giao hàng dựa trên nền tảng Web-Application.

Đối với việc xây dựng ứng dụng Client chuyên dụng để phục vụ cho tài xế giao hàng trong hệ thống hỗ trợ giao hàng cần phải được cài đặt trên Smartphone chạy hệ điều hành Android [12] với các chức năng sau:

- Giao tiếp với Server quản lý điều phối thông qua Web Service.
- Cập nhật trạng thái của xe với Server như: xe đang giao hàng, xe đang về vị trí trực, xe đang chờ nhận nhiệm vụ, xe gặp sự cố, xe chạy quá tốc độ.
- Cập nhật vị trí và tốc độ xe.
- Cập nhật trạng thái của địa chỉ giao hàng trên bản đồ như: chưa giao, đang giao, đã giao.
- Hiện thị danh sách các thời điểm giao trong ngày.
- Hiện thị danh sách vị trí, thông tin khách hàng.
- Ứng với mỗi thời điểm, hiện thị hành trình đường đi ngắn nhất qua các địa chỉ khách hàng trên bản đồ mà đã được người điều phối thiết lập sẵn từ hệ thống quản lý.
- Thực hiện liên lạc thông qua tổng đài trên mạng 3G.

2.2.2. Các đối tượng trong hệ thống

Sau khi nắm được các yêu cầu cần thiết của hệ thống hỗ trợ giao hàng, ta sẽ phân tích những đối tượng hoạt động chủ yếu trong hệ thống này bao gồm: tài xế, phương tiện chuyên chở, khách hàng, nhân viên điều phối. Các tài xế, nhân viên điều phối sẽ được quản lý theo từng đội và chi nhánh giao hàng riêng. Mỗi nhân viên điều phối và tài xế sẽ được cấp một tài khoản để thực hiện các công việc của mình đối với hệ thống hỗ trợ giao hàng.

2.2.2.1. Nhân viên điều phối

Nhân viên điều phối là người trực tiếp làm việc với hệ thống quản lý thông qua trình duyệt web để thực hiện việc giám sát, điều phối, lập lịch giao hàng, thiết lập hành trình giao hàng cho mỗi tài xế. Mỗi nhân viên điều phối sẽ được cấp một tài khoản riêng để thực hiện đúng chức năng công việc của mình được giao. Nhân

viên điều phối chịu trách nhiệm nhận khách hàng thông qua điện thoại, xác định vị trí của khách hàng và chọn ra chi nhánh giao hàng gần nhất. Điều phối tài xế giao hàng thích hợp đối với khách hàng. Quá trình điều phối diễn ra như sau:

- Nhân viên điều phối tiếp nhận yêu cầu từ khách hàng thông qua điện thoại.
- Yêu cầu cung cấp thông tin phục vụ việc giao hàng gồm: họ tên, địa chỉ, số điện thoại, email...
- Từ địa chỉ của khách hàng được nhập vào sẽ xác định được tọa độ vị trí khách hàng, chọn ra được chi nhánh giao hàng gần khách hàng nhất.
- Thực hiện việc điều phối các địa chỉ khách hàng cần giao tại một thời điểm cho mỗi tài xế như sau: nhân viên điều phối chọn các địa chỉ khách hàng trực tiếp trên bản đồ trong chức năng thiết lập hành trình giao hàng, và tiến hành định tuyến hành trình giao hàng ngắn nhất qua các địa chỉ đó. Danh sách các tài xế có khả năng phục vụ tại một thời điểm nhất định sẽ được liệt kê trong phân lập lịch giao cho khách hàng.
- Sau khi thực hiện xong việc điều phối, nếu có thay đổi về lịch trình giao hàng thì quá trình thiết lập hành trình giao hàng này sẽ phải thực hiện lại từ đầu.
- Ngoài ra, nhân viên điều phối còn được cung cấp thêm chức năng gọi điện đến từng tài xế hoặc đội xe theo khu vực tham gia hội thoại.
- Xem được lịch sử hoạt động của các tài xế giao hàng, các khách hàng phục vụ theo ngày tháng.

2.2.2.2. Tài xế giao hàng

Tài xế là người liên hệ trực tiếp phục vụ khách hàng trong hệ thống hỗ trợ giao hàng. Mỗi tài xế phải đăng nhập vào hệ thống qua ứng dụng Client được cài đặt trên Smartphone. Sau khi tài xế đăng nhập, Client sẽ tự động cập nhật danh sách các thời điểm giao hàng trong ngày, ứng với mỗi thời điểm là danh sách các thông tin khách hàng được sắp thứ tự theo hành trình giao. Khi chọn một thời điểm giao hàng, Client sẽ hiển thị hành trình đường đi ngắn nhất trên bản đồ, điểm xuất phát và kết thúc hành trình là tại vị trí chi nhánh giao hàng. Khi giao hàng xong tại tọa

độ địa chỉ khách hàng nào thì tài xế có nhiệm vụ cập nhật lại trạng thái cho tọa độ địa chỉ khách hàng đó. Cập nhật trạng thái xe như: khi đang làm nhiệm vụ, khi về vị trí trực, khi đang chờ nhận nhiệm vụ hoặc khi xe gặp sự cố. Client sẽ tự động cập nhật trạng thái xe chạy quá tốc độ, và trạng thái đăng nhập sau mỗi lần đăng nhập và thoát khỏi chương trình.

Trong quá trình giao hàng, Client sẽ liên tục cập nhật vị trí và tốc độ xe về hệ thống quản lý để nhân viên điều phối quan sát.

Mỗi tài xế cũng sẽ được cấp một đầu số dùng trong việc liên lạc nội bộ giữa tài xế và trung tâm điều khiển, tài xế và tài xế, tham gia vào các phòng hội thoại khi có yêu cầu.

2.2.2.3. Phương tiện chuyên chở

Phương tiện chuyên chở bao gồm tất cả các xe được phân thành các nhóm trọng tải khác nhau dựa trên số lượng hàng hóa cần giao. Xe được gắn với những thông tin như vị trí, tốc độ, trạng thái.

Vị trí và tốc độ xe sẽ được Smartphone ghi nhận lại bằng GPS trang bị trên điện thoại. Vị trí là tọa độ bao gồm vĩ độ (Latitude) và kinh độ (Longitude), các tọa độ này sẽ được gửi về trung tâm điều khiển để hiển thị vị trí xe trên bản đồ.

Trạng thái xe giúp cho nhân viên điều phối có thể giám sát hoạt động xe trên bản đồ. Các trạng thái xe được quy định như: xe đang giao hàng, xe đang về vị trí trực, xe đang chờ nhận nhiệm vụ, xe gặp sự cố, xe chạy quá tốc độ, tài xế chưa đăng nhập vào hệ thống.

2.2.2.4. Khách hàng

Khách hàng là đối tượng phục vụ chính. Khách hàng liên lạc bằng điện thoại và được yêu cầu cung cấp thông tin giao hàng cho nhân viên điều phối.

Khách hàng cũng có hai loại được quy định như: khách lạ, khách quen.

Mỗi khách hàng có thể đặt hàng với nhiều thời điểm giao hàng khác nhau. Mỗi thời điểm giao hàng của khách bao gồm: tài xế, ngày, giờ bắt đầu, giờ kết thúc, thứ tự giao hàng được nhân viên điều phối sử dụng chức năng định tuyến... và có hai trạng thái giao hàng: chưa giao, đang giao, đã giao.

2.3. Thiết kế hệ thống hỗ trợ giao hàng

Hệ thống hỗ trợ giao hàng tận nhà HODESS là hệ thống hoạt động theo mô hình Client-Server bao gồm hai thành phần sau:

2.3.1. Server - ứng dụng Web

Được cung cấp các chức năng:

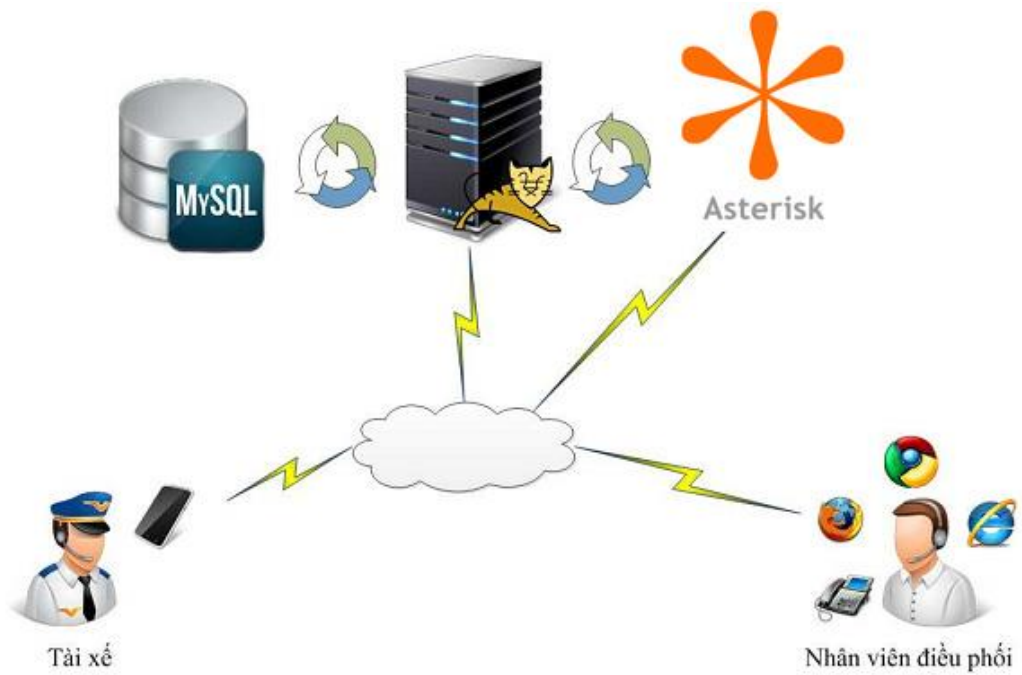
- Web Application bao gồm các chức năng quản lý và điều hành hệ thống, cung cấp cho nhân viên điều phối có cái nhìn tổng thể về vị trí, trạng thái hoạt động của các xe trên bản đồ, trạng thái địa chỉ giao hàng của khách, lập lịch ngày giờ giao cũng như lên hành trình đường đi giao hàng ngắn nhất cho từng tài xế, tiếp nhận thông tin yêu cầu phục vụ từ khách hàng, lưu trữ và theo dõi lịch sử hoạt động của tài xế xe theo ngày tháng,...
- Hệ thống RESTful Web Service [3][14] cho phép giao tiếp với ứng dụng Client được cài đặt trên Smartphone giúp tài xế cập nhật trạng thái xe, trạng thái địa chỉ giao hàng của khách sau khi hoàn thành, cập nhật vị trí, tốc độ xe đang hoạt động lên bản đồ hệ thống hỗ trợ giao hàng.
- Triển khai hệ thống liên lạc thông qua VoIP có tích hợp ứng dụng phone trên Web Application phục vụ cho việc liên lạc giữa nhân viên và tài xế trong hệ thống thay cho việc liên lạc bằng bộ đàm truyền thống.

2.3.2. Client - ứng dụng cài đặt trên Smartphone

Ứng dụng được cài đặt trên hệ điều hành Android bao gồm các chức năng:

- Hiển thị và cung cấp thông tin vị trí, trạng thái xe, trạng thái địa chỉ giao hàng tới trung tâm điều khiển.
- Tiếp nhận danh sách các thời điểm giao hàng cũng như danh sách thông tin khách hàng tương ứng mỗi thời điểm trong ngày.
- Hiển thị hành trình đường đi ngắn nhất được sắp xếp theo thứ tự đã được thiết lập bởi nhân viên điều phối.
- Chức năng gọi VoIP với tài khoản được cung cấp.

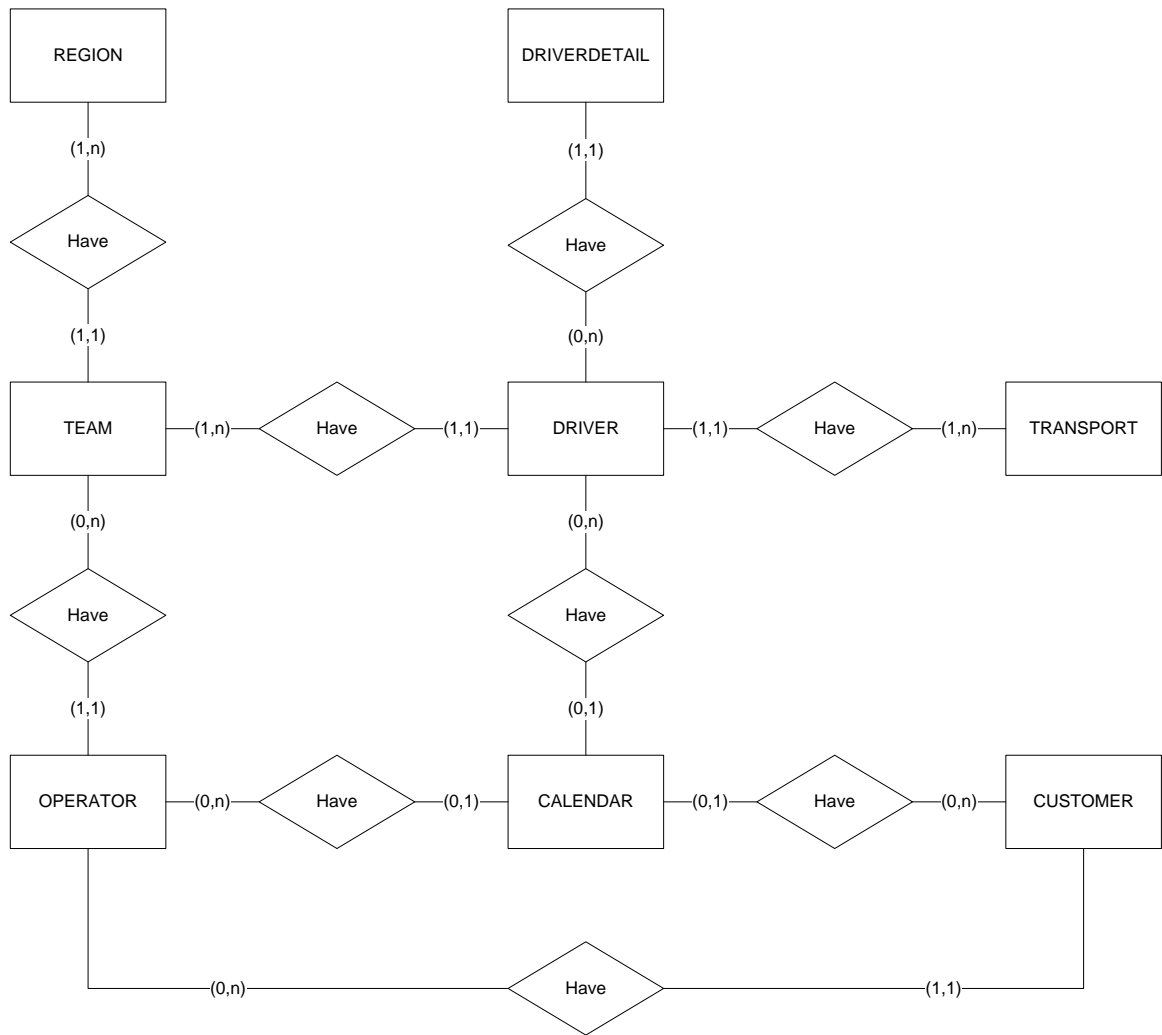
Hệ thống hỗ trợ giao hàng thiết kế được mô tả như trong hình 2.5.



Hình 2.5: Mô tả thiết kế hệ thống hỗ trợ giao hàng

2.4. Mô hình dữ liệu

Qua quá trình phân tích đối tượng và yêu cầu đối với hệ thống hỗ trợ giao hàng, em đưa ra mô hình thực thể kết hợp ở mức quan niệm để biểu diễn mối quan hệ giữa các đối tượng trong hệ thống.



Hình 2.6: Mô hình thực thể kết hợp

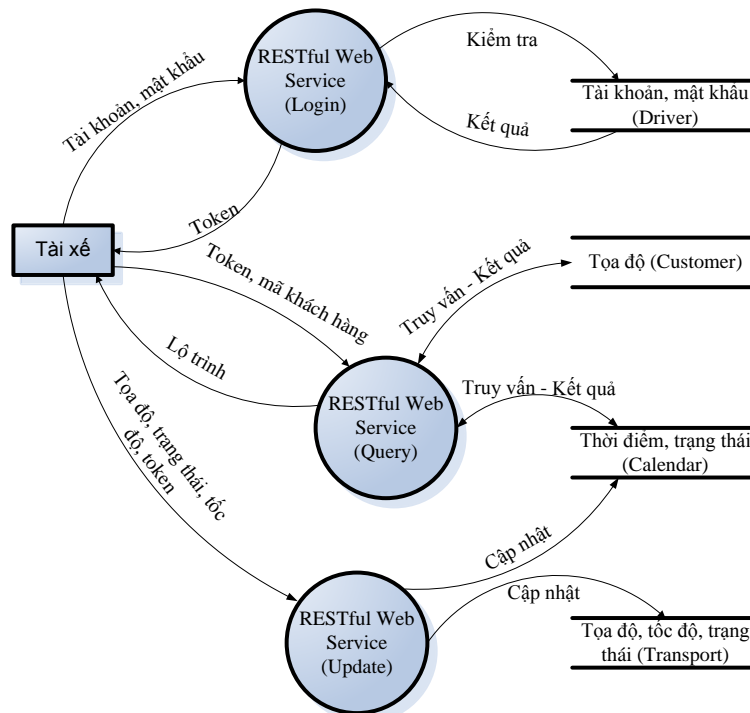
Theo như mô hình (Hình 2.6) ta thấy, các tài xế (DRIVER) và nhân viên điều phối (OPERATOR) được quản lý trong các đội (TEAM), các đội sẽ thuộc một vùng chi nhánh quản lý (REGION) cụ thể. Mỗi tài xế được trang bị một phương tiện (TRANSPORT), tuy nhiên một phương tiện có thể có nhiều tài xế sử dụng tùy theo ca làm việc của tài xế. Mọi hoạt động của tài xế sẽ được lưu lại để nhân viên điều phối theo dõi lịch sử hoạt động (DRIVERDETAIL). Các khách hàng (CUSTOMER) trong hệ thống sẽ được nhân viên điều phối tiếp nhận yêu cầu, mỗi khách hàng sẽ có nhiều lần đặt hàng vào từng thời điểm (CALENDAR) khác nhau, và sẽ được nhân viên điều phối lập lịch giao hàng cho từng thời điểm đặt giao hàng này của khách.

2.5. Thiết kế Web Service

Trong hệ thống hỗ trợ giao hàng HODESS này đòi hỏi phải thực hiện việc liên kết giữa trung tâm điều khiển và Client của tài xế. Để thực hiện được việc này em sử dụng RESTful Web Service [5][8] để xây dựng các chức năng phục vụ cho công việc như sau:

- Kiểm soát đăng nhập, đăng xuất hệ thống của tài xế.
- Truy vấn dữ liệu về các phương tiện của tài xế, thông tin địa chỉ khách hàng theo thời điểm giao hàng lên bản đồ hệ thống hỗ trợ giao hàng.
- Ghi nhận mọi thông tin được cập nhật từ Client như vị trí, trạng thái giao hàng cho khách, trạng thái và tốc độ của phương tiện...nhằm mục đích kiểm tra tài xế đã nhận được danh sách các thời điểm giao hàng kèm theo danh sách khách hàng hay chưa. Ngoài ra, theo dõi hành trình đường đi ngắn nhất theo thứ tự qua từng địa chỉ khách hàng.
- Sau mỗi lần hoàn thành việc giao hàng cho từng điểm địa chỉ khách hàng, tài xế sẽ cập nhật lại trạng thái đã giao vào cơ sở dữ liệu.

Qua mô tả thiết kế, ta xây dựng mô hình xử lý của hệ thống trong hình 2.7.



Hình 2.7: Mô hình xử lý Web service của hệ thống hỗ trợ giao hàng

Hệ thống hoạt động như sau:

- Tài xế đăng nhập vào hệ thống bằng tài khoản và mật khẩu được cung cấp sẵn để Web Service kiểm tra đăng nhập.
- Nếu thông tin đăng nhập hợp lệ tài xế sẽ được hệ thống hỗ trợ giao hàng cấp phát một token, và tài xế sẽ sử dụng token này để cập nhật thông tin và truy xuất các tài nguyên của hệ thống hỗ trợ giao hàng.
- Web Service sẽ cung cấp danh sách các thời điểm giao hàng gồm danh sách các khách hàng được sắp xếp theo hành trình.
- Tài xế sẽ phải cập nhật trạng thái, vị trí định kỳ về trung tâm điều khiển thông qua Web Service.
- Khi có khách hàng gọi điện yêu cầu giao hàng đến trung tâm điều khiển, nhân viên điều phối tiếp nhận thông tin và lựa chọn tài xế giao hàng phù hợp, và thiết lập lại hành trình giao hàng cho tài xế này.
- Trong suốt quá trình giao hàng, tài xế sẽ phải cập nhật thông tin trạng thái khách hàng về trung tâm điều khiển sau mỗi lần giao hàng xong.
- Mọi thông tin tài xế gửi sẽ được lưu trữ, cập nhật trong cơ sở dữ liệu. Nhân viên điều phối có thể theo dõi vị trí, trạng thái của xe qua bản đồ. Bản đồ sẽ cập nhật liên tục trạng thái, vị trí của các xe đang hoạt động giao hàng trong hệ thống hỗ trợ giao hàng.

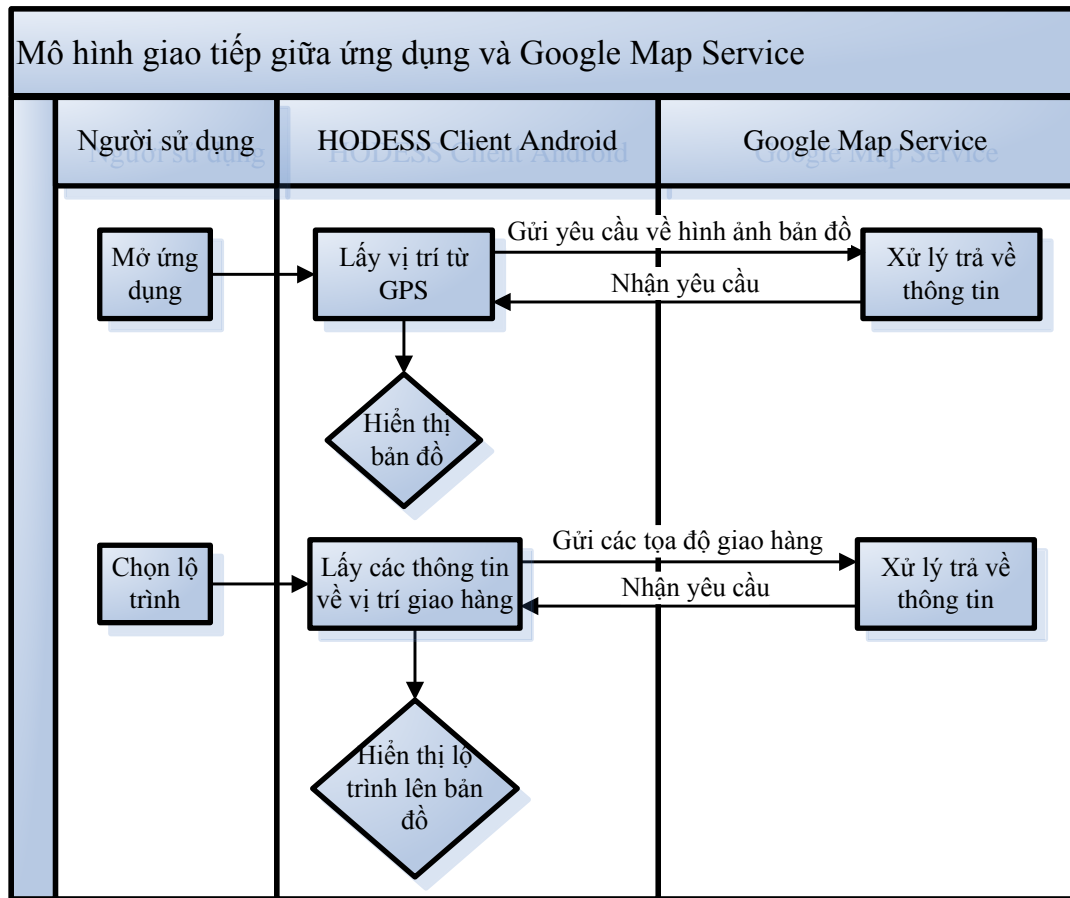
Định dạng dữ liệu sử dụng trong các Web Service này là json.

2.6. Mô tả ứng dụng Client trên Android

Ứng dụng được kết nối đến 2 server là Google Map Service và HODESS Web Service. Vì vậy, thiết bị cần phải được kết nối Internet thì sẽ hoạt động được.

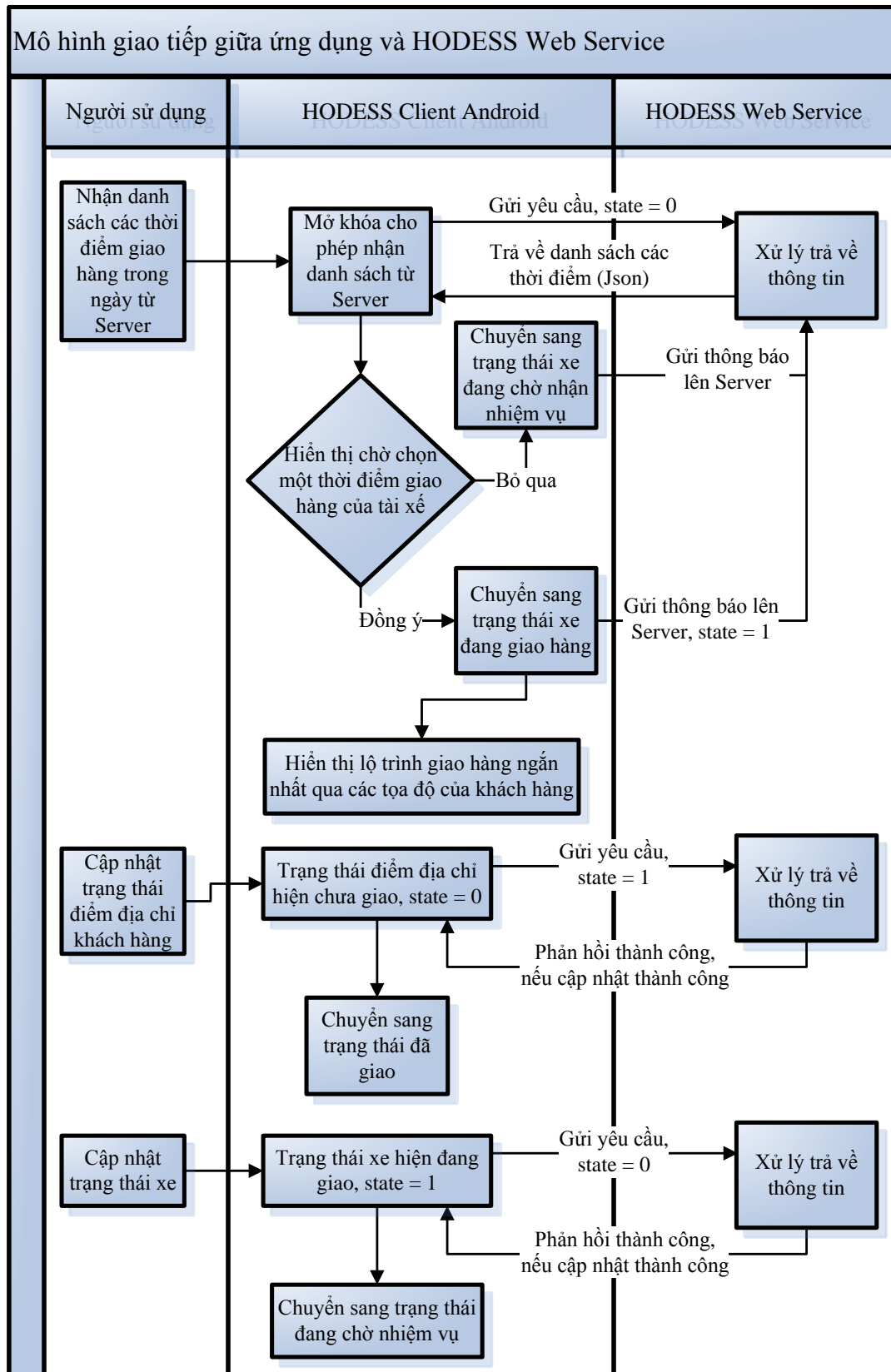
Quá trình xử lý của hai web service hoạt động sau:

- Google Map Service cung cấp tất cả dữ liệu liên quan để hiển thị lên bản đồ, các chức năng đi kèm như: hiển thị hành trình, chỉ đường... Ta có thể tóm lược qua quy trình trong hình 2.8.



Hình 2.8: Quy trình giao tiếp giữa ứng dụng và Google Map Service

- HODESS Web Service cung cấp các dữ liệu bao gồm: danh sách các thời điểm giao hàng trong ngày kèm theo danh sách khách hàng, hành trình đường đi ngắn nhất đã được thiết lập sẵn cho từng thời điểm, trạng thái xe, trạng thái các điểm địa chỉ giao hàng. Ngoài ra, ứng dụng gửi liên tục thông tin về vị trí, trạng thái đến server. Ta có thể tóm lược qua quy trình trong hình 2.9.



Hình 2.9: Quy trình giao tiếp giữa ứng dụng và HODESS Web Service

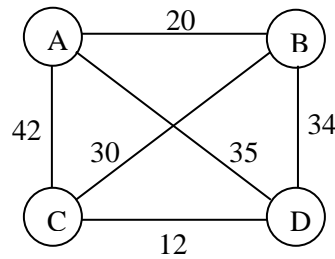
2.7. Giải pháp tìm hành trình giao hàng ngắn nhất

Hệ thống hỗ trợ giao hàng HODESS đặt yêu cầu tìm hành trình giao hàng ngắn nhất dựa trên bài toán người du lịch – Traveling Salesman problem (TSP) [11]. Đây là một bài toán khá nổi tiếng trong lĩnh vực tối ưu hóa tổ hợp được nghiên cứu trong lý thuyết khoa học máy tính. Nội dung của bài toán đơn giản được phát biểu như sau: Cho một danh sách các thành phố và khoảng cách giữa chúng, nhiệm vụ là tìm đường đi ngắn nhất có thể mà chỉ thăm mỗi thành phố đúng 1 lần. Bài toán lần đầu tiên đưa ra như một vấn đề toán học vào năm 1930 và là một trong số những bài toán được nghiên cứu chuyên sâu trong lĩnh vực tổ hợp thời đó. Nó được sử dụng như một sự đánh giá cho nhiều phương thức tối ưu khác nhau. Thậm chí bài toán thuộc lớp NP-khó (NP-hard), một lượng rất lớn các heuristic và phương thức tìm kiếm cụ thể đã được biết đến vì vậy một vài trường hợp của bài toán với khoảng chục nghìn thành phố đã được giải quyết.

Trong ứng dụng của hệ thống hỗ trợ giao hàng HODESS, khái niệm thành phố sẽ thay đổi thành địa chỉ khách hàng cần giao hàng trong khu vực thành phố.

Trong lý thuyết của độ phức tạp tính toán, phiên bản quyết định của bài toán TSP thuộc lớp NP-đầy đủ (NP-complete). Vì vậy, không có giải thuật hiệu quả nào cho việc giải bài toán TSP. Điều này có nghĩa giống như thời gian chạy tồi nhất cho bất kỳ giải thuật nào cho bài toán TSP tăng theo hàm mũ với số lượng thành phố, vì vậy thậm chí nhiều trường hợp với vài trăm thành phố cũng đã mất vài năm CPU để cho ra lời giải chính xác.

TSP có thể được mô tả như đồ thị bao gồm các đỉnh tương ứng như các thành phố, các cạnh tương ứng đường đi giữa các thành phố, độ dài cạnh tương ứng với khoảng cách giữa hai thành phố (Hình 2.10). Đường đi trong bài toán TSP là chu trình Hamilton trên đồ thị và lời giải tối ưu của bài toán là chu trình Hamilton ngắn nhất.



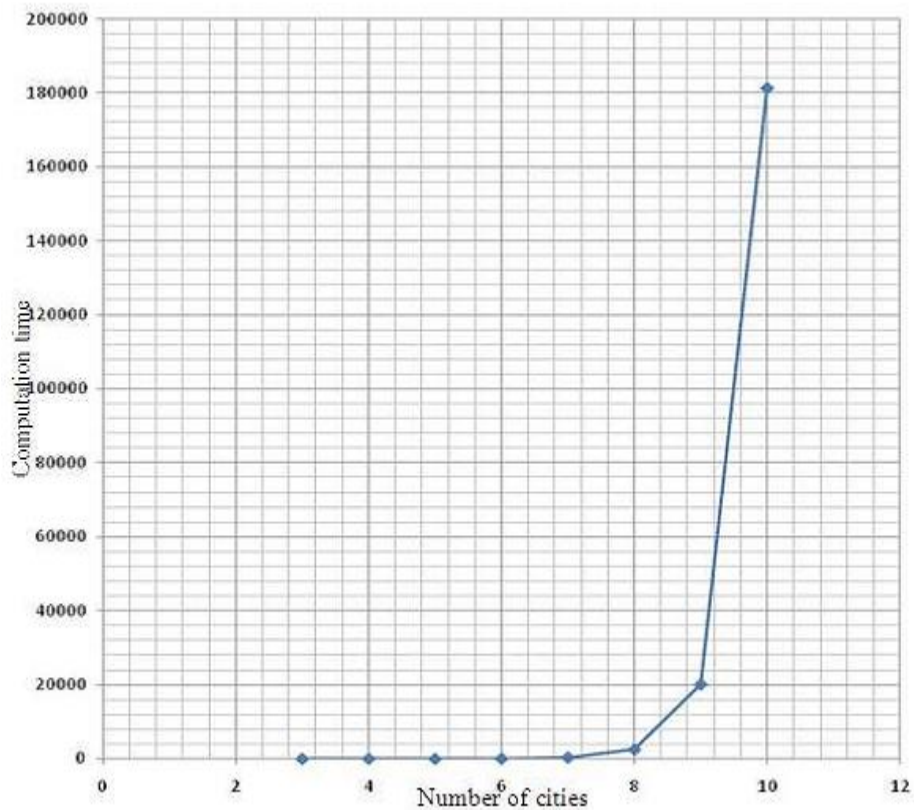
Hình 2.10: Mô tả bài toán TSP dạng đồ thị

Thường thì đồ thị là đồ thị đầy đủ, vì vậy mọi cặp cạnh đều được nối bởi các cạnh. Đây là bước đơn giản hóa bài toán vì việc tìm chu trình Hamilton trong một đồ thị đầy đủ là dễ. Các bài toán mà không phải 2 thành phố nào cũng được nối với nhau có thể được chuyển đổi thành đồ thị đầy đủ bằng cách thêm những cạnh có độ dài lớn giữa các thành phố này, các cạnh sẽ không xuất hiện trong chu trình tối ưu.

Một tính toán nhanh chóng cho thấy độ phức tạp của bài toán TSP là $O((n-1)!/2)$, với n là số thành phố (Bảng 2.1 và hình 2.11)

Bảng 2.1: Số khả năng và thời gian tính toán dựa trên số lượng thành phố [2]

Number of cities	Number of possibilities	Computation time
5	12	12 μ s
10	181440	0,18 ms
15	43 billions	12 hours
20	60 E+15	1928 years
25	310 E+21	9,8 billions of years



Hình 2.11: Sự bùng nổ nhanh chóng của bài toán tổ hợp TSP [2]

Trong ứng dụng của hệ thống hỗ trợ giao hàng HODESS, mỗi tài xế giao hàng được điều phối cho một lượt giao hàng không quá 20 địa chỉ, nên cần có giải pháp tối ưu về mặt thời gian hơn cho lời giải chính xác. Theo như biểu đồ hình 2.11, hệ thống hỗ trợ giao hàng không thể sử dụng phép tính hoán vị để đưa ra lời giải chính xác hơn 10 thành phố trong khoảng thời gian cho phép. Vấn đề đặt ra là ta sẽ sử dụng một thuật toán nào khác để thực hiện việc tìm hành trình ngắn nhất như đã đề cập trước đó, và thuật toán nhánh cận (BnB) là giải pháp được chọn. Ta sẽ xem qua phân tích của thuật toán BnB và các giải thuật tìm kiếm xấp xỉ khác với lượng lớn các thành phố trong định hướng phát triển sau này của đề tài ở chương sau.

2.8. Giải pháp VoIP [1]

Trong quá trình quản lý và điều phối phương tiện chuyên chở giao hàng, quá trình liên lạc giữa tài xế và nhân viên điều phối được thực hiện thường xuyên. Thông thường, việc liên lạc giữa trung tâm điều khiển và tài xế được thực hiện thông qua bộ đàm, việc này sẽ gây tốn kém chi phí trong việc lắp đặt cũng như bảo

trì hệ thống. Ngoài ra, sử dụng bộ đàm gặp khó khăn khi liên lạc kiểu một gọi một, gọi theo nhóm chỉ dựa trên từng khu vực cụ thể. Trong hệ thống hỗ trợ giao hàng này, em sẽ đưa ra giải pháp thay thế việc sử dụng bộ đàm bằng cách triển khai một tổng đài VoIP để nhân viên điều phối và tài xế thực hiện việc liên lạc. Việc triển khai tổng đài này có những lợi ích sau:

- Chi phí thấp vì là tổng đài mã nguồn mở. Không cần phải trang bị hệ thống bộ đàm trên mỗi phương tiện chuyên chở, vì Client hoạt động dựa trên Smartphone có thể thực hiện cuộc gọi thông qua tổng đài VoIP.
- Cung cấp đầu số cho mỗi nhân viên trong hệ thống hỗ trợ giao hàng giúp việc liên lạc nội bộ, từng cá nhân gọi nhau dễ dàng hơn.
- Nhân viên điều phối có thể họp nhóm các tài xế trong cùng khu vực để đề nghị tham gia vào phòng hội thoại.

2.9. Kết luận

Thông qua chương này, ta đã rõ được phần nào quá trình hoạt động của hệ thống giao hàng tận nhà HODESS sử dụng công nghệ GPS dựa trên những phân tích tổng quan hệ thống, mô hình hoạt động, đưa ra hành trình giao hàng ngắn nhất dựa trên bài toán TSP, kỹ thuật thoại VoIP trên mạng 3G.

Trong chương kế tiếp, em sẽ giới thiệu lịch sử của bài toán TSP cổ điển, các giải thuật liên quan, và việc thiết lập hành trình ngắn nhất trong hệ thống hỗ trợ giao hàng HODESS từ quá trình thiết lập danh sách khoảng cách giữa các cặp địa chỉ giao hàng dựa vào API tính khoảng cách giữa 2 tọa độ của Google Maps cung cấp đến áp dụng thuật toán giải bài toán TSP.

CHƯƠNG 3. BÀI TOÁN TRAVELING SALESMAN PROBLEM

Trong chương này, em sẽ giới thiệu lịch sử và một số heuristic của bài toán TSP cổ điển. Sau đó, phân tích và xây dựng chức năng tìm hành trình giao hàng ngắn nhất trong hệ thống hỗ trợ giao hàng dựa trên bài toán này.

3.1. Nguồn gốc bài toán

Bài toán Traveling salesman problem (TSP) là một bài toán tối ưu hóa tổ hợp. Các giải pháp tối ưu cho trường hợp nhỏ có thể được ra trong thời gian nhất định bằng lập trình tuyến tính. Tuy nhiên, TSP là bài toán dạng NP-khó (NP-hard), và sẽ cần rất nhiều thời gian để xử lý các trường hợp lớn hơn tối ưu về mặt thời gian. Bên cạnh việc thiết lập tối ưu, có nhiều thuật toán so sánh tốc độ thời gian chạy và cho hiệu quả xấp xỉ gần tối ưu.

Bài toán TSP đã được nghiên cứu từ thế kỷ 18 bởi nhà toán học Ireland tên là William Rowan Hamilton và nhà toán học Anh tên là Thomas Penyngton Kirkman. Cuộc thảo luận tỉ mỉ trong quá trình làm việc của Hamilton và Kirkman được trích từ quyển sách có tựa là Graph Theory [9]. Điều này cho rằng dạng chung của TSP đã từng bắt đầu được nghiên cứu bởi Kalr Menger tại Vienna và Harvard. Sau đó vấn đề này được đề xướng bởi Hassler, Whitney & Merrill tại Princeton. Mô tả chi tiết về sự kết hợp giữa Menger & Whitney và sự phát triển của TSP đã được tìm ra [16].

3.2. Định nghĩa

Ta có thể định nghĩa TSP như sau: Cho một tập các thành phố và chi phí hành trình (hoặc khoảng cách) giữa mỗi cặp điểm, TSP tìm ra đường đi tốt nhất thăm tất cả các thành phố và quay trở về điểm bắt đầu sao cho tổng chi phí (hoặc khoảng cách) của hành trình là thấp nhất [10].

3.3. Độ phức tạp [10]

Cho n là số các thành phố được thăm, tổng số các tuyến khả thi chứa tất cả thành phố đã cho trong tập các giải pháp có thể thực hiện được của TSP là $(n-1)!/2$.

3.4. Xấp xỉ [10]

Xử lý TSP tối ưu mất nhiều thời gian, thay vì thông thường sử dụng các thuật toán xấp xỉ hoặc các heuristic. Điều khác biệt là các thuật toán xấp xỉ cho ta một sự đảm bảo để so sánh các giải pháp mà ta có. Thông thường được đo bằng c lần giá trị tối ưu. Thuật toán xấp xỉ tốt nhất đã được công bố là của Sanjeev Arora [4]. Thuật toán này cho rằng sự xấp xỉ là $(1+1/c)$ với mọi $c > 1$. Mặc dù theo lý thuyết c có thể rất lớn, nó sẽ có ảnh hưởng ngược lại so với thời gian chạy của nó ($O(n(\log_2 n)^{O(c)})$ đối với các trường hợp bài toán 2 chiều).

3.5. Xây dựng hành trình [13]

Các thuật toán xây dựng hành trình đều có một điểm chung là chúng dừng lại khi có giải pháp được tìm thấy và sẽ không còn giải pháp nào khác được cải thiện hơn. Các thuật toán xây dựng hành trình tốt nhất thường nằm trong khoảng 10-15% tối ưu.

Các dạng thuật toán xây dựng gồm:

- Thuật toán láng giềng gần nhất (Nearest Neighbor): Thuật toán này có thể là TSP đơn giản và dễ thực hiện nhất. Chìa khóa cho thuật toán này là luôn luôn thăm các thành phố gần nhất. Độ phức tạp của Nearest Neighbor là $O(n^2)$.
- Thuật toán tham lam (Greedy): Heuristic thuật toán tham lam (Greedy) từng bước một xây dựng một hành trình bằng cách liên tục lựa chọn các cạnh ngắn nhất và thêm nó vào các hành trình miễn là nó không tạo ra một chu trình có ít hơn N cạnh, hoặc làm tăng bậc của bất kỳ nút nào lớn hơn 2. Độ phức tạp của Greedy là $O(n^2 \log_2(n))$.
- Thuật toán chèn (Insertion): Cơ bản của phương pháp chèn là bắt đầu từ một hành trình trong tập con của tất cả thành phố, và sau đó chèn phần còn lại bằng một số phương pháp. Hành trình con ban đầu thường là một dạng tam giác hoặc bao lồi. Cách khác cũng có thể bắt đầu bằng một cạnh đơn đặt là

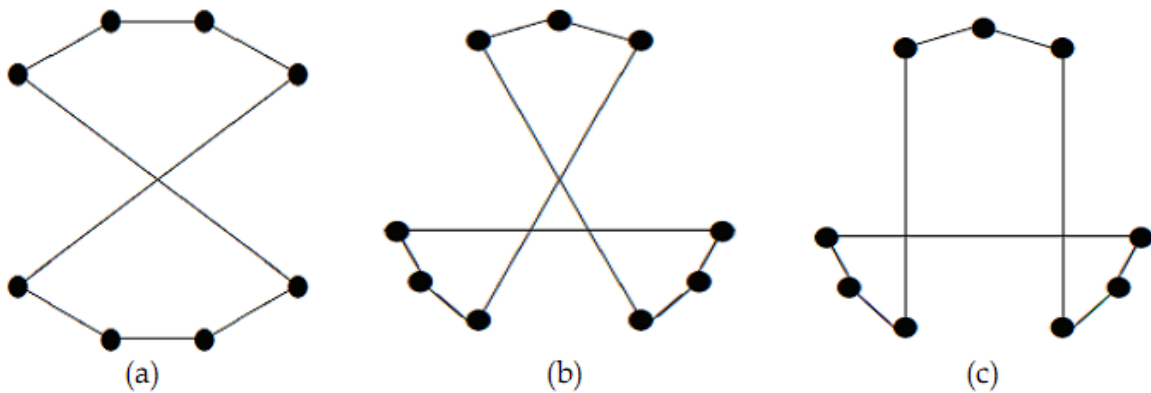
hành trình con. Độ phức tạp chèn gần nhất là $O(n^2)$ và chèn bao lỗi là $O(n^2 \log_2(n))$.

- Thuật toán Christofides: Giáo sư Nicos Christofides đã mở rộng một trong các thuật toán và kết luận rằng tỉ lệ trường hợp xấu trong thuật toán mở rộng là $3/2$. Thuật toán này có độ phức tạp là $O(n^3)$.

3.6. Cải thiện hành trình [13]

Một khi hành trình đã được tạo ra bằng một số heuristic xây dựng hành trình, có lẽ ta mong muốn cải thiện giải pháp đó. Có nhiều cách để thực hiện việc này, nhưng hầu hết đều sử dụng phương pháp tìm kiếm cục bộ 2-opt và 3-opt. Việc thực thi của chúng có phần nào liên quan đến heuristic xây dựng được sử dụng.

Cách khác để cải thiện giải pháp là thực hiện kiểu tìm kiếm tabu sử dụng chuyển đổi 2-opt và chuyển đổi 3-opt. Thuật toán mô phỏng luyện kim (SA) cũng sử dụng các kiểu chuyển đổi này để tìm giải pháp lân cận. Thuật toán di truyền (GA) thường sử dụng chuyển cặp 2-opt như một phương tiện trong biến đổi dân số.



Hình 3.1: Phương pháp chuyển đổi 2-opt và 3-opt

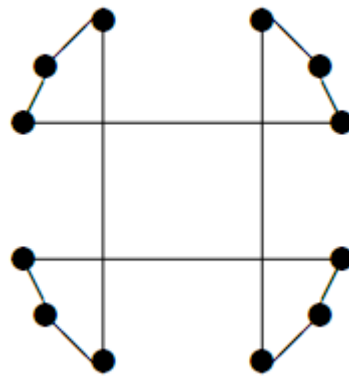
Thuật toán 2-opt về cơ bản loại bỏ 2 cạnh từ hành trình, và nối lại 2 đường đã tạo ra. Điều này thường liên quan như chuyển cặp 2-opt. Chỉ có một cách để nối lại 2 đường sao cho ta vẫn có một hành trình có giá trị (Hình 3.1(a)). Ta chỉ có thể làm được điều này nếu hành trình mới ngắn hơn. Tiếp tục loại bỏ và kết nối lại hành trình cho đến khi không còn cải thiện 2-opt nào có thể được tìm thấy. Hành trình bây giờ là 2-optimal.

Thuật toán 3-opt làm việc trong một cấu trúc tương tự, nhưng thay vì loại bỏ 2 cạnh ta sẽ loại bỏ 3 cạnh. Điều này có nghĩa là ta có 2 cách nối lại 3 đường thành một hành trình có giá trị (Không bao gồm kết nối giống như chuyển cặp 2-opt đơn giản) (Hình 3.1 (b) và hình 3.1 (c)). Một kiểu chuyển đổi 3-opt thực sự có thể được xem như là hai hoặc ba chuyển cặp 2-opt.

Ta hoàn thành tìm kiếm khi không còn chuyển đổi 3-opt nào có thể cải thiện hành trình tốt hơn. Nếu một hành trình là 3-optimal thì nó cũng là 2-optimal.

Nếu ta xem hành trình như một hoán vị tất cả các thành phố thì chuyển cặp 2-opt sẽ cho kết quả bằng việc xáo trộn thứ tự một đoạn nào đó trong hoán vị. Chuyển đổi 3-opt có thể được xem như là 2 hoặc 3 đoạn xáo trộn đó.

k-opt: Ta không nhất thiết chỉ dừng ở chuyển đổi 3-opt, ta có thể tiếp tục với chuyển đổi 4-opt và nhiều hơn thế, nhưng mỗi kiểu trong số đó sẽ cần ngày càng nhiều thời gian hơn và sẽ chỉ mang lại một ít cải thiện so với kiểu heuristic 2-opt và 3-opt. Phần lớn chuyển đổi 4-opt được sử dụng được gọi là “the crossing bridges” (Hình 3.2). Chuyển đổi đặc biệt này không thể được xây dựng một cách tuần tự bằng chuyển đổi 2-opt. Đối với việc này để có được hai trong số các kiểu chuyển đổi khả thi thì phải trái luật [7].



Hình 3.2: Phương pháp chuyển đổi double-bridge

3.7. Một số giải thuật tiêu biểu

3.7.1. *Lin-Kernighan*

Thuật toán Lin-Kernighan (LK) là một thuật toán k-opt biến số. Nó quyết định k là thích hợp nhất tại mỗi bước lặp. Điều này làm thuật toán trở nên phức tạp, và vài trong số đó đã có thể được cải thiện.

Độ phức tạp về mặt thời gian của LK xấp xỉ là $O(n^{2.2})$ làm cho nó chậm hơn so với cải thiện của 2-opt đơn giản. Tuy nhiên, kết quả của LK thì tốt hơn nhiều, và đã đưa ra một đề xuất bởi Helsgaun [7].

3.7.2. *Tabu-Search* [13]

Thuật toán tìm kiếm vùng lân cận (neighborhood-search) tìm kiếm giữa các láng giềng của một tập được chọn để tìm ra cái tốt nhất. Khi chạy neighborhood-search trên TSP, chuyển đổi láng giềng thường là các kiểu chuyển đổi 2-opt.

Vấn đề với neighborhood-search là một trong những vấn đề dễ dàng gặp phải cản trở trong tối ưu cục bộ. Điều này có thể tránh được bằng cách sử dụng tabu-search.

Tabu-search sẽ chấp nhận chuyển đổi với lời giải bị cấm nếu ta không thể tìm ra một kết quả nào khả thi. Bằng việc chấp nhận lời giải bị cấm ta có thể kết thúc quá trình chạy lặp vòng như một chuyển đổi có thể kháng lại trước đó. Để tránh mắc phải điều này thì tabu-search giữ một danh sách tabu (tabu-list) bao gồm các chuyển đổi trái luật. Sau khi đến một giải pháp lân cận, chuyển đổi sẽ được đặt vào tabu-list và sau đó sẽ không bao giờ được áp dụng lần nữa nếu nó không có cải thiện hành trình tốt nhất hoặc tabu đã được xén bớt từ danh sách.

Một vấn đề lớn với tìm kiếm tabu là thời gian chạy của nó. Hầu hết các trường hợp cho TSP sẽ mất $O(n^3)$, làm nó chậm hơn nhiều so với tìm kiếm cục bộ 2-opt. Cho rằng ta sử dụng các chuyển đổi 2-opt thì độ dài của hành trình sẽ tốt hơn được phần nào so với chuẩn tìm kiếm 2-opt.

3.7.3. *Simulated Annealing*

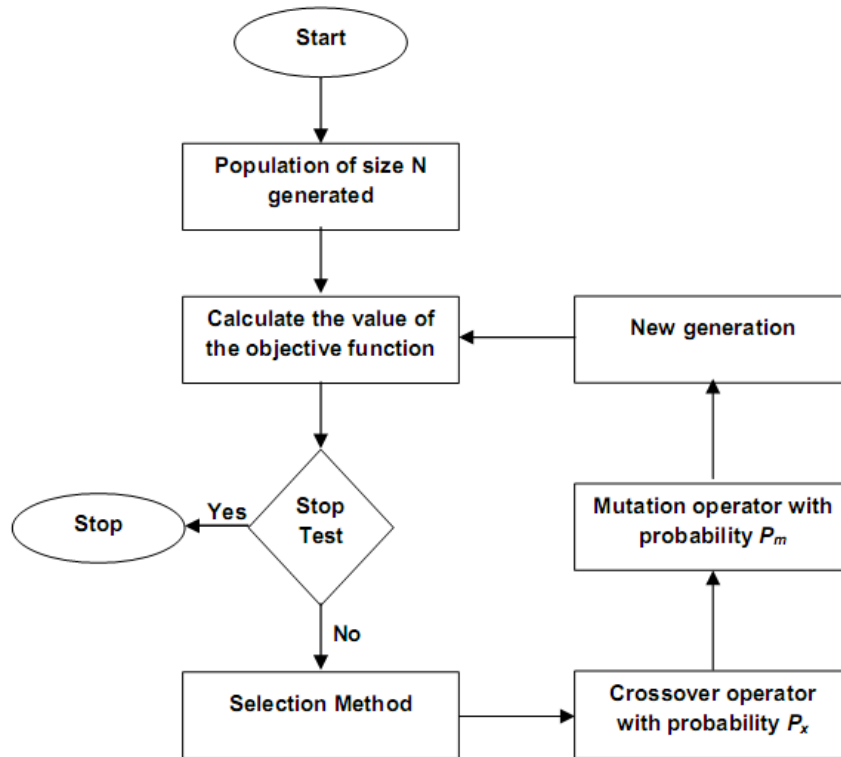
Giải thuật mô phỏng luyện kim (SA) chọn một bước chuyển đổi ngẫu nhiên trong quá trình tìm kiếm và xem xét: nếu bước chuyển này làm cho hàm chi phí tốt

hơn so với trạng thái hiện hành thì bước chuyển này sẽ được chọn. Nếu bước chuyển này có hàm chi phí xấu hơn thì có thể được chấp nhận với một xác suất nào đó.

Mô phỏng luyện kim đã được chuyển thể thành công để đưa ra các giải pháp xấp xỉ cho TSP.

3.7.4. Genetic Algorithm

Giải thuật di truyền (GA) là một kỹ thuật của khoa học máy tính nhằm tìm kiếm giải pháp thích hợp cho các bài toán tối ưu tổ hợp. Giải thuật di truyền là một phân ngành của giải thuật tiến hóa vận dụng các nguyên lý của tiến hóa như di truyền, đột biến, chọn lọc tự nhiên, trao đổi chéo. Giải thuật di truyền thường được áp dụng nhằm sử dụng ngôn ngữ máy tính để mô phỏng quá trình tiến hóa của một tập hợp những đại diện trừu tượng (các nhiễm sắc thể) của các giải pháp có thể (các cá thể) cho bài toán tối ưu hóa vấn đề. Tập hợp này sẽ tiến triển theo hướng chọn lọc những giải pháp tốt hơn. Thông thường các giải pháp được thể hiện dưới dạng nhị phân, mang nhiều thông tin được mã hóa khác nhau. Quá trình tiến hóa xảy ra từ một tập các cá thể hoàn toàn ngẫu nhiên ở tất cả các thế hệ. Trong từng thế hệ, tính thích nghi của tập hợp này được ước lượng, nhiều cá thể được chọn lọc định hướng từ tập hợp hiện thời (dựa vào thể trạng), được sửa đổi (bằng đột biến hoặc tổ hợp lại) để hình thành một tập hợp mới. Tập hợp này sẽ tiếp tục được chọn lọc lặp đi lặp lại trong các thế hệ kế tiếp của giải thuật. Giải thuật được mô tả trong hình 3.3



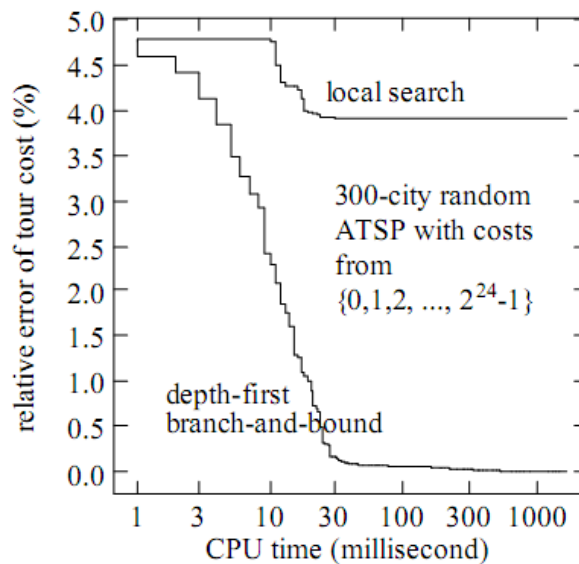
Hình 3.3: Mô tả giải thuật di truyền [2]

3.7.5. Branch and Bound

Các giải thuật nhánh cận (BnB) thường được sử dụng để tìm kiếm giải pháp tối ưu cho bài toán tối ưu hóa tổ hợp [22][23]. Phương thức này có thể dễ dàng được áp dụng cho TSP đối xứng (STSP) và kể cả TSP bất đối xứng (ATSP). Một phương thức để giải quyết ATSP là sử dụng giải thuật Depth-First Branch & Bound (DFBnB) [24]. DFBnB bắt đầu với ATSP gốc và giải quyết bài toán phân công (AP). Bài toán phân công là để kết nối mỗi thành phố với thành phố gần nhất của nó sao cho tổng chi phí của tất cả các kết nối là tối thiểu.

Ta có một giải pháp tối ưu được tìm ra đối với ATSP nếu giải pháp này là một hành trình hoàn thiện. Nếu giải pháp không phải là hành trình hoàn thiện thì ta cần phải tìm một hành trình con trong giải pháp của AP và loại trừ các cạnh từ đó. Mỗi phép loại trừ sẽ là một nhánh tìm kiếm. Thực tế việc cắt xén này được thực hiện bằng việc chọn chiều dài α đầu tiên trong giải pháp tốt nhất hiện tại được biết đến. Tất cả các nhánh được cắt nếu chi phí của giải pháp AP loại trừ α .

Depth-first branch-and-bound (DFBnB) là thuật toán đầy đủ, là loại thuật toán được sử dụng để việc tìm các giải pháp tối ưu trong các bài toán tối ưu tổ hợp khó. Tuy nhiên, nó cũng có thể được sử dụng như một thuật toán xấp xỉ. Ta so sánh DFBnB dựa vào thuật toán tìm kiếm cục bộ Kanellakis-Papadimitriou, thuật toán xấp xỉ được biết đến là tốt nhất trong ATSP, một bài toán quan trọng dạng NP-khó. Các kết quả thực nghiệm cho thấy rằng DFBnB làm tốt đáng kể hơn tìm kiếm cục bộ trên bài toán lớn ATSP và các cấu trúc ATSP khác nhau (Hình 3.4), DFBnB tìm ra các giải pháp tốt hơn nhanh hơn so với tìm kiếm cục bộ [24].



Hình 3.4: Kết quả giữa DFBnB và tìm kiếm cục bộ với 300 thành phố ATSP ngẫu nhiên [24]

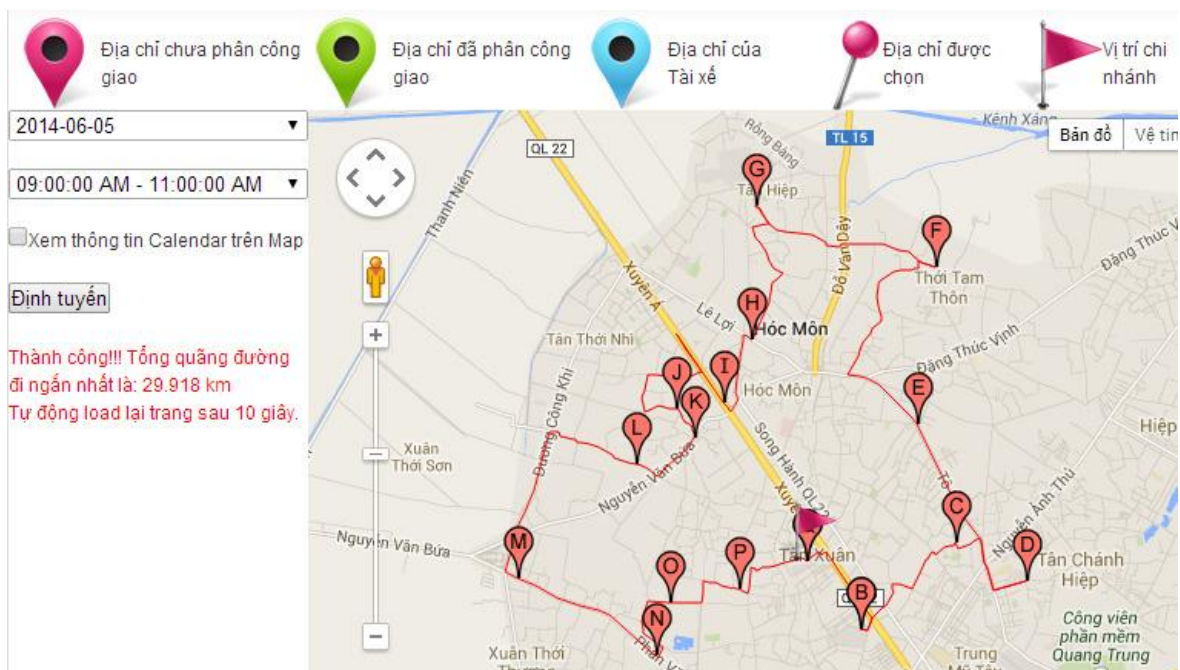
3.7.6. *Ant colony optimization*

Các thuật toán bầy kiến (ACO) lần đầu tiên được giới thiệu bởi Dorigo và các cộng sự như là cách tiếp cận đa tác tử tới các vấn đề về tối ưu tổ hợp dạng NP-khó như bài toán TSP. Các thuật toán bầy kiến dựa vào sự quan sát các bầy kiến thực. Kiến giao tiếp với nhau thông qua mùi của chúng để lại trên hành trình mà chúng đi qua. Mỗi con khi đi qua một đoạn đường sẽ để lại trên đó một chất gọi là mùi. Số lượng mùi sẽ tăng lên khi có nhiều con cùng đi qua. Các con khác sẽ tìm đường dựa vào mật độ mùi trên đường, mật độ mùi càng lớn thì chúng càng có xu

hướng chọn nhiều. Dựa theo cách này, đàn kiến sẽ tìm được đường đi ngắn nhất từ tổ đến nguồn thức ăn và sau đó quay trở về. Ý tưởng này đơn giản đã hoàn thành khi áp dụng cho bài toán TSP.

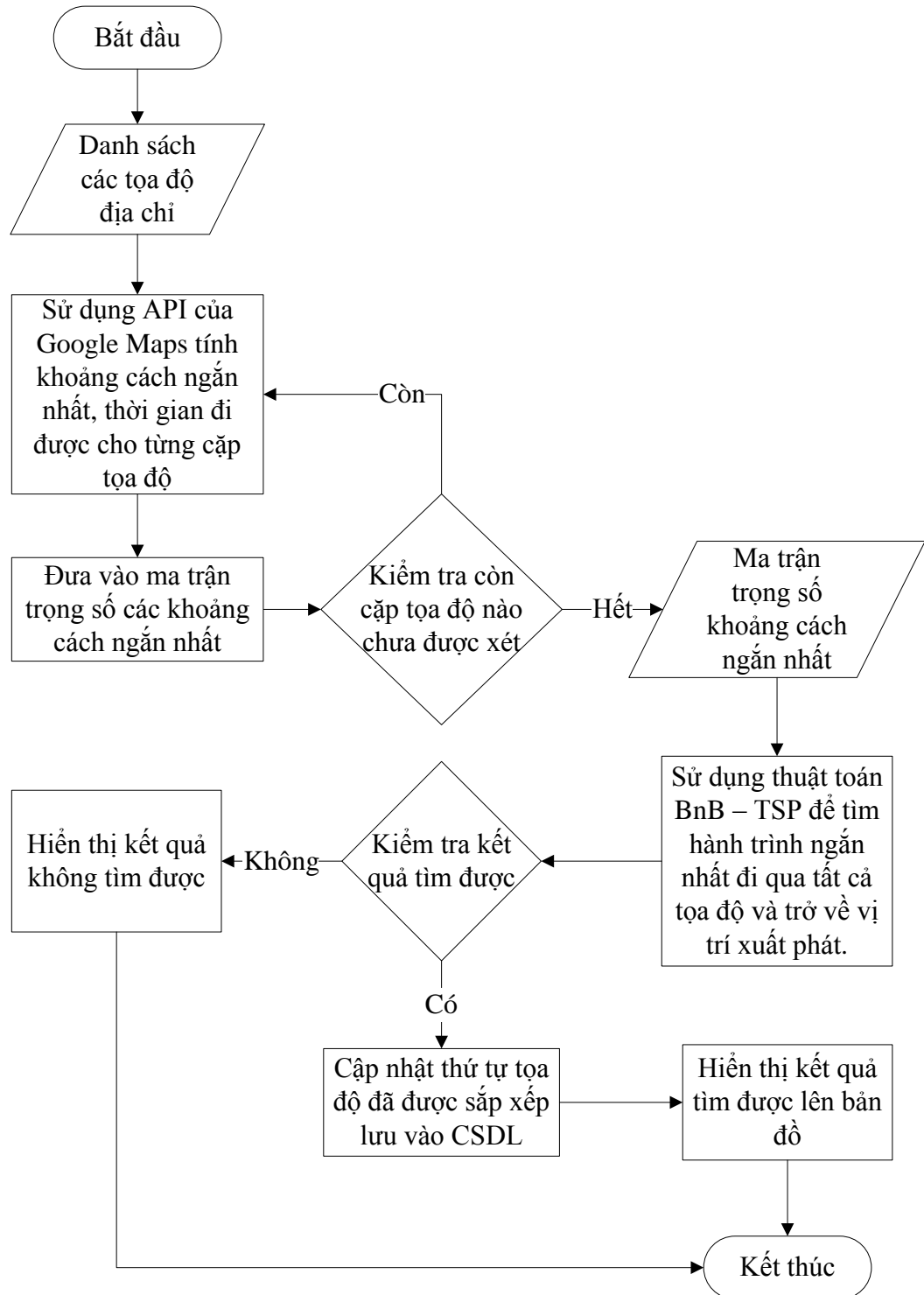
3.8. Xây dựng chức năng tìm hành trình giao hàng ngắn nhất

Qua phân tích các giải thuật cho bài toán TSP, ta sẽ xây dựng chức năng tìm hành trình giao hàng ngắn nhất trong hệ thống hỗ trợ giao hàng HODESS. Trong Google Maps chỉ cung cấp một API tìm đường đi ngắn nhất giữa 2 tọa độ địa chỉ [19]. Trên cơ sở đó, ta xây dựng chức năng tìm hành trình qua tất cả tọa độ địa chỉ sao cho quãng đường đi được là ngắn nhất. Ban đầu, ta sẽ sử dụng API của Google Maps để tìm khoảng cách ngắn nhất giữa từng cặp tọa độ trong danh sách các tọa độ địa chỉ được chọn, khoảng cách của Google ta nhắc đến có nhiều loại chẳng hạn như: khoảng cách đường đi của người đi bộ, của xe máy, của ô tô, xe buýt, nhưng trong chức năng tìm hành trình giao hàng của hệ thống hỗ trợ giao hàng này chỉ sử dụng khoảng cách đường đi của ô tô. Sau mỗi lần tính được khoảng cách cho mỗi cặp tọa độ, hệ thống sẽ đưa vào ma trận trọng số. Sau khi thiết lập ma trận trọng số khoảng cách ngắn nhất xong, ta bắt đầu áp dụng thuật toán BnB giải bài toán TSP để tìm hành trình ngắn nhất qua tất cả các tọa độ địa chỉ (Hình 3.5).



Hình 3.5: Kết quả tìm được hành trình ngắn nhất sử dụng thuật toán BnB của TSP

Quy trình xử lý của chức năng tìm đường trong hệ thống hỗ trợ giao hàng HODESS được tóm gọn trong mô hình hình 3.6.



Hình 3.6: Quy trình xử lý tìm hành trình ngắn nhất dựa trên Google API và thuật toán BnB của bài toán TSP

CHƯƠNG 4. HIỆN THỰC HỆ THỐNG HODESS

Qua những phân tích thiết kế hệ thống hỗ trợ giao hàng ở chương trước, chương này sẽ nêu nổi bật những công nghệ sẽ áp dụng. Đồng thời, em sẽ làm nổi bật thuật toán TSP được áp dụng cho hệ thống hỗ trợ giao hàng trong việc điều phối hành trình giao hàng cho mỗi tài xế. Cuối cùng, em sẽ trình bày thiết kế cơ sở dữ liệu, web service và mô tả hoạt động của hệ thống quản lý và ứng dụng được cài trên Smartphone.

4.1. Những công nghệ sử dụng trong quá trình xây dựng hệ thống hỗ trợ giao hàng

4.1.1. *Google Map*

Ngày nay, trên mọi hệ thống website, bản đồ là một thành phần xuất hiện khá phổ biến. Ta sử dụng chúng để hiển thị vị trí mọi địa điểm, để tìm kiếm một vị trí nào đó theo địa chỉ, để được chỉ dẫn đường đi, và ngoài ra còn nhiều tiện ích khác nữa được bản đồ hỗ trợ. Hầu hết mỗi thông tin có một vị trí xác định đều có thể được hiển thị trên bản đồ.

Hiện nay, có nhiều nhà cung cấp về giải pháp bản đồ như Yahoo! Maps, Bing Maps, nhưng sử dụng phổ biến nhất là Google Maps. Thực tế, theo như Programmableweb.com là API phổ biến nhất trên Internet, thống kê tháng 5 năm 2014, Google Maps có đến 26% các nội dung được sử dụng trong các mashup. Các ứng dụng và website được kết hợp dữ liệu hoặc chức năng từ hai hay nhiều nguồn khác nhau thông thường được xem như là mashup. Mashup ngày càng trở nên phổ biến trong các ứng dụng website

4.1.2. *Công nghệ GPS [6]*

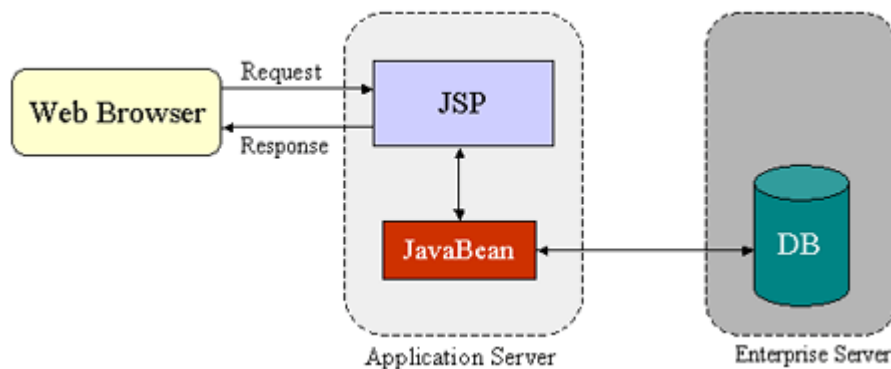
Hệ thống định vị toàn cầu GPS là một hệ thống vị tinh điều hướng được phát triển bởi bộ quốc phòng Mỹ trong những năm đầu thập niên 70. Ban đầu, GPS được phát triển như một hệ thống sử dụng trong mục đích quân sự. Tuy nhiên, thời gian sau nó được sử dụng thêm vào mục đích dân sự.

GPS cũng cấp liên tục thông tin về vị trí và thời gian bất kỳ nơi nào trên trái đất trong mọi điều kiện thời tiết. Nó không hạn chế về lượng người dùng cũng như được sử dụng cho các lý do an ninh. GPS là hệ thống một chiều thụ động có nghĩa là người dùng chỉ có thể nhận tín hiệu từ vệ tinh mà không thể gửi trả lại. Để sử dụng được GPS người dùng chỉ cần có tín hiệu thu sóng và không phải trả bất kỳ chi phí nào khác.

4.1.3. Java Server Page và Java Servlet

4.1.3.1. Java Server Page [17]

Java Server Page (JSP) là một công nghệ hỗ trợ phát sinh nội dung động vào một tài liệu HTML bằng các câu lệnh Java được nhúng vào trong một tài liệu HTML bằng các thẻ đặc biệt. Các câu lệnh Java sẽ được thực thi ở phía server.



Hình 4.1: Quy trình thực thi một trang JSP

Hai quá trình quan trọng trong thực thi một trang JSP là:

- Biên dịch: Trang JSP được biên dịch thành servlet ở lần đầu tiên phục vụ yêu cầu từ client.
- Thực thi: Servlet được thực thi ở server và kết quả thực thi được gửi về cho client (không phải nội dung của trang JSP).

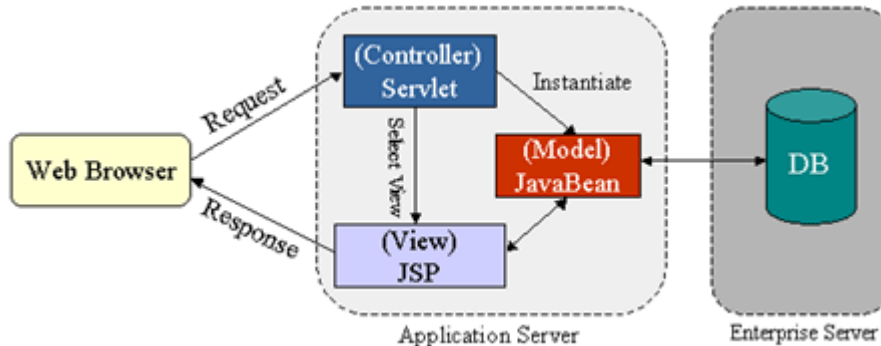
Có hai loại dữ liệu trong một trang JSP

- Dữ liệu tĩnh: được chuyển trực tiếp vào response.
- Thẻ JSP: mã lệnh được biên dịch và thực thi ở server.

4.1.3.2. Servlet [17]

Servlet là các chương trình chạy trên máy server (servlet container). Hoạt động như lớp trung gian giữa máy client và CSDL hoặc các ứng dụng khác. Servlet

có thể phát sinh các nội dung (HTML) động. Servlet không có GUI mà chỉ cài đặt những thao tác xử lý business logic.



Hình 4.2: Servlet

4.1.4. RESTful Web Service [15]

Representational State Transfer (REST) hiện nay đã được sử dụng rộng rãi trên các hệ thống Web và được coi như giải pháp thay thế cho Web Service kiến trúc SOAP và Web Service Description Language (WSDL). Các nhà cung cấp dịch vụ Web 2.0 điển hình như Google, Yahoo, Facebook đã bỏ qua giao thức dựa trên SOAP và WSDL mà tập trung vào việc sử dụng REST trong việc triển khai các dịch vụ của họ.

REST xuất phát từ luận án tiến sỹ của Roy Fielding được xuất bản năm 2000 “Architectural Styles and the Design of Network-based Software Architectures”, là viết tắt của Representational State Transfer. REST không phải là một kiến trúc mềm khi chúng được áp dụng trong việc thiết kết các hệ thống. REST chú trọng đến tài nguyên hệ thống và việc định dạng và phương thức trao đổi dữ liệu thông qua giao thức HTTP đến một số lượng lớn các client mà không phải quan tâm đến vấn đề ngôn ngữ lập trình.

4.1.5. MySQL 5.5 và Apache Tomcat 7

4.1.5.1. MySQL 5.5 [21]

MySQL là hệ quản trị cơ sở dữ liệu mã nguồn mở phổ biến nhất 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 kết 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 5.5 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 5.5 là sự kết hợp giữa cơ sở dữ liệu MySQL và InnoDB storage engine cung cấp cho ta một giải pháp quản lý dữ liệu mạnh mẽ với những ưu điểm sau:

- InnoDB storage engine là mặc định trong MySQL.
- 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 5.5, 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 em quyết định chọn MySQL để triển cơ sở dữ liệu cho hệ thống này.

4.1.5.2. Apache Tomcat 7 [20]

Apache Tomcat server được xây dựng trên mã nguồn mở trên nền tảng Java, các Java Web Application Container được tạo ra để thực thi các Servlet và Java Server Pages. Với sự phổ biến của nó hiện nay, các tình nguyện viên từ cộng đồng mã nguồn mở Java đang tích cực hỗ trợ và tăng cường phát triển.

Apache Tomcat hoạt động rất ổn định và hỗ trợ đầy đủ các tính năng của một Web Application Container. Tomcat cũng cung cấp những chức năng bổ sung khác giúp nó là một sự lựa chọn tốt để phát triển các ứng dụng Web như ứng dụng quản lý (Tomcat Manager application), Specialized Realm Implementations, Tomcat valves.

Bảng 4.1: Các phiên bản Tomcat Server

Apache Tomcat	Servlet API	JSP API	JDK
7.0	3.0	2.2	1.6
6.0	2.5	2.1	1.5
5.5	2.4	2.0	1.4
4.1	2.3	1.2	1.3
3.0	2.2	1.1	1.1

4.1.6. JSON và thư viện mã nguồn mở GSON

4.1.6.1. JSON [27]

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 dựa 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.

4.1.6.2. Thư viện mã nguồn mở GSON [28]

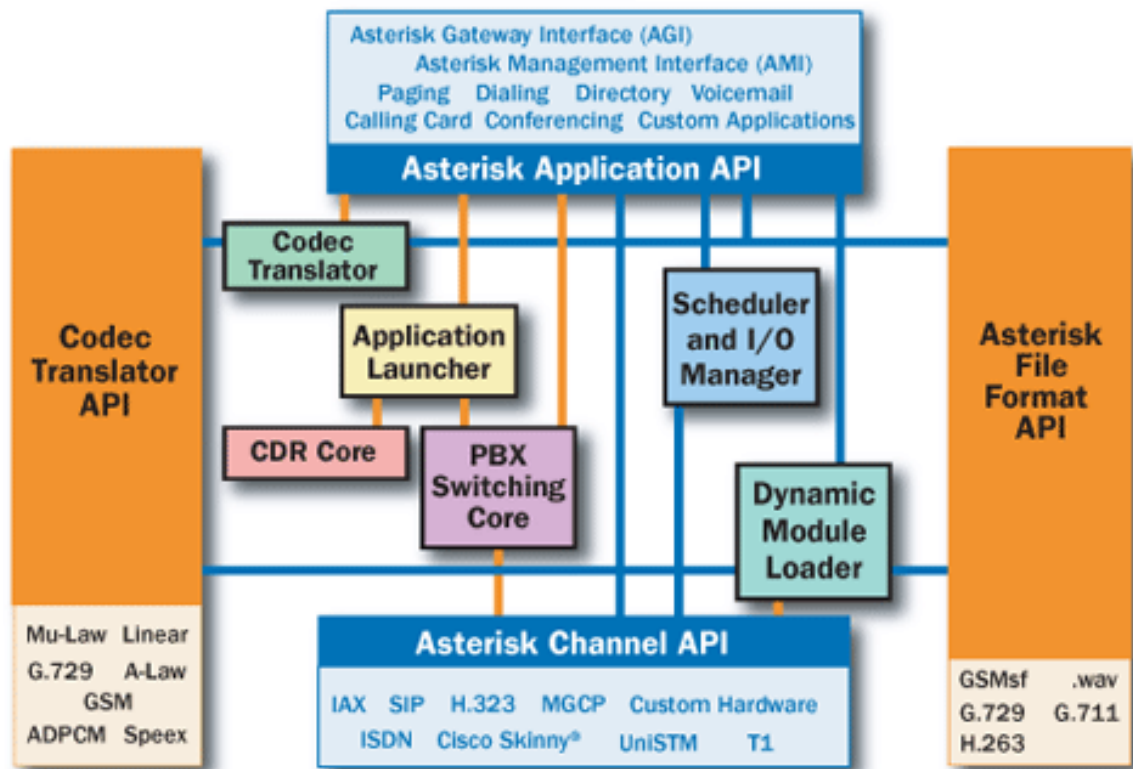
GSON là một thư viện Java được sử dụng trong việc chuyển đổi một đối tượng Java bất kỳ sang định dạng Json. GSON cũng hỗ trợ chuyển đổi từ một chuỗi định dạng Json sang đối tượng Java tương ứng. Đây là một thư viện mã nguồn mở được lưu trữ tại <http://code.google.com/p/google-gson>

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.

4.1.7. Hệ thống tổng đài Asterisk

Asterisk là hệ thống chuyển mạch mềm, phần mềm mã nguồn mở được viết bằng ngôn ngữ C chạy trên nền hệ điều hành Linux thực hiện mọi tính năng của tổng đài PBX, cho phép các máy điện thoại nhánh thực hiện cuộc gọi với nhau và kết nối các hệ thống điện thoại khác kể cả mạng điện thoại Analog thông thường (PSTN) và điện thoại IP (VoIP) [18]. Asterisk ngày càng phát triển và được rất nhiều các doanh nghiệp triển khai ứng dụng cho công ty của mình. Đơn giản các công ty đều có mạng máy tính để thực hiện cuộc gọi nội bộ, và chi phí triển khai cho hệ thống này rất thấp, thậm chí không phải tốn chi phí.



Hình 4.3: Kiến trúc tổ chức của Asterisk

Mục đích của việc sử dụng công nghệ thoại IP là:

- Tiết kiệm được chi phí so với sử dụng hệ thống điện thoại thông thường đặc biệt là khi gọi điện thoại đường dài.
- Có thể đưa vào nhiều loại dịch vụ một cách dễ dàng như: quản lý cuộc gọi, hội thoại hội nghị ...
- PBX (Private Branch Exchange) là một tổng đài nội bộ có nhiệm vụ dịch số điện thoại và quản lý tất cả các cuộc gọi ra vào trong hệ thống.

Một số đặc điểm của PBX:

- Chức năng chính của tổng đài này là chuyển mạch, phân phối cuộc gọi trong toàn hệ thống.
- Những người sử dụng PBX dùng chung một số đường điện thoại ngoài để thực hiện các cuộc gọi ra bên ngoài.
- Có các dịch vụ hướng dẫn, dịch vụ Voice Mail, hệ thống tính cước.

IP PBX là hệ thống PBX chạy bằng phần mềm thực hiện một số nhiệm vụ nhất định và cung cấp những dịch vụ thoại trên mạng IP.

4.1.8. Tổng quan về SIP

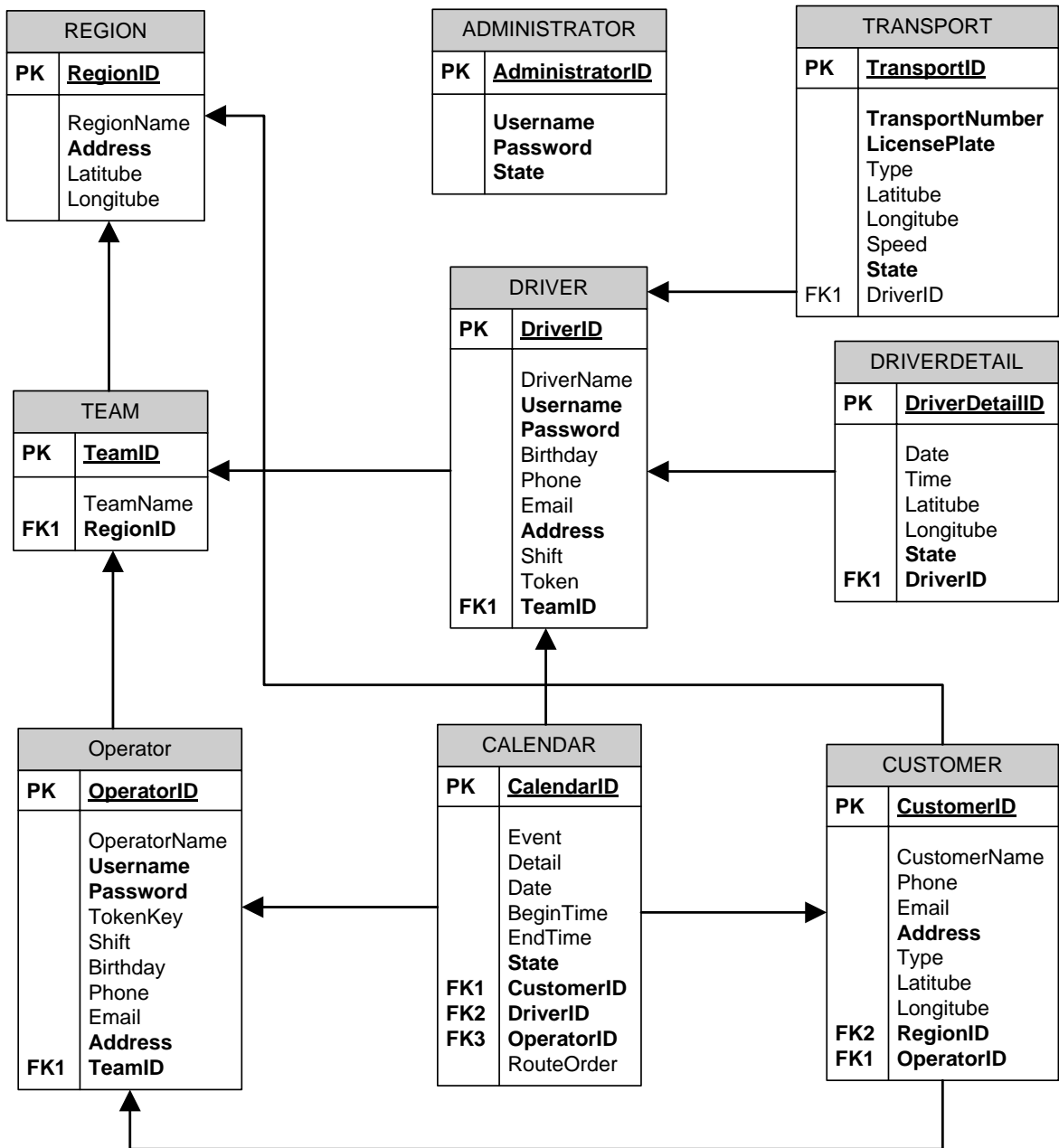
Session Initiation Protocol (SIP) là một giao thức báo hiệu lớp ứng dụng dùng để điều khiển các phiên giao tiếp như gọi thoại và gọi video thông qua internet. Giao thức này được sử dụng để tạo, chỉnh sửa, duy trì và kết thúc một hay nhiều phiên truyền thông đa phương tiện với một hay nhiều người tham gia. Các phiên truyền thông đa phương tiện này bao gồm thoại Internet, hội nghị, và các ứng dụng tương tự có liên quan đến truyền âm thanh, hình ảnh, và dữ liệu.

Sip là một giao thức thuộc tầng ứng dụng, nó có thể hoạt động trên giao thức TCP, UDP hay SCTP. SIP dựa theo ý tưởng của SMTP và HTTP. Nó được định nghĩa theo mô hình client-server trong đó bên gọi sẽ đóng vai trò là client để gửi các yêu cầu lên bên bị gọi đóng vai trò là server đáp ứng các yêu cầu này.

4.2. Xây dựng cơ sở dữ liệu

4.2.1. Mô hình dữ liệu

Dữ liệu của hệ thống hỗ trợ giao hàng HODESS trên hệ quản trị cơ sở dữ liệu MySQL được thiết kế như trong hình 4.4.



Hình 4.4: Cơ sở dữ liệu

4.2.2. Chi tiết cơ sở dữ liệu

4.2.2.1. Quan hệ ADMINISTRATOR

Quan hệ này lưu trữ thông tin chứng thực đăng nhập của người quản trị hệ thống hỗ trợ giao hàng

Bảng 4.2: Quan hệ ADMINISTRATOR

Tên thuộc tính	Diễn giải	Kiểu dữ liệu	Giá trị
<u>Username</u>	Tài khoản đăng nhập hệ thống	varchar(100)	notNull
Password	Mật khẩu đăng nhập hệ thống	varchar(100)	notNull
State	Trạng thái đăng nhập	int	notNull

4.2.2.2. Quan hệ REGION

Quan hệ này lưu trữ thông tin các vùng giao hàng trong hệ thống hỗ trợ giao hàng.

Bảng 4.3: Quan hệ REGION

Tên thuộc tính	Diễn giải	Kiểu dữ liệu	Giá trị
<u>RegionID</u>	Mã vùng giao hàng	varchar(30)	notNull
RegionName	Tên vùng giao hàng	varchar(255)	
Address	Địa chỉ vùng giao hàng	varchar(255)	notNull
Latitude	Vĩ độ của vùng giao hàng	double	notNull
Longitude	Kinh độ của vùng giao hàng	double	notNull

4.2.2.3. Quan hệ TEAM

Quan hệ này lưu trữ thông tin về các đội xe trong hệ thống hỗ trợ giao hàng.

Bảng 4.4: Quan hệ TEAM

Tên thuộc tính	Diễn giải	Kiểu dữ liệu	Giá trị
<u>TeamID</u>	Mã đội	varchar(30)	notNull
TeamName	Tên đội	varchar(255)	
RegionID	Mã vùng giao hàng	varchar(255)	notNull

4.2.2.4. Quan hệ DRIVER

Quan hệ này lưu trữ danh sách các tài xế giao hàng bao gồm thông tin cá nhân và thông tin tài khoản để đăng nhập hệ thống.

Bảng 4.5: Quan hệ DRIVER

Tên thuộc tính	Diễn giải	Kiểu dữ liệu	Giá trị
<u>DriverID</u>	Mã tài xế	varchar(30)	notNull
DriverName	Tên tài xế	varchar(255)	notNull
Username	Tài khoản đăng nhập vào hệ thống	varchar(100)	notNull
Password	Mật khẩu đăng nhập vào hệ thống	varchar(100)	notNull
Birthday	Ngày sinh tài xế	date	
Phone	Số điện thoại liên lạc tài xế	varchar(30)	
Email	Địa chỉ email liên lạc tài xế	varchar(255)	
Address	Địa chỉ tài xế	varchar(255)	notNull
Shift	Ca làm việc của tài xế	int	notNull
Token	Dấu hiệu kiểm tra trạng thái đăng nhập của tài xế	varchar(100)	notNull
TeamID	Mã đội	varchar(30)	notNull

4.2.2.5. Quan hệ DRIVERDETAIL

Quan hệ này lưu trữ thông tin lịch sử hoạt động của tài xế bao gồm ngày, giờ, tọa độ vị trí, trạng thái hoạt động của phương tiện chuyên chở của tài xế. Thông tin này được cập nhật khi tài xế đăng nhập hệ thống.

Bảng 4.6: Quan hệ DRIVERDETAIL

Tên thuộc tính	Diễn giải	Kiểu dữ liệu	Giá trị
<u>DriverDetailID</u>	Mã thông tin chi tiết tài xế	varchar(30)	notNull
Date	Ngày hoạt động của tài xế	date	notNull
Time	Thời gian hoạt động của tài xế	time	notNull
Latitude	Vĩ độ vị trí của tài xế	double	notNull

Longitude	Kinh độ vị trí của taxi	double	notNull
State	Trạng thái của taxi	int	notNull
DriverID	Mã taxi	varchar(30)	notNull

4.2.2.6. Quan hệ OPERATOR

Quan hệ này lưu trữ thông tin cá nhân, tài khoản đăng nhập hệ thống của các nhân viên điều phối.

Bảng 4.7: Quan hệ OPERATOR

Tên thuộc tính	Diễn giải	Kiểu dữ liệu	Giá trị
<u>OperatorID</u>	Mã nhân viên điều phối	varchar(30)	notNull
OperatorName	Tên nhân viên điều phối	varchar(255)	notNull
Username	Tài khoản đăng nhập vào hệ thống	varchar(100)	notNull
Password	Mật khẩu đăng nhập vào hệ thống	varchar(100)	notNull
TokenKey	Khóa dấu hiệu để giải mã mật khẩu	varchar(100)	notNull
Shift	Ca trực của nhân viên điều phối	int	notNull
Birthday	Ngày sinh	date	
Phone	Số điện thoại liên lạc	varchar(30)	
Email	Địa chỉ email liên lạc	varchar(255)	
Address	Địa chỉ	varchar(255)	notNull
TeamID	Mã đội	varchar(30)	notNull

4.2.2.7. Quan hệ TRANSPORT

Quan hệ này lưu trữ thông tin các phương tiện giao hàng của taxi. Vị trí của các xe sẽ được cập nhật liên tục khi taxi điều khiển trong quá trình giao hàng.

Bảng 4.8: Quan hệ TRANSPORT

Tên thuộc tính	Diễn giải	Kiểu dữ liệu	Giá trị
<u>TransportID</u>	Mã phương tiện	varchar(30)	notNull
TransportNumber	Số hiệu xe	varchar(100)	notNull

LicensePlate	Biển số xe	varchar(100)	notNull
Type	Loại xe	int	notNull
Laitude	Vĩ độ vị trí của xe	double	notNull
Longitude	Kinh độ vị trí của xe	double	notNull
Speed	Tốc độ di chuyển của xe	float	notNull
State	Trạng thái hoạt động của xe	int	notNull
DriverID	Mã tài xế điều khiển xe	varchar(30)	notNull

4.2.2.8. Quan hệ CUSTOMER

Quan hệ này lưu trữ thông tin của khách hàng để hỗ trợ trong quá trình giao hàng.

Bảng 4.9: Quan hệ CUSTOMER

Tên thuộc tính	Diễn giải	Kiểu dữ liệu	Giá trị
<u>CustomerID</u>	Mã khách hàng	varchar(30)	notNull
CustomerName	Tên khách hàng	varchar(255)	notNull
Phone	Số điện thoại liên lạc	varchar(30)	notNull
Email	Địa chỉ email liên lạc	varchar(255)	
Address	Địa chỉ khách hàng	varchar(255)	notNull
Type	Loại khách hàng	int	
Latitude	Vĩ độ vị trí khách hàng	double	notNull
Longitude	Kinh độ vị trí khách hàng	double	notNull
OperatorID	Mã nhân viên điều phối	varchar(30)	notNull
RegionID	Mã vùng giao hàng	varchar(30)	notNull

4.2.2.9. Quan hệ CALENDAR

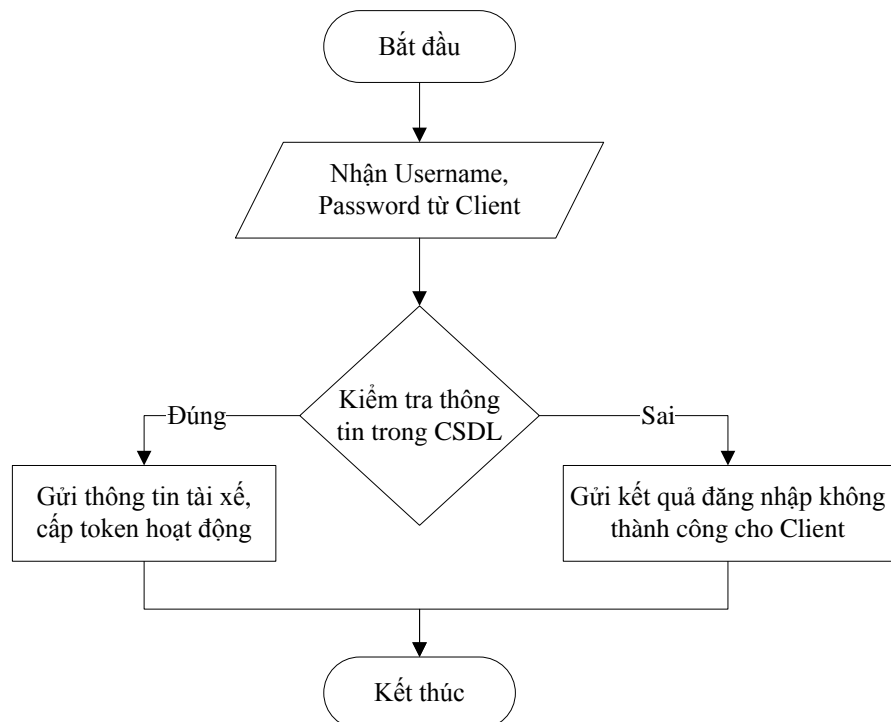
Quan hệ này lưu trữ các thông tin thời điểm giao hàng của khách hàng như ngày giờ, thời gian bắt đầu, thời gian kết thúc, trạng thái giao hàng...

Bảng 4.10: Quan hệ CALENDAR

Tên thuộc tính	Diễn giải	Kiểu dữ liệu	Giá trị
<u>CalendarID</u>	Mã lịch giao hàng	varchar(30)	notNull
Date	Ngày giao hàng	date	notNull
BeginTime	Thời gian bắt đầu giao	time	notNull
EndTime	Thời gian kết thúc giao	time	notNull
State	Trạng thái giao hàng	int	notNull
CustomerID	Mã khách hàng	varchar(30)	notNull
DriverID	Mã tài xế	varchar(30)	notNull
OperatorID	Mã nhân viên điều phối	varchar(30)	notNull
RouteOrder	Thứ tự giao hàng	int	notNull

4.3. Xây dựng Web Service

4.3.1. Chức năng đăng nhập

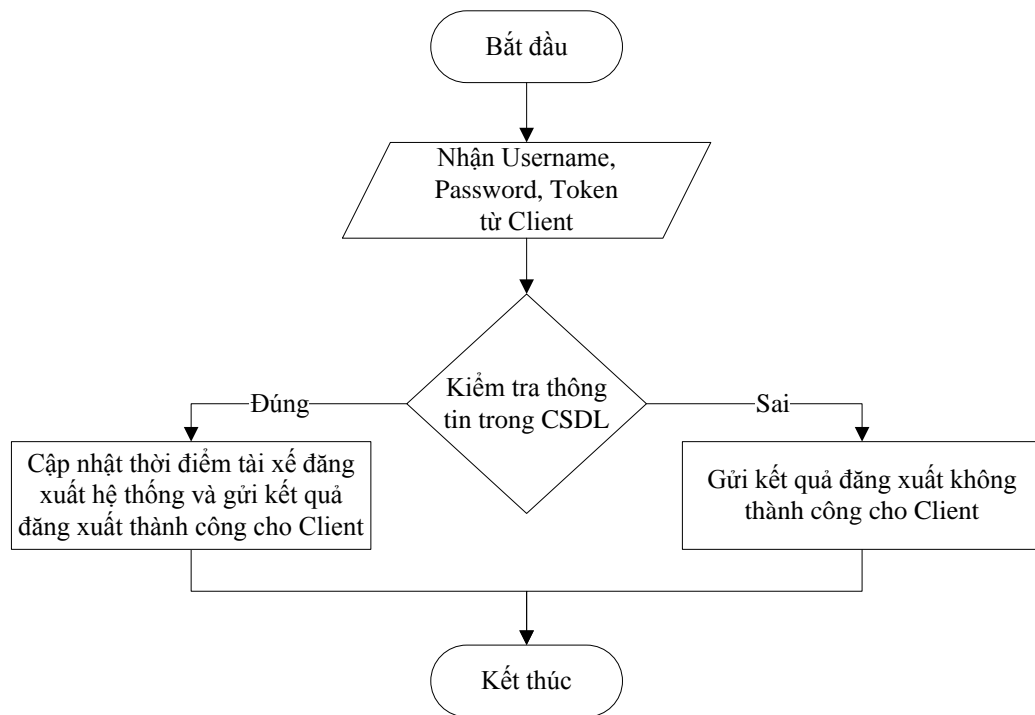
**Hình 4.5: Chức năng đăng nhập**

Chi tiết Web Service:

- Phương thức: GET
- URI: /driver/login/{username}/{password}
- Danh sách tham số:

Tham số	Mô tả
Username	Tài khoản đăng nhập hệ thống
Password	Mật khẩu đăng nhập hệ thống

4.3.2. Chức năng đăng xuất



Hình 4.6: Chức năng đăng xuất

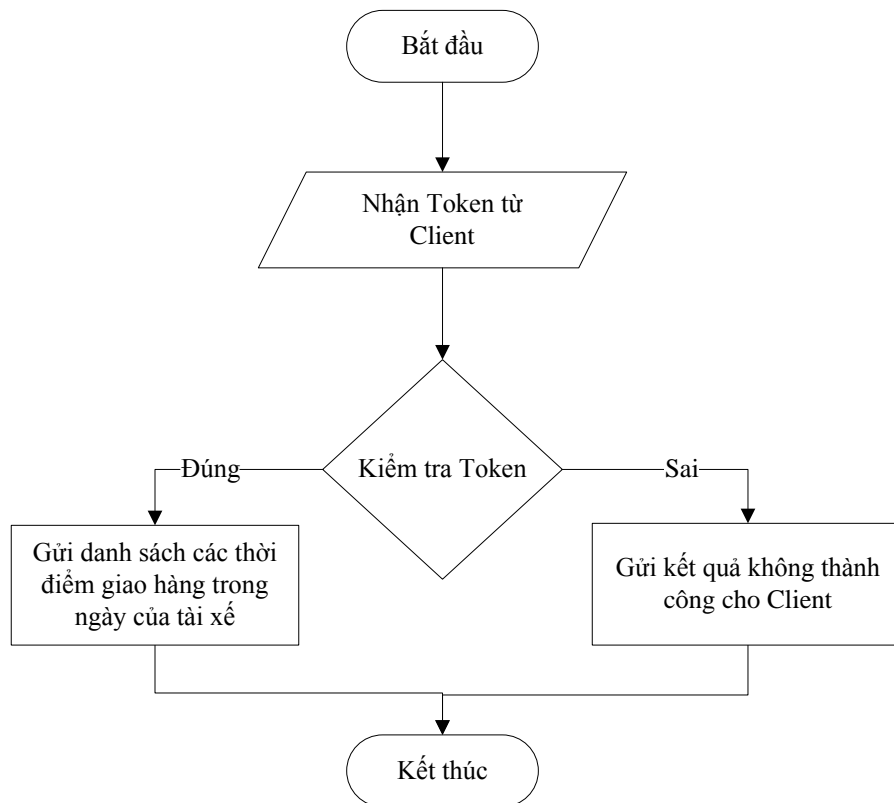
Chi tiết Web Service:

- Phương thức: GET
- URI: /driver/logout/{username}/{password}/{token}

- Danh sách tham số:

Tham số	Mô tả
Username	Tài khoản đăng nhập hệ thống
Password	Mật khẩu đăng nhập hệ thống
Token	Token được cấp phát khi đăng nhập

4.3.3. Chức năng hiển thị các thời điểm giao hàng trong ngày



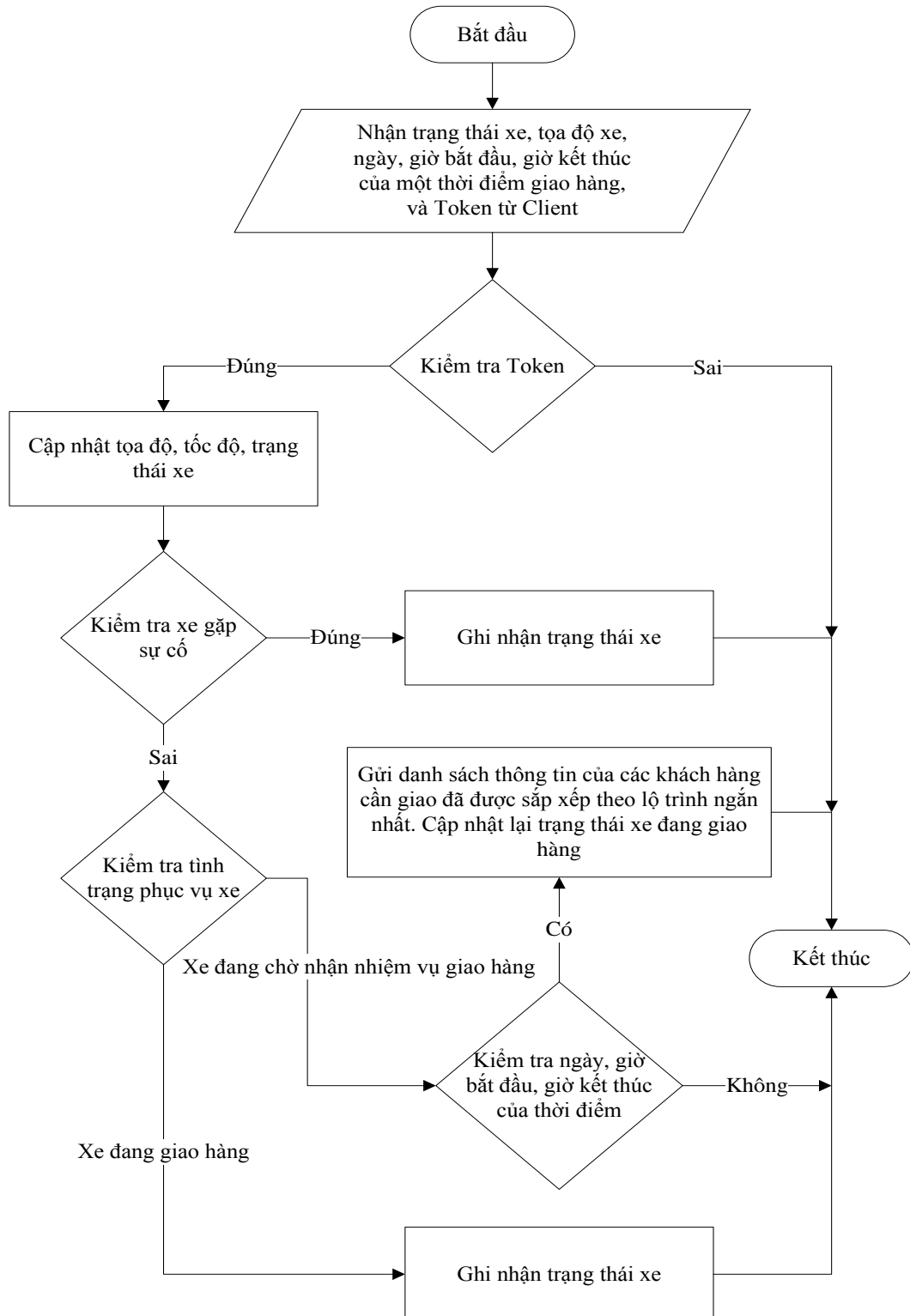
Hình 4.7: Chức năng hiển thị các danh sách thời điểm giao hàng trong ngày

Chi tiết Web Service:

- Phương thức: GET
- URI: /calendar/calendar_info/{token}
- Danh sách tham số:

Tham số	Mô tả
Token	Token được cấp phát khi đăng nhập

4.3.4. Chức năng cập nhật trạng thái và hiển thị hành trình giao hàng ngắn nhất



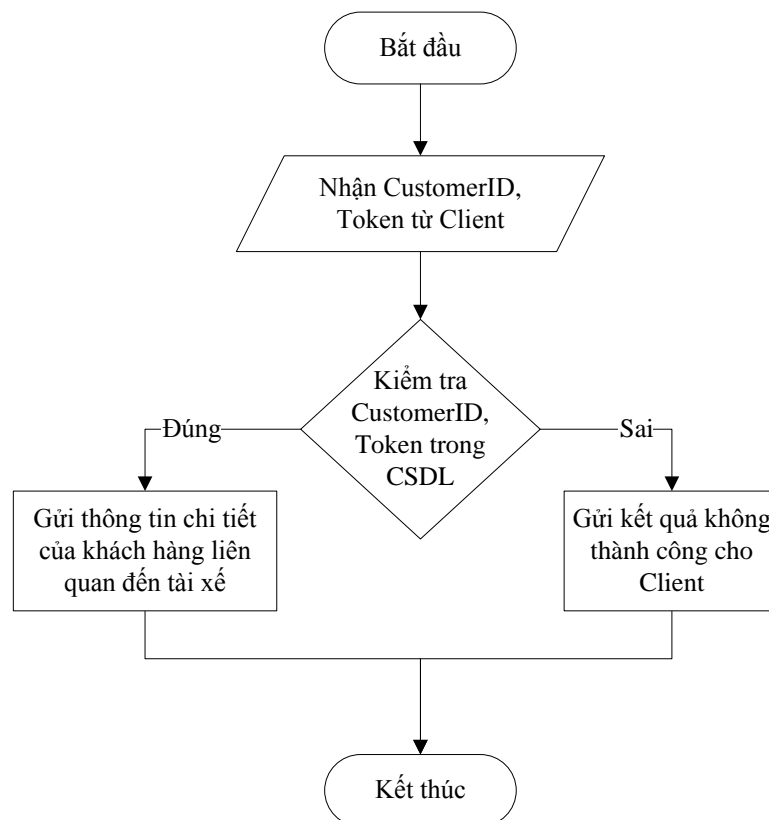
Hình 4.8: Chức năng cập nhật trạng thái và hiển thị hành trình giao hàng ngắn nhất

Chi tiết Web Service:

- Phương thức: GET
- URI: /calendar/routeInfo/{token}/{date}/{beginTime}/{endTime}
- Danh sách tham số:

Tham số	Mô tả
Token	Token được cấp phát khi đăng nhập
Date	Ngày giao hàng
beginTime	Thời gian bắt đầu
endTime	Thời gian kết thúc

4.3.5. Chức năng hiển thị thông tin chi tiết khách hàng



Hình 4.9: Chức năng hiển thị thông tin chi tiết khách hàng

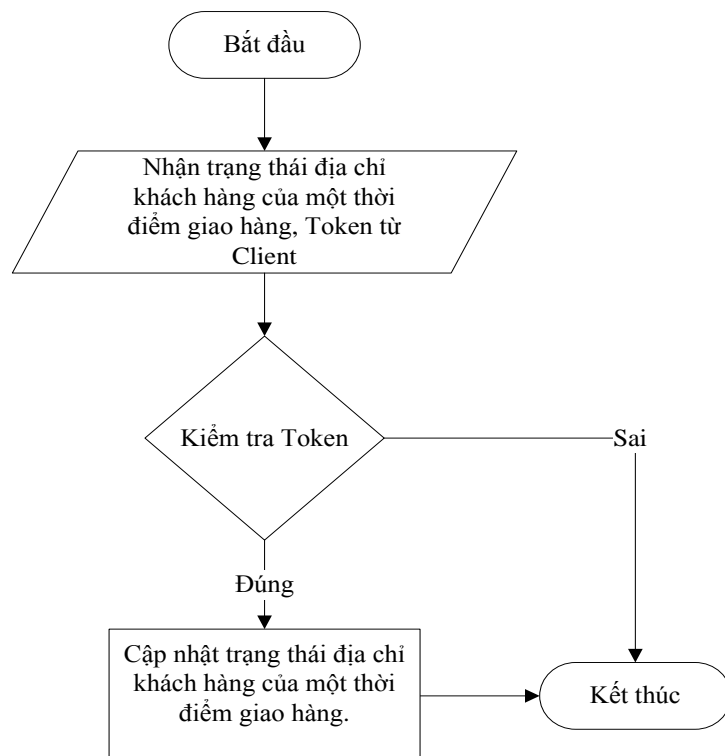
Chi tiết Web Service:

- Phương thức: GET
- URI: /customer/customer_info/{token}/{customerID}

- Danh sách tham số:

Tham số	Mô tả
Token	Token được cấp phát khi đăng nhập
customerID	Mã khách hàng

4.3.6. Chức năng cập nhật trạng thái địa chỉ khách hàng trong một thời điểm giao hàng



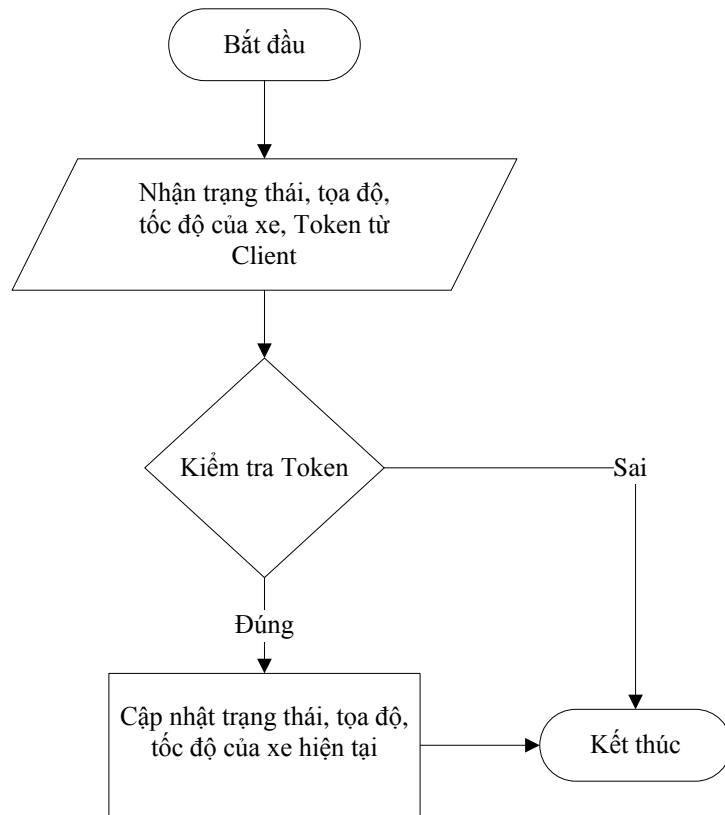
Hình 4.10: Chức năng cập nhật trạng thái địa chỉ trong một thời điểm giao hàng

Chi tiết Web Service:

- Phương thức: GET
- URI: /calendar/updateState/{token}/{calendarID}/{state}
- Danh sách tham số:

Tham số	Mô tả
Token	Token được cấp phát khi đăng nhập
calendared	Mã một thời điểm giao hàng
State	Trạng thái giao hàng

4.3.7. Chức năng cập nhật thông tin phương tiện giao hàng



Hình 4.11: Chức năng cập nhật thông tin phương tiện giao hàng

Chi tiết Web Service:

- Phương thức: GET
- URI:
/transport/update/{token}/{latitude}/{longitude}/{speed}/{state}/{driverID}
- Danh sách tham số:

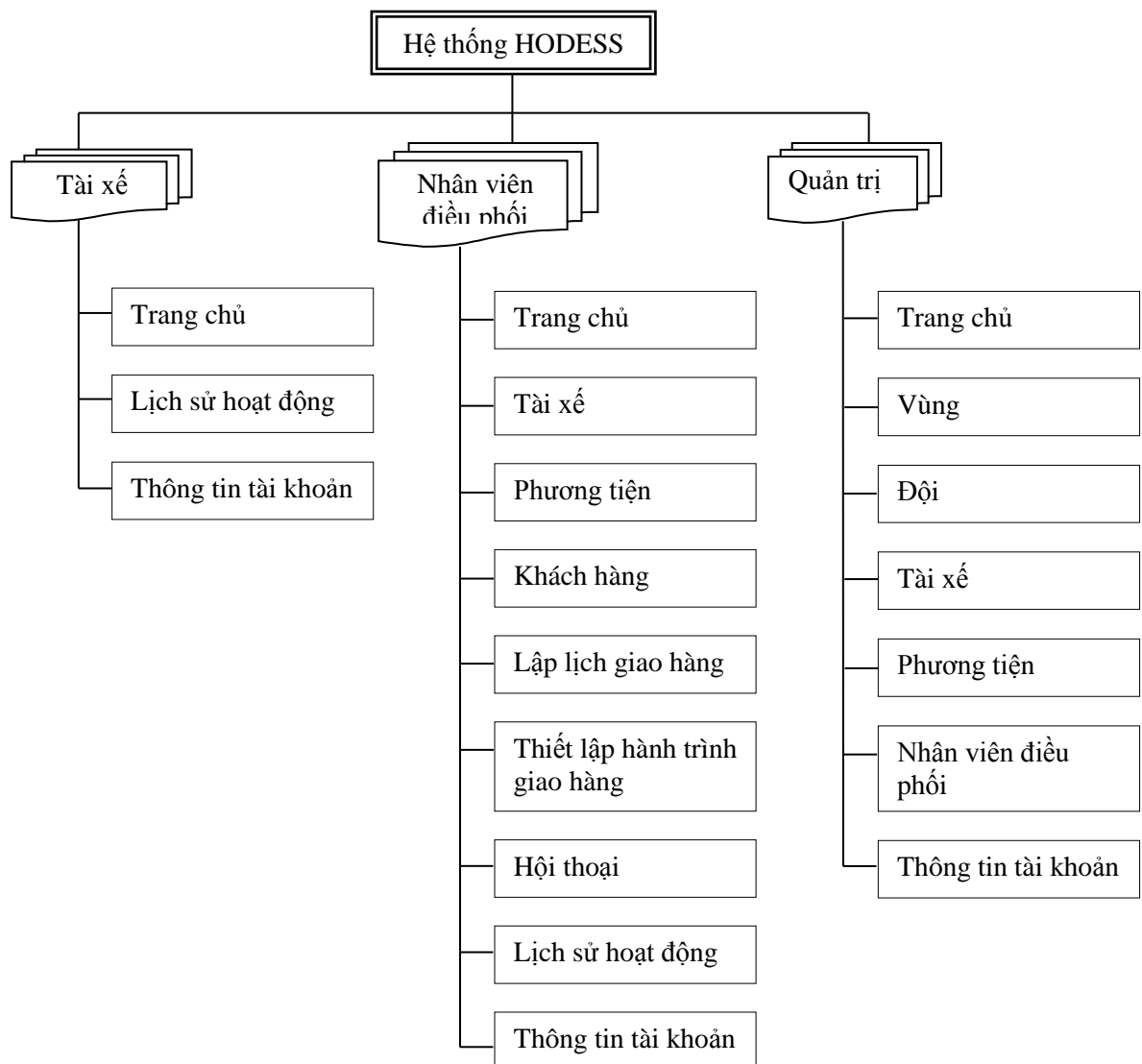
Tham số	Mô tả
Token	Token được cấp phát khi đăng nhập
Latitude	Vĩ độ vị trí của xe
Longitude	Kinh độ vị trí của xe
Speed	Tốc độ xe
driverID	Mã tài xế
State	Trạng thái của xe

4.4. Xây dựng Web Application điều hành giao hàng

Phần này là hệ thống quản lý của người điều phối hỗ trợ trong quá trình giám sát, tiếp nhận yêu cầu giao hàng của khách hàng, điều phối phương tiện chuyên chở, lập lịch giao hàng và thiết lập hành trình giao hàng ngắn nhất cho mỗi tài xế.

Tài nguyên hỗ trợ để xây dựng hệ thống hỗ trợ giao hàng này gồm: IDE (Eclipse Juno), Web Server (Apache Tomcat 7.0), Java EE version (Java EE 6 Web), Java platform (JDK 1.6), cơ sở dữ liệu (MySQL).

Hệ thống hỗ trợ giao hàng chủ yếu phục vụ ba đối tượng được mô tả như trong hình 4.12.



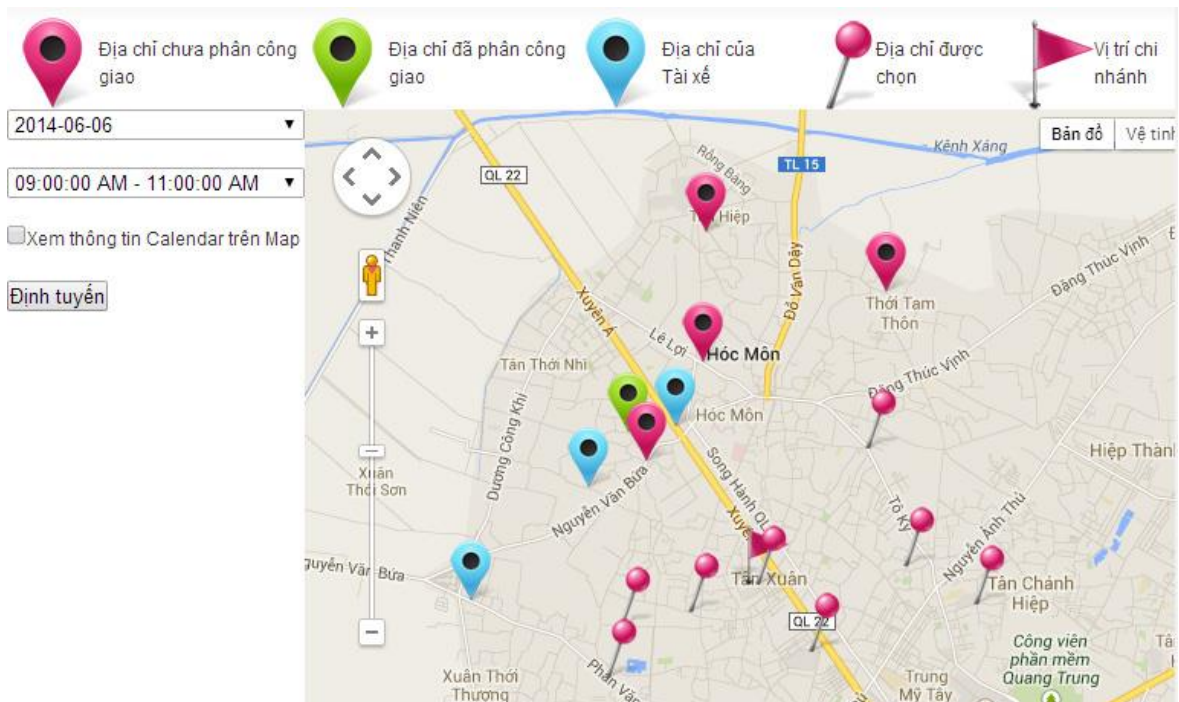
Hình 4.12: Ứng dụng web quản lý hỗ trợ giao hàng

Mỗi đối tượng được cấp quyền sử dụng các chức năng cụ thể như sau:

- Tài xế: kiểm tra trạng thái, vị trí hiện tại của xe, xem lịch sử hoạt động giao hàng đến địa chỉ khách hàng nào theo ngày hoặc tháng, chỉnh sửa thông tin tài khoản.
- Nhân viên điều phối: kiểm tra, giám sát vị trí, trạng thái của các xe chuyên chở hoặc riêng từng xe, tiếp nhận yêu cầu giao hàng của khách hàng, lập lịch giao hàng, điều phối các địa chỉ cần giao hàng cho mỗi tài xế, thiết lập hành trình giao hàng ngắn nhất cho mỗi tài xế, theo dõi lịch sử hoạt động giao hàng của tài xế, chọn phương tiện chuyên chở cho tài xế, quản lý phòng hội thoại.
- Quản trị: quản lý mọi đối tượng trong hệ thống.

4.4.1. Xây dựng chức năng điều phối và thiết lập hành trình

Đây là phần có vai trò quan trọng trong hệ thống hỗ trợ giao hàng giúp cho hoạt động điều phối, giám sát của nhân viên. Nhân viên điều phối sử dụng các công cụ trên form để lựa chọn các địa điểm giao hàng cho từng tài xế theo từng thời điểm cụ thể. Hình 4.13 là giao diện điều phối địa điểm giao hàng cho một tài xế.








Hình 4.13: Thành phần điều phối địa chỉ giao hàng cho một tài xế

Trên bản đồ hiển thị tất cả các địa điểm đang chờ giao hàng của một thời điểm cụ thể trong cùng một vùng giao hàng với tài xế. Việc chọn hoặc xem thông tin một địa chỉ nào đó trong việc thiết lập hành trình chỉ cần click chọn Marker trên bản đồ, icon của Marker sẽ được thay đổi kiểu sau mỗi lần click chọn. Sau khi chọn xong các địa chỉ giao hàng cho tài xế, người điều phối sẽ định tuyến đường giao hàng ngắn nhất để hoàn thành việc điều phối.

Các biểu tượng tương ứng với trạng thái của địa chỉ giao hàng của tài xế được mô tả trong bảng 4.11.

Bảng 4.11: Trạng thái địa chỉ giao hàng

Icon	Mô tả
	Địa chỉ chưa được phân công giao hàng
	Địa chỉ đã có tài xế khác được phân công
	Địa chỉ của tài xế đã được phân công
	Địa chỉ nhân viên điều phối vừa chọn để chờ thiết lập hành trình.
	Vị trí của chi nhánh giao hàng, là điểm khởi hành của tài xế

Thời điểm thiết lập hành trình cho từng tài xế dựa vào quá trình lập lịch giao hàng cho khách hàng của nhân viên điều phối. Thời điểm khách hàng chọn trùng với các địa chỉ khách hàng khác trong cùng một vùng tại cùng một thời điểm sẽ được hệ thống sàng lọc, liệt kê danh sách các tài xế có khả năng giao hàng với điều kiện các tài xế này đã được điều phối ít địa chỉ giao hàng hoặc chưa nhận nhiệm vụ tại thời điểm này (Hình 4.14).

Ngày giao
 Giờ giao -
 Trạng thái
 Tài xế

	Tài xế ▾	SĐT ▾	Đội ▾
<input type="radio"/>	Đào Thị A	0923451565	Đội quận 1
<input type="radio"/>	Đào Thị B	0987655674	Đội quận 1
<input type="radio"/>	Đào Thị C	0985842951	Đội quận 1
<input type="radio"/>	Đào Thị D	0923607050	Đội quận 1
<input type="radio"/>	Đào Thị F	0923607050	Đội quận 1

Hình 4.14: Danh sách tài xế có khả năng giao trong việc lập lịch giao hàng cho khách

Ngoài ra, nhân viên điều phối có thể theo dõi danh sách khách hàng trong ngày (Hình 4.15) để biết được còn khách hàng nào chưa nhận được giao.

KHÁCH HÀNG HÔM NAY

<input type="checkbox"/>	Họ và tên ▾	Địa chỉ ▾	Trạng thái ▾	Bắt đầu ▾	Kết thúc ▾	Tài xế ▾	Tùy chọn ▾
<input type="checkbox"/>	Đỗ Sang	9 Đường số 4, 7, Gò Vấp, HCM	Chưa giao	19:00:00	22:30:00	Giroud	<input type="button" value="⚙"/> <input type="button" value="✕"/>
<input type="checkbox"/>	Tấn Lợi	Nguyễn Văn Linh, Bình Chánh, HCM	Chưa giao	13:00:00	16:30:00	Bao	<input type="button" value="⚙"/> <input type="button" value="✕"/>
<input type="checkbox"/>	Trần Thanh Bình	18 Cộng Hòa, 4, Tân Bình, HCM	Chưa giao	08:00:00	10:00:00	Iniesta	<input type="button" value="⚙"/> <input type="button" value="✕"/>
<input type="checkbox"/>	Nguyễn Minh Tâm	29 Nguyễn Văn Săng, HCM	Chưa giao	13:00:00	16:30:00	Bale	<input type="button" value="⚙"/> <input type="button" value="✕"/>
<input type="checkbox"/>	Đinh Tiến Đạt	339 Lê Văn Sỹ, quận 3, HCM	Hoàn thành	13:00:00	16:30:00	Đào Thị A	<input type="button" value="⚙"/> <input type="button" value="✕"/>

Page 1 / 4

5

Hình 4.15: Danh sách khách hàng trong ngày


Trong trường hợp xấu các địa điểm giao hàng không thể thực hiện được do tài xế giao hàng gặp sự cố, nhân viên điều phối sẽ nhận được thông báo để điều phối tài xế khác hoặc thay đổi lịch giao hàng cho các khách hàng.

4.4.2. Xây dựng hệ thống bản đồ

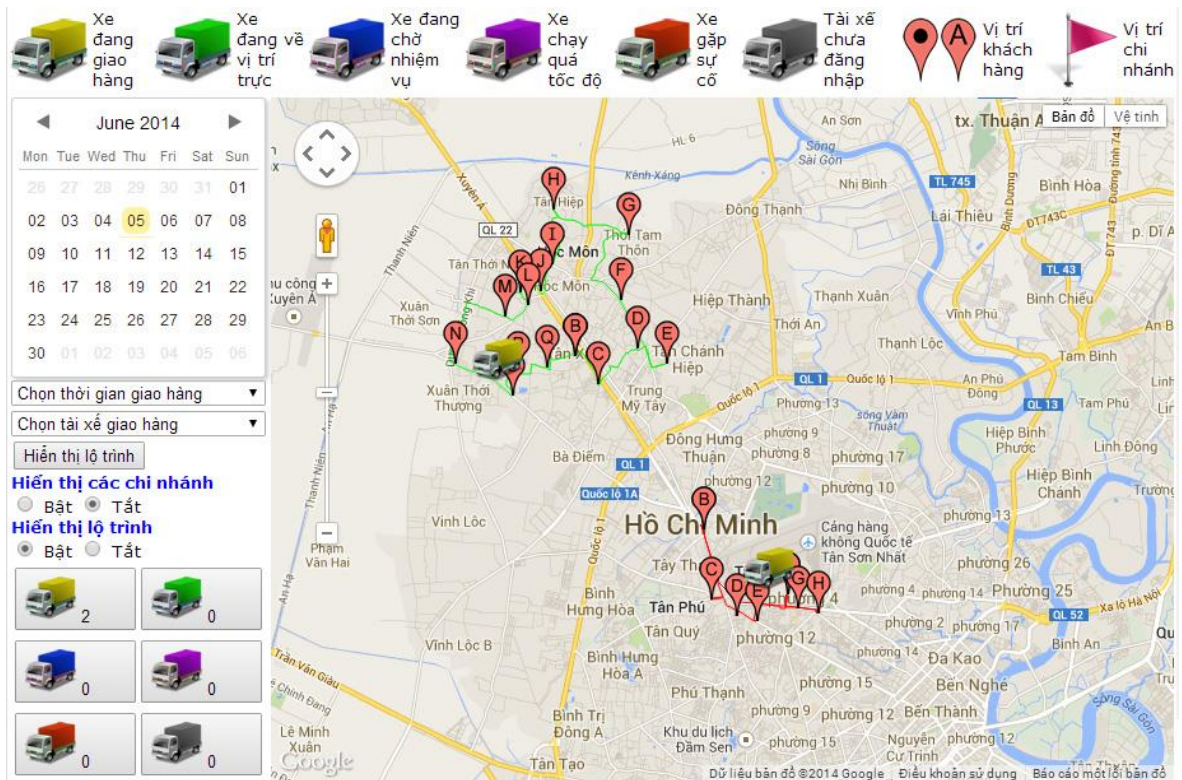
Đây là thành phần bản đồ hiển thị tổng thể quá trình hoạt động của các xe trong một thời điểm. Việc tương tác với bản đồ, yêu cầu, xử lý, hiển thị dữ liệu được thực hiện hoàn toàn bằng JavaScript. Để hiển thị vị trí và trạng thái của các

xe trên bản đồ ta sẽ sử dụng các loại Marker. Trạng thái của xe hiển thị bằng cách thay đổi icon của Marker được quy định trong bảng 4.12.

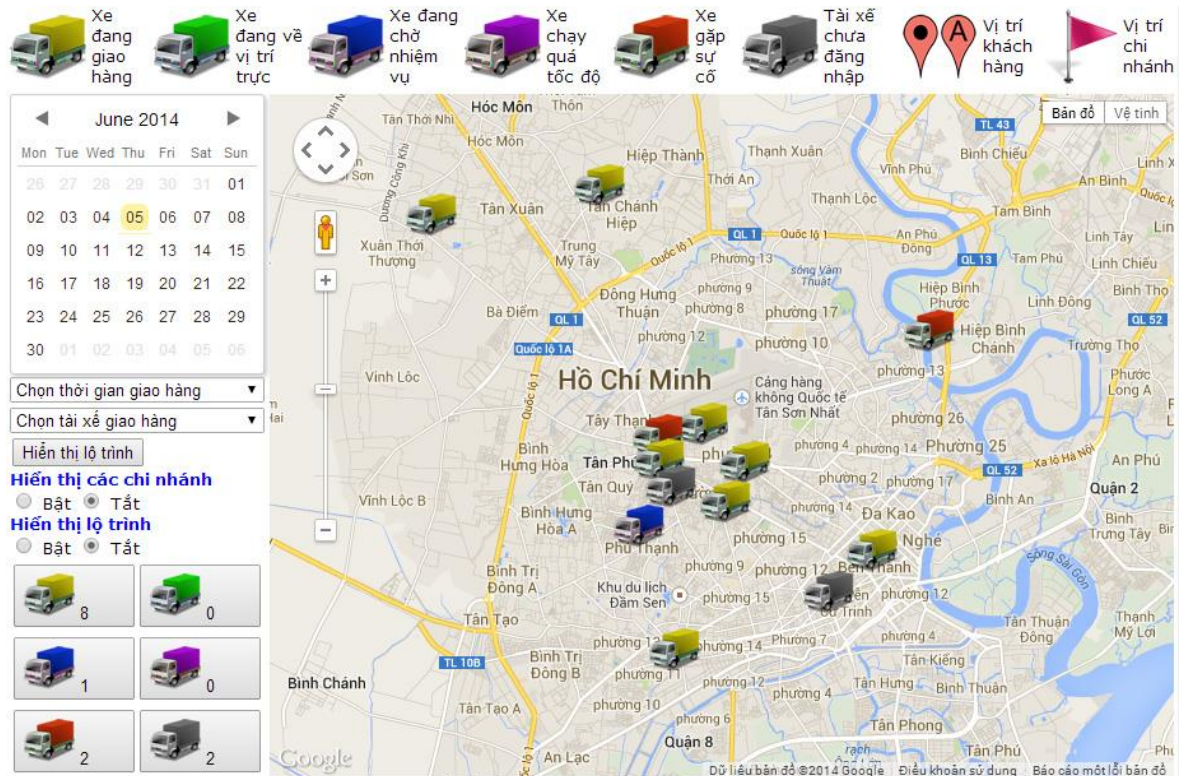
Bảng 4.12: Trạng thái xe trong hệ thống hỗ trợ giao hàng

Icon	Mô tả
	Xe đang giao hàng
	Xe đang về vị trí trực
	Xe đang chờ được giao nhiệm vụ
	Xe chạy quá tốc độ cho phép
	Xe gặp sự cố
	Tài xế chưa đăng nhập hệ thống
	Vị trí địa chỉ của khách hàng
	Vị trí của chi nhánh giao hàng, là điểm khởi hành của tài xế

Ngoài ra, trên bản đồ sẽ hiển thị hành trình giao hàng ngắn nhất của mỗi tài xế đã được thiết lập sẵn, hành trình của mỗi xe được hiển thị trực tiếp lên bản đồ (Hình 4.16), và trạng thái hoạt động hiện hành của mỗi xe (Hình 4.17).



Hình 4.16: Hệ thống bản đồ giám sát hành trình giao hàng

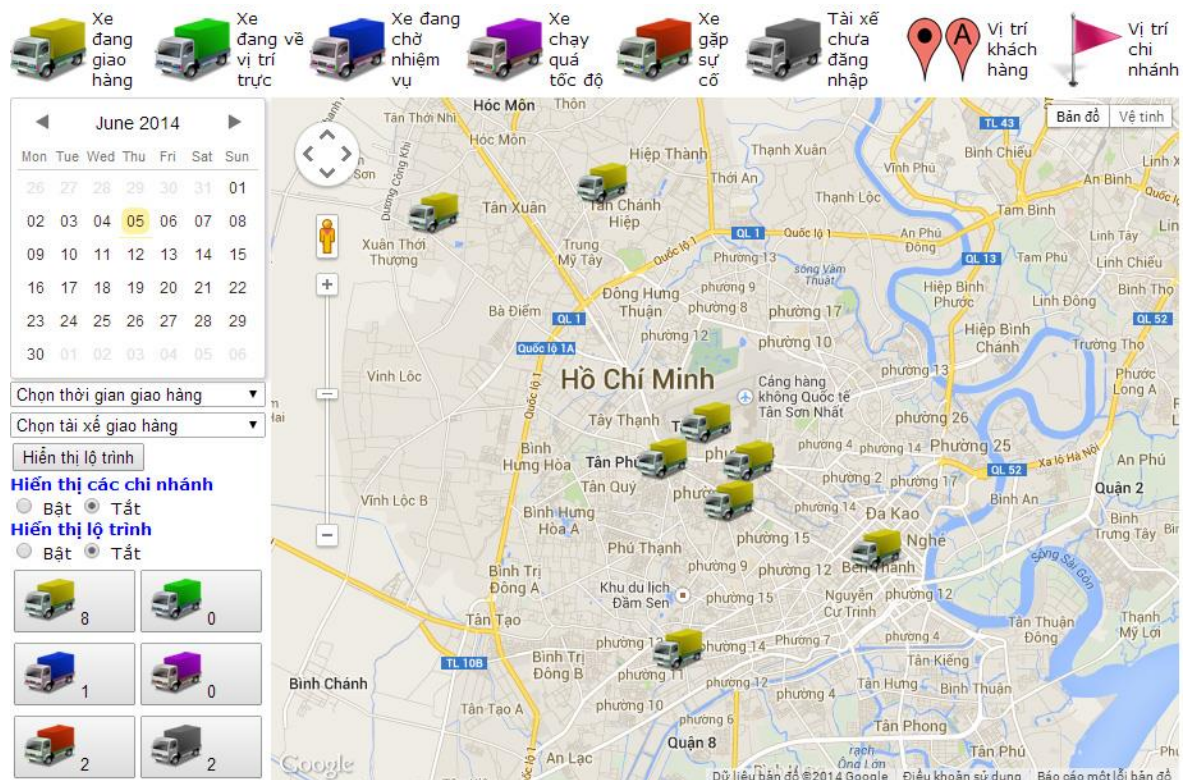


Hình 4.17: Hệ thống giám sát trạng thái các xe

Dựa trên trạng thái của xe, hệ thống có khả năng cho biết số lượng xe theo trạng thái trong hệ thống (Hình 4.18), phân loại để hiển thị các xe lên bản đồ theo trạng thái (Hình 4.19).

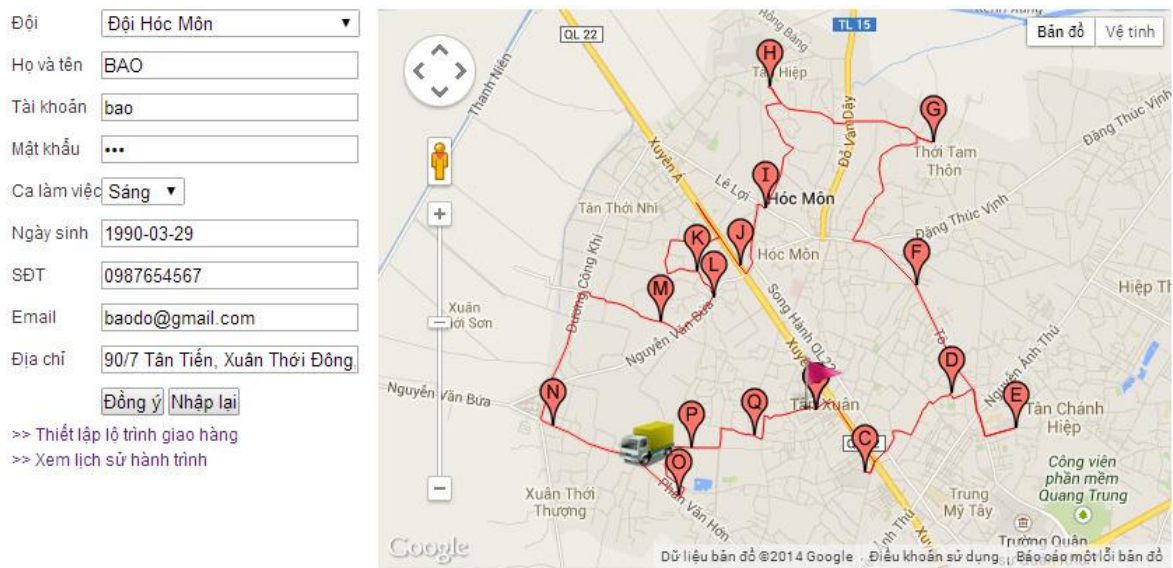


Hình 4.18: Số lượng xe trong hệ thống



Hình 4.19: Hiển thị theo phân loại trạng thái xe

Hiện thị vị trí và trạng thái của một xe: ta có thể chọn một xe cụ thể để tiến hành giám sát hành trình (Hình 4.20), theo dõi lịch giao hàng (Hình 4.21).



Hình 4.20: Giám sát hành trình của một xe

	Giờ bắt đầu ▾	Giờ kết thúc ▾	Trạng thái ▾	Khách hàng ▾	Tùy chọn ▾
Ngày giao hàng: 2014-06-05. Số địa chỉ giao: 16					
<input type="checkbox"/>	09:00:00	11:00:00	Chưa giao	Đỗ Thiên Ân	
<input type="checkbox"/>	09:00:00	11:00:00	Chưa giao	Khách HM2	
<input type="checkbox"/>	09:00:00	11:00:00	Chưa giao	Khách HM3	
<input type="checkbox"/>	09:00:00	11:00:00	Chưa giao	Khách HM4	
<input type="checkbox"/>	09:00:00	11:00:00	Chưa giao	Khách HM5	

Hình 4.21: Theo dõi lịch giao hàng của một xe

4.5. Xây dựng ứng dụng Client

Hầu hết các Smartphone sử dụng hệ điều hành Android có tích hợp chip xử lý có khả năng nhận tín hiệu GPS từ vệ tinh. Android còn cung cấp một thư viện dùng để giao tiếp và xử lý các thông tin lấy từ GPS nằm trong package android.location. Trong package này gồm các thành phần như LocationManager, Location... Nhưng lớp location là quan trọng nhất, vì nó đại diện cho tập hợp các

thông tin thu từ vệ tinh gồm: vĩ độ, kinh độ, cao độ, tốc độ, độ chính xác của vị trí... Cách thức lấy thông tin GPS được thực hiện thông qua hai nhà cung cấp là GPS_PROVIDER và NETWORK_PROVIDER được đặc trưng bởi lớp provider.




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

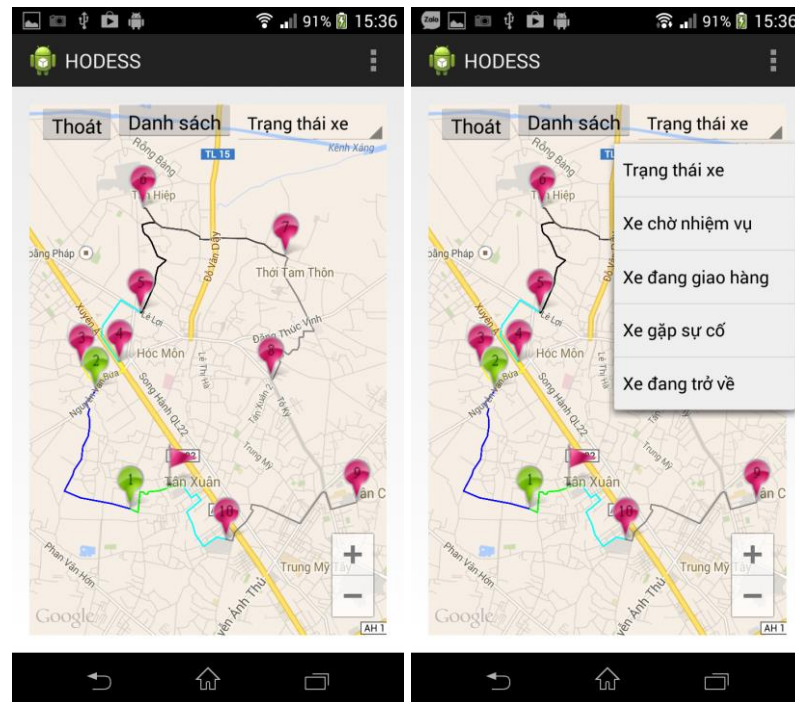
- Android.permission.ACCESS_FINE_LOCATION: quyền này giúp ta có thể sử dụng LocationManager.GPS_PROVIDER để xác định vị trí thông qua GPS.
- Android.permission.ACCESS_FINE_LOCATION hay android.permission.ACCESS_COARSE_LOCATION: giúp ta có thể sử dụng LocationManager.NETWORK_PROVIDER để xác định vị trí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

Các giao diện sau sẽ mô tả các chức năng hoạt động của Client:

- Màn hình chính: giao diện của Google Map v2 gồm các nút kiểm tra danh sách phân công giao hàng và thay đổi trạng thái xe, bản đồ hiển thị hành trình giao hàng khi chọn một thời điểm giao hàng nào đó trong danh sách phân công (Hình 4.22), các Marker trên bản đồ được quy định sẵn (Bảng 4.13).

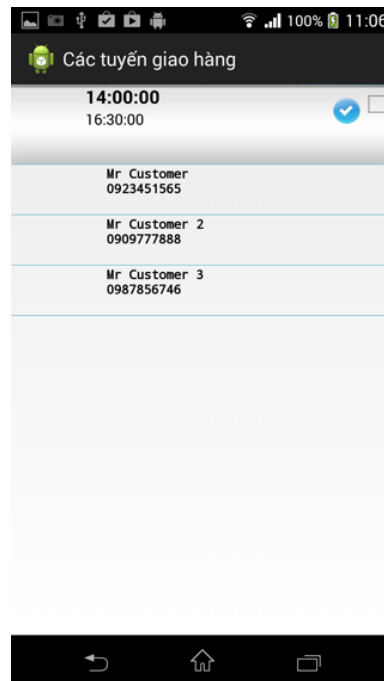
Bảng 4.13: Danh sách Marker trên bản đồ Client

Icon	Mô tả
	Địa chỉ đang chờ giao hàng
	Địa chỉ đã giao hàng
	Vị trí của chi nhánh giao hàng, là điểm khởi hành của tài xế



Hình 4.22: Màn hình chính Client

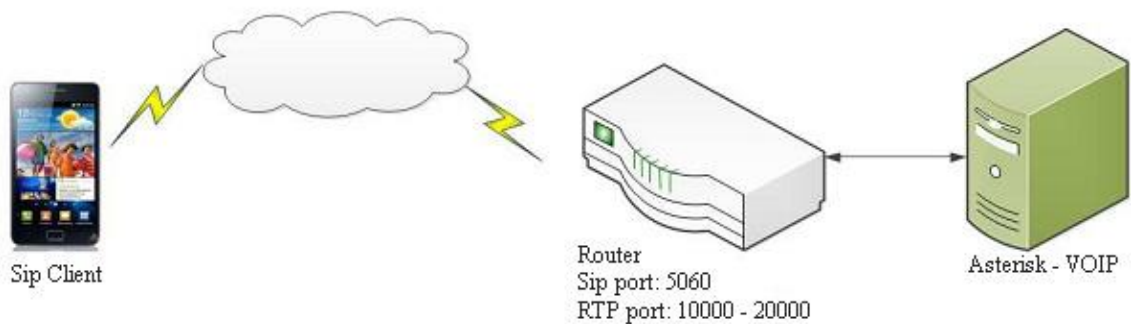
- Màn hình kiểm tra các thời điểm giao hàng: Bao gồm danh sách thời điểm giao với thứ tự khách hàng được sắp xếp theo thứ tự giao của hành trình, và xem chi tiết thông tin của một khách hàng (Hình 4.23).



Hình 4.23: Danh sách các thời điểm giao hàng kèm thông tin khách hàng

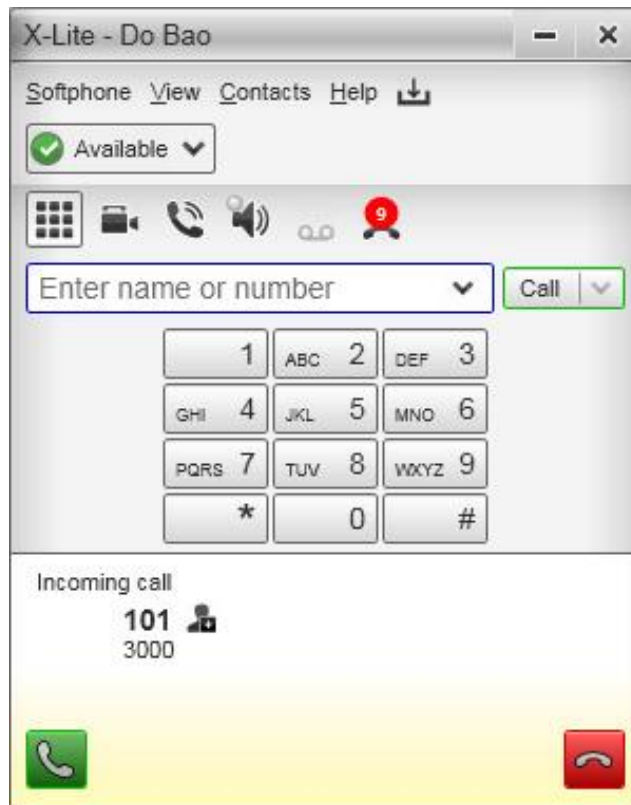
4.6. Triển khai tổng đài VoIP

Tổng đài phục vụ trong việc liên lạc giữa tài xế và nhân viên điều phối trong hệ thống được triển khai là tổng đài mã nguồn mở Asterisk được cài đặt trên hệ thống máy ảo VMware. Hệ thống này sẽ thay thế cho việc liên lạc bằng bộ đàm truyền thống vốn tốn nhiều chi phí và đang được sử dụng nhiều hiện nay. Để tổng đài này hoạt động được thông qua mạng 3G, ta phải tiến hành public tổng đài này để cho các Sip Client có thể liên lạc được qua mạng Internet.



Hình 4.24: Mô hình triển khai tổng đài

Mỗi nhân viên điều phối và tài xế đều được cung cấp một tài khoản Sip. Thông tin tài khoản này sẽ được cung cấp cho tài xế khi đăng nhập, tài xế sẽ sử dụng chức năng gọi VoIP của Client để thực hiện cuộc gọi. Trên hệ thống quản lý, nhân viên điều phối sẽ thực hiện cuộc gọi đến tài xế thông qua X-Lite.



Hình 4.25: X-Lite

Ngoài ra, nhân viên điều phối có thể sử dụng các phần mềm khác trên máy tính hỗ trợ giao thức Sip để thực hiện cuộc gọi đến tài xế.

4.7. Kết luận

Chương này đã đi sâu vào hệ thống hỗ trợ giao hàng tận nhà HODESS sử dụng công nghệ GPS thông qua từng mô hình thiết kế chi tiết các chức năng hoạt động, và đã mô tả được quá trình truyền nhận dữ liệu thông qua web service giữa hệ thống quản lý điều phối giám sát và ứng dụng client trên smartphone, thực hiện việc liên lạc qua mạng 3G thay thế cho bộ đàm truyền thống. Phần quan trọng nhất trong chương này đã mô tả cách thức hoạt động điều phối, lập lịch giao hàng, thiết lập hành trình giao hàng ngắn nhất dựa trên bài toán cổ điển Traveling Salesman Problem.

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN ĐỀ TÀI

1. Kết quả đạt được

Luận văn đã trình bày cụ thể các kiến thức về công nghệ, giải thuật để xây dựng hệ thống hỗ trợ giao hàng dựa trên nền tảng Google Map Service và thuật toán tìm hành trình giao hàng ngắn nhất giải bài toán TSP. Đề tài xây dựng ứng dụng trên cả Client và Server gồm các chức năng như:

- Web Application cung cấp các chức năng quản lý, điều phối thông qua trình duyệt Web tiện dụng và nhanh chóng. Cung cấp cho nhân viên điều phối có cái nhìn tổng thể về hệ thống bằng việc hiển thị vị trí, trạng thái hoạt động, hành trình giao hàng của mỗi xe trong hệ thống, phân loại hiển thị mỗi trạng thái xe, giám sát hành trình cũng như lịch giao hàng của một tài xế. Tiếp nhận, xử lý yêu cầu giao hàng của khách, chọn xe giao hàng phù hợp, thiết lập hành trình giao hàng theo từng thời điểm cho mỗi tài xế. Triển khai hệ thống liên lạc qua VoIP, tích hợp Soft Phone trên Web Application giúp cho việc liên lạc giữa tài xế và nhân viên điều phối thuận tiện hơn.
- Ứng dụng Client chuyên dụng trên Android cho tài xế giao hàng. Hiển thị vị trí xe, trạng thái hoạt động. Hiển thị danh sách các thời điểm giao hàng kèm theo danh sách khách hàng được sắp xếp theo hành trình giao hàng. Hiển thị hành trình giao hàng ngắn nhất qua tất cả địa chỉ giao hàng đã được nhân viên điều phối thiết lập sẵn. Liên tục cập nhật trạng thái, vị trí, tốc độ về trung tâm để nhân viên điều phối theo dõi. Ứng dụng còn cung cấp chức năng gọi VoIP trên mạng 3G.

Tóm lại, đề tài “*Nghiên cứu và xây dựng hệ thống hỗ trợ giao hàng tận nhà sử dụng công nghệ GPS*” đã hoàn thành các yêu cầu đặt ra và có thể triển khai trong thực tế. Hệ thống hỗ trợ giao hàng được xây dựng dựa trên bối cảnh hiện nay bằng việc tận dụng sự phát triển mạnh mẽ của công nghệ và các thiết bị di động, điển hình trong hệ thống này là Smartphone Android và công nghệ truyền tải thông tin qua mạng 3G.

2. Những khó khăn và hạn chế

Trong quá trình thực hiện đề tài, có một số khó khăn, hạn chế khác nhau về công nghệ, thiết bị và thử nghiệm thực tế cụ thể như sau:

- Những công nghệ áp dụng còn mới, nguồn tài liệu nghiên cứu một số chức năng không nhiều.
- Hạn chế về dịch vụ của Google Maps như độ chính xác có một số sai lệch trong việc tìm kiếm và mã hóa địa lý địa chỉ ở Việt Nam.
- Smartphone Android cần chạy trên nền tảng 3.0 trở lên thì mới có thể thực hiện chức năng VoIP.

3. Hướng phát triển đề tài

Đề tài hiện chỉ xây dựng hệ thống hỗ trợ giao hàng thông qua thiết bị di động nền tảng Android, nên cần thêm điều kiện như nguồn kinh phí và thời gian để có thể xây dựng hoàn chỉnh hơn. Một số hướng phát triển:

- Thêm chức năng giám sát thông qua camera của thiết bị di động.
- Xây dựng bản đồ riêng tại một khu vực thành phố để tăng tốc độ xử lý và giảm sai sót trong quá trình mã hóa địa lý cho địa chỉ thành tọa độ trên bản đồ.
- Tăng số lượng giao hàng quy mô lớn, thiết lập hành trình giao hàng ngắn nhất qua nhiều địa chỉ hơn cho đội xe chuyên chở. Hiện tại, trong hệ thống hỗ trợ giao hàng chỉ điều phối tối đa 20 địa chỉ giao hàng cho một tài xế tại cùng thời điểm nhất định.

TÀI LIỆU THAM KHẢO

Tiếng việt

[1] Trần Công Hùng (2012), *Kỹ thuật thoại trên IP-VoIP*, Nhà XB Thông tin và Truyền thông.

Tiếng Anh

[2] Otman Abdoun, Jaafar Abouchabaka, Chakir Tajani (2012), “Analyzing the Performance of Mutation Operators to Solve the Travelling Salesman Problem”, IJES, International Journal of Emerging Sciences , 2(1), pp. 61-77.

[3] Subbu Allamaraju (2010), *RESTful Web Services Cookbook*, O'Reilly Media, Inc.

[4] Sanjeev Arora (1998), "Polynomial Time Approximation Schemes for Euclidian Traveling Salesman and Other Geometric Problems", Journal of the ACM, 45(5), pp. 753-782.

[5] Bill Burke (2010), *RESTful Java with JAX-RS*, O'Reilly Media, Inc.

[6] Ahmed El-Rabbany (2002), "*Chapter 1: Introduction to GPS*", in *Introduction to GPS - The Global Positioning System*, Boston, London: Artech House.

[7] Keld Helsgaun (2000), “An Effective Implementation of the Lin-Kernighan Traveling Salesman Heuristic”, European Journal of Operational Research, 126, pp. 106-130.

[8] Martin Kalin (2009), *Java Web Services: Up and Running*, O'Reilly Media, Inc.

[9] Norman L. Biggs, E. Keith Lloyd, Robin J. Wilson (1986), *Graph Theory 1736-1936*, Clarendon Press, Oxford.

[10] Matai Rajesh, Singh Surya Prakash, Mittal Murari Lal (2010), “*Traveling Salesman Problem: An Overview of Applications, Formulations, and*

Solution Approaches”, *Traveling Salesman Problem, Theory and Applications*, INTECH.

[11] David S. Johnson, Lyle A. McGeoch (1997), “The Traveling Salesman Problem: A Case Study in Local Optimization”, *Local search in combinatorial optimization*, 1, pp. 215-310.

[12] Reto Meier (2012), *Professional Android 4 Application Development*, John Wiley & Sons, Inc.

[13] Christian Nilsson (2003), “Heuristics for the Traveling Salesman Problem”, *Journal of Theoretical Computer Science Reports*, Linköping University, Sweden.

[14] Leonard Richardson, Sam Ruby (2007), *RESTful Web Services*, O'Reilly Media, Inc.

[15] Jose Sandoval (2009), *RESTful Java Web Services*, Birmingham: Packt Publishing Ltd.

[16] A. Schrijver (2005), “On the history of combinatorial optimization (till 1960)”, in *Handbook of Discrete Optimization* (K. Aardal, G. L. Nemhauser, R. Weismantel, eds.), Elsevier, Amsterdam, pp.1-68.

[17] Kathy Sierra, Bert Bates (2008), *Head First Servlet and JSP*, O'Reilly Media, Inc.

[18] Jesse Russell, Ronald Cohn (2012), *Asterisk (PBX)*, Bookvika Publishing.

[19] Gabriel Svennerberg (2010), *Beginning Google Maps API 3*, Apress.

[20] Aleksa Vukotic, James Goodwill (2011), *Apache Tomcat 7*, Apress.

[21] 2010. *What's New in MySQL 5.5 Performance and Scalability* [Trực tuyến]. Oracle Corporation.

Địa chỉ : <http://www.oracle.com/partners/en/most-popular-resources/whats-new-in-mysql-5-5-468281.pdf> [Truy cập: 10/04/2014]

[22] Richard Wiener (2003), “Branch and Bound Implementations for the Traveling Salesperson Problem - Part 1: A solution with nodes containing partial

tours with constraints”, ETH Zurich, Chair of Software Engineering ©JOT, 2(2), pp. 65-86.

[23] Richard Wiener (2003), “Branch and Bound Implementations for the Traveling Salesperson Problem - Part 2: Single threaded solution with many inexpensive nodes”, ETH Zurich, Chair of Software Engineering ©JOT, 2(3), pp. 65-76.

[24] Weixiong Zang (2000), “Depth-First Branch-and-Bound versus Local Search: A case Study”, Proc. 17-th National Conf. on Artificial Intelligence (AAAI-2000), Austin, Texas, 5, pp.930.

Website

[25] <http://opengts.sourceforge.net>.

[26] <http://www.advantrack.com/index.php/vehicle-tracking-system>.

[27] <http://www.json.org>.

[28] <https://sites.google.com/site/gson/gson-user-guide>.