

ĐẠI HỌC QUỐC GIA TP.HCM
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC & KỸ THUẬT MÁY TÍNH



LUẬN VĂN TỐT NGHIỆP ĐẠI HỌC

**XÂY DỰNG ỨNG DỤNG HIỂN THỊ TÌNH HÌNH GIAO
THÔNG DỰA TRÊN GOOGLE MAP CHO THIẾT BỊ DI
ĐỘNG SỬ DỤNG HỆ ĐIỀU HÀNH ANDROID**

HỘI ĐỒNG: KHOA HỌC MÁY TÍNH

GVHD: TS. PHẠM TRẦN VŨ

GVPB: TS. LÊ THANH VÂN

SVTH: NGUYỄN VĂN ĐỨC 50800506

TP. HỒ CHÍ MINH, 01/2013

Lời cam đoan



Tôi xin cam đoan luận văn **“XÂY DỰNG ỨNG DỤNG HIỂN THỊ TÌNH HÌNH GIAO THÔNG DỰA TRÊN GOOGLE MAP CHO THIẾT BỊ DI ĐỘNG SỬ DỤNG HỆ ĐIỀU HÀNH ANDROID”** là phần nghiên cứu và thể hiện của tôi và bạn Nguyễn Thành Trí dưới sự hướng dẫn của thầy Phạm Trần Vũ, không sao chép từ bất kỳ luận văn nào khác. Các kết quả đạt được trong luận văn là trung thực, chưa từng được ai công bố trong bất kỳ công trình nào khác.

Lời cảm ơn



Sau 4 năm học tập chuyên ngành Khoa Học và Kỹ Thuật Máy Tính tại trường Đại Học Bách Khoa thành phố Hồ Chí Minh, được sự cho phép của nhà trường, em thực hiện báo cáo luận văn hoàn thành khóa học.

Em xin chân thành cảm ơn các thầy cô trong bộ môn Khoa Học và Kỹ Thuật Máy Tính, các anh chị hướng dẫn đã tận tình giúp đỡ và truyền đạt kiến thức để em có thể hoàn thành luận văn này. Đặc biệt, em xin gửi lời cảm ơn sâu sắc nhất đến thầy Phạm Trần Vũ đã nhiệt tình hướng dẫn em trong suốt quá trình thực hiện luận văn.

Cảm ơn tất cả các bạn cùng khóa đã nhiệt tình chia sẻ kinh nghiệm và những kiến thức quý báu giúp em hoàn thành tốt luận văn tốt nghiệp của mình.

Mặc dù em đã cố gắng hoàn thành luận văn với tất cả sự nỗ lực, nhưng luận văn chắc chắn không tránh khỏi những thiếu sót. Em kính mong quý thầy cô tận tình chỉ bảo.

Cuối cùng em xin gửi đến quý thầy cô lời chúc sức khỏe và lời cảm ơn chân thành nhất!

TP.HCM, tháng 1 năm 2013

Sinh viên thực hiện đề tài

Tóm tắt luận văn

Hệ thống giao thông đóng một vai trò quan trọng trong việc xây dựng và phát triển đất nước. Nó giúp cho việc giao thương, đi lại được thông suốt, dễ dàng, góp phần vào việc phát triển kinh tế. Với tốc độ phát triển đô thị, dân cư như hiện nay ở Việt Nam, chúng ta có thể thấy rõ sự gia tăng kẹt xe, ùn tắc giao thông hàng giờ liền tại thành phố lớn như Hồ Chí Minh và Hà Nội. Đây là một vấn đề nan giải và hậu quả của nó là không nhỏ đối với từng cá nhân cũng như các tổ chức.

Để góp phần giảm tình trạng kẹt xe cũng như bảo vệ lợi ích của người dân, nhóm chúng tôi kết hợp với một số thành viên nhóm khác quyết định xây hệ thống "**Dự báo và hướng dẫn giao thông trên thiết bị di động**". Hệ thống có chức năng thu thập dữ liệu từ người dùng và các nguồn khác nhau, từ dữ liệu thu nhận được sẽ phân tích và xử lý, ứng dụng có thể cho người dùng biết tình trạng giao thông tại một thời điểm hiện tại, dự báo kẹt xe trong thời gian từ 10 đến 120 phút, người sử dụng cũng có thể tìm được đường đi với thời gian ngắn nhất, hiển thị vị trí hiện tại đang đứng của người dùng, hiển thị đường đã đi.

Hệ thống trên được hiện thực bằng bốn đề tài, trong đó nhóm chúng tôi chịu trách nhiệm hiện thực đề tài "**Xây dựng ứng dụng hiển thị tình hình giao thông dựa trên GoogleMap cho thiết bị di động sử dụng hệ điều hành Android**".

Đề tài có nhiệm vụ xây dựng một ứng dụng trên Android với những chức năng chính như sau:

- Ứng dụng có giao diện người dùng, để người dùng tùy chọn chức năng, cấu hình.
- Hiển thị hiện trạng giao thông hiện tại.
- Hiển thị đường đi cho chức năng tìm đường và hiển thị đường đã đi.
- Hiển thị vị trí hiện tại của người dùng.
- Hiển thị dự báo kẹt xe trong thời gian từ 10 đến 120 phút sắp tới.
- Ứng dụng kết nối được với server, và hiển thị dữ liệu trên thời gian thực.

Yêu cầu của ứng dụng là phải có giao diện thân thiện với người dùng, việc hiển thị dữ liệu lên màn hình phải hiệu quả để đảm bảo ứng dụng có thể hoạt động được trên những smartphone cấu hình thấp. Ứng dụng phải chạy được trên các phiên bản hệ điều hành Android hiện tại.

Abstract

Transport system is very important role for the construction and development of the country. It makes trade and travel smooth and easy, contribute to economic development. With urban growth, the current population in Viet Nam, we can clearly see the increase in traffic congestion, traffic jams for hours in big cities like Ho Chi Minh City and Hanoi Capital. This is a big problem and its consequences is not small for individuals as well as organizations.

To reduce traffic congestion as well as to protect the interests of the people, our group combined with other groups decided to build the system " Traffic forecast and traffic guidance on mobile devices." The system has the function to collect data from users, and other sources, from data acquisition and analysis process. Application can let the user know the traffic situation at the present time, predict traffic jam in between 10 minutes and 120 minutes, users can also find the path in shortest time, showing the current location of user, show tracking of user.

The system is built by four groups, of which we shall make the thesis's subject is "Building Android application show the traffic situation based on GoogleMap.

The tasks of the thesis is building an application on Android with following main functions:

- The application should have user interface, so that the user can configure options.
- Display the current traffic situation .
- Display the prediction traffic jam.
- Show the path found in the shortest time and display the route.
- Show the current location of the user.
- The application can connect to the server, and displays traffic data in real-time.

It is required that the application have a friendly user interface, and the data should be displayed efficiently on the screen to ensure the application could run on low-performance hardware. The application is also required to have a ability to run on all current Android versions.

MỤC LỤC

Đề mục	Trang
Lời cam đoan	ii
Lời cảm ơn	iii
Tóm tắt luận văn	iv
Abstract	v
MỤC LỤC	vi
Danh mục hình	ix
Danh mục bảng	x
CHƯƠNG 1. GIỚI THIỆU	1
1.1 Giới thiệu đề tài	1
1.2 Mục tiêu đề tài	3
1.3 Ý nghĩa của đề tài	4
1.4 Kế hoạch thực hiện đề tài.....	4
1.4.1 Giai đoạn Thực Tập Tốt Nghiệp	4
1.4.2 Giai đoạn Luận Văn Tốt Nghiệp	5
1.5 Một số giải pháp khác đã thực hiện thực	6
1.5.1 Google Map.....	6
1.5.2 Intrix traffic	7
1.5.3 Yandex Traffic Widget.....	7
1.5.4 Giao thông 24	8
CHƯƠNG 2: KIẾN THỨC NỀN TẢNG	10
2.1 Hệ điều hành Android :	10
2.1.1 Một số đặc điểm của hệ điều hành Android :	10
2.1.2 Lịch sử :	11
2.1.3 Delving với máy ảo Dalvik :	12
2.1.4 Kiến trúc của Android	13
2.1.5 Các thành phần của ứng dụng Android	16
2.2 GooleMap API	19
2.3 Hệ cơ sở dữ liệu SQLite.....	20
2.4 Giao thức TCP/IP	21
2.4.1 Các tầng trong TCP/IP	22

2.4.2 Ưu nhược điểm và bảo mật	24
2.5 Định dạng dữ liệu GPX.....	25
2.6 OpenStreetMap	26
PHẦN 3 : THIẾT KẾ ỨNG DỤNG	27
3.1 Phân tích hệ thống chính.....	27
3.1.1 Các thành phần tham gia hệ thống	27
3.1.2 Sơ đồ hoạt động của hệ thống	28
3.2 Phân tích ứng dụng trên Android.....	30
3.2.1 Lược đồ User-Case cho module ứng dụng.....	30
3.2.2 Lược đồ lược đồ hoạt động cho các chức năng của ứng dụng	32
3.3 Lược đồ tuần tự.....	36
PHẦN 4 : HIỆN THỰC ỨNG DỤNG	40
4.1 Dữ liệu về đường đi	40
4.2 Thiết kế cơ sở dữ liệu cho ứng dụng.....	41
4.3 Cách truy xuất dữ liệu nhanh	42
4.4 Gửi nhận dữ liệu trên Server.....	43
4.6 Hiển thị dữ liệu lên màn hình	44
4.6.1 Những layout chính của chương trình.....	44
4.6.2 Lấy dữ liệu theo đúng màn hình.....	44
4.6.3 Hiển thị dữ liệu theo độ zoom của bản đồ:	45
4.6.4 Xác định màu sắc và vận tốc	46
4.6.5 Cách vẽ điểm và đường đi.....	47
4.7 Làm mới dữ liệu.....	48
4.8 Hoàn thành ứng dụng.....	49
4.9 Cài đặt ứng dụng.....	53
4.10 Đánh giá ứng dụng.....	54
CHƯƠNG 5 : TỔNG KẾT	55
5.1 Kết quả đạt được	55
5.2 Hạn chế	55
5.3 Hướng phát triển	56
TÀI LIỆU THAM KHẢO.....	57
PHỤ LỤC	58

Thuật ngữ Anh Việt	58
Nội dung CD	59

Danh mục hình

Hình 1 Ứng dụng giao thông	8
Hình 2 Kiến trúc hệ điều hành Android	13
Hình 3 Chu trình sống của Activity	17
Hình 4 Cách tạo Service	18
Hình 5 Mô hình của TCP/IP	22
Hình 6 Các thành phần tham gia hệ thống	27
Hình 7 Mô hình hoạt động của hệ thống	28
Hình 8 Lược đồ Use-Case của ứng dụng	30
Hình 9 Lược đồ hoạt động hiển thị vị trí hiện tại	32
Hình 10 Lược đồ hoạt động hiển thị hiện trạng giao thông	33
Hình 11 Lược đồ hoạt động hiển thị dự báo kẹt xe	33
Hình 12 Lược đồ hoạt động chức năng hiển thị tìm đường đi	34
Hình 13 Lược đồ hoạt động chức năng hiển thị đường đã đi	34
Hình 14 Lược đồ tuần tự hiển thị hiện trạng giao thông	36
Hình 15 Lược đồ tuần tự hiển thị dự báo kẹt xe	37
Hình 16 Lược đồ tuần tự chức năng hiển thị tìm đường	38
Hình 17 Lược đồ tuần tự hiển thị đường đã đi	39
Hình 18 Lược đồ cơ sở dữ liệu	41
Hình 19 Cơ sở dữ liệu ứng dụng – server	42
Hình 20 Sử dụng bộ đệm đọc dữ liệu	42
Hình 21 Các loại đường trên bản đồ	46
Hình 22 Công thức nội suy tính màu	46
Hình 23 Màn hình chính của ứng dụng	49
Hình 24 Hiển thị trạng thái giao thông	50
Hình 25 Danh sách các đường đã đi theo thời gian	51
Hình 26 Hiển thị đường đã đi	52
Hình 27 Đường đi cho chức năng tìm đường	53

Danh mục bảng

Bảng 1 Công việc giai đoạn Thực Tập Tốt Nghiệp.....	5
Bảng 2 Công việc giai đoạn Luận Văn Tốt Nghiệp	6
Bảng 3 Định dạng tập tin GPX.....	26
Bảng 4 Định dạng node trong OpenStreetMap	40
Bảng 5 Định dạng đường trong OpenStreetMap	41
Bảng 6 Kết nối đến Server sử dụng TCP/IP	43
Bảng 7 Xử lý dữ liệu nhận được.....	43
Bảng 8 Cách vẽ điểm lên trên bản đồ.....	47
Bảng 9 Các vẽ đoạn thẳng	48
Bảng 10 Sử dụng bộ đệm để đọc dữ liệu.....	48

CHƯƠNG 1. GIỚI THIỆU

1.1 Giới thiệu đề tài

Hệ thống giao thông đóng một vai trò vô cùng quan trọng trong việc xây dựng và phát triển đất nước. Nó giúp cho việc giao thương, vận chuyển hàng hóa, đi lại được thông suốt, dễ dàng, góp phần vào việc phát triển kinh tế.

Với tốc độ phát triển đô thị, dân cư như hiện nay ở Việt Nam, tại các thành phố lớn, dân cư ngày càng đông đúc và có nhiều người tham gia giao thông với những phương tiện khác nhau. Vì vậy, trong vài năm gần đây, chúng ta có thể thấy rõ sự gia tăng kẹt xe, ùn tắc giao thông hàng giờ liền tại thành phố Hồ Chí Minh và Hà Nội. Đây là một vấn đề nan giải và hậu quả của nó là không nhỏ đối với từng cá nhân cũng như các tổ chức. Ngoài hậu quả trước mắt là lãng phí tiền bạc và thời gian, kẹt xe còn làm tăng mức độ ô nhiễm do khí thải xe máy, ô tô. Các bác sĩ về sức khỏe lao động và môi trường khẳng định điều này ảnh hưởng xấu đến sức khỏe người dân và gây nên tâm lý nặng nề, bức bối, ảnh hưởng đến năng suất lao động của người dân, từ đó ảnh hưởng tới sự phát triển kinh tế đất nước. Vấn nạn ùn tắc giao thông đã trở thành một bài toán cấp thiết cần được giải quyết.

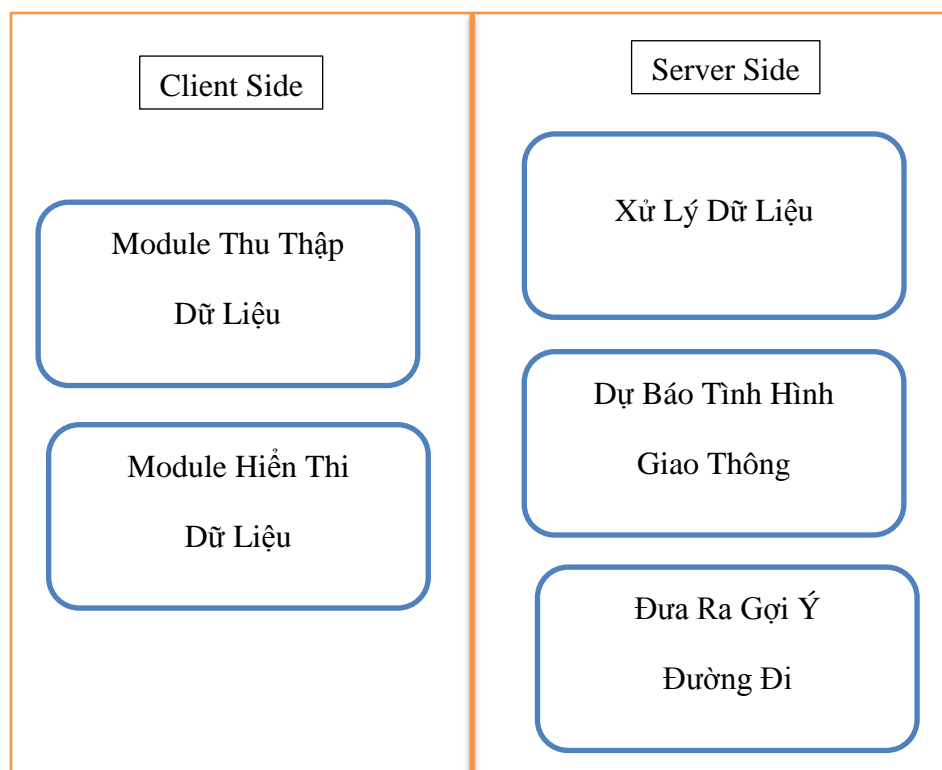
Nhằm góp phần giảm tình trạng kẹt xe cũng như bảo vệ lợi ích của người dân, nhóm chúng em kết hợp với một số nhóm khác xây dựng một hệ thống có chức năng thông báo đến người dùng tình trạng giao thông tại một thời điểm bất kì đồng thời đưa ra các dự báo về tình trạng giao thông. Khi biết được tình trạng giao thông hiện tại và dự báo tình hình trong thời gian gần, người dùng có thể chọn các con đường phù hợp hơn để tránh các khu vực đang hoặc sắp xảy ra kẹt xe. Nhờ đó người dùng có thể tiết kiệm được thời gian, bảo vệ sức khỏe, có tâm lý tốt đồng thời giúp cho tình trạng giao thông trên đường tốt hơn.

Để giải quyết bài toán giao thông trên nhóm chúng tôi và ba nhóm nữa thực hiện xây dựng một hệ thống “**Dự báo và hướng dẫn giao thông trên thiết bị di động** “. Hệ thống được xây dựng dựa trên bốn đề tài. Trong đó có hai đề tài luận văn Cao Học và đề tài luận văn cấp Đại Học

Hệ thống của chúng tôi gồm các chức năng chính sau:

- Thu thập dữ liệu và gửi lên Server: Nếu người dùng đang tham gia giao thông, và đang sử dụng ứng dụng để thu thập dữ liệu, ứng dụng sẽ ghi nhận địa điểm, vận tốc trong một khoảng thời gian xác định của người dùng, và gửi lên server.
- Hiển thị vị trí hiện tại của người dùng: Ứng dụng sẽ hiển thị vị trí hiện tại của người dùng, dựa trên thông tin GPS của thiết bị.
- Dự báo tình trạng giao thông tại một địa điểm theo yêu cầu của người dùng: ứng dụng sẽ đưa ra dự báo tình trạng giao thông ở xung quanh vị trí của người dùng nhờ vào sự tính toán của server từ dữ liệu đã thu thập được.
- Gợi ý đường đi cho người dùng: Ứng dụng đưa ra các phương án tốt nhằm giúp người đến được địa điểm mong đợi mà không phải đi qua các điểm kẹt xe.
- Website theo dõi tình trạng giao thông.

Sơ đồ các module chung của hệ thống :



Trong hệ thống nói trên, nhóm chúng tôi chịu trách nhiệm nghiên cứu và hiện thực module hiển thị, với tên đề tài là :” **Xây dựng ứng dụng hiển thị tình hình giao thông dựa trên Google Map cho thiết bị di động sử dụng hệ điều hành Android**”.

Ứng dụng của chúng tôi gồm các chức năng chính sau:

- Hiển thị vị trí hiện tại của người dùng: Ứng dụng sẽ hiển thị vị trí hiện tại của người dùng, dựa trên thông tin GPS của thiết bị.
- Hiển thị dự báo tình trạng giao thông tại một địa điểm theo yêu cầu của người dùng: ứng dụng sẽ đưa ra dự báo tình trạng giao thông ở xung quanh vị trí của người dùng nhờ vào sự tính toán của server từ dữ liệu đã thu thập được.
- Hiển thị dữ liệu dự đoán kẹt xe trong thời gian từ 10 đến 120 phút trong sắp tới. Người dùng sẽ dùng chức năng này để tìm đường đi cho thích hợp.
- Hiển thị đường đi đã được tìm thấy từ server cho người dùng: Ứng dụng đưa ra các phương án tốt nhằm giúp người đến được địa điểm mong đợi mà không phải đi qua các điểm kẹt xe.
- Hiển thị đường người dùng đã đi qua sử dụng khi người dùng sử dụng chức năng tracking của ứng dụng.
- Cho phép người dùng tương tác tùy chỉnh cầu hình GPS để tăng độ chính xác của GPS.

1.2 Mục tiêu đề tài

Mục tiêu của đề tài là hoàn thành module hiển thị dữ liệu cho hệ thống. Đề tài sẽ xây dựng ứng dụng chạy trên hệ điều hành Android. Ứng dụng có các chức năng chính như sau :

- Ứng dụng kết nối được với hệ thống “**Dự báo và hướng dẫn giao thông trên thiết bị di động**”. Tương tác với server trên thời gian thực.
- Nhận yêu cầu của người dùng từ giao diện ứng dụng , gửi yêu cầu đến server , nhận lại dữ liệu và hiển thị hiện trạng giao thông lên bản đồ cho người dùng.
- Nhận yêu cầu dự báo kẹt xe của người dùng từ giao diện của ứng dụng , gửi yêu cầu đến server , nhận lại dữ liệu và hiển thị dự báo kẹt xe tại các điểm trên bản đồ cho người dùng.
- Nhận yêu cầu tìm đường đi của người dùng từ giao diện của ứng dụng , gửi yêu cầu đến server , nhận lại dữ liệu là đường đi tìm thấy và hiển thị lên ứng dụng.
- Ứng dụng phải hiển thị được vị trí hiện tại của người dùng trên bản đồ.
- Ứng dụng hiển thị đường đã đi của người dùng , với dữ liệu đã được lưu trữ.

- Ứng dụng có thể cài được trên tất cả các phiên bản phổ biến nhất hiện nay của Android.
- Giao diện người dùng phải thân thiện dễ dùng. Với mục đích của hệ thống là càng có nhiều người sử dụng thì dữ liệu xử lý sẽ càng cho ra kết quả chính xác.

1.3 Ý nghĩa của đề tài

Đề tài là một trong những module quan trọng góp phần hoàn thành hệ thống “**Dự báo và hướng dẫn giao thông**”.

Đề tài giúp hệ thống đến được với người sử dụng đó cũng là mục đích chính của cả hệ thống, Từ đó tăng lượng dữ liệu thu thập được từ người dùng, dự trên đó hệ thống sẽ tính toán chính xác hơn các tính năng dự báo, tìm đường, và hiển thị hiện trạng giao thông.

Hiển thị dữ liệu lấy được từ server lên trên giao diện người dùng sẽ giúp người dùng tương tác tốt hơn với hệ thống.

1.4 Kế hoạch thực hiện đề tài

Đề tài được thực hiện hai Thực Tập Tốt Nghiệp và Luận Văn Tốt Nghiệp dưới đây là phân công công việc cho từng giai đoạn:

1.4.1 Giai đoạn Thực Tập Tốt Nghiệp

Giai đoạn này chủ yếu là cùng các nhóm khác phân tích được các yêu cầu của hệ thống và các công nghệ cũng như phương pháp thực hệ thống, từ đó thực hiện đề tài như thế nào. Vì đây là một hệ thống lớn cần phải có sự phối hợp giữa nhiều đề tài nên cần phải có sự phân tích thiết kế thống nhất ngay từ ban đầu.

Nhiệm vụ của giai đoạn này là :Xây dựng được giao diện cơ bản cho ứng dụng, demo được việc nhận dữ liệu từ server phân tích dữ liệu và hiển thị lên Google Map ở một vùng nhỏ (do module phía bên server chưa hoàn thành), hoàn thành một số tính năng khác cho ứng dụng như hiển thị tracking, hiển thị được hiện trạng giao thông với file dữ liệu giả lập.

Bảng phân chia công việc :

STT	DETAILS	DEADLINE	ASSIGN
1	Thu thập và phân tích các yêu cầu của ứng dụng cùng với các nhóm còn lại.	01/02	Team
2	Tìm hiểu về hệ điều hành Android và Google Map API cho Android.	01/03	Đức
3	Cùng với các nhóm khác vẽ lược đồ hệ thống	01/04	Team
4	Xây dựng giao diện cho ứng dụng	01/05	Đức
5	<ul style="list-style-type: none"> Vẽ demo trên map sử dụng lớp Overlay. Hiển thị tracking qua dữ liệu thu được từ nhóm thu thập. Giả lập dữ liệu hiển thị hiện trạng giao thông. 	01/06	Trí
6	Viết báo cáo cho giai đoạn Thực Tập Tốt Nghiệp	01/07	Đức

Bảng 1 Công việc giai đoạn Thực Tập Tốt Nghiệp.

1.4.2 Giai đoạn Luận Văn Tốt Nghiệp

Chỉnh lại giao diện cho phù hợp với thiết bị di động và trải nghiệm người dùng hơn.

Tối ưu hóa quá trình vẽ để cải thiện tốc độ của ứng dụng. Ứng dụng chạy trên thiết bị di động nên cần phải tối ưu hóa việc xử lý dữ liệu.

Hoàn thiện việc gửi nhận dữ liệu từ server để có thể hoạt động tốt trong môi trường rộng lớn , môi trường có nhiều người sử dụng. Sử dụng dữ liệu thực từ hệ thống.

Xây dựng cơ sở dữ liệu để lưu trữ dữ liệu cho ứng dụng.

Tiếp tục hoàn thiện các thành phần của ứng dụng , nhằm hướng tới một sản phẩm có khả năng phục vụ cộng đồng như tiêu chí của đề tài.

Bảng phân chia công việc

STT	DETAILS	DEADLINE	ASSIGN
01	Thu thập phân tích yêu cầu của ứng dụng Vẽ các lược đồ	25/09	Team
02	Tùy chỉnh lại giao diện cho phù hợp hơn với ứng dụng di động và người dùng.	30/09	Đức
03	Tối ưu hóa hiển thị dữ liệu lên google map <ul style="list-style-type: none"> – Hiển thị theo mức độ zoom – Hiển thị theo loại đường 	01/10	Trí
04	Thiết kế cơ sở dữ liệu cho ứng dụng Thiết kế cơ sở dữ liệu cho phía server Hiện thực phần giao thức phía server	15/10	Đức
05	Hiện thực phần kết nối giữa server và ứng dụng sử dụng giao thức TCP/IP	15/10	Trí
06	Tối ưu hóa viết truy xuất dữ liệu từ cơ sở dữ liệu và từ server	01/11	Đức
07	Tạo dữ liệu giả lập để ứng chạy toàn bộ các chức năng của ứng dụng	15/11	Đức
08	Xem lại thiết kế của toàn bộ chương trình , tối ưu hóa code	30/11	Trí
09	Viết báo cáo	15/12	Đức

Bảng 2 Công việc giai đoạn Luận Văn Tốt Nghiệp

1.5 Một số giải pháp khác đã thực hiện thực

1.5.1 Google Map

Là một phần mềm được cài mặc định trong hệ điều hành Android.

Tính năng của Google Maps:

- Xem 3D đối với các tòa nhà.
- Chỉ đường từng bước từng bằng giọng nói và sử dụng GPS.

- Chỉ đường cho lái xe ô tô, các phương tiện giao thông công cộng, xe đạp, đi bộ.
- Hiện thị thông tin hiện trạng giao thông tránh tắc nghẽn.
- Tìm các vị trí , địa điểm , các khu vực kinh doanh.
- Sử dụng Google Map Street View.
- Map trong nhà cho sân bay , khách sạn , cửa hàng ...

Hỗ trợ các tìm đường đã đi qua ,người sử dụng có thể lựa chọn các vị trí như vị trí nơi làm việc , vị trí nhà. Và lưu nó lại, khi người sử dụng cần dụng đến thì các vị trí đó sẽ hiện ra cho người dùng chọn.

Đồng bộ hóa tài khoản Google với các tài khoản khác như Google Mail , Google Talk... với các chức năng xem lịch sử tìm kiếm, tự đồng điền một số thông tin sau khi thông tin đó đã được điền trước trong quá khứ.

Tính năng hiển thị thông tin hiện trạng giao thông , giúp người dùng có thể tránh được các điểm kẹt xe. Chức năng này chỉ dùng được ở một số thành phố lớn trên thế giới và chưa hỗ trợ tại Việt Nam.

1.5.2 Intrix traffic

Là phần mềm hữu ích về cung cấp thông tin về giao thông cho lái xe ô tô :

- Giúp người dùng có thể tìm được đường đi nhanh nhất từ nhà đến công sở. Dễ dàng xác định được đường đi tốt nhất dựa trên dữ liệu được cung cấp từ hàng trăm nghìn lái xe.
- Cung cấp định vị thời gian thực dựa trên cảnh báo liên tục.
- Đề xuất thời gian khởi hành :INRIX Traffic sẽ dự báo và đưa ra cho người dùng những lựa chọn thời gian tốt nhất để khởi hành.
- Chia sẻ thời gian đến : Dễ dàng chia sẻ thời gian đến với bất cứ tài khoản nào chỉ với thao tác đơn giản

Hạn chế :phần mềm này chỉ hoạt động trên các thành phố và quốc gia ở châu Âu.

1.5.3 Yandex Traffic Widget

Là phần mềm viết dưới dạng widget với xuất xứ từ Nga.Đây là phần mềm hiển thị hiện trạng giao thông , giúp người dùng có thể tìm được đường đi với thời gian ngắn nhất .

Phần mềm đã hỗ trợ được khoảng 60 thành phố trên nước Nga , Ukraine , Kazakhstan và Belarus.

Hạn chế : Phần mềm này chỉ hoạt động được ở bên Nga và một số quốc gia Đông Âu.

1.5.4 Giao thông 24

Giaothong247 (Giao Thông 247) là một trong những phần mềm định vị đầu tiên của Việt Nam viết cho iPhone. Mục tiêu là giúp sức cùng cộng đồng giảm thiểu tình trạng kẹt xe tại các đô thị ở Việt Nam, đặc biệt là Thành phố Hồ Chí Minh.

Một trong những tính năng có giá trị nhất của ứng dụng này là khả năng tương tác người dùng. Người dùng có thể thông báo cho cộng đồng các sự kiện giao thông trên cả nước thông qua kết nối internet của điện thoại di động. Ứng dụng cũng hiển thị trực tiếp các thông tin về giao thông chẳng hạn như các sự cố giao thông, ùn tắc giao thông đang xảy ra trên màn hình bản đồ số.

Ngoài ra, Giaothong247 cũng cung cấp các tính năng khác như:



Hình 1 Ứng dụng giaothong24

Những tính năng của chương trình:

- Định vị và hướng dẫn lộ trình bằng cách sử dụng hệ thống định vị toàn cầu GPS.
- Dữ liệu bản đồ chi tiết 63 tỉnh của Việt Nam (phiên bản hiện đang sử dụng Gmap).
- Giao diện người dùng rõ ràng và thân thiện.
- Khi kích hoạt chức năng Chia sẻ thông tin vị trí chạy chế độ nền, người dùng sẽ cho phép Giaothong247 thu thập thông tin vô danh về vị trí, thời gian, tốc độ chạy xe để đóng góp xây dựng cơ sở dữ liệu bản đồ thông tin giao thông đầu tiên cho Việt Nam trong những phiên bản tới. Tuy nhiên, sử dụng tính năng chạy nền GPS sẽ làm giảm đáng kể tuổi thọ pin của máy.
- Và cuối cùng nhưng không kém phần quan trọng, Giaothong247 là miễn phí và sẽ được TMA Solutions liên tục cập nhật.

Điểm hạn chế : Ứng dụng này chạy trên iOS , các thiết bị này giá khá cao. Và không phổ biến bằng các thiết bị chạy Android.

Hiện nay các giải pháp được hiện thực để giải quyết vấn nạn giao thông rất nhiều. Nhưng ở Việt Nam thì hầu như không có hoặc chỉ hoạt động trên các thiết bị chạy iOS hoặc các phiên bản thu phí. Việc phát triển một hệ thống dự báo hiện trạng giao thông chạy trên các thiết bị Android là thực sự cần thiết, giúp ích rất nhiều cho người tham gia giao thông. Các thiết bị Android ngày càng trở nên phổ biến và đang ở mức giá rất gần với mức thu nhập của người Việt Nam. Vì vậy tính khả thi cũng như ứng dụng của đề tài là rất lớn.

CHƯƠNG 2: KIẾN THỨC NỀN TẢNG

2.1 Hệ điều hành Android :

2.1.1 Một số đặc điểm của hệ điều hành Android :

Tính Năng Mở của hệ điều hành Android là gì ?

Android được xây dựng để cho phép các nhà phát triển để tạo ra các ứng dụng di động hấp dẫn tận dụng tất cả phần cứng của chiếc điện thoại nhà sản xuất cung cấp. Nó được xây dựng để được thực sự mở. Ví dụ, một ứng dụng có thể kêu gọi bất kỳ chức năng lõi của điện thoại như thực hiện cuộc gọi, gửi tin nhắn văn bản, hoặc bằng cách sử dụng máy ảnh, cho phép các nhà phát triển có thể tạo ra những ứng dụng phong phú hơn, mang đến cho người dùng nhiều trải nghiệm hơn. Android là hệ điều hành nguồn mở dựa trên Linux Kernel. Hơn nữa, nó sử dụng một máy ảo được tùy chỉnh thiết kế để tối ưu hóa bộ nhớ và tài nguyên phần cứng trong môi trường di động. Android là mã nguồn mở, nó có thể được tự do mở rộng. Nền tảng này sẽ tiếp tục tiến triển, các cộng đồng nhà phát triển sẽ làm việc cùng nhau để xây dựng các ứng dụng di động ngày càng sáng tạo.

Có thể tạo bất cứ ứng dụng nào cho Android :

Android không phân biệt giữa các ứng dụng gốc của điện thoại với các ứng dụng của nhà phát triển thứ ba. Tất cả có thể được xây dựng để có thể truy cập bằng khả năng của một người cung cấp cho người sử dụng điện thoại với một dải rộng các ứng dụng và dịch vụ. Với các thiết bị xây dựng trên Hệ điều hành Android, người dùng có thể hoàn toàn thích ứng với điện thoại đến lợi ích của họ. Họ có thể trao đổi trên màn hình của điện thoại, những phong cách của dialer, hoặc bất kỳ ứng dụng. Họ thậm chí có thể hướng dẫn điện thoại của họ để sử dụng hình ảnh ưa thích của họ xem các ứng dụng để xử lý xem tất cả các hình ảnh.

Android phá bỏ các rào cản của ứng dụng

Android phá bỏ rào cản để xây dựng các ứng dụng mới và sáng tạo. Ví dụ, một nhà phát triển có thể kết hợp thông tin từ các trang web với dữ liệu trên điện thoại di động của một cá nhân – ví dụ như địa chỉ liên hệ của người dùng, lịch, hoặc vị trí địa lý - để cung cấp cho người dùng một trải nghiệm đầy đủ hơn. Với Android, một nhà phát triển có thể xây dựng một ứng dụng cho phép

người dùng xem vị trí của bạn bè của họ và có thể gửi gợi ý kết bạn cho họ nếu có những người ở gần họ sử dụng cùng phần mềm.

Với Android ứng dụng được phát triển dễ dàng và nhanh chóng

Android cung cấp truy cập đến một loạt các thư viện công cụ hữu ích và có thể được sử dụng để xây dựng các ứng dụng phong phú. Ví dụ, Android cho phép các nhà phát triển để có được vị trí của điện thoại, và cho phép các thiết bị để giao tiếp với nhau tạo điều kiện cho các ứng dụng tương tác phong phú hơn. Ngoài ra, Android bao gồm một tập hợp đầy đủ công cụ đã được xây dựng từ cấp thấp đến cấp cao, cùng với việc cung cấp nền tảng phát triển, với năng suất cao và cái nhìn phát triển lâu dài vào các ứng dụng của họ.

2.1.2 Lịch sử :

Ban đầu, Android là hệ điều hành cho các thiết bị cầm tay dựa trên nhân Linux do công ty Android Inc (California, Mỹ) thiết kế. Công ty này sau đó được Google mua lại vào năm 2005 và bắt đầu xây dựng Android Platform. Các thành viên chủ chốt tại Android Inc gồm có: Andy Rubin, Rich Miner, Nick Scars và Chris White.

Và sau tiếp, vào cuối năm 2007, thuộc về Open Handset Alliance gồm các thành viên nổi bật trong ngành viễn thông và thiết bị cầm tay như :Texas Instruments, Broadcom Corporation, Google, HTC, Intel, LG, Marvell Technology Group, Motorola, Nvidia, Qualcomm, Samsung Electronics, Sprint Nextel, T-Mobile, ARM Holdings, Atheros Communications, Asutek Computer Inc, Garmin Ltd, Softbank, Sony Ericsson,...

Mục tiêu của liên minh này là nhanh chóng đổi mới để đáp ứng tốt hơn cho nhu cầu người tiêu dùng và kết quả đầu tiên của nó chính là nền tảng Android. Android được thiết kế để phục vụ nhu cầu của các nhà sản xuất, các nhà khai thác và các lập trình viên thiết bị cầm tay.

Phiên bản SDK đầu tiên phát hành vào tháng 11 năm 2007, hãng T-Mobile cũng công bố chiếc điện thoại Android đầu tiên đó là chiếc T-Mobile G1, chiếc smartphone đầu tiên dựa trên nền tảng Android. Một vài ngày sau đó, Google lại tiếp tục công bố sự ra mắt phiên bản Android SDK realease Candidate 1.0. Trong tháng 10 năm 2008, Google được cấp phép mã nguồn mở cho Android Platform.

Khi Android được phát hành thì một trong những mục tiêu trong kiến trúc của nó là cho phép các ứng dụng có thể tương tác được với nhau và có thể sử dụng lại các thành phần từ các ứng dụng khác. Việc tái sử dụng không chỉ được áp dụng cho các dịch vụ mà nó còn được áp dụng cho cả các thành phần dữ liệu và giao tiếp người dùng.

Vào cuối năm 2008, Google cho phát hành một thiết bị cầm tay được gọi là Android Dev Phone 1 có thể chạy được các ứng dụng Android mà không bị ràng buộc vào các nhà chuyên cung cấp mạng điện thoại di động. Mục tiêu của thiết bị này là cho phép các nhà phát triển thực hiện các cuộc thí nghiệm trên một thiết bị thực có thể chạy hệ điều hành Android mà không phải ký một bản hợp đồng nào. Vào khoảng cùng thời gian đó thì Google cũng cho phát hành một phiên bản vá lỗi 1.1 cho hệ điều hành này. Ở cả 2 phiên bản 1.0 và 1.1 Android chưa hỗ trợ soft-keyboard mà đòi hỏi các thiết bị phải sử dụng bàn phím vật lý, Android cố định vấn đề này bằng cách phát hành SDK 1.5 vào tháng 4 năm 2009, cùng với một số chức năng khác. Chẳng hạn như nâng cao khả năng ghi âm truyền thông, vật dụng và các live folder.

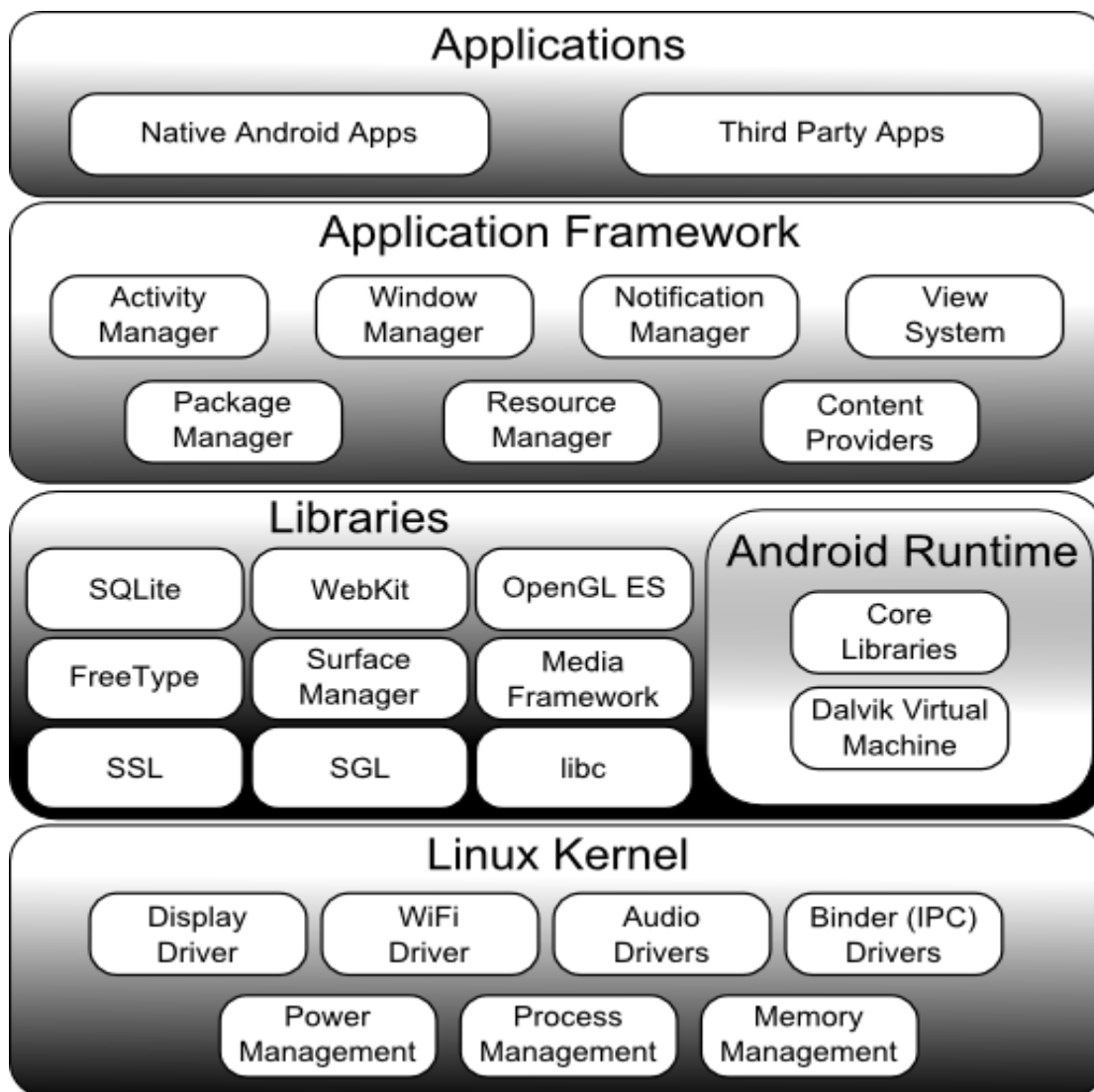
2.1.3 Delving với máy ảo Dalvik :

Dalvik là máy ảo giúp các ứng dụng java chạy trên các thiết bị di động nền tảng Android. Nó chạy các ứng dụng đã được chuyển đổi thành file thực thi Dalvik (.dex). Định dạng này phù hợp cho các hệ thống mà thường bị hạn chế về bộ nhớ và tốc độ xử lý. Dalvik đã được thiết kế và viết bởi Dan Bornstein, người đã đặt tên cho nó sau khi đến thăm một ngôi làng đánh cá có tên Dalvik ở đảo Eyiafiorur, nơi mà một số tổ tiên ông sống.

Từ góc nhìn của một nhà phát triển, Dalvik trông giống như máy ảo Java (Java Virtual Machine) nhưng thực tế thì hoàn toàn khác nhau. Khi nhà phát triển viết một ứng dụng dành cho Android, anh ta thực hiện các đoạn mã trên môi trường Java. Sau đó, nó sẽ biên dịch sang các bytecode của Java, tuy nhiên để thực thi được ứng dụng này trên Android thì nhà phát triển phải thực thi một công cụ tên là DX. Đây là công cụ dùng để chuyển đổi bytecode sang một dạng gọi là dex bytecode. “Dex” là từ viết tắt của “Dalvik executable” đóng vai trò như cơ chế ảo thực thi các ứng dụng Android.

2.1.4 Kiến trúc của Android

Mô hình sau thể hiện một cách tổng quát các thành phần của hệ điều hành Android. Mỗi một phần sẽ được đặt tả chi tiết dưới đây.



Hình 2 Kiến trúc hệ điều hành Android

Tầng ứng dụng:

Android được tích hợp sẵn một số ứng dụng cần thiết, cơ bản như: Contacts, Browser, Camera, Phone,... Tất cả các ứng dụng trên đều được viết bằng Java.

Tầng Application Framework:

Android cung cấp một nền tảng phát triển ứng dụng mở, xây dựng các API hỗ trợ người lập trình để phát triển các ứng dụng.

Người lập trình được phép sử dụng các API trong các ứng dụng tích hợp sẵn của Android. Kiến trúc ứng dụng được thiết kế nhằm mục đích đơn giản hóa việc sử dụng lại các thành phần; bất kỳ ứng dụng nào cũng có thể công bố tính năng (capability) của nó, và các ứng dụng khác có thể sử dụng tính năng này. Cơ bản, tất cả các ứng dụng là một bộ dịch vụ và các hệ thống, bao gồm:

- Một tập hợp phong phú và có thể mở rộng các View để xây dựng ứng dụng, gồm danh sách, hiển thị dạng lưới, hiển thị chữ dạng hộp, nút bấm và kể cả trình duyệt web được tích hợp.
- Trình cung cấp nội dung (Content Provider) cho phép ứng dụng truy xuất dữ liệu từ các ứng dụng khác, hay chia sẻ dữ liệu.
- Quản lý tài nguyên (Resource Manager) cung cấp quyền truy xuất đến tài nguyên không phải mã nguồn như chuỗi định vị (localized strings), các tập tin layout, hay hình ảnh.
- Quản lý thông báo (Notification Manager) cung cấp khả năng hiển thị các cảnh báo lên thanh trạng thái
- Quản lý hoạt động (Activity Manager) quản lý vòng đời của ứng dụng.

Thư viện (Library):

Android bao gồm một tập các thư viện C/C++ được sử dụng bởi nhiều thành phần khác nhau trong hệ thống Android. Điều này được thể hiện thông qua nền tảng ứng dụng Android. Một số các thư viện cơ bản được liệt kê dưới đây:

- System C library: có nguồn gốc từ hệ thống thư viện C chuẩn (libc), được điều chỉnh để nhúng vào các thiết bị dựa trên Linux.
- Media Libraries: dựa trên PacketVideo của OpenCORE, các thư viện hỗ trợ phát và ghi các định dạng audio và video phổ biến.
- Surface Manager: Quản lý hiển thị nội dung 2D và 3D.
- LibWebCore: Công cụ cho các trình duyệt web hiện đại.
- SGL: Công cụ đồ họa 2D cơ sở.
- 3D libraries: Một thể hiện được xây dựng dựa trên OpenGL ES 1.0 APIs. Những thư viện này dùng để tối ưu hóa hiển thị 3D.

- Free Type: Biểu diễn font vector và bitmap.
- QLite: Cơ sở dữ liệu.
- Webkit – Thư viện hiển thị HTML

Android Runtime :

Android bao gồm một tập hợp các thư viện cơ bản mà cung cấp hầu hết các chức năng có sẵn trong các thư viện kernel của ngôn ngữ lập trình Java. Tất cả các ứng dụng Android đều chạy trong tiến trình riêng. Máy ảo Dalvik đã được viết lại để cho một thiết bị có thể chạy nhiều máy ảo hiệu quả. Các VM Dalvik thực thi các tập tin thực thi Dalvik (dex). Định dạng được tối ưu hóa cho bộ nhớ tối thiểu. VM là dựa trên register-based và chạy các lớp đã được biên dịch bởi một trình biên dịch Java để chuyển đổi thành các định dạng dex. Các VM Dalvik dựa vào nhân Linux cho các chức năng cơ bản như luồng và quản lý bộ nhớ cấp thấp.

Linux Kernel :

Android dựa trên Linux phiên bản 2.6 cho hệ thống dịch vụ cốt lõi như bảo mật , quản lý bộ nhớ , quản lý các tiến trình , quản lý mạng , các driver cho các thiết bị. Kernel Linux hoạt động như một lớp trừu tượng giữa phần cứng và phần còn lại của hệ thống (Các thành phần ở trên).

Một vài con số về Android sau 5 năm ra đời và phát triển:

- Theo Google, hiện mỗi ngày có trung bình khoảng 1,3 triệu sản phẩm chạy Android mới được kích hoạt trên toàn cầu.
- Tính đến thời điểm hiện tại, trên toàn cầu có khoảng 570 triệu thiết bị sử dụng Android.
- Trung bình mỗi người Mỹ sở hữu 2 thiết bị chạy Android (smartphone và máy tính bảng) và gần 1/10 dân số thế giới đang sở hữu thiết bị chạy Android.
- Theo Google, số lượng thiết bị Android sẽ cán mốc 1 tỷ trước khi nền tảng di động này cán mốc 6 tuổi. Như vậy Android sẽ mất ít thời gian hơn so với Facebook để cán mốc 1 tỷ người dùng (Facebook mất gần 8 năm để đạt được điều này). Điều này thực sự ấn tượng bởi lẽ tài khoản Facebook có thể đăng ký miễn phí, trong khi muốn sở hữu Android, người dùng sẽ phải mất khoản tiền để sở hữu thiết bị.
- Mặc dù đã ra mắt khá lâu tuy nhiên Android 2.3 Gingerbread vẫn là phiên bản Android phổ biến nhất thế giới với 54% số lượng thiết bị Android. Đứng thứ 2 là Android 4.0 Ice Cream Sandwich với 25,8%. Nền tảng mới nhất Android 4.1 Jelly Bean hiện chỉ mới chiếm 2.7%.

- Hiện tại số lượng ứng dụng trên kho ứng dụng Google Play của Google dành cho nền tảng Android là 700.000 ứng dụng.

2.1.5 Các thành phần của ứng dụng Android

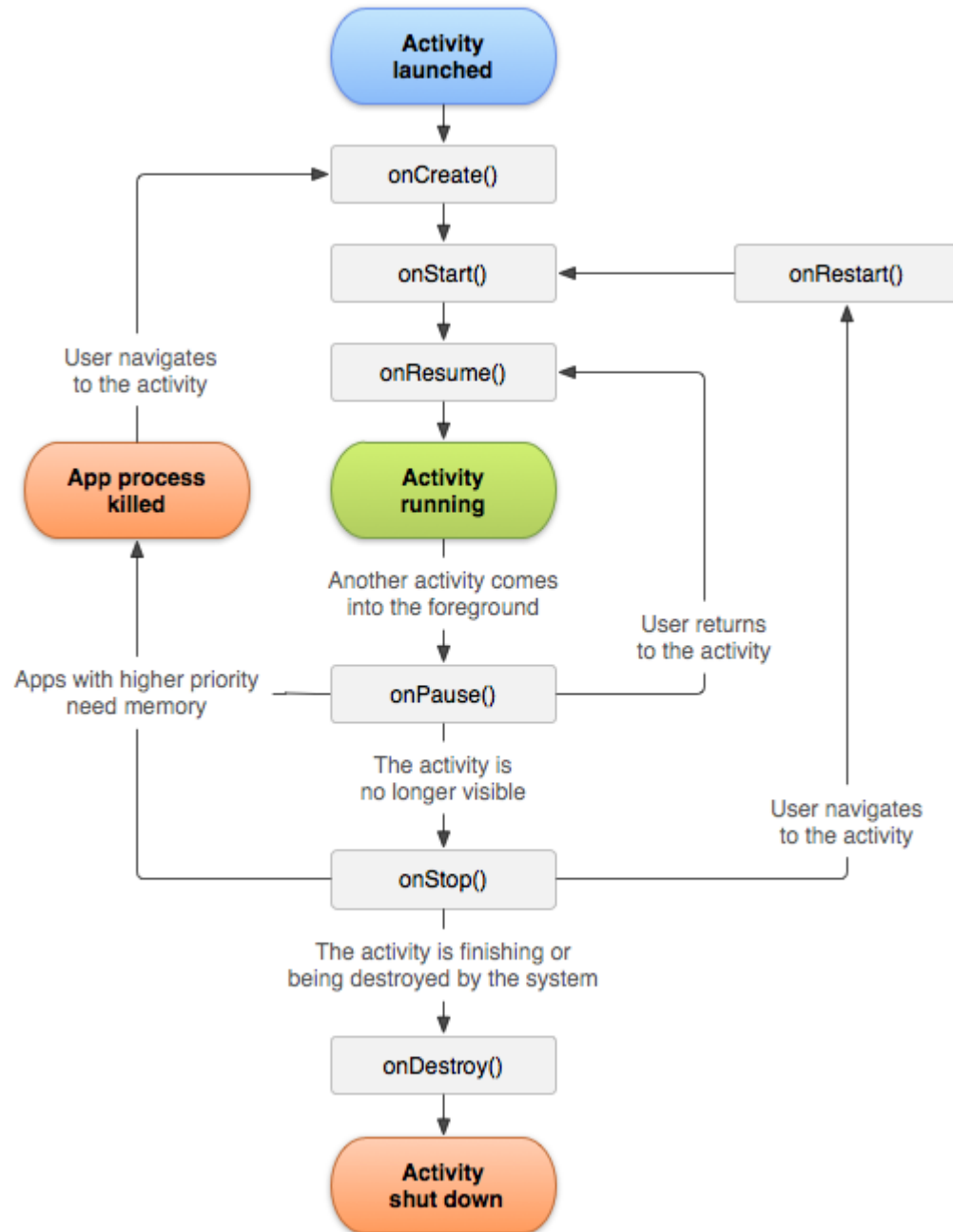
Activity

Activity là thành phần của ứng dụng, hiển thị màn hình giao diện, cho phép người dùng tương tác.

Các Activity được quản lý dưới dạng Activity Stack-Last In First Out. Khi một Activity mới được khởi tạo, nó nằm trên đỉnh stack, các Activity trước đó muốn chạy trên nền trở lại cần phải chờ Activity mới này kết thúc.

Một Activity có 4 trạng thái :

- Active/Running: Khi Activity đang chạy trên màn hình.
- Paused: Khi một Activity vẫn đang chạy trên màn hình nhưng bị một Activity khác chiếm toàn màn hình hiển thị phía trên.
- Stopped: Khi một Activity bị che khuất hoàn toàn bởi một Activity khác. Nó vẫn có thể được đưa trở lại để trở thành một Activity đang chạy (running) hoặc nó cũng có thể bị phá hủy và loại bỏ khỏi bộ nhớ.
- Destroyed hay Shut down: Khi một Activity đang paused hay stopped, hệ thống sẽ xóa Activity ấy ra khỏi bộ nhớ.



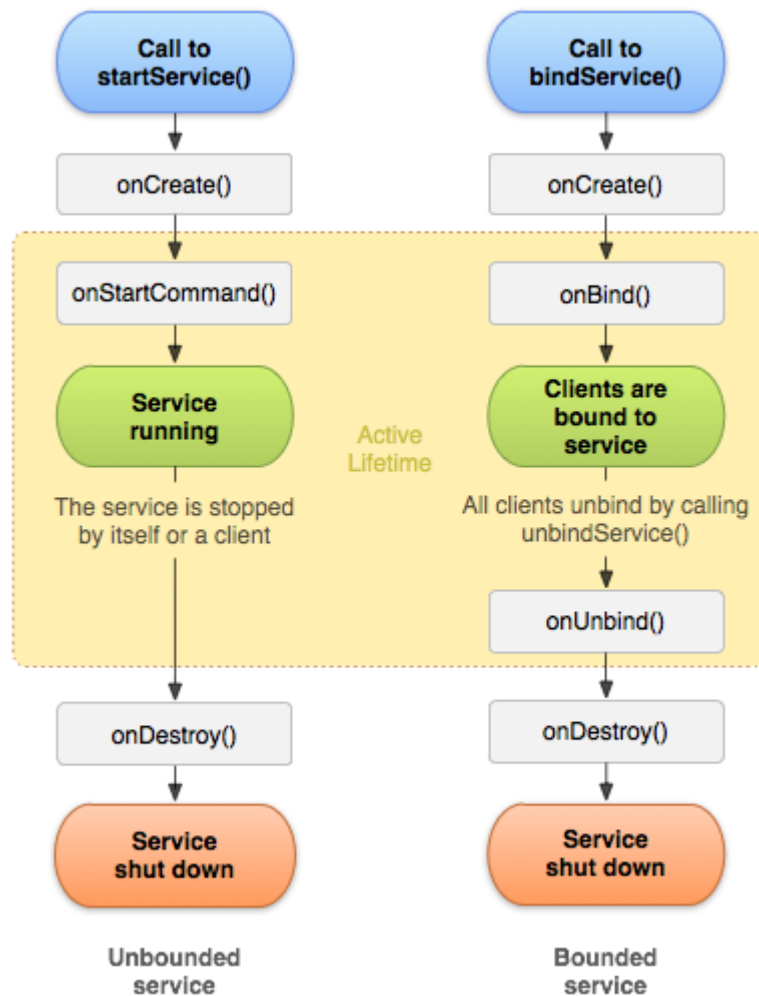
○

Hình 3 Chu trình sống của Activity

Service

Service là các tác vụ chạy ngầm dưới hệ thống nhằm thực hiện một nhiệm vụ nào đó. Service được sử dụng để cập nhật dữ liệu, đưa ra các cảnh báo.

Có 2 cách để tạo Service: bằng cách gọi phương thức Context.startService() hoặc Context.bindService().



Hình 4 Cách tạo Service

- Service được bắt đầu bởi phương thức `startService()` dùng để thực hiện các tác vụ ngầm dưới nền.
- Service được bắt đầu bởi phương thức `bindService()` dùng để cung cấp các chức năng cho các chương trình khác.

Broadcast receiver :

Broadcast receiver là một thành phần thu nhận các Intent bên ngoài gửi tới. Có thể tạo instance cho lớp này bằng 2 cách: Context.registerReceiver() hoặc thông qua tag <receiver> trong file AndroidManifest.xml.

Content Provider :

Content Provider được sử dụng để quản lý và chia sẻ dữ liệu giữa các ứng dụng.

Android cung cấp sẵn content providers cho 1 số kiểu dữ liệu thông dụng như âm thanh, video, danh bạ điện thoại,... Người lập trình cũng có thể tự tạo ra các class con (subclass) của Content Provider để lưu trữ kiểu dữ liệu của riêng mình.[7]

2.2 GooleMap API

Google Map một ứng dụng bản đồ tuyệt vời mà phổ biến nhất hiện nay được phát triển từ Google. Thêm một điểm cộng nữa đó là nó hoàn toàn miễn phí. Google Map hầu như có mặt trên tất cả các thiết bị di động thông minh. Và trên các ứng dụng bản đồ trên web tại Việt Nam. Chính vì vậy việc lựa chọn GooleMap sẽ có được sự thiện cảm với người dùng khi sử dụng ứng dụng.

Google Maps API giúp dễ dàng thêm các chức năng bản đồ vào trong ứng dụng. Google Maps API là một tiện ích được thêm vào trong thư viện phụ (external library). Các thư viện viết sẵn sẽ hỗ trợ các tính năng như tải bản đồ về, lưu , vẽ và rất nhiều các tùy chọn về hiển thị và điều khiển.

Lớp quan trọng nhất trong thư viện là lớp MapView, là subclass của lớp ViewGroup trong thư viện chuẩn của Android. MapView sẽ hiển thị bản đồ với dữ liệu nhận được từ dịch vụ Google Maps. Khi MapView được chọn , nó có thể được xoay và phóng to thu nhỏ bằng cách nhấn các phím một cách tự động , bao gồm cả việc xử lý các yêu cầu của mạng cho bản đồ. Bên cạnh đó nó cũng cung cấp tất cả các yếu tố giao diện người dùng cần thiết cho người sử dụng để kiểm soát bản đồ. Ứng dụng của bạn cũng có thể sử dụng các phương pháp điều khiển lớp để điều khiển lớp MapView và có thể vẽ lên trên lớp MapView nhiều lớp Overlay khác. Đây là một tính năng rất quan trọng.

Nói chung , lớp MapView là lớp bao quanh Google Maps API , nó có thể giúp người lập trình điều khiển dữ liệu của Google Maps thông qua các phương thức của lớp. Giúp cho bạn làm việc với bản đồ như thể là làm việc với các View trong Android.

Thư viện bản đồ do Google Maps API cung cấp không phải là thư viện chuẩn của Android, vì thế nó có thể không hỗ trợ một số thiết bị Android. Tương tự như vậy thư viện Google Maps API cũng không bao gồm thư viện chuẩn của Android trong bộ công cụ phát triển (SDK). Các Google APIs cung cấp các tiện ích, các thư viện về bản đồ giúp cho các lập trình viên có thể phát triển, xây dựng, và chạy các ứng dụng dựa trên công cụ phát triển Android SDK với đầy đủ quyền truy cập vào dữ liệu của Google Maps.

Google Maps tuy cung cấp API cho người phát triển nhưng cũng có một luật khá khắt khe. Đó là không được sử dụng API để xây dựng các ứng dụng để kinh doanh, hoặc xây dựng các ứng dụng có mục đích cạnh tranh với các sản phẩm của Google. Bị giới hạn về thông tin, và phải có logo của Google.[2]

2.3 Hệ cơ sở dữ liệu SQLite

Nói một cách đơn giản SQLite là phần mềm quản lý cơ sở dữ liệu (DBMS) tương tự như Mysql, PostgreSQL... Đặc điểm của SQLite là gọn, nhẹ, đơn giản. Chương trình gồm một file duy nhất vốn vẹn chưa đến 500kB, không cần cài đặt, không cần cấu hình hay khởi động mà có thể sử dụng ngay. Dữ liệu database cũng được lưu ở một file duy nhất. Không có khái niệm user, password hay quyền hạn trong SQLite database.

SQLite không thích hợp với những hệ thống lớn nhưng ở quy mô vừa tầm thì SQLite phát huy uy lực và không hề yếu kém về mặt chức năng hay tốc độ.

Mỗi ứng dụng đều sử dụng dữ liệu, dữ liệu có thể đơn giản hay đôi khi là cả một cấu trúc. Trong Android thì hệ cơ sở dữ liệu được sử dụng là SQLite Database, đây là hệ thống mã nguồn mở được sử dụng rộng rãi trong các ứng dụng và các hệ điều hành di động để lưu trữ thông tin vì tính hiệu quả của nó.

Trong Android, cơ sở dữ liệu mà bạn tạo cho một ứng dụng thì chỉ ứng dụng đó có quyền truy cập và sử dụng, các ứng dụng khác thì không. Điều này có ý nghĩa quan trọng trong việc bảo mật thông tin cho các ứng dụng. Nó cũng giúp phân chia cơ sở dữ liệu của các ứng dụng khác nhau ra riêng biệt. Tránh được những nhập nhằng khi truy xuất dữ liệu.

Một vài nét chính :

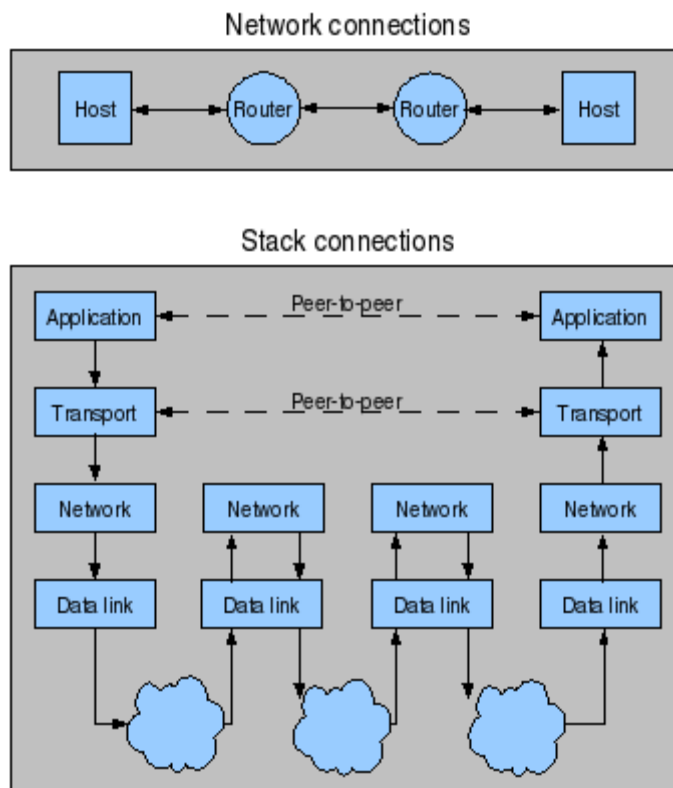
- Cơ sở dữ liệu dạng quan hệ (RDBMS), hỗ trợ chuẩn SQL-92

- Sử dụng dưới dạng thư viện nhúng, không chạy ở theo kiểu server độc lập
- Hỗ trợ các ngôn ngữ phổ biến: C, C++, C#, Basic, Perl, Ruby, Python, PHP, Java ...
- Không cần chỉ định kiểu dữ liệu (SQLite is typeless)
- Hỗ trợ mã UTF8
- Hỗ trợ command line
- Hỗ trợ transaction
- Hỗ trợ view
- Hỗ trợ C extensions
- Có các công cụ quản lý bằng đồ hoạ (xem chi tiết ở dưới)
- Download & sử dụng miễn phí

2.4 Giao thức TCP/IP

Bộ giao thức TCP/IP, ngắn gọn là TCP/IP (tiếng Anh: Internet protocol suite hoặc IP suite hoặc TCP/IP protocol suite - bộ giao thức liên mạng), là một bộ các giao thức truyền thông cài đặt chồng giao thức mà Internet và hầu hết các mạng máy tính thương mại đang chạy trên đó. Bộ giao thức này được đặt tên theo hai giao thức chính của nó là TCP (Giao thức Điều khiển Giao vận) và IP (Giao thức Liên mạng). Chúng cũng là hai giao thức đầu tiên được định nghĩa.

Mô hình OSI miêu tả một tập cố định gồm 7 tầng mà một số nhà sản xuất lựa chọn và nó có thể được so sánh tương đối với bộ giao thức TCP/IP. Sự so sánh này có thể gây nhầm lẫn hoặc mang lại sự hiểu biết sâu hơn về bộ giao thức TCP/IP.



Hình 5 Mô hình của TCP/IP

2.4.1 Các tầng trong TCP/IP

Sau đây là miêu tả từng tầng trong bộ giao thức TCP/IP.

Tầng ứng dụng

Tầng ứng dụng là nơi các chương trình mạng thường dùng nhất làm việc nhằm liên lạc giữa các nút trong một mạng.

Giao tiếp xảy ra trong tầng này là tùy theo các ứng dụng cụ thể và dữ liệu được truyền từ chương trình, trong định dạng được sử dụng nội bộ bởi ứng dụng này, và được đóng gói theo một giao thức tầng giao vận.

Hai giao thức tầng thấp thông dụng nhất là TCP và UDP. Mỗi ứng dụng sử dụng dịch vụ của một trong hai giao thức trên đều cần có cổng. Hầu hết các ứng dụng thông dụng có các cổng đặc biệt được cấp sẵn cho các chương trình phục vụ (server)(HTTP - Giao thức truyền siêu văn bản dùng cổng 80; FTP - Giao thức truyền tệp dùng cổng 21, v.v..) trong khi các trình khách (client) sử dụng các cổng tạm thời (ephemeral port).

Tầng giao vận

Trách nhiệm của tầng giao vận là kết hợp các khả năng truyền thông điệp trực tiếp (end-to-end) không phụ thuộc vào mạng bên dưới, kèm theo kiểm soát lỗi (error control), phân mảnh (fragmentation) và điều khiển lưu lượng. Việc truyền thông điệp trực tiếp hay kết nối các ứng dụng tại tầng giao vận có thể được phân loại như sau:

- định hướng kết nối (connection-oriented), ví dụ TCP
- phi kết nối (connectionless), ví dụ UDP

Tầng giao vận có thể được xem như một cơ chế vận chuyển thông thường, nghĩa là trách nhiệm của một phương tiện vận tải là đảm bảo rằng hàng hóa/hành khách của nó đến đích an toàn và đầy đủ.

Tầng giao vận cung cấp dịch vụ kết nối các ứng dụng với nhau thông qua việc sử dụng các cổng TCP và UDP. Do IP chỉ cung cấp dịch vụ phát chuyển nỗ lực tối đa (best effort delivery), tầng giao vận là tầng đầu tiên giải quyết vấn đề độ tin cậy.

Giao thức mới hơn, SCTP (Stream Control Transmission Protocol), cũng là một cơ chế giao vận định hướng kết nối "đáng tin cậy". Giao thức này định hướng dòng (stream-oriented), chứ không định hướng byte như TCP, và cung cấp nhiều dòng đa công (multiplexed) trên một kết nối. Nó còn hỗ trợ multi-homed, trong đó một đầu của kết nối có thể được đại diện bởi nhiều địa chỉ IP (đại diện cho nhiều giao diện vật lý), sao cho, nếu một giao diện vật lý thất bại thì kết nối vẫn không bị gián đoạn. Giao thức này ban đầu được phát triển dành cho các ứng dụng điện thoại (để vận chuyển SS7 trên giao thức IP), nhưng nó cũng có thể được sử dụng cho các ứng dụng khác.

Tầng mạng

Theo định nghĩa ban đầu, tầng mạng giải quyết các vấn đề dẫn các gói tin qua một mạng đơn. Một số ví dụ về các giao thức như vậy là X.25, và giao thức Host/IMP của mạng ARPANET.

Với sự xuất hiện của khái niệm liên mạng, các chức năng mới đã được bổ sung cho tầng này, đó là chức năng dẫn đường cho dữ liệu từ mạng nguồn đến mạng đích. Nhiệm vụ này thường đòi hỏi việc định tuyến cho gói tin qua một mạng lưới của các mạng máy tính, đó là liên mạng.

Trong bộ giao thức liên mạng, giao thức IP thực hiện nhiệm vụ cơ bản dẫn đường dữ liệu từ nguồn tới đích. IP có thể chuyển dữ liệu theo yêu cầu của nhiều giao thức tầng trên khác nhau; mỗi giao thức trong đó được định danh bởi một số hiệu giao thức duy nhất: giao thức ICMP (Internet Control Message Protocol) là giao thức 1 và giao thức IGMP (Internet Group Management Protocol) là giao thức 2.

Một số giao thức truyền bởi IP, chẳng hạn ICMP (dùng để gửi thông tin chẩn đoán về truyền dữ liệu bằng IP) và IGMP (dùng để quản lý dữ liệu đa truyền (multicast)), được đặt lên trên IP nhưng thực hiện các chức năng của tầng liên mạng, điều này minh họa một sự bất tương thích giữa liên mạng và chồng TCP/IP và mô hình OSI. Tất cả các giao thức định tuyến, chẳng hạn giao thức BGP (Border Gateway Protocol), giao thức OSPF, và giao thức RIP (Routing information protocol), đều thực sự là một phần của tầng mạng, mặc dù chúng có thể có vẻ thuộc về phần trên của chồng giao thức.

Tầng liên kết

Tầng liên kết - phương pháp được sử dụng để chuyển các gói tin từ tầng mạng tới các máy chủ (host) khác nhau - không hẳn là một phần của bộ giao thức TCP/IP, vì giao thức IP có thể chạy trên nhiều tầng liên kết khác nhau. Các quá trình truyền các gói tin trên một liên kết cho trước và nhận các gói tin từ một liên kết cho trước có thể được điều khiển cả trong phần mềm điều vận thiết bị (device driver) dành cho các mạng, cũng như trong phần sụn (firmware) hay các chipset chuyên dụng. Những thứ đó sẽ thực hiện các chức năng liên kết dữ liệu chẳng hạn như bổ sung một tín đầu (packet header) để chuẩn bị cho việc truyền gói tin đó, rồi thực sự truyền frame dữ liệu qua một môi trường vật lý.

Tầng liên kết còn có thể được xem là bao gồm cả tầng vật lý - tầng là kết hợp của các thành phần mạng vật lý thực sự (hub, các bộ lặp (repeater), cáp mạng, cáp quang, cáp đồng trục (coaxial cable), các mạng, các HBA (Host Bus Adapter) và các thiết bị nối mạng có liên quan: RJ-45, BNC, etc), và các đặc tả mức thấp về các tín hiệu (mức hiệu điện thế, tần số, v.v..).

2.4.2 Ưu nhược điểm và bảo mật

Ưu điểm

Là phương thức truyền thông tin cậy, đảm bảo cho các gói tin đến đúng thứ tự như đáp ứng yêu cầu của ứng dụng.

- Được dùng để thiết lập kết nối giữ nhiều loại thiết bị khác nhau theo mô hình server-client. Đồng thời là khả năng mở rộng của hệ thống.
- Là một chuẩn giao thức công nghiệp mở, không bị quản lý bởi một tổ chức công nghiệp nào.
- Gồm nhiều giao thức định tuyến cho phép truyền dữ liệu thông qua các mạng khác nhau.
- Hoạt động độc lập với hệ điều hành.
- Cung cấp rất nhiều dịch vụ cần thiết như phân giải tên miền, cấu hình địa chỉ máy chủ linh hoạt...

Nhược điểm

- Chi phí để duy kết nối TCP/IP là rất cao.
- TCP/IP khá phức tạp để có thể thiết lập và quản lý.

Bảo mật

Có rất nhiều phương pháp tấn công dựa trên TCP/IP để lấy được dữ liệu từ gói tin hoặc là giả mạo gói tin như : IP Spoofing , Connection Hijacking , tấn công vào DNS , hoặc tấn công DDOS...

Một số biện pháp bảo mật đã được thực hiện và đến nay vẫn đạt được độ tin cậy nhất định như mã hóa , hoặc dùng các phương pháp xác thực truy cập ...[4]

2.5 Định dạng dữ liệu GPX

GPX là định dạng dữ liệu XML cho việc trao đổi dữ liệu GPS giữa các ứng dụng và dịch vụ Web trên Internet. GPX là một định dạng mở, và có thể được sử dụng miễn phí.

```
<?xml version='1.0' encoding='UTF-8' standalone='yes' ?>
<gpx version="1.1" creator="Traffic Direction"
xsi:schemaLocation="http://www.topografix.com/GPX/1/1
http://www.topografix.com/gpx/1/1/gpx.xsd"
```

```

xmlns="http://www.topografix.com/GPX/1/1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:gpox10="http://www.topografix.com/GPX/1/0"
xmlns:ogt10="http://gpstracker.android.sogeti.nl/GPX/1/0">
<metadata>
  <time>2012-12-09T11:15:00+0700</time>
</metadata>
<trk>
  <name>Track2012-12-9 11-15</name>
  <trkpt lat="10.753101666666664" lon="106.663136666666669">
    <ele>0.0</ele>
    <time>2012-12-09T10:54:23+0700</time>
    <extensions>
      <ogt10:accuracy>0</ogt10:accuracy>
      <speed>0.0</speed>
    </extensions>
  </trkpt>
</trk>
</gpx>

```

Bảng 3 Định dạng tập tin GPX

- Thẻ xml: bắt đầu file GPX
- Thẻ gpx: là phần tử gốc trong file GPX
- Thẻ trk: chứa thông tin track, bao gồm nhiều điểm.
- Thẻ trkpt: chứa thông tin điểm, bao gồm kinh độ, vĩ độ, độ cao, thời gian track, và các thông số do người dùng định nghĩa như vận tốc, độ chính xác.

2.6 OpenStreetMap

OpenStreetMap là một dự án bản đồ mã nguồn mở ra đời vào năm 2004. Nó tồn tại theo cách mà Wikipedia đã sống tới ngày hôm nay tức những tình nguyện viên sẽ cung cấp thông tin bản đồ để xây dựng một dịch vụ bản đồ miễn phí và mã nguồn mở. Vào năm 2006, quỹ OpenStreetMap (OSMF) đã được thành lập để khuyến khích sự phát triển và phân phối OpenStreetMap. Quỹ này cho biết có khoảng nửa triệu tình nguyện viên đã cung cấp dữ liệu bản đồ cho OpenStreetMap.

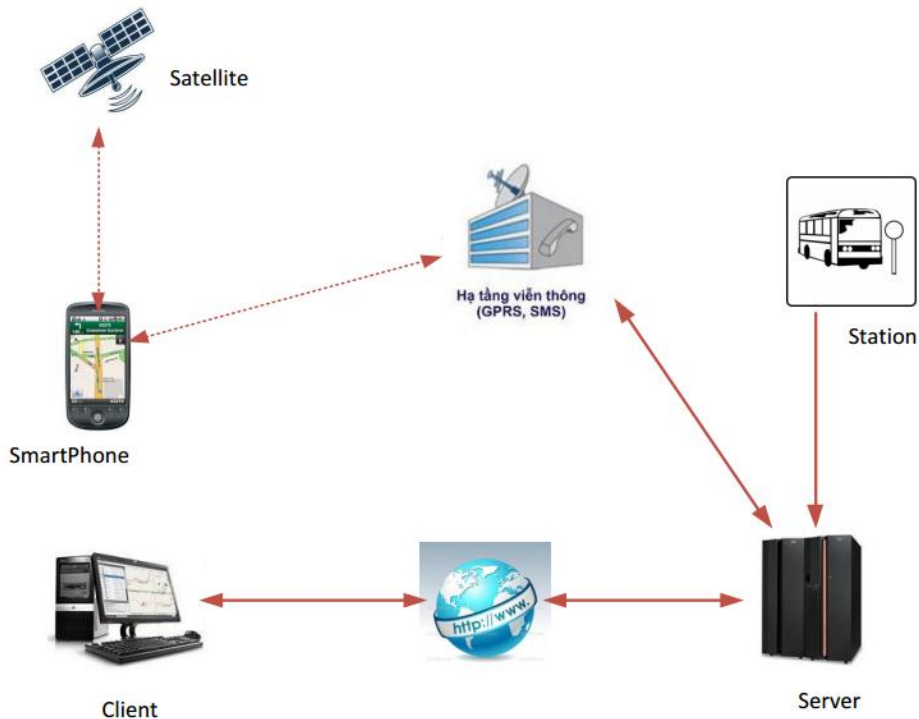
Người dùng có thể tải tất cả dữ liệu trên OpenStreetMap xuống để phát triển các ứng dụng cho riêng mình hoặc để kinh doanh mà không phải trả bất cứ chi phí nào.

Tuy nhiên dịch vụ bản đồ này cũng có nhiều hạn chế đó là dữ liệu của nó chưa lớn, và chưa thể cạnh tranh với các dịch vụ nổi tiếng hiện nay như GoogleMap hoặc bản đồ của Microsoft, hoặc Nokia ... Thêm vào đó dịch vụ bản đồ này hầu như xa lạ với đại đa số người sử dụng tại Việt Nam.

PHẦN 3 : THIẾT KẾ ỨNG DỤNG

3.1 Phân tích hệ thống chính

3.1.1 Các thành phần tham gia hệ thống



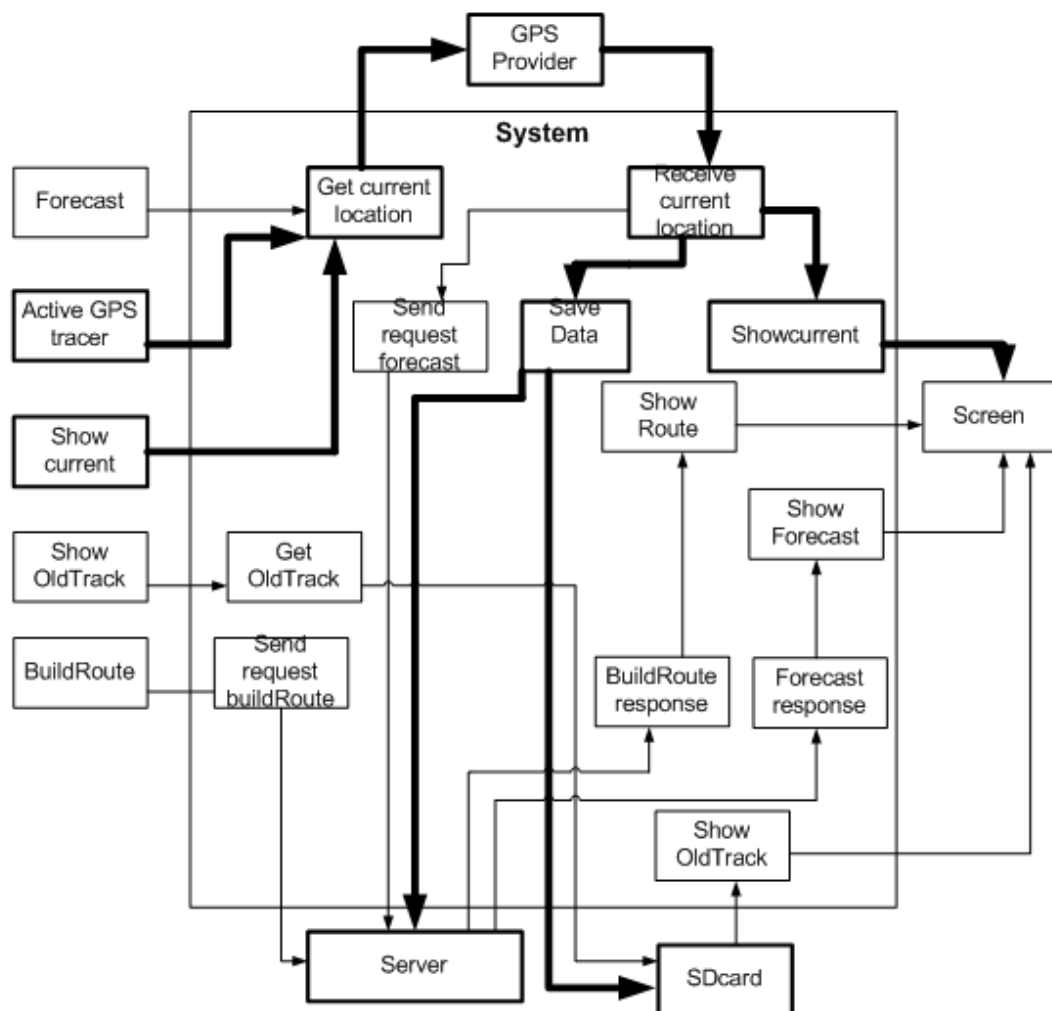
Hình 6 Các thành phần tham gia hệ thống

Hệ thống bao gồm :

- Smart Phone : là các smartphone chạy hệ điều hành Android có hỗ trợ GPS. Các smart phone với nhiệm vụ chính là thu thập dữ liệu thu được từ GPS gửi về cho Server thông qua các hạ tầng viễn thông như Viettel , MobilePhone.. dữ liệu được gửi về thông qua 3G hoặc GPRS.Ngoài ra các smart phone cũng có nhiệm vụ hiển thị bản đồ dưới dạng GUI cho người dùng, nhận yêu cầu từ người dùng gửi về Server. Và nhận dữ liệu trả về từ Server và hiển thị lại cho người dùng.
- Satellite : Cung cấp định vị cho thiết bị di động , và các phương tiện giao thông có trang bị GPS.
- Hạ tầng viễn thông : Các nhà cung cấp dịch vụ viễn thông với nhiệm vụ cung cấp kết nối giữa các thiết bị smartphone với Server.

- Station : Ở đây là các trung tâm xe bus, xe taxi ... hoặc các hãng vận tải. Sẽ làm nhiệm vụ cung cấp dữ liệu cho Server.
- Server: Đây là thành phần tính toán chính của hệ thống .Nhiệm vụ xử lý dữ liệu nhập vào , vì dữ liệu từ nhiều nguồn khác nhau nên phải xử lý để thống nhất, chính xác.Chạy các giải thuật để dự đoán tình hình giao thông trong khoảng thời gian xác định.Tìm ra đường đi nhanh nhất dựa vào các dữ liệu đã nhận được...Đáp ứng các yêu cầu của người dùng.
- Client : Là các máy tính cá nhân thông thường.Với chức năng có thể thị tình hình giao thông , dự báo kẹt xe , tìm đường đi với thời gian ngắn nhất.

3.1.2 Sơ đồ hoạt động của hệ thống



Hình 7 Mô hình hoạt động của hệ thống

Phía bên trái của lược đồ là các chức năng chính của hệ thống.

Phía bên phải của lược đồ là màn hình hiển thị lên các kết quả sau khi tính toán được của hệ thống.

Ở giữa của lược đồ là toàn bộ quá trình xử lý của hệ thống khi nhận được các yêu cầu từ phần chức năng của hệ thống.

Các bước hoạt động các chức năng của hệ thống :

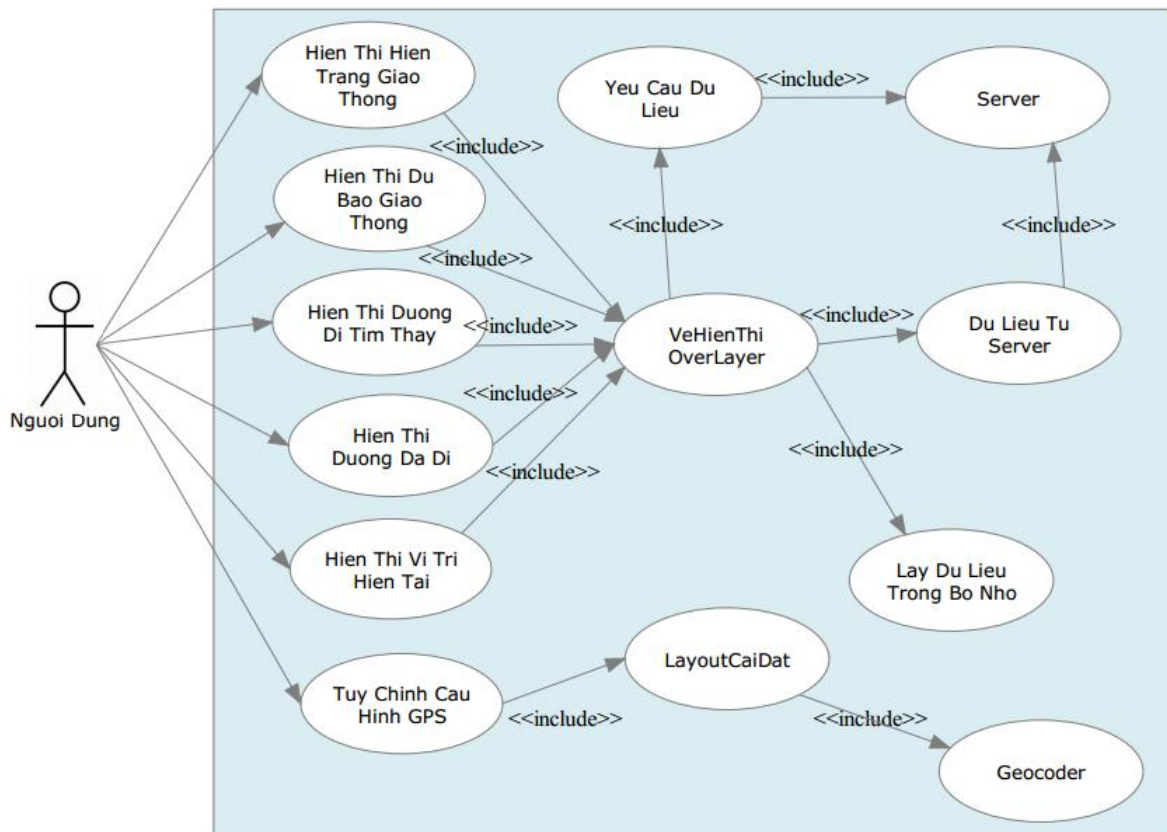
- Chức năng hiển thị dự đoán (forecast). Khi chức năng này được chọn. Bộ phận định vị sẽ lấy vị trí hiện tại của người dùng thông qua định vị GPS và nhà cung cấp GPS. Hệ thống sẽ nhận yêu cầu và gửi một yêu cầu dự đoán hiện trạng giao thông đến Server. Sau khi xử lý xong dữ liệu phía server, server sẽ gửi trả dữ liệu lại cho ứng dụng phía người dùng, cũng chính là module hiển thị. Sau khi Module này xử lý xong dữ liệu nhận về sẽ hiển thị lên trên màn hình cho người dùng.
- Chức năng lưu trữ đường đi của người dùng (Active GPS tracer). Đầu tiên hệ thống sẽ thực hiện việc lấy vị trí hiện tại của người dùng thông qua các hệ thống GPS được các nhà cung cấp hỗ trợ. Hệ thống sẽ nhận yêu cầu của người dùng. Sau đó thực hiện việc lưu trữ dữ liệu. Dữ liệu sẽ được lưu trữ ở hai phía cả server và client. Ở phía client dữ liệu sẽ được lưu vào các thẻ nhớ SD.
- Chức năng hiển thị các đường đã đi qua. Về phía ứng dụng người dùng di động. Ứng dụng sẽ thực hiện yêu cầu bằng cách lấy danh sách các tracklist đã được lưu trữ trong thẻ nhớ hiển thị lên màn hình cho người dùng chọn. Sau khi người dùng chọn được đường đã đi trong danh sách. Ứng dụng sẽ hiển thị lên màn hình cho người dùng. Về phía server, sau khi người dùng chọn chức năng này, ứng dụng phía server (ứng dụng viết trên nền web) cũng sẽ hiển thị ra một danh sách các đường mà người dùng đã đi khi sử dụng ứng dụng trên thiết bị di động. Sau khi người dùng chọn được đường đi đã lưu. Ứng dụng sẽ hiển thị lên màn hình cho người dùng.
- Chức năng hiển thị vị trí hiện tại. Khi người dùng mở ứng dụng lên. Nếu GPS đã được bật sẵn. Hệ thống sẽ lấy vị trí người dùng thông qua định vị GPS được các nhà cung cấp hỗ trợ. Sau khi nhận được vị trí hiện tại của người dùng. Ứng dụng trên web sẽ hiển thị vị trí của người dùng lên giao diện web. Ứng dụng trên thiết bị di động sẽ hiển thị vị trí của người

dùng lên thiết bị di động với lớp nền là Google Map. Nếu GPS chưa được bật ứng dụng sẽ hiển thị vị trí mặc định là cổng trường Đại Học Bách Khoa Thành Phố Hồ Chí Minh.

- Chức năng tìm đường (build route) .Hệ thống có thể nhận được yêu cầu từ phía người dùng từ ứng dụng trên nền web hoặc trên thiết bị di động. Sau khi nhận được yêu cầu với chức năng tìm đường đi của người dùng. Hệ thống sẽ nhận yêu cầu tìm đường gửi lên phía server . Server sẽ xử lý yêu cầu và trả về dữ liệu hiển thị lên ứng dụng web hoặc ứng dụng trên di động tương ứng.

3.2 Phân tích ứng dụng trên Android

3.2.1 Lược đồ User-Case cho module ứng dụng



Hình 8 Lược đồ Use-Case của ứng dụng

Các chức năng chính của ứng dụng :

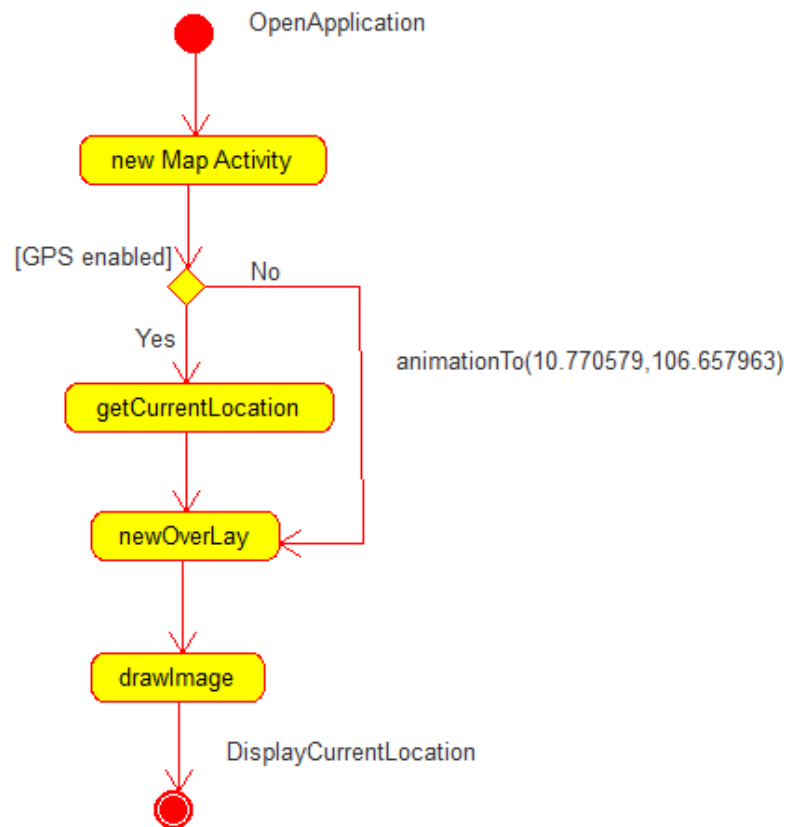
- Chức năng hiển thị hiện trạng giao thông : Chức năng này để xem hiện trạng giao thông của thành phố tại một thời điểm. Mục đích của tính năng này giúp người dùng có thể xác

định được đâu là các điểm có lưu lượng xe tham gia giao thông đông, để tránh các điểm đó khi di chuyển.

- Chức năng hiển thị dự báo tình trạng kẹt xe : Chức năng này giúp người dùng có thể biết trước được các điểm kẹt xe trong khoảng thời gian từ 10 phút đến 120 phút trong tương lai.
- Chức năng hiển thị đường đi tìm được : Khi người dùng chọn chức năng này , ứng dụng sẽ đưa ra hộp thoại để người dùng nhập địa chỉ điểm bắt đầu và điểm kết thúc bằng cách nhấn chọn trên bản đồ. Sau khi nhập xong ứng dụng sẽ gửi dữ liệu lên server. Và nhận kết quả trả về từ server hiển thị lên bản đồ cho người dùng.
- Hiển thị vị trí hiện tại người dùng : Chức năng này giúp người dùng có thể biết được vị trí hiện tại của mình trên bản đồ.
- Hiển thị đường đã đi qua(tracking) : Chức năng này giúp người dùng có thể xem lại đoạn đường mình đã đi qua, với vận tốc của từng quãng đường. Từ chức năng này người dùng có thể biết được đoạn đường nào đi với tốc độ cao, đoạn đường nào với tốc độ thấp, và rút kinh nghiệm cho lần sau khi đi qua đoạn đường đó.

3.2.2 Lược đồ lược đồ hoạt động cho các chức năng của ứng dụng

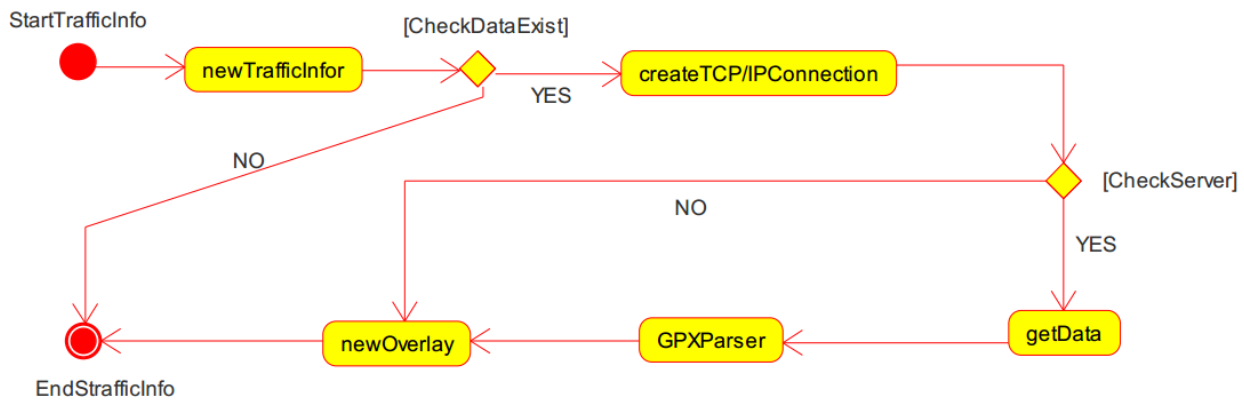
Lược đồ hoạt động hiển thị vị trí hiện tại của người dùng



Hình 9 Lược đồ hoạt động hiển thị vị trí hiện tại.

Khi ứng dụng mới được mở lên , ứng dụng sẽ kiểm tra xem GPS được bật lên chưa , nếu GPS đã được bật lên, ứng dụng sẽ dùng GPS lấy vị trí hiện tại của người dùng ,hiển thị lên Overlay. Nếu GPS chưa được bật , Ứng dụng mặc định sẽ hiển thị ngã tư Tô Hiến Thành – Lý Thường Kiệt.

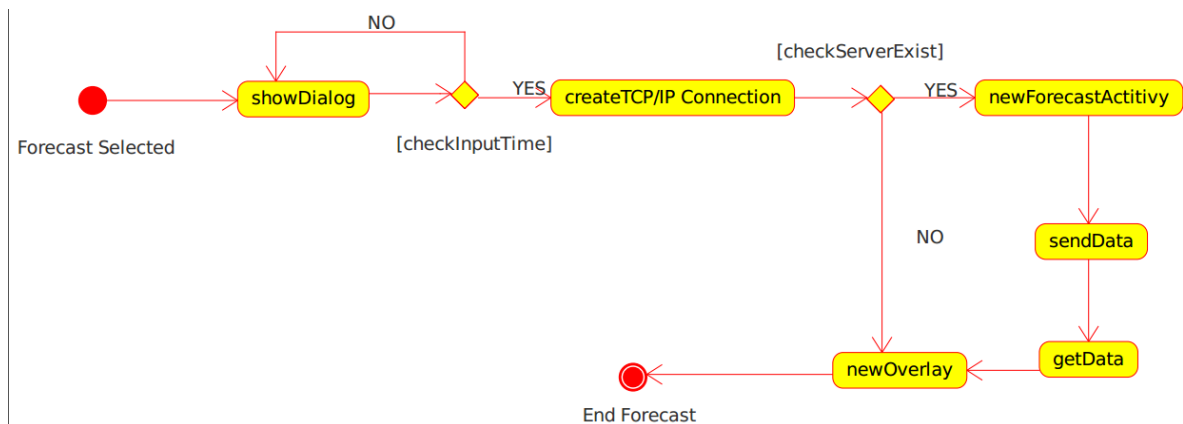
Lược đồ hoạt động cho chức năng hiển thị hiện trạng giao thông



Hình 10 Lược hoạt động hiển thị hiện trạng giao thông.

Khi người dùng chọn chức năng xem hiện trạng giao thông từ menu của ứng dụng. Một activity mới sẽ được cấp phát, Ứng dụng sẽ kiểm tra xem dữ liệu đã tồn tại chưa, nếu chưa tồn tại ứng dụng sẽ quay trở lại menu ban màn hình chính của ứng dụng. Nếu tồn tại ứng dụng sẽ thực hiện kết nối TCP/IP đến server nếu kết nối thành công ứng dụng sẽ gửi yêu cầu lên server, và nhận lại dữ liệu, sẽ dùng GPX để đọc dữ liệu và hiển thị dữ liệu lên Overlay, nếu không thực hiện được kết nối đến server ứng dụng sẽ hiển thị thông báo không kết nối đến server được trên Overlay mới.

Lược đồ hoạt động cho chức năng hiển thị dự báo kẹt xe

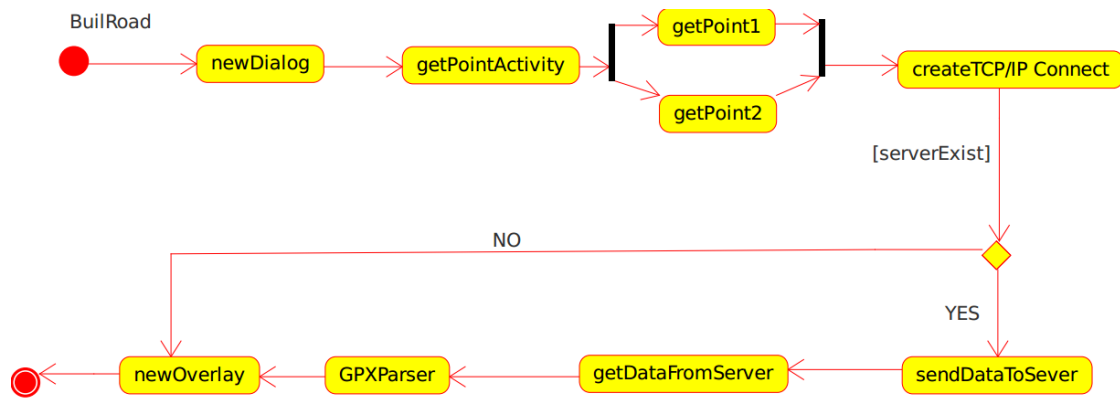


Hình 11 Lược đồ hoạt động hiển thị dự báo kẹt xe.

Khi người dùng chọn chức năng dự báo kẹt xe thông qua menu của ứng dụng. Ứng dụng sẽ cho phép người dùng chọn khoảng thời từ 10 phút đến 120 phút trên hộp thoại, và gửi yêu cầu lên

server, sau khi xử lý xong server sẽ gửi thông tin xuống. Hoạt động nhận dữ liệu sẽ đọc dữ liệu và hiển thị lên lớp Overlay của bản đồ.

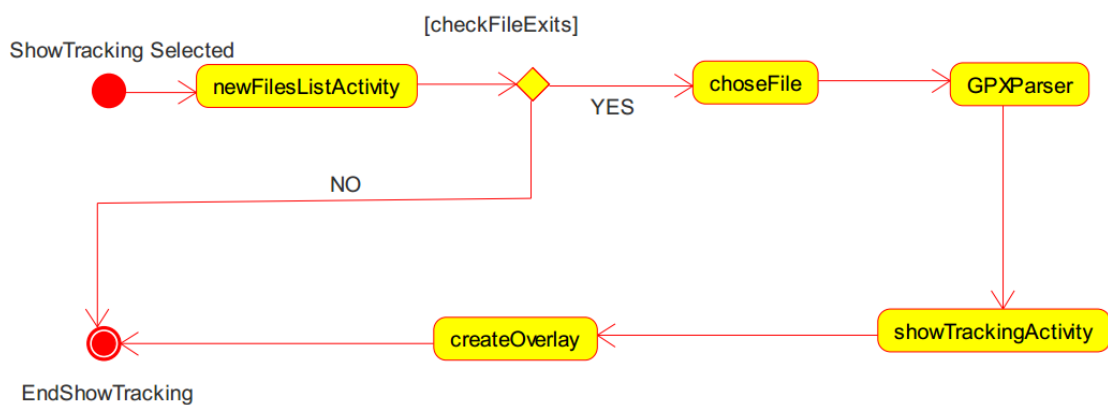
Lược đồ hoạt động cho chức năng hiển thị tìm đường đi



Hình 12 Lược đồ hoạt động chức năng hiển thị tìm đường đi.

Khi người dùng chọn chức năng tìm đường đi. Tại menu của ứng dụng. Ứng dụng sẽ hiển thị lên hộp thoại chọn đường đi cho người dùng, người dùng sẽ phải chọn điểm bắt đầu và điểm kết thúc bằng cách chọn trên map. Hoạt động tạo kết nối TCP/IP sẽ tạo kết nối đến server, nếu kết nối thành công. Sau đó ứng dụng sẽ gửi yêu cầu lên server. Server sẽ nhận và xử lý dữ liệu sau đó gửi lại ứng dụng. Ứng dụng hiển thị lên đường đi cho người dùng trên map.

Lược đồ hoạt động chức năng hiển thị Tracking

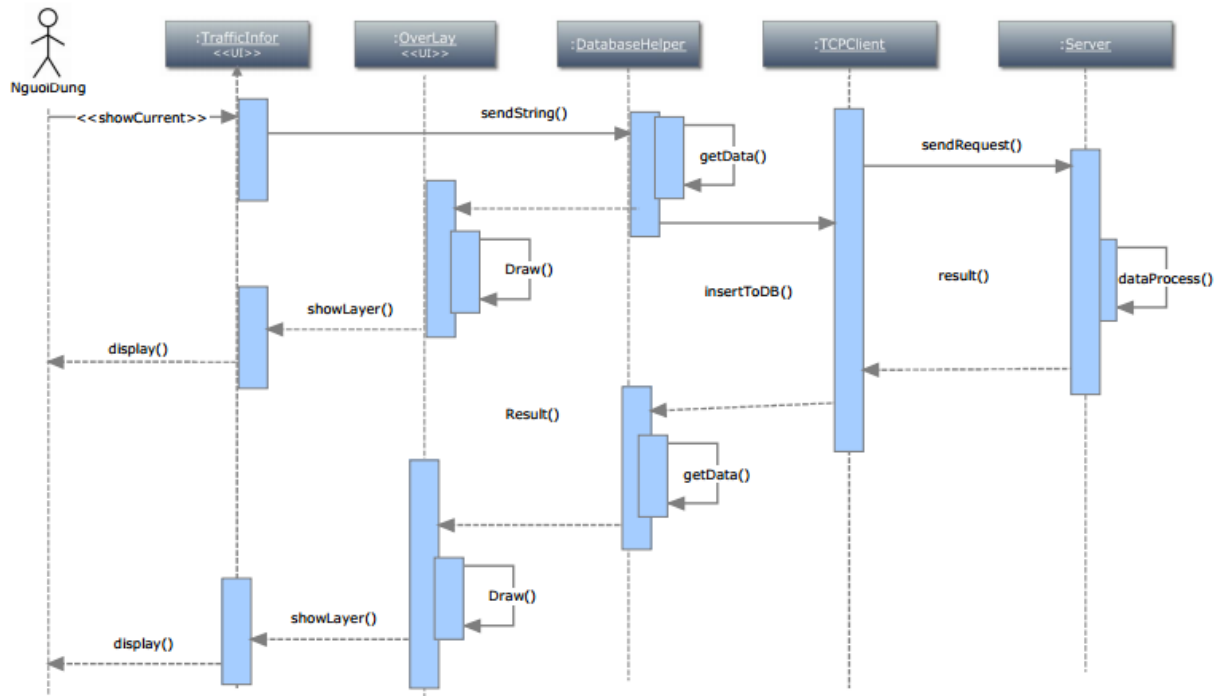


Hình 13 Lược đồ hoạt động chức năng hiển thị đường đã đi.

Khi người dùng chọn chức năng hiển thị tracking từ menu của ứng dụng. Một màn hình mới sẽ được tạo ra, màn hình này sẽ hiển thị danh sách các tập tin theo vết đã được người dùng lưu trữ. Ứng dụng sẽ hiển thị ra một danh sách các file tracking mà người dùng đã track lại. Người dùng sẽ chọn tập tin lưu đường đã đi muốn hiển thị. Kết quả là quãng đường đã đi qua của người dùng với vận tốc được tính trong từng khoảng. Từ đó người dùng có thể xem lại các vận tốc các đoạn đường đã đi qua để có thể có lựa chọn tốt hơn trong lần sau đi trên đoạn đường đó.

3.3 Lược đồ tuần tự

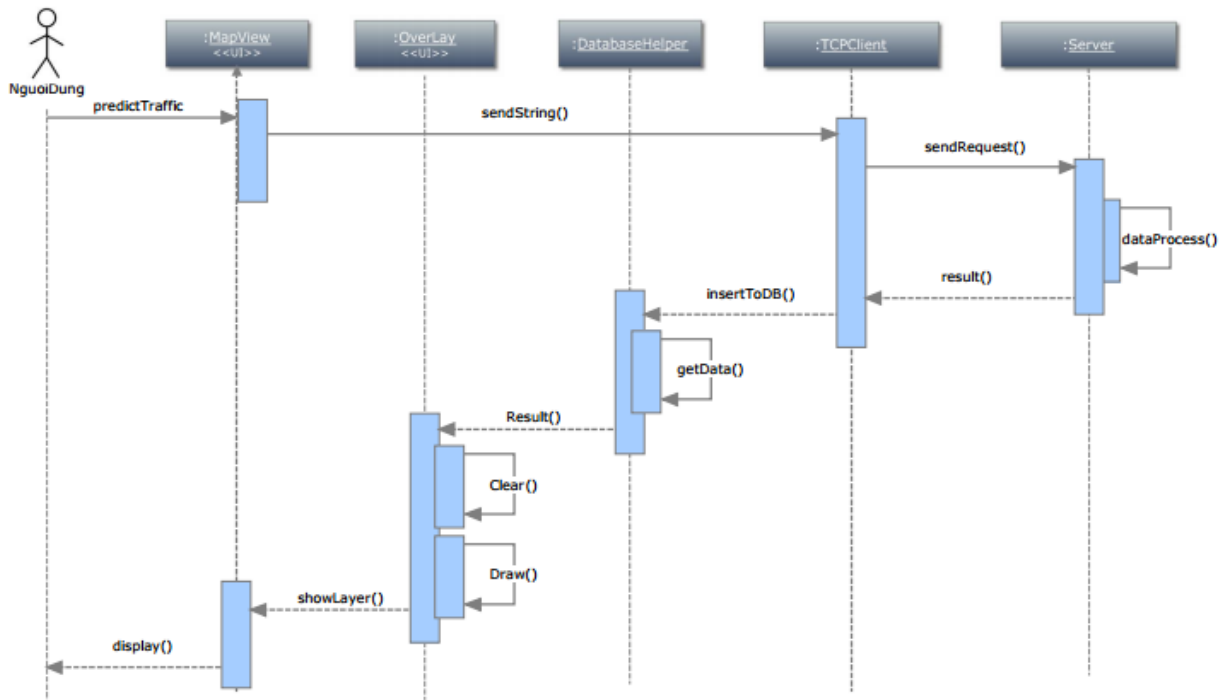
Chức năng hiển thị hiện trạng giao thông.



Hình 14 Lược đồ tuần tự hiển thị hiện trạng giao thông

Sau khi người dùng chọn chức năng hiển thị trạng thái hiện tại của tình trạng giao thông trong menu của lớp MapView. ,Lớp TrafficInfor sẽ thực hiện việc kiểm tra dữ liệu bằng lớp DatabaseHelper nếu dữ liệu mới hơn 15 phút . Thì sẽ lấy dữ liệu đó hiển thị lên lớp Overlay, Ngược lại nếu dữ liệu đã quá 15 phút .Lớp TrafficInfor sẽ gửi yêu cầu TCPClient gửi về cho Server dưới dạng String. thông qua giao thức TCP/IP. Server nhận được yêu cầu sẽ xử lý và gửi kết quả lại cho ứng dụng thông qua phương thức TCP/IP. Lớp TCPClient sẽ nhận yêu cầu và xử lý dữ liệu sau đó gửi cho lớp DatabaseHelper().Lớp này có nhiệm vụ ghi xuống cơ sở dữ liệu . Nếu dữ liệu mới nhận được có thay đổi so với dữ liệu trong CSDL thì dữ liệu trong CSDL sẽ được cập nhật lại. Sau khi cập nhật lại CSDL.Dữ liệu sẽ được lớp OverLayer truy xuất. Lớp OverLay sẽ gọi hàm Clear() để xóa toàn bộ các Layer có trước và vẽ Layer hiển thị hiện trạng giao thông bằng dữ liệu mới nhận được lên.Sau khi vẽ xong lớp OverLayer sẽ được hiển thị bên trên lớp MapView.

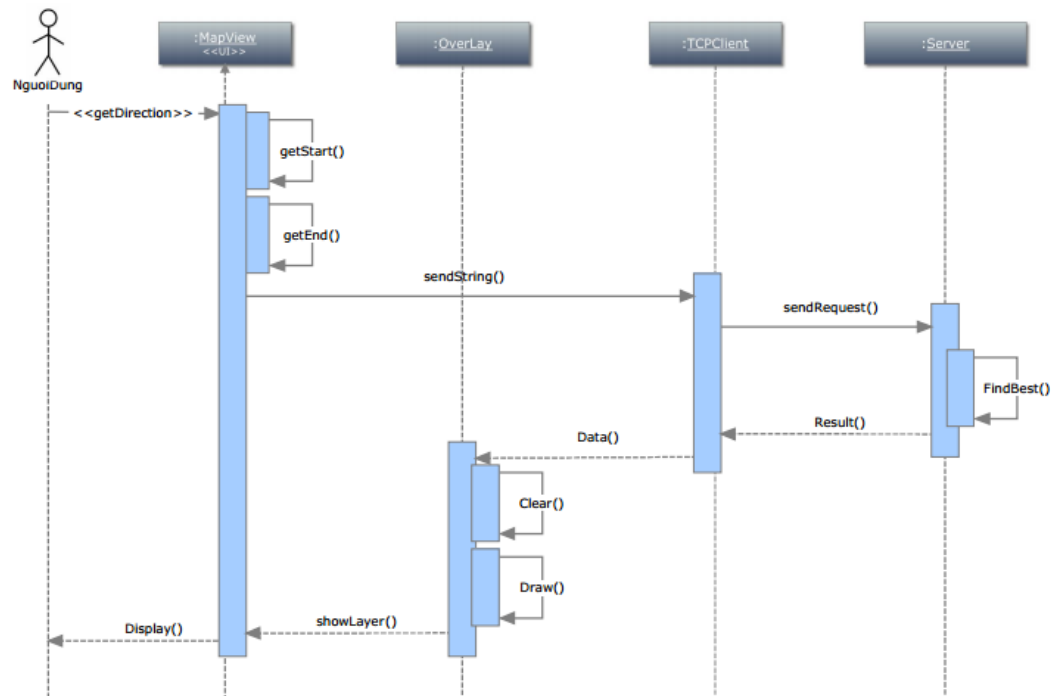
Chức năng dự báo kẹt xe



Hình 15 Lược đồ tuần tự hiển thị dự báo kẹt xe.

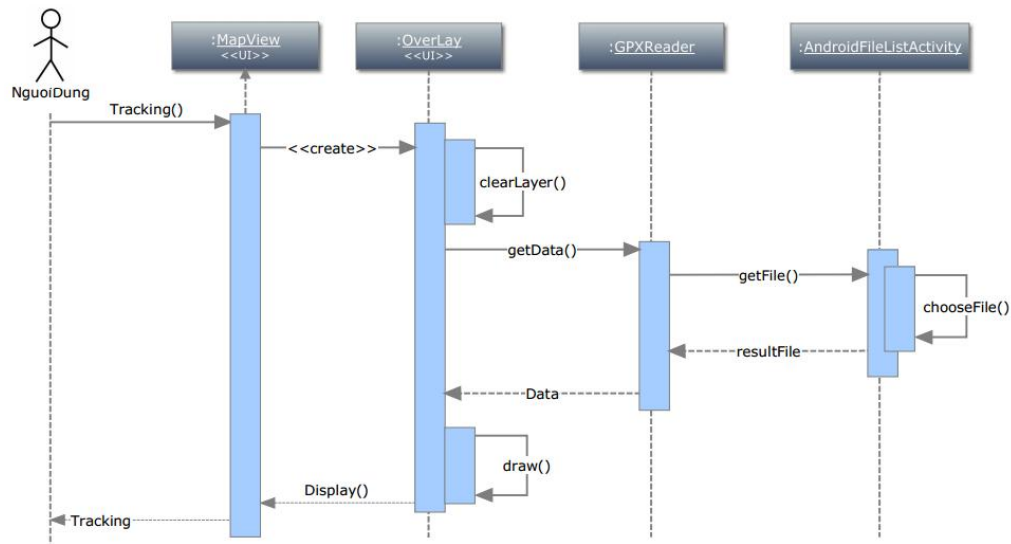
Sau khi người dùng chọn chức năng dự báo hiện trạng giao thông trong một khoảng thời gian gần trong tương lai trong menu của lớp MapView. Yêu cầu sẽ được ghi nhận và gửi về cho lớp TCPClient. Dữ liệu sẽ được lớp TCPClient gửi về cho Server dưới dạng String thông qua giao thức TCP/IP. Server nhận được yêu cầu sẽ xử lý và gửi kết quả lại cho ứng dụng thông qua phương thức TCP/IP. Lớp TCPClient sẽ nhận yêu cầu và xử lý dữ liệu sau đó gửi cho lớp DatabaseHelper(). Lớp này có nhiệm vụ so sánh dữ liệu hiện có trong cơ sở dữ liệu với dữ liệu mới nhận được. Nếu dữ liệu mới nhận được có thay đổi so với dữ liệu trong CSDL thì dữ liệu trong CSDL sẽ được cập nhật lại. Sau khi cập nhật lại CSDL. Dữ liệu sẽ được lớp Overlay truy xuất. Lớp Overlay sẽ gọi hàm Clear() để xóa toàn bộ các Layer có trước và vẽ Layer hiển thị hiện trạng giao thông bằng dữ liệu mới nhận được lên. Sau khi vẽ xong lớp Overlay sẽ được hiển thị bên trên lớp MapView.

Về căn bản yêu cầu dự báo hiện trạng giao thông trong tương lai cũng giống như yêu cầu dự báo hiện trạng giao thông hiện tại nhưng có khác thời gian.

Chức năng tìm đường.

Hình 16 Lược đồ tuần tự chức năng hiển thị tìm đường

Khi người dùng chọn chức năng tìm đường khi nhấn nút menu. Menu hoạt động trên lớp MapView. Từ menu người dùng sẽ chọn điểm bắt đầu và điểm kết thúc. Sau đó sẽ gửi yêu cầu dạng String lại cho lớp TCP Client xử lý. Lớp TCPClient sẽ thực hiện gửi một yêu cầu lên trên server. Sau khi Server nhận yêu cầu. Server sẽ xử lý dữ liệu và tính toán ra đường đi với thời gian ngắn nhất, Dữ liệu tìm được sẽ được chuyển về dạng String sau đó gửi xuống lại cho Ứng dụng. Lớp TCPClient sẽ làm nhiệm vụ nhận thông tin trả về từ phía Server, sau khi xử lý dữ liệu. Dữ liệu sẽ được gửi xong lớp Overlay. Lớp Overlay sẽ thực hiện hàm Clear() với mục đích là xóa tất cả các Layer có trước đó. Sau đó sẽ vẽ đường đi dựa trên dữ liệu mới nhận được và hiển thị lên lớp MapView.

Chức năng hiển thị tracking của người dùng.

Hình 17 Lược đồ tuần tự hiển thị đường đã đi

Người dùng chọn chức năng hiển thị tracking từ menu trong lớp MapView . Lớp MapView sẽ gửi yêu cầu tạo Overlay() và xóa hết các Overlay cũ đi . Sau đó yêu cầu được lớp MapView gửi lên GPXReader() để lấy dữ liệu vẽ. Tại lớp GPXReader() sẽ gửi yêu cầu đến lớp AndroidListFilesActiviy tại lớp này người dùng sẽ chọn file Tracking trong quá khứ để hiển thị. Sau khi file được chọn xong sẽ được chuyển về cho lớp GPXReader() lớp này sẽ dùng SAX để parser dữ liệu GPX dạng XML ra dữ liệu thông thường (dữ liệu bao gồm các trường kinh độ , vĩ độ , thời gian, vận tốc) và gửi về cho lớp Overlay(). Lớp Overlay() sẽ làm nhiệm vụ nhận dữ liệu gửi về và vẽ lên Overlay(). Sau khi vẽ xong sẽ hiển thị lên MapView cho người dùng.

PHẦN 4 : HIỆN THỰC ỨNG DỤNG

Công cụ hiện thực ứng dụng : Eclipse IDE , Android SDK cho Eclipse, SQLite, GoogleMapAPI, OpenStreetMap.

4.1 Dữ liệu về đường đi

GoogleMap API chỉ cung cấp cho người dùng một số chức năng như :

- Chiều một điểm có tọa độ xuống bản đồ.
- Cho người dùng lấy dữ liệu về độ zoom của bản đồ.

Để hiện thực việc vẽ đường đi chúng tôi lựa chọn giải pháp lấy dữ liệu từ OpenStreetMap để có được tọa độ các điểm dùng để vẽ đường đi.

Dữ liệu được lấy xuống trên OpenStreetMap sẽ là file XML có chứa dữ liệu thông tin về đường đi của toàn bản đồ.

```
<bounds minlat="10.7693300" minlon="106.6559980" maxlat="10.7729300" maxlon="106.6601280"/>
<node id="366408389" lat="10.7728190" lon="106.6565017" user="Ivan Garcia" uid="94270" visi
<node id="366475907" lat="10.7694897" lon="106.6659820" user="Ivan Garcia" uid="94270" visi
<node id="2061032839" lat="10.7901592" lon="106.6525375" user="Prenn" uid="1097298" visible
<node id="366421015" lat="10.7759359" lon="106.6566859" user="Ivan Garcia" uid="94270" visi
<node id="1524323178" lat="10.7850938" lon="106.6540083" user="Quark" uid="560735" visible=
<node id="366397569" lat="10.7693231" lon="106.6616158" user="Ivan Garcia" uid="94270" visi
<node id="366385215" lat="10.7714474" lon="106.6604336" user="Ivan Garcia" uid="94270" visi
<node id="366385195" lat="10.7821074" lon="106.6548835" user="Quark" uid="560735" visible="
<node id="366409168" lat="10.7696639" lon="106.6653736" user="Ivan Garcia" uid="94270" visi
<node id="1552973814" lat="10.7751082" lon="106.6624130" user="David F" uid="214311" visibl
<node id="366428560" lat="10.7919752" lon="106.6528853" user="Prenn" uid="1097298" visible=
<node id="366457953" lat="10.7691340" lon="106.6634266" user="Ivan Garcia" uid="94270" visi
<node id="366406964" lat="10.7864127" lon="106.6536513" user="Quark" uid="560735" visible="
```

Bảng 4 Định dạng node trong OpenStreetMap

Một thẻ node sẽ chứa thông tin về , định danh của node ,kinh độ , vĩ độ, người tạo, định danh của người tạo, có hiển thị hay không , thời gian cập nhật gần nhất

```

<way id="32587110" user="OSMF Redaction Account"
  <nd ref="366473797"/>
  <nd ref="366451748"/>
  <nd ref="366425238"/>
  <nd ref="366403724"/>
  <nd ref="366478509"/>
  <tag k="highway" v="residential"/>
  <tag k="name" v="Đường Số 56"/>
</way>

```

Bảng 5 Định dạng đường trong OpenStreetMap

Một thẻ way sẽ chứa định danh của đường đi, tên đường đi, và loại đường đi. Và tham khảo đến các node của nó.

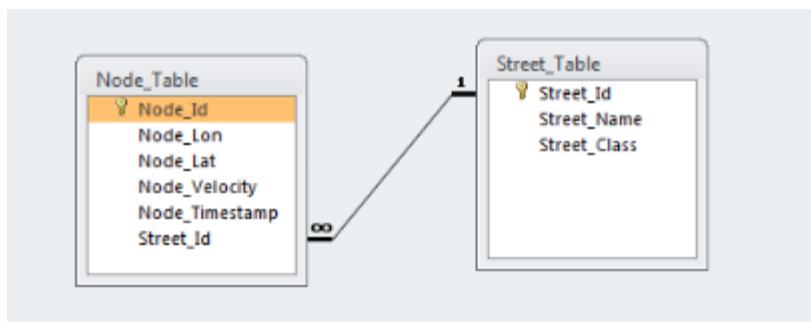
Ngoài ra còn nhiều thông tin khác. Nhưng để vẽ hiện trạng giao thông chúng ta chỉ cần chú ý đến hai loại dữ liệu nêu trên.

Sau khi lấy được file XML chứa dữ liệu đường đi của toàn thành phố một cách thủ công. Dữ liệu sẽ được đọc bằng SAX(Simple API for XML), chỉ lấy các giá trị liên quan đến kinh độ vĩ độ, loại đường, chỉ số đường, chỉ số node. Mục đích là để cho dữ liệu thu được là ít nhất. Vừa đủ để phục vụ yêu cầu ứng dụng. Dữ liệu sau khi được xử lý sẽ được lưu xuống cơ sở dữ liệu được trình bày ở bên dưới. Dữ liệu này là một trong những dữ liệu quan trọng cho ứng dụng.

4.2 Thiết kế cơ sở dữ liệu cho ứng dụng

Cơ sở dữ liệu sẽ được thiết kế cả ở server và trong ứng dụng.

Cơ sở dữ liệu :



Hình 18 Lược đồ cơ sở dữ liệu

Thực hiện việc đánh index trên các cột NodeId , StreetId, StreetClass để có truy cập dữ liệu nhanh hơn.

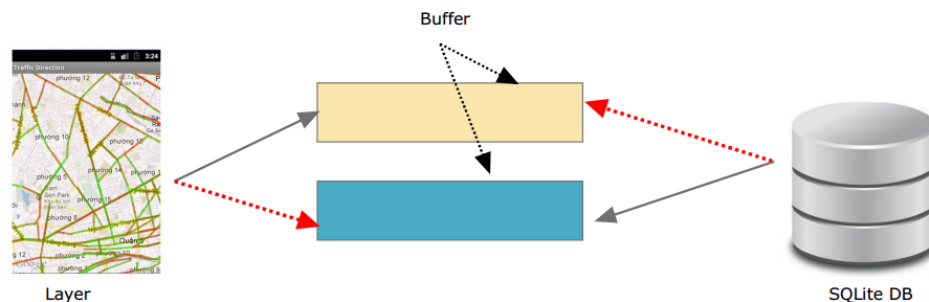
Cơ sở dữ liệu trên ứng dụng và cơ sở dữ liệu ở Server



Hình 19 Cơ sở dữ liệu ứng dụng – server

4.3 Cách truy xuất dữ liệu nhanh

Dùng hai buffer để truy xuất dữ liệu. Trong quá trình đọc dữ liệu từ dưới cơ sở dữ liệu lên , sẽ sử dụng hai thread. Một thread dùng để đọc dữ liệu từ cơ sở dữ liệu và lưu vào buffer. Một thread dùng để đọc dữ liệu từ buffer còn lại và vẽ lên bản Overlay. Hai thread này sẽ hoạt động liên tục. Và hoán đổi buffer với nhau để giảm thời gian chờ khi đọc dữ liệu.



Hình 20 Sử dụng bộ đệm đọc dữ liệu

4.4 Gửi nhận dữ liệu trên Server

Gửi nhận dữ liệu server và ứng dụng dùng TCP/IP . Mặc dù phương pháp này tốn rất nhiều chi phí để duy trì kết nối. Nhưng nó cũng cho phép việc liên lạc giữa server được với ứng dụng.

```
public void run() {  
    try {  
        InetAddress.getByname("www.vre.cse.hcmut.edu.vn");  
        Socket socket = new Socket(serverAddr, 6655);  
        out = new PrintWriter(new BufferedWriter(new OutputStreamWriter(  
            socket.getOutputStream()), true);  
        BufferedReader in = new BufferedReader(new InputStreamReader(  
            socket.getInputStream()));  
        // Handling incoming messages  
    } catch (UnknownHostException e) {  
        // catch exception  
    }  
}
```

Bảng 6 Kết nối đến Server sử dụng TCP/IP

Dùng thread để kết nối với cơ sở dữ liệu server. Ứng dụng sẽ tạo ra một thread kết nối với server theo địa chỉ **www.vre.cse.hcmut.edu.vn** , thông qua cổng 6655.

Xử lý dữ liệu nhận được :

```
while (true) {  
    String serverMessage = in.readLine();  
    if (serverMessage != null) {  
        update = "UPDATE " + DatabaseHelper.NODE_TABLE_NAME  
            + " SET Speed = " + serverMessage.split(":")[2]  
            + ", Density = " + serverMessage.split(":")[3]  
            + ", TimeStamp = " + serverMessage.split(":")[4]  
            + " WHERE ID = " + serverMessage.split(":")[1];  
        md.getWritableDatabase().execSQL(update);  
    }  
}
```

Bảng 7 Xử lý dữ liệu nhận được

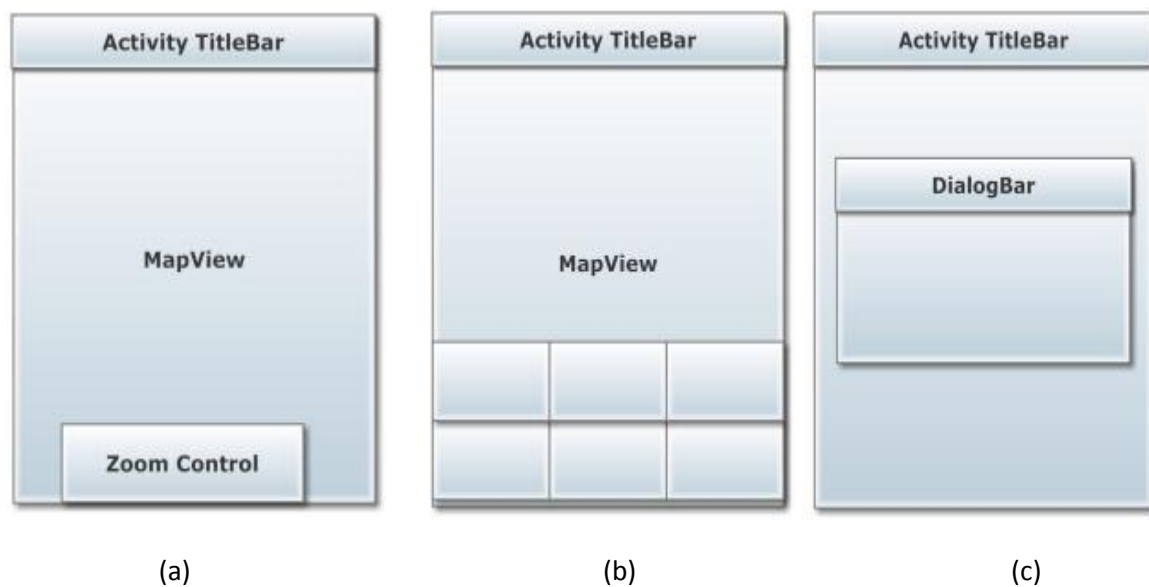
Dữ liệu nhận được dưới dạng chuỗi , các nhau bằng dấu “:” vì thế sẽ dùng hàm String.split để tách dữ liệu, sau đó sẽ lưu dữ liệu vào SQLite.

Để giảm chi phí overhead khi thiết lập kết nối TCP/IP. Sẽ sử dụng các biện pháp như sau.

- Sử dụng dữ liệu dạng String để gửi nhận dữ liệu trên với server. Mục đích là để đơn giản quá trình xử lý dữ liệu vì dữ liệu gửi nhận đơn giản sẽ giảm được thời gian truy xuất dữ liệu.
- Vì hiện trạng giao thông không thể thay đổi trong thời gian tức thì, nên ứng dụng sẽ gửi nhận dữ liệu một cách tự động sau khoảng 15-20 phút một lần.
- Thời gian để gửi và nhận dữ liệu từ server đến ứng dụng chỉ dưới hạn dưới 1 phút.

4.6 Hiện thị dữ liệu lên màn hình

4.6.1 Những layout chính của chương trình



Layout a : là layout chính của ứng dụng , Activity TitleBar là thanh tiêu đề của các màn hình, MapView là lớp nền hiển thị , ở đây dùng GoogleMap.ZoomControll dùng để phóng to thu nhỏ bản đồ.

Layout b: là layout chính của ứng dụng , khi nhấn nút menu để hiện các chức năng chính của ứng dụng cho người sử dụng chọn.

Layout c: là layout khi người dùng chọn các chức năng có hộp thoại.

4.6.2 Lấy dữ liệu theo đúng màn hình

Đầu tiên sẽ lấy kích thước của màn hình sau đó chiếu lên bản đồ để biết được kích thước bản đồ được lấy tương ứng với với màn hình hiện tại, từ đó suy ra các điểm cần lấy để vừa với kích thước màn hình.

Đối với chức năng hiển thị hiện trạng giao thông. Khi có yêu cầu từ người dùng. Ứng dụng sẽ thực hiện truy vấn đến cơ sở dữ liệu SQLite trong điện thoại, đọc dữ liệu so sánh với thời gian hiện tại nếu dữ liệu trong cơ sở dữ liệu được cập nhật nhỏ hơn so với dữ liệu được cập nhật là 20 phút. Thì ứng dụng sẽ lấy thực hiện lấy các thông số của màn hình, sau đó lấy các điểm trong bản đồ của màn hình, và gửi yêu cầu cập nhật đến server. Server sẽ xử lý và gửi lại dữ liệu cho ứng dụng. Với kích thước người vừa đúng so với kích thước được yêu cầu. Điều này sẽ làm giảm thời gian tính toán của ứng dụng. Làm cho ứng dụng hoạt động hiệu quả hơn.

4.6.3 Hiển thị dữ liệu theo độ zoom của bản đồ:

Với mỗi cấp độ zoom khác nhau của bản đồ. Dữ liệu về hiện trạng giao thông sẽ được vẽ khác nhau.

Đường đi ở mức zoom cao : (những chấm đỏ là số node cần để hiển thị)



Đường đi ở mức zoom thấp hơn :



Hiển thị theo mức độ zoom và các loại đường. Ta có các loại đường như sau.

- Loại số 4 hẻm nhỏ, bao gồm các con hẻm.
- Loại số 3 là các đường nhỏ. Ví dụ : đường Thành Thái, Tô Hiến Thành ...
- Loại số 2 là các đường lớn. Ví dụ : đường Lý Thường Kiệt, 3 tháng 2
- Loại số 1 là các đường quốc lộ.

Tùy theo mức độ zoom của bản đồ mà ta sẽ vẽ các thêm các loại đường hoặc loại bỏ đi một số đường mục đích là làm giảm lượng dữ liệu cần tải, cải thiện tốc độ cho ứng dụng.

Khoảng cách giữa hai node trên thực tế là từ khoảng 50-100(m) ở mức độ zoom lớn nhất. Vì trên thực tế với khoảng cách này thì hiện trạng giao thông không thay đổi nhiều.



Hình 21 Các loại đường trên bản đồ.

4.6.4 Xác định màu sắc và vận tốc



0 (km/h)

>40 (km/h)

Màu sắc được dùng cho ứng dụng sẽ nằm trong khoảng từ màu đỏ cho đến màu xanh non (green). Để hiển thị vận tốc của trên các đoạn đường. Ứng dụng sẽ hiển thị vận tốc từ 0(km/h) tới 40(km/h). Đây là dải vận tốc chính của các phương tiện lưu thông trong thành phố. Đối với các vận tốc lớn hơn 40km/h ứng dụng vẫn hiển thị màu xanh.

Đối với các dải vận tốc từ 0 (km/h) đến 40(km/h). Để biểu diễn các khoảng vận tốc trong đó chúng ta sẽ sử dụng công thức nội suy dưới đây (dạng code java).

```
int redColor = (int) ((1.0 - speed / 40.0) * 255);
int greenColor = (int) ((speed / 40.0) * 255);
mPaint.setColor(Color.argb(100, redColor, greenColor, 0));
```

Hình 22 Công thức nội suy tính màu.

4.6.5 Cách vẽ điểm và đường đi

Sử dụng các hàm draw() trong lớp Overlay . Đây là một hàm callback sẽ được gọi liên tục để vẽ liên tục lên lớp Overlay.

```
public boolean draw(Canvas canvas, MapView mapView, boolean shadow,
    long when) {
    Bitmap bmp1 = BitmapFactory.decodeResource(getResources(),
        R.drawable.startpoint);
    Point screenPts1 = new Point();
    GeoPoint p1 = new
    GeoPoint(Integer.parseInt(Point1.split(",")[0]),Integer.parseInt(Point1
    .split(",")[1]));
    mapView.getProjection().toPixels(p1, screenPts1);
    canvas.drawBitmap(bmp1, screenPts1.x - dip, screenPts1.y - dip,null);
    return super.draw(canvas, mapView, shadow, when);
}
```

Bảng 8 Cách vẽ điểm lên trên bản đồ.

Trong hàm draw sẽ sử dụng các đối tượng Bitmap để lấy lên ảnh từ nguồn resource của chương trình. Đối tượng Point sẽ dùng để lưu điểm để vẽ trên bản đồ. GeoPoint sẽ lấy điểm trên bản đồ.

Sau khi lấy được điểm trên bản đồ. Ta dùng phép chiếu getProjection() để lấy được tọa độ của điểm trên màn hình. Và dùng hàm drawBitmap() để vẽ lên Overlay.

```

public boolean draw(Canvas canvas, MapView mapView, boolean shadow,
                    long when) {
    Paint mPaint = new Paint();
    Paint.setAntiAlias(true);
    mPaint.setStyle(Paint.Style.FILL_AND_STROKE);
    mPaint.setStrokeJoin(Paint.Join.ROUND);
    mPaint.setStrokeCap(Paint.Cap.ROUND);
    mPaint.setStrokeWidth(6);
    //get point on map
    Point p1 = new Point();
    Point p2 = new Point();
    //project geopoint to point
    Path path = new Path();
    //calcula color for path
    mPaint.setColor(Color.argb(100, redColor, greenColor, 0));
    path.moveTo(p2.x, p2.y);
    path.lineTo(p1.x, p1.y);
    canvas.drawPath(path, mPaint);
}

```

Bảng 9 Các vẽ đoạn thẳng

Vẽ đoạn thẳng cũng tương tự như vẽ một điểm lên trên Overlay. Sử dụng đối tượng Paint để định nghĩa các phương thức khi vẽ, màu vẽ và góc cạnh, độ dày của nét vẽ... Sử dụng đối tượng Path để xác định đoạn thẳng dựa vào hai điểm đã cho. Khi vẽ sử dụng drawPath để vẽ lên đoạn thẳng trên Overlay của bản đồ.

4.7 Làm mới dữ liệu

Mặc định dữ liệu thời gian yêu cầu dữ liệu sẽ nằm trong khoảng một phút. Sau một khoảng thời gian là 15-20 phút thì ứng dụng sẽ gửi yêu cầu tới server để server cập nhật dữ liệu giao thông và gửi dữ liệu mới nhất về.

```

public ArrayList<DrawableStreet> streetList = new ArrayList<DrawableStreet>();
public ArrayList<DrawableStreet> streetList2 = new ArrayList<DrawableStreet>();
// add street to buffer
if (count == 0)
    streetList.add(street);
else
    streetList2.add(street);
// remove street from buffer
if (count == 0)
    streetList.clear();
else
    streetList2.clear();

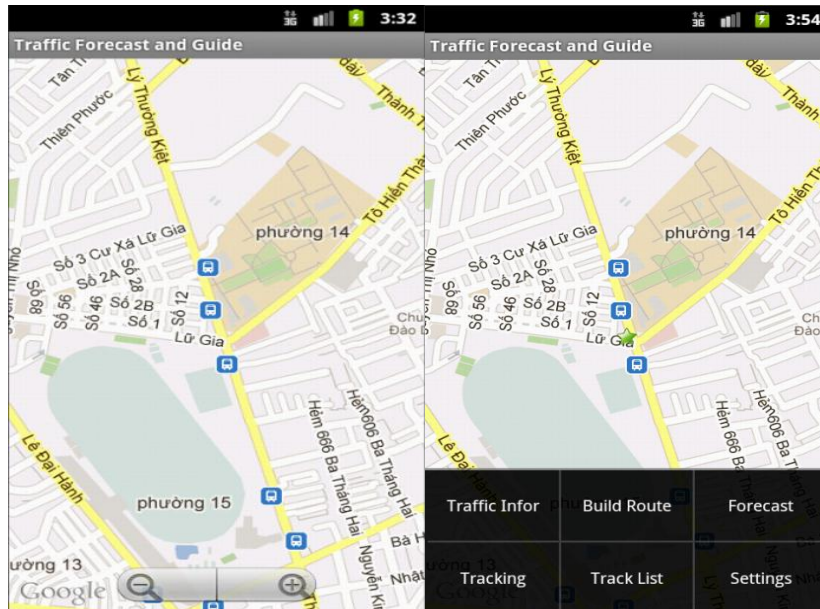
```

Bảng 10 Sử dụng bộ đệm để đọc dữ liệu.

Việc hiển thị dữ liệu giao thông lấy được từ server lên trên màn hình ứng dụng cũng sử dụng buffer để có được tốc độ nhanh hơn.

4.8 Hoàn thành ứng dụng

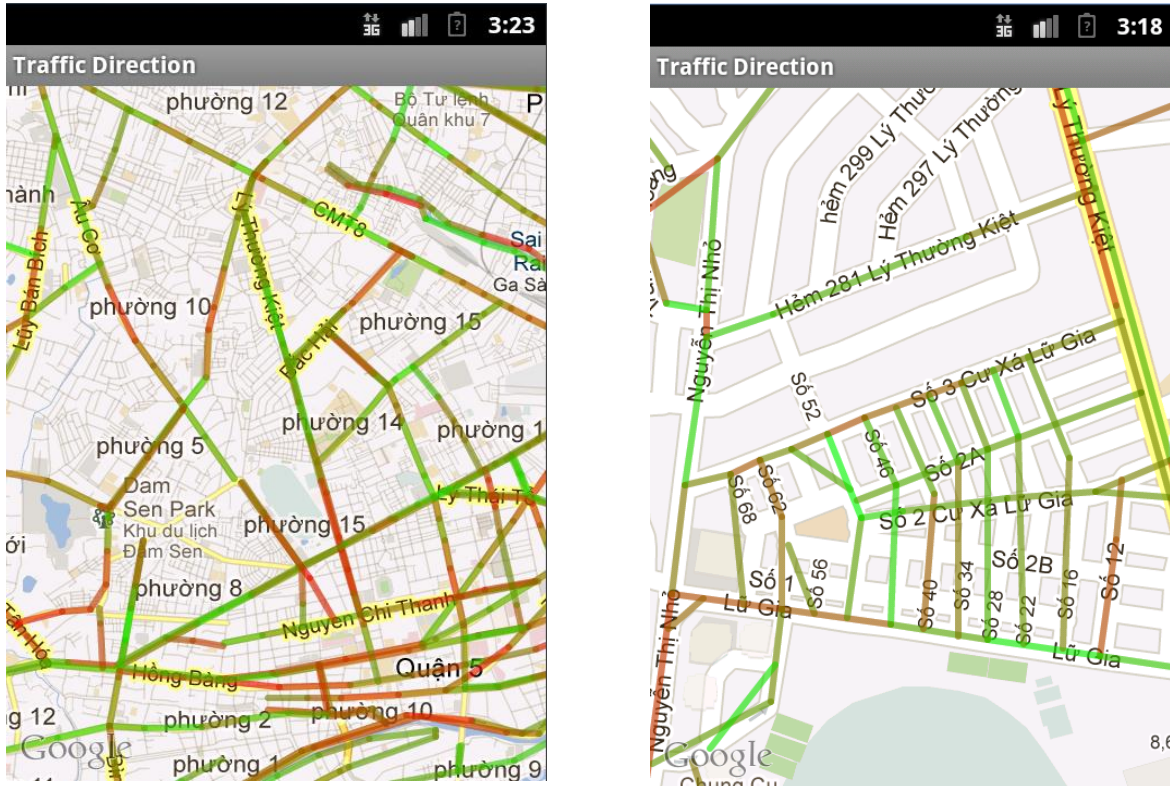
Màn hình chính của ứng dụng



Hình 23 Màn hình chính của ứng dụng

Về giao diện của ứng dụng. Ở đây vì màn hình của các thiết bị di động nhỏ, giải pháp được đặt ra là ứng dụng sẽ hiển thị lên một màn hình cơ bản, tất cả các chức năng của ứng dụng sẽ nằm trong menu để người dùng chọn.

Hiện thị trạng thái giao thông của một vùng



Hình 24 Hiện thị trạng thái giao thông

Để hiện thị hiện trạng giao thông. Sẽ có một lớp OverLay lên trên GoogleMap. Lớp Overlay này sẽ hiện thị đường đi với các màu khác nhau tương ứng với vận tốc khác nhau.

Ở mỗi mức độ phóng to hoặc thu nhỏ khác nhau của bản đồ, dữ liệu giao thông sẽ được vẽ tương thích :

Với các mức zoom thấp, diện tích nhìn thấy của bản đồ lớn. Hiện trạng giao thông chỉ vẽ trên các tuyến đường chính.

Đối với các mức zoom cao , diện tích nhìn thấy nhỏ . Hiện trạng giao thông sẽ được vẽ chi tiết.

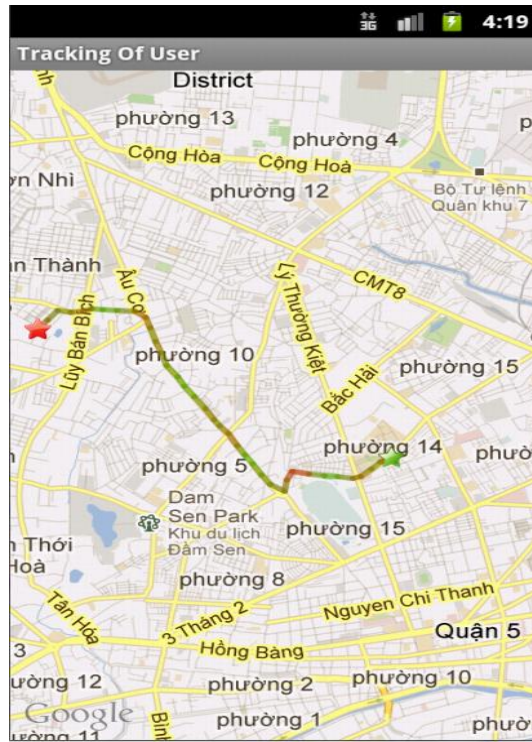
Với cách hiện thực trên , sẽ tiết kiệm được rất nhiều chi phí vẽ đồng thời không gây rối cho người sử dụng.

Hiện thị đường đã đi qua (Tracking)

Thời gian kết thúc track	Ngày tương ứng
09 : 30	10/12/2012
15 : 53	12/12/2012
07 : 27	14/12/2012
07 : 08	09/12/2012
17 : 18	04/11/2012
07 : 31	03/11/2012
07 : 27	18/10/2012

Hình 25 Danh sách các đường đã đi theo thời gian

Khi người dùng chọn chức năng hiển thị tracking một màn hình khác sẽ hiện ra, với danh sách các tập tin tracking mà người sử dụng đã tạo bằng chức năng tracking. Danh sách này hiển thị theo định dạng thời gian kết thúc track và ngày tương ứng. Người dùng chỉ cần chọn trên danh sách một tập tin hiển thị bằng cách bấm lên ô chọn.



Hình 26 Hiện thị đường đã đi

Sau khi người dùng nhấn vào ô chọn , kết quả sẽ hiển thị cho người dùng như hình trên.Với các đoạn màu sắc khác nhau khi hiển thị đường như hình trên , người dùng có thể biết được đoạn đường nào mình đã đi qua với tốc độ bao nhiêu.Từ đó giúp ích cho người dùng trong việc chọn đường khi tham gia giao thông.

Hiện thị đường đi cho chức năng tìm đường đi.

Hình 27 Đường đi cho chức năng tìm đường.

Sau khi nhập dữ liệu với điểm bắt đầu và điểm kết thúc. Ứng dụng sẽ xuất ra đường đi với thời gian ngắn nhất. Đường đi được tìm thấy sẽ được vẽ như hình trên. Đường đi với vận tốc từng đoạn cũng được server tính toán trước. Giúp người dùng có thể chuẩn bị tốt hơn trước khi tham gia giao thông.

4.9 Cài đặt ứng dụng

Khi ứng dụng được cài đặt và một smartphone thì việc đầu tiên sẽ là ghi dữ liệu về đường đi của bản đồ xuống dưới cơ sở dữ liệu SQLite. Việc ghi một lượng lớn dữ liệu sẽ rất tốn thời gian, đặc biệt là đối với các smartphone có cấu hình thấp. Giải pháp được thực hiện ở đây là sẽ sao chép sẵn tập tin cơ sở dữ liệu trong máy đã cài rồi, và lưu vào trong tập tin cài đặt của ứng dụng. Khi cài đặt ứng dụng sẽ thực hiện việc chép tập tin chứa dữ liệu đường đi của bản đồ vào trong bộ nhớ của smartphone. Đồng thời sẽ không phải lưu tập tin chứa thông tin đường đi vào tập tin cài đặt. Làm giảm dung lượng cho tập tin cài đặt. Giúp việc cài đặt đơn giản hơn.

Ứng dụng được hiện thực trên nền tảng Android 2.3.3 cũng là phiên bản phổ biến nhất của hệ điều hành Android hiện nay với chiếm hơn 50% tổng số smartphone chạy Android. Các phiên bản thấp hơn 2.3.3 hiện nay còn rất ít chiếm chưa tới 5% .

Hơn thế nữa theo như thiết kế của hệ điều hành Android thì hầu hết các ứng dụng của các phiên bản trước đều có thể chạy được trên các phiên bản mới. Chính vì vậy mà ứng dụng có thể chạy được trên tất cả các smartphone hiện nay

4.10 Đánh giá ứng dụng

Những điểm đã làm được :

Đã hoàn thành các chức năng chính của ứng dụng : hiển thị hiện trạng giao thông , hiển thị tìm đường , hiển thị đường người dùng đã đi qua, hiển thị vị trí của người dùng , hiển thị dự báo kẹt xe. Giao diện của ứng dụng đơn giản dễ dùng.

Tập tin cài đặt có dung lượng 1,4 Megabyte tương đối nhỏ, tốc độ cài đặt nhanh thuận tiện cho người dùng.

Những chức năng hiển thị tìm đường đi đường đi , hiển thị vị trí hiện tại , hiển thị dự báo kẹt xe. Điều hoạt động tốt , khả năng đáp ứng chấp nhận được, chạy ổn định.

Ứng hoạt động ổn định trên các phiên bản 2.3.3 của hệ điều hành Android trở về sau.

Những hạn chế của ứng dụng :

Ở một số đoạn đường dữ liệu GoogleMap và OpenStreetMap không khớp nhau nên hiển thị hiện trạng giao thông không được đẹp.

Nét vẽ hiện trạng giao thông và vẽ đường đi là cố định nên khi người dùng phóng to bản đồ lên , đường vẽ hiển thị hiện trạng sẽ không khớp nhiều với đường đi trên Google Map.

Chức năng hiển thị hiện trạng giao thông hoạt động khá nặng nề và chậm chạp , mặc dù đã có nhiều cải tiến. Nhưng do lượng dữ liệu quá lớn.

Giao diện của ứng dụng chưa thực sự ấn tượng và đẹp như GoogleMap. Chức năng của ứng dụng khá đơn giản.

.

CHƯƠNG 5 : TỔNG KẾT

5.1 Kết quả đạt được

Xây dựng thành công ứng dụng chạy trên hệ điều hành Android với các chức năng chính đưa ra :

- Hiển thị hiện trạng giao thông
- Hiển thị vị trí hiện tại của người dùng.
- Hiển thị dự báo kẹt xe trong khoảng thời gian ngắn.
- Hiển thị đường người dùng đã đi , lưu đường đi của người dùng.
- Hiển thị chức năng tìm đường , hiển thị đường đi đã được server tính toán.
- Ứng dụng có thể hiển thị được bản đồ bằng hai dạng : dạng đường đi , và dạng vệ tinh, hai tính năng này do GoogleMap cung cấp.

Ứng dụng cũng đã kết nối được với server để chạy trên thời gian thực.

Những tính năng hiển thị đường đi , hiển thị dự báo kẹt xe , hiển thị đường đi người dùng có tốc độ đáp ứng nhanh thỏa mãn yêu cầu.

5.2 Hạn chế

Việc sử dụng bản đồ Google làm lớp nền cho ứng dụng mang lại cảm giác thân thiện cho người dùng , tuy nhiên Google Map API đã giới hạn quá nhiều tính năng dẫn đến việc ứng dụng phải sử dụng nguồn dữ liệu từ nhà cung cấp bản đồ khác , và dữ liệu của hai bản đồ này mặc dù không sai lệch nhiều nhưng cũng gây khá nhiều phiền toái cho người dùng. Và có thể không thân thiện đối với những người dùng khó tính.

Việc hiển thị hiện trạng giao thông bằng cách dùng các hàm vẽ mặc định trong java là khá tốn chi phí , làm cho ứng dụng chạy chậm ở chức năng hiển thị hiện trạng giao thông. Điều này có thể làm ứng dụng bị chậm đối với một số smartphone cấu hình thấp.

Tuy ứng dụng đã hoàn thành các chức năng cơ bản nhưng để đến được với người dùng cần phải hoàn thiện nhiều thêm nữa .Ứng dụng cũng chưa được đưa lên Google Play để có thể đến với người dùng rộng rãi.

5.3 Hướng phát triển

Trong tương lai để khắc phục những hạn chế của GoogleMap API , có thể ứng dụng sẽ chuyển qua sử dụng OpenStreetMap để dữ liệu hiển thị giao thông khớp hoàn toàn với lớp nền.

Để cải thiện tốc độ vẽ hiện tại , trong tương lai ứng dụng có thể sử dụng vẽ hình ảnh như cách mà các bản đồ nổi tiếng như GoogleMap đã làm , để cải thiện tốc độ cho ứng dụng.

Để ứng dụng thêm thân thiện với người dùng với người dùng. trong tương lai có thêm nhiều tính năng tìm đường đến các ATM , nhà hàng , quán , các công viên địa điểm vui chơi giải trí...

Các tính năng hiển thị hiện trạng giao thông theo chế độ 3D đối với bản đồ 3D .

Để người dùng không phải vào ứng dụng, việc phát triển thêm widget cho ứng dụng giúp người dùng có thể tiếp cận với ứng dụng ở nhiều cách khác nhau.

Ứng dụng còn thể có các chức năng chia sẻ hình ảnh video khi người dùng đi đến các điểm kẹt xe từ đó có thể cập nhật được hình ảnh cho tất cả các người dùng khác về hiện trạng giao thông.

Hệ thống có thể mở rộng thêm ứng dụng trên các thiết bị chạy iOS và Window Phone đây là hai hệ điều hành di động cũng rất phổ biến và sẽ phát triển trong tương lai . Với mục đích chính của hệ thống là càng có nhiều thiết bị cung cấp thông tin thì hệ thống của chúng ta sẽ cho ra kết quả càng chính xác.

TÀI LIỆU THAM KHẢO

- [1].Reto Meier , Professional Android Application Development, Wiley Publishing , 2009.
- [2].Google Map API version 1th , Google Inc, Last updated December 3, 2012.
- [3].Thầy Phạm Trần Vũ , Bài giảng đại học Computer Network 1 Khoa Khoa Học Và Kỹ Thuật Máy Tính ĐH Bách Khoa TP.HCM, 2009.
- [4].Jim Kurose, Keith Ross, Computer Networking: A Top Down Approach , 5th Edition, Addison-Wesley , April 2009.
- [5].Thầy Lê Ngọc Minh , Bài giảng đại học môn Hệ Điều Hành , Khoa Khoa Học Và Kỹ Thuật Máy Tính ĐH Bách Khoa TP HCM.
- [6].Thầy Nguyễn Trung Trực , Bài giảng đại học môn Database , Khoa Khoa Học Và Kỹ Thuật Máy Tính ĐH Bách Khoa TP.HCM.
- [7].Developer Android Documentation tại **<http://developer.android.com>** , Google Inc.
- [8].Lê Ngọc Quốc Khánh ,OpenStreetMap Introduction , **Nov 2009**
- [9].SQLite Documentation tại **<http://www.sqlite.org/docs.html>** .
- [10]. Tổng quan về giải pháp phần mềm định vị tại **<http://www.thietbidinhvi.vn/giai-phap-phan-mem.html>**

PHỤ LỤC**Thuật ngữ Anh Việt**

Buffer	Bộ đệm
Database	Cơ sở dữ liệu
Density	Mật độ
Forecast	Dự đoán
GPS	Thiết bị định vị
Map	Bản đồ
Node	Nút, điểm
Overhead	Chi phí duy trì kết nối
OverLay	Lớp trên dùng để vẽ
SD	Một loại thẻ nhớ trên điện thoại
Server	Máy chủ
Show current	Hiển thị trạng thái hiện tại
Smartphone	Điện thoại thông minh
Street	Đường đi
String	Chuỗi ký tự
Tracking	Theo vết, theo dõi
Velocity	Vận tốc
Way	Đường
Zoom	Phóng to hoặc thu nhỏ

Nội dung CD

Một bản CD được đính kèm theo tài liệu. Các folder trong CD bao gồm :

- Blueprint : Chứa các tập tin thiết kế của chương trình.
- Database : Chứa các tập tin mã cơ sở dữ liệu , và dữ liệu bản đồ.
- Report : Bao gồm các tập tin báo cáo , slide ...
- Server : Mã nguồn server để demo dữ liệu cho ứng dụng.
- TrafficDirection : Mã nguồn ứng dụng.
- Screenshot : Các hình ảnh về ứng dụng.
- Tập tin readme.txt : hướng dẫn sử dụng CD