ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC BÁCH KHOA

KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH

----------o0o----------

LUẬN VĂN TỐT NGHIỆP ĐẠI HỌC

XÂY DỰNG ỨNG DỤNG THU THẬP DỮ LIỆU GIAO THÔNG CHO ĐIỆN THOẠI THÔNG MINH SỬ DỤNG HỆ ĐIỀU HÀNH ANDROID CÓ CHỨC NĂNG GPS

HỘI ĐỒNG: KHOA HỌC MÁY TÍNH

GVHD: TS. Phạm Trần Vũ

GVPB: TS. Lê Thanh Vân

SVTH:	Trương Vũ Minh Trí	50802348
	Nguyễn Anh Tiến	50802216
	Chu Quang Thiện	50802076

TP. HỒ CHÍ MINH, Tháng 1/2013

LỜI CAM ĐOAN

Chúng tôi xin cam đoan luận văn **"XÂY DỰNG ỨNG DỤNG THU THẬP DỮ LIỆU GIAO THÔNG CHO ĐIỆN THOẠI THÔNG MINH SỬ DỤNG HỆ ĐIỀU HÀNH ANDROID CÓ CHỨC NĂNG GPS"** là phần nghiên cứu và thể hiện của riêng chúng tôi, không sao chép từ bất kì 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, nhóm em thực hiện báo cáo luận văn hoàn thành khóa học.

Nhóm 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 nhóm em có thể hoàn thành luận văn này.Đặc biệt, nhóm em xin gửi lời cảm ơn sâu sắc nhất đến thầy Phạm Trần Vũ đã nhiệt tình hướng dẫn nhóm 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 báo giúp nhóm em hoàn thành tốt luận văn tốt nghiệp của mình.

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

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

TP.HCM, tháng 1 năm 2013

Nhóm sinh viên thực hiện đề tài

TÓM TẮT

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 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 dựng ứng dụng 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 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, chi phí, bảo vệ sức khỏe, có tâm lý thoải mái đồng thời giúp cho tình trạng giao thông trên đường tốt hơn. Trong đó, việc thu thập dữ liệu từ người tham gia giao thông là một bài toán quan trọng của hệ thống, đó cũng là đề tài mà nhóm chúng tôi thực hiện.

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

- Thu thập dữ liệu GPS từ điện thoại người dùng và gửi lên server lưu trữ. Thông số thu thập là số imei, vĩ độ, kinh độ, độ cao, thời gian, độ chính xác, tốc độ đi, được lưu lại dưới dạng file GPX trong thẻ nhớ điện thoại của người dùng, đồng thời gửi dữ liệu vừa nhận được lên server theo giao thức UDP. Dữ liệu GPS được lưu trữ trong DB của server sẽ được chuyển thành dữ liệu giao thông và lưu vào một bảng khác để sử dụng cho các mục đích sau này.
- Hiển thị dữ liệu thu thập được thông qua website được thiết kế trên server. Website này cho phép người dùng truy cập để xem đường đi của họ bằng cách tạo tài khoản với số imei điện thoại cá nhân. Bên cạnh đó, website còn cho phép admin xem đường đi của nhiều user khác nhau cùng một lúc, để đưa ra các tính toán cũng như dự báo chính xác hơn.

ABSTRACT

Transportation system playsanimportantrolein the development of the country.Itmakestradeandtravelmore convenient, easier and it also contributes to the growth of the national economy. In recent years, since the industrial revolution, people have been moving from countryside to large cities. The increasing number of people migrating has affected the world's cities in negative ways and resulted in many problems. For example, wecan clearly noticethe increase intrafficcongestions, traffic jams, lasting for hoursin HoChiMinhCityandHanoi, Vietnam. This isabig problemanditsconsequences are extremely significant toindividualsas well asorganizations.

In order to conserve the fresh environment and preserve people from harmful effects resulted from massive traffic congestion, our team associates with other groups in developing an application which could inform customers the traffic status at any time around their current location. By knowing the present traffic conditions that also being forecast in the near future, users can choose the appropriate alternative way to avoid the traffic jam areas. Thanks to this application, users could save time, cost, have good health and psychological comfort that lead to better traffic conditions on the street. In particular, the collection progress of traffic data from participants is one of the most remarkable features of the whole system; it is also a topic that our group carried out.

Our application comprises two main functions as follows:

- Collecting GPSdatafromthe user's mobileandsending themtotheserver storage.
  Data collectionconsists oftheimeinumber, latitude, longitude, elevation, current time, accuracy, speed, savedasaGPXfilein the memory cardof thephone; at the same time, those data are sent totheserverthroughUDPprotocol. Having done that, GPSdata stored in theDBserverwillbeconvertedtotraffic dataandstoredintoanothertable for further use.

- Displaying datacollectedthroughthewebsitedesignedontheserver.
  Thiswebsiteallows users topreviewtheir old routes (the roads they passed through when going outside)by creatinganaccountwiththeirpersonalphoneimei. Furthermore, thewebsitealsoletadministrators observe thepathsofmanydifferentusersat the same time; this helps them to makecalculationsand predictionsmoreaccurate.

MỤC LỤC

CHƯƠNG 1:GIỚI THIỆU

1.1 Giới thiệu đề tài

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.

1.1.1Vấn đề kẹt xe

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. Hậu quả của kẹt xe là vô cùng to lớn:

- Ô nhiễm không khí
  Các tác nhân gây ô nhiễm chủ yếu là các loại loại xe có động cơ thải ra khí đốt nhiên liệu, bụi và tiếng ồn.Đặc biệt mật độ ô nhiễm càng tăng lên ở những nút kẹt xe.Theo các chuyên gia về môi trường,chỉ số CO_2 bình thường ở thành phố Hồ Chí Minh đã ở mức báo động,song tại những nơi kẹt xe nồng độ CO_2 tăng lên hàng chục lần so với bình thường.
- Ảnh hưởng đến sức khỏe con người
  Các khí thải độc hại như cacbon monoxit CO, cacbon dioxit CO_2 ,Hydro cacbon (C_nH_m), các oxit nitơ (NO_x), Sunfua dioxit (SO_2), khói đen, chì và các dạng hạt gây ảnh hưởng cực kì nghiêm trong tới sức khỏe con người.Chúng trực tiếp và gián tiếp gây ra nhiều loại bệnh và triệu chứng như rối loạn hô hấp,gây ngạt, viêm mắt, viêm mũi,rối loạn tiêu hóa.

  Khi dòng xe lưu thông trên đường, đặc biệt là khi hãm phanh, các lốp xe sẽ ma sát mạnh với mặt đường làm mòn đường, mòn các lốp xe và tạo ra bụi đá, bụi cao su và bụi sợi. Các bộ phận ma sátcủa phanh bị mòn cũng thải ra bụi kẽm, đồng, niken, crom, sắt và cadmi.Ngoài ra quá trình cháy không hết nhiên liệu cũng thải ra bụi cacbon. Bên cạnh các nguồn bụi sinh ra từ xe, còn có bụi đất đá, cát tồn đọng trên đường do chất lượng đường kém, do đường bẩn và do chuyên chở các vật liệu xây dựng, chuyên chở rác. Chúng có thể gây nên các bệnh đường hô hấp, bệnh hen suyễn, viêm cuống phổi, bệnh khí thũng, bệnh viêm cơ phổi, trước hết là các dạng bệnh bụi phổi.

  Tiếng ồn là dạng ô nhiễm phổ biến ở các đô thị. Trong các nguồn sinh ra tiếng ồn ở đô thị thì các phương tiện giao thông vận tải đóng vai trò chủ yếu: 60 – 80% nguồn sinh ra ồn đô thị là phương tiện giao thông bởi động cơ, ống xả hay

các rung động của các bộ phận xe,do còi xe.Tiếng ồn gây tác hại rất lớn đến toàn bộ cơ thể nói chung và cơ quan thính giác nói riêng.Tiếng ồn mạnh, thường xuyên gây nên bệnh đau đầu, chóng mặt, cảm giác sợ hãi, bực tức vô cớ, trạng thái tâm thần bất ổn, mệt mỏi. Tiếng ồn gây ra những thay đổi trong hệ thống tim – mạch, kèm theo sự rối loạn trương lực mạch máu, rối loạn nhịp tim. Tiếng ồn còn làm rối loạn chức năng bình thường của dạ dày, làm giảm bớt sự tiết dịch vị, ảnh hưởng đến sự co bóp bình thường của dạ dày.

- Ảnh hưởng đến các công việc mang tính ưu tiên

  Các xe cấp cứu, chữa cháy, hộ đê … khi chấp hành nhiệm vụ được quyền ưu tiên so với các phương tiện giao thông khác vì mức độ quan trọng trong công việc. Tuy nhiên, các nút kẹt xe sẽ gây ảnh hưởng lớn đến những công việc này, và mức độ nghiêm trọng do nó mang lại là không tránh khỏi.
- Ảnh hưởng kinh tế
  - Sụt giảm đầu tư

    Đầu tư vào một thành phố, một đất nước không chỉ phụ thuộc vào điều kiện kinh tế mà còn ảnh hưởng nhiều từ các điều kiện của môi trường nơi đầu tư. Nạn kẹt xe ảnh hưởng xấu cho ngành vận chuyển, kéo theo sự sụt giảm của một số ngành kinh tế khác. Đó cũng là nguyên nhân khiến các nhà đầu tư e dè trong việc đầu tư vào một thành phố có điều kiện kinh tế thuận lợi nhưng lại khó khăn trong giao thông, điển hình là thành phố Hồ Chí Minh, những năm gần đây đang mất dần đầu tư nước ngoài.
  - Hao tốn thời gian của người tham gia giao thông

    Dù khó quy ra tiền, nhưng tính thời gian bị hao phí do kẹt xe là không nhỏ. Thời gian ấy đã có thể làm ra bao của cải vật chất, chưa kể những thiệt hại không thể tính được do trễ giờ hẹn làm việc theo kế hoạch của người dân.

1.1.2 Nguyên nhân kẹt xe

- Tầm nhìn về quản lý giao thông, quản lý đô thị kém

  Cách đây hơn mười năm, rất nhiều lời cảnh báo rằng nếu không qui hoạch, không tính toán thì mười năm sau sẽ kẹt đường, ngập nước.Đến bây giờ nhìn lại, thấy rằng mười năm qua có quá ít những biện pháp, giải pháp hữu hiệu.

  Biết sẽ tắc đường, kẹt xe mà lãnh đạo TP cứ cho xây chung cư cao tầng, xây cao ốc vào giữa trung tâm TP? Chưa kể những người sẽ vào ở làm việc, mỗi một công trình đang xây dựng có vài trăm công nhân và hàng trăm lượt xe vận tải phục vụ công trình mỗi ngày. Tại sao TP không đưa những công trình như vậy ra các TP vệ tinh. Rồi bến xe, nhà ga sao cứ bố trí trong nội thành...?

  Kế hoạch phát triển xe buýt, quản lý, điều hành và tổ chức hoạt động cho xe buýt một cách không hợp lý dẫn đến có tuyến đường, xe buýt chạy bít cả đường, dẫn đến ùn tắt giao thông.

Việc quy hoạch phân luồng tuyến cho các loại xe ôtô lại không gắn với tình hình thực tế nên đã dẫn tới tình trạng gia tăng kẹt xe, có làn đường thì không có xe chạy, trong khi đó có làn đường kẹt cứng.

- Hạ tầng giao thông không theo kịp sự phát triển kinh tế
  Do sự không thống nhất giữa các cơ quan như: Cấp nước, điện thoại…cũng là một cản trở giao thông vì cùng một con đường nhưng hôm nay bên này đào lên lắp cáp ngầm, mai bên cấp nước đào lắp ống nước, hôm kia tới bên làm cống đào…dẫn đến không có chỗ cho xe lưu thông.
- Ý thức lưu thông xe của người dân chưa cao đôi khi tự mình gây khó cho mình
  Điều này thấy rõ ràng nhất khi lưu thông vào giờ cao điểm, nhất là các ngã ba, ngã tư. Lượng người tham gia giao thông rất đông, nên chỉ một xe bị sự cố, một va quẹt nhỏ của người đi đường với nhau mà không sử lý kịp thì sẽ tạo thành nút, thắt chặt luồng giao thông gây ra kẹt xe. Ở các ngã ba, thường xuyên xảy ra trường hợp các xe trong đường phụ đâm thẳng ra đường chính cắt mạch giao thông gây ra hiện tượng ùn tắt. Không ai nhường ai, nên dễ dàng xảy ra kẹt xe hàng giờ đồng hồ.
- Số lượng xe bus taxi và diện tích đường
  Chiếm đến 51% đường phố có lòng đường rộng từ 7-12m và chỉ phù hợp cho ôtô con, xe buýt nhỏ lưu thông; 35% đường phố có lòng đường rộng dưới 7m chỉ phù hợp với xe 2 bánh và chỉ có 14% số đường có chiều rộng trên 12m để có thể tổ chức vận chuyển hành khách bằng xe buýt loại lớn được thuận lợi, 3.300 đầu xe buýt, loại xe lớn 80 chỗ chiếm đến 40%, xe nhỏ dưới 17 chỗ chỉ có 26%, còn lại là những đầu xe 40-55 chỗ, trên 7.000 đầu xe taxi (chưa kể lượng taxi dù), vài chục ngàn đầu xe hơi và xe tải tập trung về mỗi ngày.(Nguồn: http://ca.cand.com.vn/vi-VN/thoisuxahoi/tintucsukien/2007/10/116702.cand).
- Dân số đông
  Điều dễ dàng nhận thấylà mỗi sáng ra đường đi làm, mỗi chiều tan sở về là mọi ngả đường đều chật kín người. Ai ai cũng hối hả cho kịp giờ làm, kịp về nhà nấu bữa tối cho gia đình. Một thành phố ở ngưỡng 10 triệu dân, việc giải quyết việc đi lại, lưu thông cho tất cả 10 triệu người này không phải là dễ. Hậu quả đó là tình trạng kẹt xe ngày càng tăng ở TP HCM. 10 năm trước, kẹt xe vẫn xảy ra chứ không phải là không, nhưng bây giờ, khi nhắc đến kẹt xe là nhắc đến mọi nơi ở TP HCM này. Và 10 năm nữa, với tốc độ người nhập cư về TP HCM vẫn không giảm, thì vấn đề kẹt xe vẫn là nỗi đáng ngại nhất.

1.1.3 Giải pháp:

- Giải pháp lâu dài
  - Giao thông công cộng: Khi một Thành phố có số dân vượt 1 triệu người, nhất thiết phải tính đến việc đầu tư giao thông công cộng để bảo đảm sự đi

lại của nhân dân bằng những phương thức vận tải được lựa chọn thích hợp như: xe buýt, xe taxi, xe điện (tramway), xe chạy trên ray ở tầng cao (monorail hay sky-train) và hệ thống tàu điện ngầm (metro hay subway)

- Hệ thống cơ sở hạ tầng: Khi đường sá đô thị bị quá tải, không đủ cho phương tiện giao thông đi lại (tỷ lệ diện tích mặt đường trên diện tích chung của đô thị dưới 10%), lúc đó cần mở rộng không gian để tăng thêm đường, thiết lập các nút giao thông lập thể, hệ thống đường vượt trên cao
- Mở rộng diện tích, giảm độ tập trung dân cư: để đối phó với tốc độ tăng trưởng dân số ở những đô thị không thể dung nạp thêm dân hoặc duy trì nguyên trạng cảnh quan của "thành phố lịch sử" cần bảo tồn, bảo tàng, người ta thường xây dựng một số đô thị vệ tinh để giãn dân. Trường hợp này rất phổ biến ở Thủ đô hay các trung tâm kinh tế – thương mại sầm uất của một số nước phát triển nhằm tránh phải cải tạo, chỉnh trang đô thị rất tốn kém.
- Ý thức người dân: Giáo dục ý thức người dân theo kế hoạch dài hạn, qui mô toàn dân, về chấp hành giao thông (Một quốc gia có giáo dục ý thức điển hình là Singapore).Thực hiện các hoạt động vận động,tuyên truyền.

- Giải pháp tình thế
  - Phân luồng giao thông.
  - Uốn nắn tuyến.
  - Điều chỉnh dải phân cách.
  - Hạn chế lưu lượng và tốc độ lưu thông.
  - Tăng cường CSGT điều khiển các nút kẹt xe.
  - "Dẹp" càng nhanh càng tốt các công trình xây dựng đang "nghênh chiến" trên đường.
  - Dừng ngay điệp khúc "đào - lấp, đào - bỏ đó… "
  - Phải kiên quyết chuyển các hoạt động quét hốt rác, vận chuyển rác, sơn đường, rửa đường, vẽ đường, nạo vét cống… vào ban đêm khi có ít phương tiện lưu thông.
  - Thực hiện nghiêm các biện pháp chống buôn bán lấn chiếm lòng lề đường.
- Giải pháp của chúng tôi

  Với tình trạng giao thông trên,chúng tôi xây dựng hệ thống thu thập,thông báo và dự báo trực tiếp đến người tham gia giao thông.

  Hiện nay, có nhiều phương pháp thu thập dữ liệu giao thông như thu thập từ những vị trí cố định trên các tuyến đường, dữ liệu hình ảnh qua các camera, hay các thiết bị thu thập dữ liệu được gắn trên phương tiện giao thông (taxi, xe buýt,…). Tuy nhiên với những phương pháp này, việc duy trì cũng như triển khai tốn rất nhiều công sức tiền bạc.

Trên thế giới đã có nhiều nước xây dựng hệ thống ước lượng và dự đoán tình trạnh giao thông như Mỹ, Anh, Ý, Singapore,.. Hiện tại ở Việt Nam, chúng ta chỉ có một hệ thống thông báo tình trạng giao thông, kẹt xe trên sóng FM nhằm giúp người tham gia giao thông nắm được tình trạng giao thông và tránh những điểm kẹt xe hiện tại. Tuy nhiên, tình trạng giao thông được thông báo thông qua các nhân viên dựa trên dữ liệu hiện tại thu thập được từ camera, hay ở một số vị trí xác định. Cách làm này còn thủ công và tốn tài nguyên công sức.

Ngoài những cách thu thập dữ liệu trên, việc thu thập thông tin về giao thông thông qua hệ thống định vị GPS được tích hợp trên điện thoại di động ngày càng hứa hẹn. Ngày nay với sự phát triển mạnh mẽ của kỹ thuật công nghệ, Smartphone với những tính năng phong phú tiên tiến ngày càng phổ biến và giữ vai trò quan trọng đối với người dân.Thêm vào đó, sự phát triển của mạng không dây 3G cũng góp phần hỗ trợ cho các công nghệ mới. Do vậy việc thu thập dữ liệu vị trí khi tham gia giao thông trở nên dễ dàng. Chi phí triển khai và duy trì sẽ thấp hơn rất nhiều lần, ngoài ra chúng ta có thể thu thập được một lượng dữ liệu vô cùng lớn từ người tham gia giao thông. Do đó kết quả về tình trạng giao thông hiện tại cũng như dự báo sẽ chính xác hơn.

1.2 Mục tiêu

Mục tiêu chính của nhóm chính là xây dựng thành công module thu thập dữ liệu cho ứng dụng.

Dữ liệu thu thập sẽ bao gồm các điểm người sử dụng đi qua,thời gian và vận tốc tức thời của người sử dụng tại một điểm đi qua bất kì.Hệ thống sẽ thu thập dữ liệu này bằng cách sử dụng tín hiệu GPS của người dùng.

Dữ liệu thu thập được sẽ được gửi lên server thành công và hệ thống sẽ cho lưu file gpx chứa thông tin đoạn đường người dùng đã đi qua vào thẻ nhớ của smart phone thành công.

Xây dựng web site giúp người dùng có thể theo dõi việc di chuyển của các thiết bị và hiển thị tình trạng giao thông.

1.3Khả năng ứng dụng và ý nghĩa của đề tài

1.3.1 Khả năng ứng dụng

- Hiện nay với sự phát triển mạnh mẽ của Smartphone và hệ thống mạng không dây,đề tài có thể dễ dàng được thực hiện:
    - Smartphone: hiện nay Smartphone đang phát triển mạnh mẽ với nhiều tính năng ưu việt và người dùng không phải bỏ quá chi phí lớn để sở hữu một Smartphone nên chúng đang trở nên rất phổ biến trong cộng đồng người

Việt.Từ học sinh, sinh viên đến những người đã đi làm hay doanh nhân đều có thể có Smartphone.Trong hầu hết các Smartphone hiện nay đều có GPS và hỗ trợ mạng không dây 3G.

- Mạng không dây: hiện nay hầu hết các nhà cung cấp dịch vụ mạng như MobilePhone, VinaPhone hay Viettel đều cung cấp dịch vụ mạng 3G cho thuê bao di động với giá cước rẻ khoảng 40000-60000/tháng với lượng lưu lượng sử dụng là thoải mái và tốc độ sử dụng ổn định. Với chính sách này người dùng sẽ không ngần ngại chi ra một khoảng tiền nhỏ để có thể truy cập internet trên Smartphone của mình một cách thoải mái.Bên cạnh đó,tốc độ upload của các mạng không dây là tương đối tốt,điều này sẽ giúp cho việc gửi thông tin lên server sẽ được liên tục và không bị ngắt quãng.

- Việc kẹt xe trong quá trình giao thông gây ảnh hưởng lớn đến công việc, tài sản của người dân nên người dân sẽ không ngại để sử dụng ứng dụng của chúng tôi và cung cấp thông tin cho chúng tôi để ứng dụng có thể hoạt động tốt hơn.Chi phí người dùng bỏ ra rõ ràng là không cao.
- Chi phí triển khai và duy trì sẽ thấp

Với các yếu tố trên khả năng triển khai và ứng dụng của đề tài sẽ rất khả quan.

1.3.2 Ý nghĩa

Với việc đề tài được triển khai thành công, chúng ta sẽ xây dựng được một hệ thống có thể:

- Thông báo tình trạng giao thông đến người dùng ngay khi người dùng yêu cầu giúp người dùng có thể chọn lựa đường đi một cách chính xác, giúp người dùng tiết kiệm thời gian và chi phí trong việc tham gia giao thông
- Gợi ý đường đi cho người dùng.Với việc chức năng này hoạt động tốt,chúng ta có thể góp phần vào việc điều phối giao thông ở thành phố Hồ Chí Minh.
- Giúp người dùng có thể lưu lại những đoạn đường người dùng đã đi qua trong Smartphone
- Với hệ thống web, người dùng có thể tracking được vị trí của thiết bị di động.Với chức năng này chúng ta có thể lấy được thiết bị khi bị mất hoặc hỗ trợ các ông bố, bà mẹ trong việc quản lý con mình.

1.4Giới thiệu ứng dụng

Để góp phần giảm tình trạng kẹt xe cũng như bảo vệ lợi ích của người dân, việc xây dựng một hệ thống thu thập dữ liệu để thông báo, dự đoán tình trạng kẹt xe dựa trên tín hiệu GPS mở ra một hướng tiếp cận mới cho chúng ta. Trong đó, việc thu thập dữ liệu

từ người tham gia giao thông là một bài toán quan trọng của hệ thống, đó cũng là đề tài mà nhóm chúng tôi thực hiện.

Nhóm chúng tôi kết hợp với một số thành viên nhóm khác quyết định xây dựng một ứng dụng 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 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, chi phí, bảo vệ sức khỏe, có tâm lý thoải mái đồng thời giúp cho tình trạng giao thông trên đường tốt hơn.

Ứng dụng 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.
- Xây dựng website theo dõi tình trạng giao thông.

Trong các chức năng trên, nhóm chúng tôi chịu trách nhiệm nghiên cứu và hiện thực chức năng thu thập dữ liệu giao thông của người dùng, lưu vào bộ nhớ máy và gửi lên server,lưu dữ liệu vào cơ sở dữ liệu,truy vấn dữ liệu và xây dựng website theo dõi tình trạng giao thông. Các chức năng còn lại sẽ được các nhóm khác phụ trách.

1.5 Kế hoạch

- Giai đoạn thực tập tốt nghiệp:
    - Tìm hiểu tổng quan về Android, cách lập trình trên Android
    - Tìm hiểu các công nghệ có liên quan như GPS, GoogleMaps.
    - Tìm hiểu các giải thuật về xử lý dữ liệu.
    - Tìm hiểu về GPX.
    - Tìm hiểu về cách thức trao đổi dữ liệu với server.

- Hiện thực thành công các module nhỏ. Nhóm thực hiện Module thu thập dữ liệu GPS từ những người dùng Android. Do đó, ở giai đoạn nhóm sẽ hiện thực các module nhỏ sau:
  - Hiện thực giao diện giao diện đơn giản cho ứng dụng.
  - Nhận được tín hiệu GPS từ smart phone tính toán vận tốc và thời gian tại các điểm người dùng đi qua.
  - Lưu dữ liệu vào file GPX và lưu vào thẻ nhớ.
  - Thực hiện giao tiếp giữa client và server.

- Giai đoạn luận văn: nhóm sẽ tiếp tục phát triển và hoàn thiện ứng dụng:
  - Hoàn thiện giao diện.
  - Gửi dữ liệu lên Server và lưu dữ liệu vào database của hệ thống.
  - Hoàn thành module thu thập dữ liệu
  - Xây dựng website theo dõi

CHƯƠNG2: KIẾN THỨC NỀN TẢNG

2.1 Hệ thống định vị toàn cầu

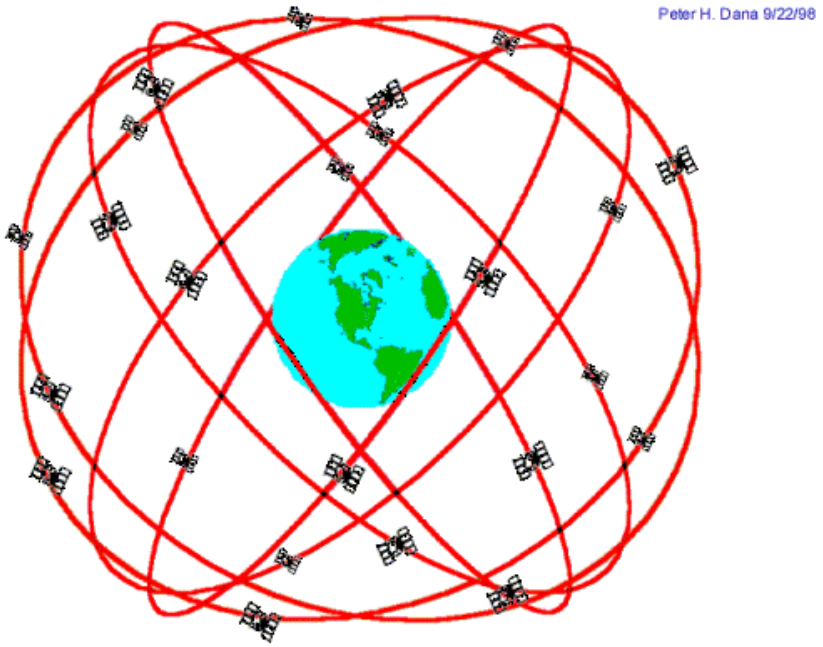
2.1.1 Định nghĩa

GPS là hệ thống xác định vị trí dựa trên vị trí của các vệ tinh nhân tạo, do Bộ Quốc phòng Hoa Kỳ thiết kế, xây dựng, vận hành và quản lý. Hệ thống này được phát triển với mục đích quân sự, nhưng hiện nay nó đã được mở rộng cho các mục đích dân sự.Trong cùng một thời điểm, tọa độ của một điểm trên mặt đất sẽ được xác định nếu xác định được khoảng cách từ điểm đó đến ít nhất ba vệ tinh.

2.1.2 Các thành phần của GPS

GPS hiện tại gồm 3 phần chính: phần không gian, phần kiểm soát và phần sử dụng.

- Phần không gian
  - Phần không gian gồm 24 vệ tinh chuyển động trên các quỹ đạo xung quanh trái đất. Chúng các mặt đất 20.200 km, bán kính quỹ đạo 26.600 km. Chúng chuyển động ổn định và quay hai vòng quỹ đạo trong khoảng thời gian gần 24 giờ với vận tốc 7 nghìn dặm một giờ. Các vệ tinh được bố trí sao cho các máy thu GPS trên mặt đất có thể nhìn thấy thối thiểu 4 vệ tinh vào bất kỳ thời điểm nào.
  - Mỗi vệ tinh nặng khoảng 2 tấn, sử dụng năng lượng Mặt trời. Ngoài ra, chúng có các nguồn pin dự phòng để duy trì hoạt động khi chạy vào vùng không có ánh sang Mặt trời.



- Phần kiểm soát
  Phần kiểm soát để kiểm soát vệ tinh đi đúng hướng theo quỹ đạo và thông tin thời gian chính xác.
- Phần sử dụng
  Phần sử dụng là các thiết bị nhận tín hiệu vệ tinh GPS và người sử dụng thiết bị.

2.1.3 Cách hoạt động của GPS

Các vệ tinh GPS bay vòng quanh Trái Đất 2 lần trong một ngày theo một quỹ đạo rất chính xác và phát tín hiệu có thông tin xuống Trái Đất. Các máy thu GPS nhận thông tin này và tính toàn bằng các phép tính lượng giác để xác định vị trí người dùng. Để thực hiện tính toán, máy thu GPS cần phải biết hai yếu tố: Vị trí của ít nhất 3 vệ tinh và khoảng cách giữa máy thu GPS đến từng vệ tinh nói trên. Về bản chất, máy thu GPS so sánh thời gian tín hiệu được phát từ vệ tinh với thời gian nhận được chúng. Sai lệch về thời gian cho biết máy thu GPS ở cách vệ tinh bao xa.

Khi nhận được tín hiệu của ít nhất 4 vệ tinh, máy thu có thể tính được vị trí ba chiều, bao gồm kinh độ, vĩ độ và độ cao của vị trí hiện tại.

2.1.4 Các yếu tố ảnh hưởng đến độ chính xác của tín hiệu GPS

- Tín hiệu vệ tinh bị chậm khi xuyên qua tầng khí quyển

- Tín hiệu đi nhiều đường: tín hiệu phản xạ từ nhà hay các đối tượng khác trước khi tới máy thu.
- Lỗi quỹ đạo, do vệ tinh thông báo vị trí không chính xác.
- Số lượng vệ tinh nhìn thấy: càng nhiều vệ tinh GPS nhìn thấy cho dữ liệu càng chính xác.
- Sự giảm tín hiệu vệ tinhcó chủ tâm: giảm tín hiệu vệ tinh do sự sắp đặt của Bộ Quốc phòng Hoa Kỳ.

2.2 Android

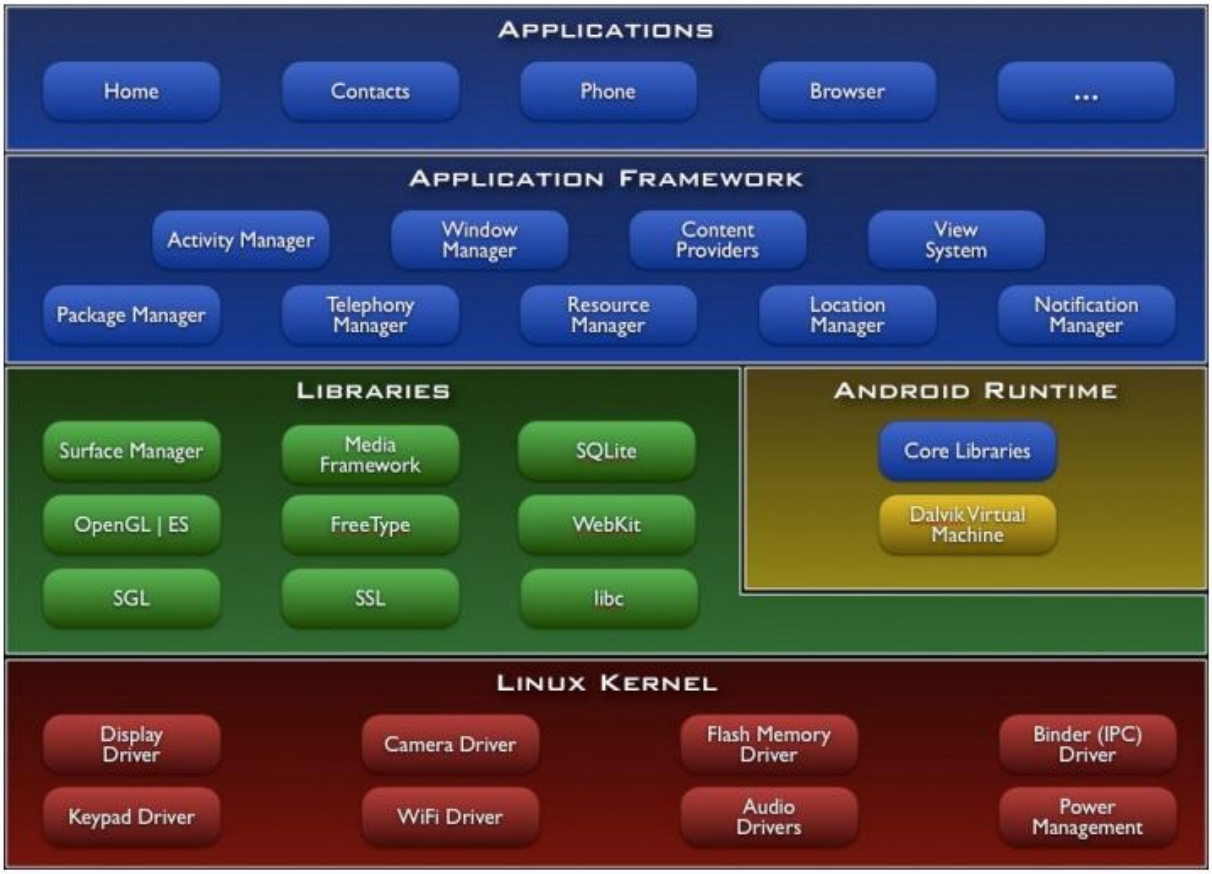
2.2.1 Định nghĩa Android

Android là nền tảng mã nguồn mở trên điện thoại di động bao gồm một hệ điều hành, middleware và một số ứng dụng chủ đạo; được phát triển bởi Google dựa trên nền tảng Linux.

2.2.2 Những tính năng mà nền tảng Android hỗ trợ

- Application framework: cho phép tái sử dụng và thay thế các thành phần có sẵn của Android
- Máy ảo Dalvik: Máy ảo java, tối ưu hóa cho thiết bị di động.
- Intergrated browser: Trình duyệt web tích hợp được xây dựng dựa trên WebKit engine
- Optimized graphics: hỗ trợ bộ thư viện 2D và 3D dựa vào đặc tả OpenGL ES 1.0.
- SQLite: Hệ quản trị cơ sở dữ liệu dùng để lưu dữ liệu có cấu trúc.
- Hỗ trợ các định dạng media phổ biến như MPEG4, h.264, MP3, AAC, JPG, PNG, GIF.
- Hỗ trợ thoại trên nền tảng GSM, camera, GPS, Bluetooth, EDGE, 3G và WiFi.

2.2.3 Kiến trúc Android



- Tầng ứng dụng

  Tầng này chứa các ứng dụng được tích hợp sẵn và các ứng dụng được viết bằng ngôn ngữ Java.

- Tầng Application Framework

  Tầng Application Framework cung cấp nền tảng phát triển ứng dụng mở, hỗ trợ các nhà phát triển ứng dụng tạo ra các ứng dụng.Các nhà phát triển ứng dụng được tự do sử dụng các API thao tác tính năng cao cấp của thiết bị phần cứng như thông tin định vị địa lý, dịch vụ chạy ngầm, thiết lập đồng hồ báo thức, thêm thông báo vào thanh trạng thái của màn hình thiết bị…

  Tầng này bao gồm một tập các services và các thành phần

  - Activity Manager: quản lý vòng đời của một ứng dụng.
  - Window Manager: quản lý việc xây dựng và hiển thị các giao diện người dùng.

- Content Provider: cho phép các ứng dụng có thể truy xuất dữ liệu từ các ứng dụng khác, hoặc chia sẻ dữ liệu của chúng.
- Resource Manager: cung cấp khả năng truy xuất các nguyên non-code như hình ảnh, file layout,…
- Notification Manager: cung cấp khả năng hiển thị các thông báo trên thanh trạng thái.
- Location Manager: cho phép xác định vị trí dựa vào hệ thống định vị toàn cầu GPS.
- Tập các đối tượng View như List, Text box, Button,…

- Tầng Libraries và Android Runtime

  Tầng này có 2 thành phần là phần Library và Android Runtime

  - Phần Library có nhiều thư viện được viết bằng ngôn ngữ C/C++.
    - System C library: Thư viện hệ thống, được xây dựng từ bộ thư viện hệ thống C chuẩn, được điều chỉnh để tối ưu hóa cho các thiết bị chạy trên nền Linux.
    - Media libraries: bộ thư viện hỗ trợ phát và ghi các định dạng âm thanh, hình ảnh phổ biến.
    - Surface manager: hỗ trợ quản lý hiển thị nội dung 2D và 3D.
    - LibWebCore: Thư viện web, được sử dụng để xây dựng phần mềm duyệt web.
    - SGL (Scalable Graphics Library): hỗ trợ đồ họa 2D.
    - 3D libraries: tối ưu hóa hiển thị 3D.
    - SQLite: hệ quản trị cơ sở dữ liệu nhỏ gọn, hỗ trợ cho ứng dụng.

  - Phần Android Runtime

    Android Runtime gồm các thư viện lõi và máy ảo Dalvik.

    - Các thư viện lõi cung cấp các chức năng có sẵn trong các thư viện lõi của ngôn ngữ Java như Java IO, truy xuất file,…
    - Máy ảo Dalvik là một biến thể của máy ảo Java, bổ sung các công nghệ đặc trưng cho thiết bị di động, mục đích giúp cho các thiết bị di động có thể chạy nhiều máy ảo. Mọi ứng dụng Android chạy trên một tiến trình riêng trên một thể hiện của máy ảo Dalvik.

- Tầng nhân Linux

  Android được phát triển dựa trên nhân Linux 2.6, hiện thực các dịch vụ hệ thống cốt lõi như bảo mật, quản lý bộ nhớ, quản lý tiến trình, giao tiếp với phần cứng,…

  - Display Driver: điều khiển việc hiển thị lên màn hình, thu nhận tương tác của người dùng.
  - Camera Driver: điều khiển hoạt động của camera, nhận luồng dữ liệu từ camera.
  - Bluetooth Driver: điều khiển thiết bị thu và phát Bluetooth.
  - USB Driver: quản lý hoạt động của các cổng USB.
  - Keypad Driver: điều khiển bàn phím.
  - Wifi Driver: quản lý thu phát Wifi.
  - Audio Driver: điều khiển các bộ thu phát âm thanh, giải mã các tín hiệu dạng audio thành tín hiệu số và ngược lại.
  - Flash Memory Driver: quản lý việc đọc ghi lên thẻ nhớ.

2.2.4 Phát triển ứng dụng trên Android

- Ngôn ngữ lập trình

  Ngôn ngữ lập trình chính thức của Android là Java. Mặc dù các ứng dụng trên Android được phát triển dựa trên nền tảng Java, nhưng Android ko hỗ trợ J2ME và J2SE, là 2 ngôn ngữ lập trình phổ dụng cho các thiết bị di động.

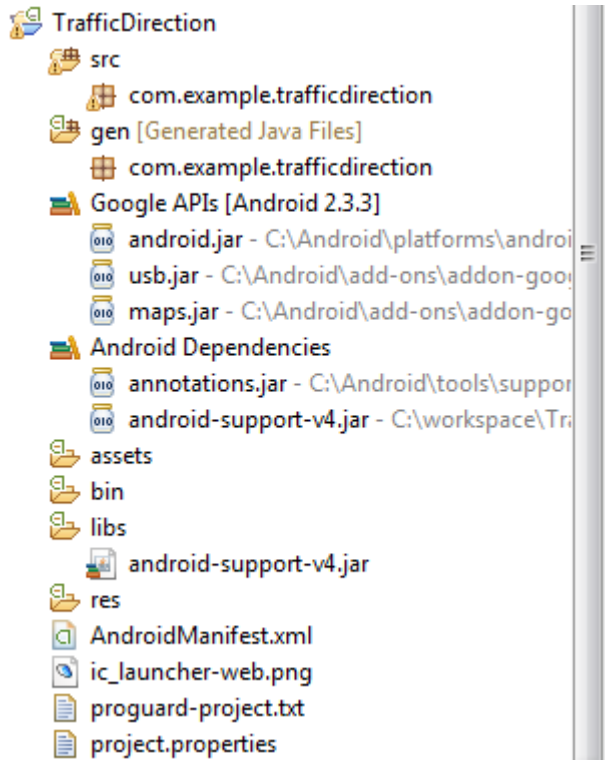
  Bằng cách sử dụng các Frameworks của các hãng thứ ba như PhoneGap, Titanium,… các nhà lập trình Web cũng có thể phát triển ứng dụng Android một cách dễ dàng bằng các ngôn ngữ Web phổ biến như HTML, CSS, Javascript. Tuy nhiên, số lượng các ứng dụng đi theo hướng này chưa nhiều.

- Môi trường lập trình cho Android

  Android SDK bao gồm các công cụ riêng lẻ như: debugger, các thư viện, trình giả lập điện thoại Android, các tài liệu hỗ trợ và code mẫu. Hiện Android cung cấp bộ công cụ này trên nhiều nền tảng hệ điều hành khác nhau (Windows, Linux, Mac,...), miễn là có sẵn Java Development Kit, Apache Ant và Python2.2 trở lên.

  Môi trường lập trình (IDE) chính thức của Android là Eclipse (từ phiên bản 3.2) với sự hỗ trợ của plugin Android Development Tools (ADT).

- Các thành phần cơ bản của một Project Android trên Eclipse



Note: Cấu trúc thư mục và file của một dự án phần mềm Android trên Eclipse

- AndroidManifest.xml: file XML mô tả ứng dụng và các thành phần được cung cấp bởi ứng dụng như Activities, Services,…
- src/: nơi chứa các file mã nguồn Java.
- gen/: nơi chứa file R.java. File R.java là một file được tự động sinh ran gay khi tạo ứng dụng, file này dùng để quản lý các thuộc tính được khai báo trong file XML và các tài nguyên hình ảnh. Mã nguồn của file R.java được tự động sinh ra khi có bất kỳ một sự kiện làm thay đổi thuộc tính trong ứng dụng.
- assets/: nơi chứa các files tính được yêu cầu đi kèm với ứng dụng.
- bin/: nơi chứa ứng dụng sau khi được biên dịch.
- libs/: nơi chứa các file thư viện JAR.
- res/: nơi chứa các file tài nguyên của ứng dụng như hình ảnh, layouts,…

2.2.5 Các thành phần chính của một ứng dụng Android

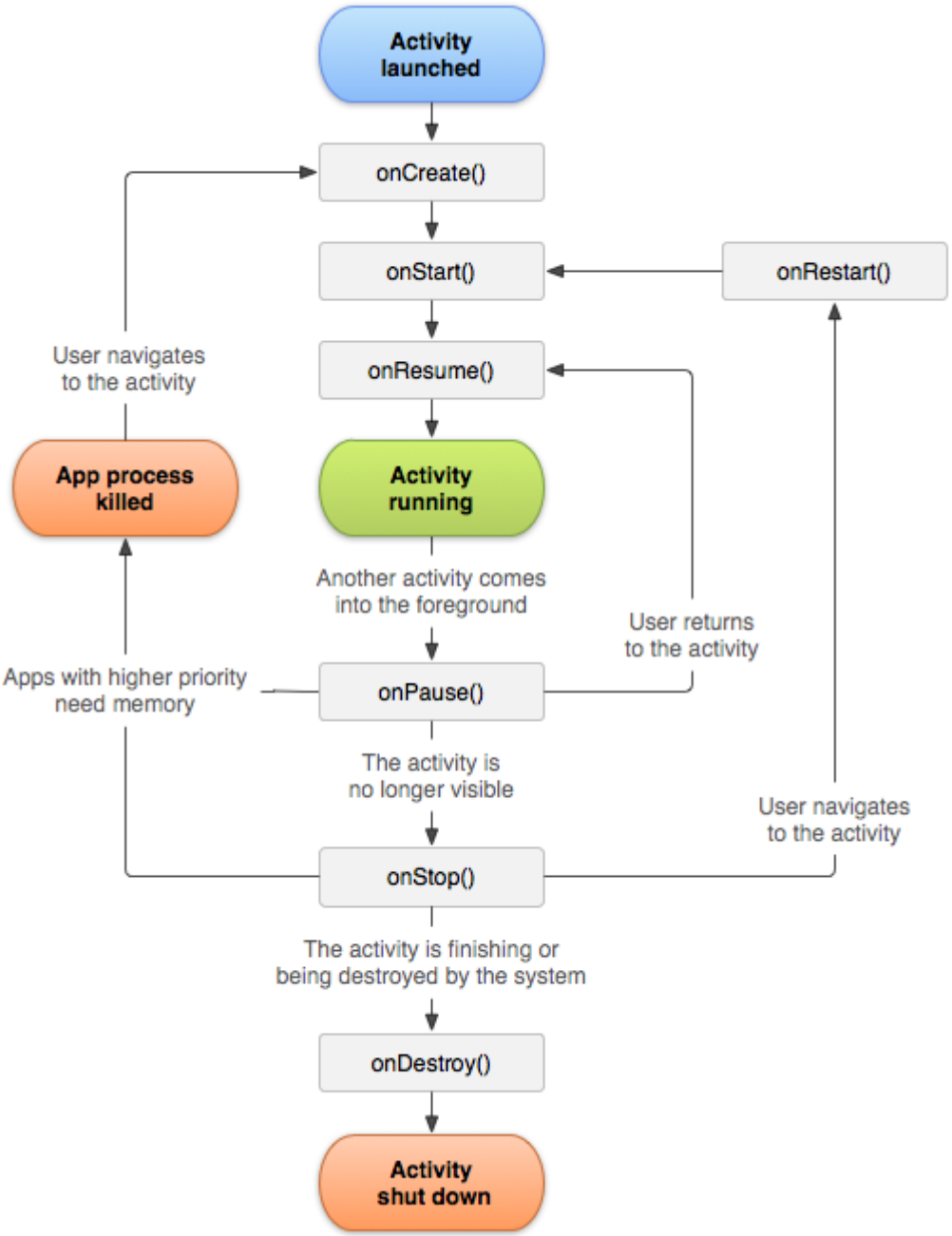
- Activity

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

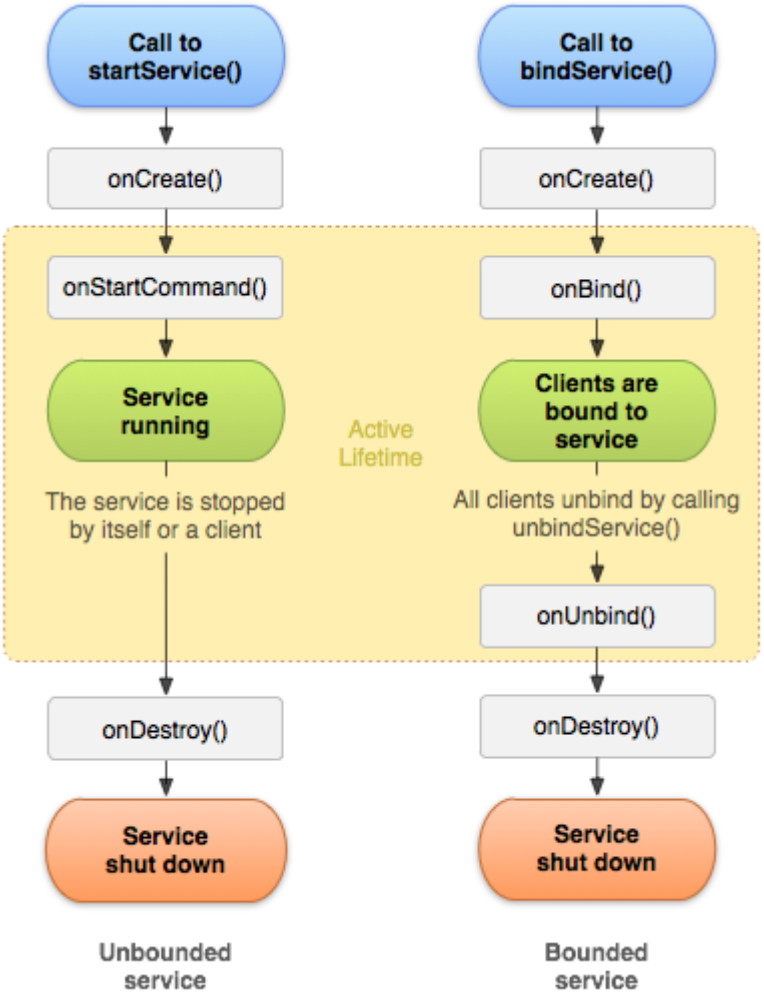
- Active/Running: Khi Activity đang chạy trên màn hình.
- Paused: Khi một Activity vẫn đang chạy trên màn hình nhưng bị một Activity không chiếm toàn màn hình hiển thị phía trên.
- Stopped: Khi một Activity bị che khuất hoàn toàn bởi một Activity khác.Nó vẫn có thể được đưa trở lại để trở thành một Activity 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.



- 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().



  - 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.

2.2.6 Android Location API

Hầu hết các thiết bị Android cho phép xác định vị trí hiện tại.Việc xác định vị trí có thể thông qua GPS, hoặc thông qua mạng Wifi, hoặc qua các trạm phát sóng điện thoại.

Gói android.location cung cấp các API để xác định vị trí hiện tại.

- Criteria: là lớp đưa ra các tiêu chuẩn để chọn nhà cung cấp địa điểm tốt nhất.
- Location: lớp biểu diễn vị trí địa lý của một điểm. Một điểm gồm kinh độ, vĩ độ, mốc thời gian, và các thông tin khác như độ cao, tốc độ.
  - getLatitude(): lấy vĩ độ, đơn vị độ.
  - getLongitude(): lấy kinh độ, đơn vị độ.
  - hasAltitude(): kiểm tra một vị trí có thông tin về độ cao hay không.
  - getAlitude(): lấy độ cao so với mực nước biển, đơn vị mét.
  - has Accuracy():kiểm tra một vị trí có thông tin về độ chính xác hay không.
  - getAccuracy(): dự đoán độ chính xác của điểm, đơn vị mét.
  - hasSpeed():kiểm tra một vị trí có thông tin về độ vận tốc hay không.
  - getSpeed(): lấy vận tốc của một vị trí, đơn vị m/s.

- LocationManager: cung cấp truy cập vào các dịch vụ định vị hệ thống. Những dịch vụ này cho phép ứng dụng cập nhật định kỳ vị trí địa lý của thiết bị.
  - getAllProviders(): lấy về danh sách các nhà cung cấp vị trí.
  - getBestProvider(Criteria criteria, Boolean enabledOnly): trả về tên nhà cung cấp thõa mãn các tiêu chí được đưa ra.

- getLastKnownLocation(String provider): trả về vị trí vừa được cập nhật từ nhà cung cấp.
- requestLocationUpdates(String provider, long minTime, float minDistance, LocationListener listener): yêu cầu cập nhật vị trí, sử dụng nhà cung cấp provider.
- removeUpdates(LocationListener listener): gỡ bỏ tất cả cập nhật vị trí cho listener được đưa ra.

- LocationListener: nhận thông báo từ LocationManager khi vị trí thay đổi. Các phương thức bên dưới sẽ được gọi nếu như LocationListener được đăng ký với dịch vụ quản lý vị trí thông qua phương thức requestLocationUpdates(String, long, float, LocationListener).
  - onLocationChanged(Location location): được gọi khi vị trí thay đổi.
  - onProviderDisabled(String provider): được gọi khi người dùng tắt provider.
  - onProviderEnabled(String provider): được gọi khi người dùng mở provider.
  - onStatusChanged(String provider, int status, Bundle extras): được gọi khi trạng thái provider thay đổi.

2.3 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:gpx10="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="106.66313666666669">
<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>
```

- 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.

2.4 Giao thức UDP

UDP (User Datagram Protocol) là một trong những giao thức cốt lõi của giao thức TCP/IP. Dùng UDP, chương trình trên mạng máy tính có thể gởi những dữ liệu ngắn gọi là datagram. Trong giao thức UDP, khi phía gửi và phía nhận giao tiếp với nhau, không cần phải tạo kết nối giữa chúng, không cần phải ghi nhớ trạng thái gửi/nhận.Các gói tin được gửi độc lập từ máy gửi tới máy nhận.

- Nhược điểm: Không cung cấp sự tin cậy cho gói dữ liệu và thứ tự truyền nhận các gói dữ liệu. Do đó, các gói dữ liệu có thể đến không đúng thứ tự, và không thể đảm bảo gói dữ liệu được gửi đến đích.
- Ưu điểm: UDPđơn giản, có độ trễ nhỏ,truyền dữ liệu nhanh và hiệu quả, có thể đáp ứng các yêu cầu nhỏ với lượng người dùng lớn.
- Bảo mật trong UDP: Mã hóa các gói dữ liệu trước khi gửi.

2.5 JDBC

2.5.1 Định nghĩa JDBC

JDBC (viết tắt của cụm: "Java Database Connectivity") là kiến trúc, các đặc tả, và giao diện ứng dụng dùng để truy cập dữ liệu. JDBC API là một giao diện lập trình sử dụng SQL, nó định nghĩa các lớp Java trừu tượng hóa các kết nối CSDL, các câu lệnh SQL, các tập hợp kết quả, các siêu dữ liệu ... Nó cho phép một người lập trình Java đưa ra các câu lệnh SQL và xử lí các kết quả được trả về.

2.5.2 Lợi ích của JDBC

JDBC tách các nhà phát triển ứng dụng khỏi sự phức tạp của việc kết nối tới một nguồn dữ liệu.Tiêu chí của JDBC là nó phải dễ dàng cho người lập trình ứng dụng có thể tạo ra các kết nối của người sử dụng cuối tới nguồn dữ liệu thích hợp mà không phải trở thành một chuyên gia về mạng. Các nhà phát triển Java có thể tạo nên các ứng dụng truy xuất cơ sở dữ liệu mà không cần phải học và sử dụng các API độc quyền do các công ty sản xuất phần mềm khác nhau ở bên thứ ba cung cấp. JDBC ứng với từng hệ quản trị csdl mà sẽ sử dụng các driver khác nhau, do đó, tính mềm dẻo của CSDL JDBC rất cao, có thể tương thích với hầu hết các RDBMS (Relational Database Management System).

2.5.3 Kiến trúc JDBC

- JDBC API (các gói java.sql & javax.sql)
  - JDBC và Java nói chung đều có đặc điểm là không có nhiều các lớp công nghệ (class) được xây dựng sẵn, mà bù lại, Java định ra nhiều Interface, từ đó các hãng, các tổ chức open source sẽ Implements các Interface này. Do đó, dù nhiều công nghệ được phát triển, nhưng Java vẫn có tính thống nhất chung.
  - JDBC cũng vậy, nó chứa một tập hợp các lớp, các giao diện Java và các exception. Đa số các API này được định nghĩa trong gói "java.sql". Các lớp và giao diện quan trọng trong gói java.sql

    - DriverManager: Nạp các JDBC driver vào trong bộ nhớ. Có thể sử dụng nó để mở các kết nối tới một nguồn dữ liệu. Tạo sẵn cơ chế phân chia đường kết nối (built-in connection pooling).
    - Connection: Biểu thị một kết nối đến một nguồn dữ liệu. Được dùng để tạo ra các đối tượng Statement, PreparedStatement và CallableStatement.
    - Statement: Biểu diễn một lệnh SQL tĩnh. Có thể sử dụng nó để thu về đối tượng ResultSet.

- PreparedStatement: Một giải pháp thay thế hoạt động tốt hơn đối tượng Statement, thực thi một câu lệnh SQL đã được biên dịch trước.
- CallableStatement: biểu diễn một thủ tục được lưu trữ. Có thể được sử dụng để thực thi các thủ tục được lưu trữ trong một RDBMS có hỗ trợ chúng.
- ResultSet: biểu diễn một tập kết quả trong cơ sở dữ liệu tạo ra bởi việc sử dụng một câu lệnh SQL là SELECT.
- SQLException: một lớp xử lý lỗi ngoại lệ chứa các lỗi truy cập cơ sở dữ liệu.

- Thiết lập kết nối đến CSDL
  Trong Java có 2 lớp chủ yếu chịu trách nhiệm về thiết lập kết nối đến một cơ sở dữ liệu:
  - Lớp đầu tiên là DriverManager. Đó là một trong rất ít các lớp thực sự do JDBC API cung cấp. DriverManager chịu trách nhiệm quản lý một nhóm (pool) các driver đã đăng kí, mà thực chất là là trừu tượng hóa các chi tiết về việc sử dụng một driver, cho nên lập trình viên không cần phải làm việc trực tiếp với driver đó. Như tên gọi của nó, nhiệm vụ của nó là quản lý sự tương tác giữa các chương trình ứng dụng và các driver, nhiều ứng dụng và nhiều driver có thể được quản lý cùng một lúc. Driver Manager cung cấp sự liên kết giữa các ứng dụng và các driver, cho phép nhiều ứng dụng truy xuất dữ liệu qua nhiều driver. Driver Manager load hay unload một hoặc nhiều driver cho một hoặc nhiều ứng dụng. Khi một ứng dụng cần truy xuất một nguồn dữ liệu, Driver Manager sẽ load đúng driver cần thiết.

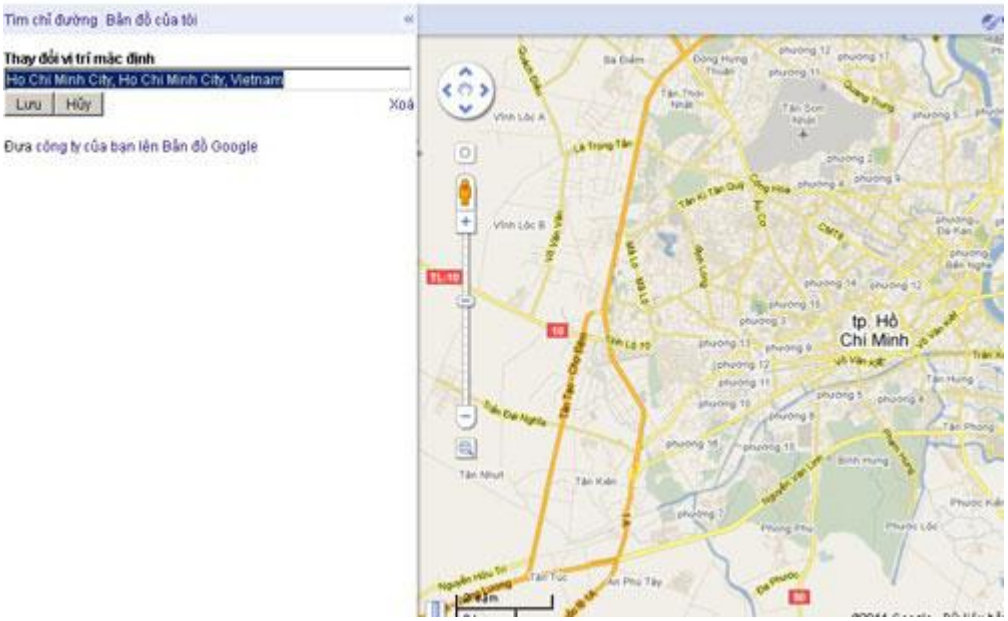
  - Lớp thứ 2 là lớp JDBC Driver. Nó được cung cấp bởi các nhà sản xuất phần mềm độc lập. Lớp JDBC Driver chịu trách nhiệm thiết lập đường kết nối cơ sở dữ liệu và xử lý tất cả các giao tiếp với cơ sở dữ liệu đó: đưa ra các yêu cầu SQL để chỉ định các nguồn dữ liệu, và trả về kết quả cho các ứng dụng. Các driver cùng đảm nhận việc tương tác với bất cứ các lớp phần mềm nào cần thiết để truy xuất nguồn dữ liệu. Các JDBC driver chia thành 4 kiểu khác nhau nhưng kiểu thông dụng nhất là JDBC Driver loại 4 - Chúng được viết thuần túy bằng Java và là loại hiệu quả nhất. Chúng cho phép kết nối trực tiếp vào cơ sở dữ liệu, cung cấp kết quả tối ưu và cho phép lập trình viên thực hiện các chức năng tùy thuộc vào cơ sở dữ liệu cụ thể. Điều này đã tạo ra tính cơ động cao nhất là khi bạn cần thay đổi cơ sở dữ liệu bên dưới một ứng dụng. Loại driver này thường được dùng cho các ứng dụng phân tán cao.

2.6 Google Maps

2.6.1 Giới thiệu về Google Maps

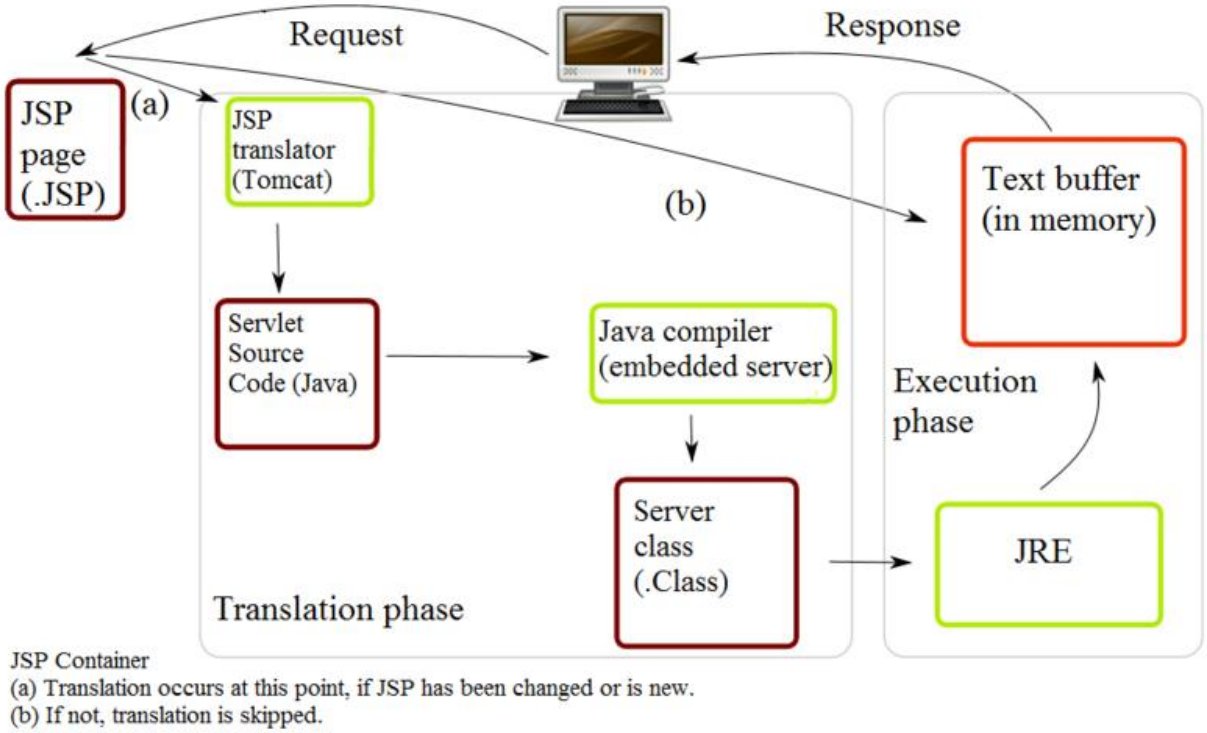
Google Maps (thời gian trước còn gọi là Google Local) là một dịch vụ ứng dụng và công nghệ bản đồ trực tuyến trên web miễn phí được cung cấp bởi Google và hỗ trợ nhiều dịch vụ dựa vào bản đồ như Google Ride Finder và một số có thể dùng để nhúng vào các trang web của bên thứ ba thông qua Google Maps API. Nó cho phép thấy bản đồ đường sá, đường đi cho xe đạp, cho người đi bộ (những đường đi ngắn hơn 6.2 dặm) và xe hơi, và những địa điểm kinh doanh trong khu vực cũng như khắp nơi trên thế giới.
Địa chỉ chính thức của Google Maps: http://maps.google.com

Google Maps cung cấp các cách thể hiện bản đồ khác nhau như bản đồ giao thông, bản đồ vệ tinh, bản đồ ba chiều.Giao diện của chương trình rất thân thiện với người dùng.



2.7 Web server

2.7.1 Java Servlet



(a) Translation occurs at this point, if JSP has been changed or is new.
(b) If not, translation is skipped.

Servlet là một kỹ thuật server của Java. Về mặt kỹ thuật, một Servlet là một lớp trong Java EE, một giao thức mà một lớp Java có thể đáp ứng yêu cầu từ phía client. Servlets về nguyên tắc có thể giao tiếp qua bất kỳ giao thức client-server, nhưng chúng được sử dụng thường xuyên nhất với giao thức HTTP. Vì vậy, từ "Servlet" thường được sử dụng trong ý nghĩa của "Servlet HTTP". [nguồn] Như vậy, một nhà phát triển phần mềm có thể sử dụng một servlet để thêm nội dung động đến một máy chủ web sử dụng nền tảng Java. Các nội dung được tạo ra thường là HTML, nhưng có thể là các dữ liệu khác như XML. Servlets có thể duy trì trạng thái trong các biến.

API Servlet, chứa trong gói javax.servlet của Java, định nghĩacác tương tác của những trang web container và servlet .

Servlet là một đối tượng nhận được yêu cầu và tạo ra một phản ứng dựa trên yêu cầu đó. Kết quả trả về có thể có nhiều kiểu khác nhau như HTML,JSON.

2.7.2 HTML

HTML được viếtbaogồmcác thẻkèm theo trongdấu ngoặc vuông góc(như <html>), trong nội dungtrang web. Thẻ HTMLthườngđitheo cặpnhư<h1>và </ h1>. Thẻđầutiêntrong một cặplàbắt đầu từ khóa, thẻ thứhailà thẻ kết thúc(còn được gọi là thẻ mởvà đóng thẻ). Tronggiữa những thẻ này,thiết kế webcóthểthêm văn

bản,thẻ,nhậnxétvà các loạinội dungdựa trên văn bản.

Mục đích của một trình duyệtweblà để đọc các fileHTMLvàhiển thị nội dung trang web. Các trình duyệt khônghiển thị các thẻHTML, nhưng sử dụngcácthẻ để biên dịchnội dung của trang.

HTMLchophépchèn hình ảnh và các đội tượng khác như video,audio. Nó cung cấp mộtphương tiện đểtạo ra các tài liệucócấutrúcbằngcáchbiểu thịcác thành phần củavăn bản nhưtiêu đề,đoạn văn,danh sách, liên kếtvà các thành phầnkhác. Nócó thể nhúngcáckịchbảnbằng các ngôn ngữnhư JavaScriptcó ảnh hưởng đếnhoạt động của các trang webHTML.

2.7.3 CSS

CSS được sử dụng để tùy chỉnh style như font chữ,kích cỡ,màu sắc và layout cho webpage. CSS thường được viết vào phần header của file HTML hoặc viết ở một file riêng.

2.7.4 Javascript

JavaScript (đôi khiđược viết tắt làJS)là một ngôn ngữkịch bản thường đượcthực hiện nhưmộtphầncủa một trình duyệtwebđểtạo ra các giao diệncó tính tương tác cao vàcác trang web động.

2.7.5 Ajax

Ajax là một nhóm kỹ thuật web được sử dụng bên phía client để tạo ra các ứng dụng web bất đồng bộ.Với ajax,các ứng dụng webcóthểgửi dữ liệu, và lấy dữ liệutừmột máy chủkhông đồng bộ (trong nền) mà không can thiệpvớimàn hình hiển thịvà hoạt độngcủa tranghiệntại. Dữ liệu có thểđược lấy rabằngcáchsửdụngđối tượng XMLHttpRequest. Mặc dù tên gọi có chưa XML, việc sử dụng XMLkhông cần thiết(JSONthường được sử dụngthay thế).

CHƯƠNG 3: PHÂN TÍCH HỆ THỐNG

3.1 Phân tích cho toàn hệ thống

3.1.1 Giới thiệu tổng quát

Ứng dụng cung cấp chức năng thu thập dữ liệu giao thông - hỗ trợ cho việc dự báo tình trạng giao thông, gợi ý đường đi, và để người dùng lưu trữ, xem lại các tuyến đường đã đi qua. Các thông tin này được hiển thị thông qua Smartphone hoặc thông qua Web.

3.1.2 Các chức năng chính của ứng dụng

- Phía Client
  - Thu thập dữ liệu từ người dùng: thông qua tín hiệu GPS, hoặc qua mạng 3G, ứng dụng sẽ thu thập dữ liệu và gửi lên Server.
  - Hiển thị vị trí hiện tại: vị trí hiện tại của người dùng được hiển thị tại trung tâm màn hình ứng dụng.
  - Hiển thị thôngtin giao thông: ứng dụng cập nhật thông tin giao thông từ Server và hiển thị lên màn hình ứng dụng, bao gồm tình trạng giao thông hiện tại, tình trạng giao thông được dự báo sau các khoảng thời gian.
  - Gợi ý đường đi: người dùng nhập vào địa điểm đích, ứng dụng đưa ra các gợi ý từ điểm hiện tại đến đích.

- Phía Server
  - Lưu trữ dữ liệu: nhận dữ liệu thô từ Client, kết nối đến Database Server, ghi dữ liệu xuống Database.
  - Lọc dữ liệu: đánh giá độ chính xác của dữ liệu, loại bỏ các dữ liệu nhiễu.
  - Dự báo tình trạng giao thông: sử dụng dữ liệu đã được làm sạch, kết hợp với dữ liệu trong quá khứ, đưa ra dự báo tình trạng giao thông.
  - Gợi ý đường đi: loại bỏ các điểm có thể kẹt xe, đưa ra các gợi ý đường đi giữa hai điểm.

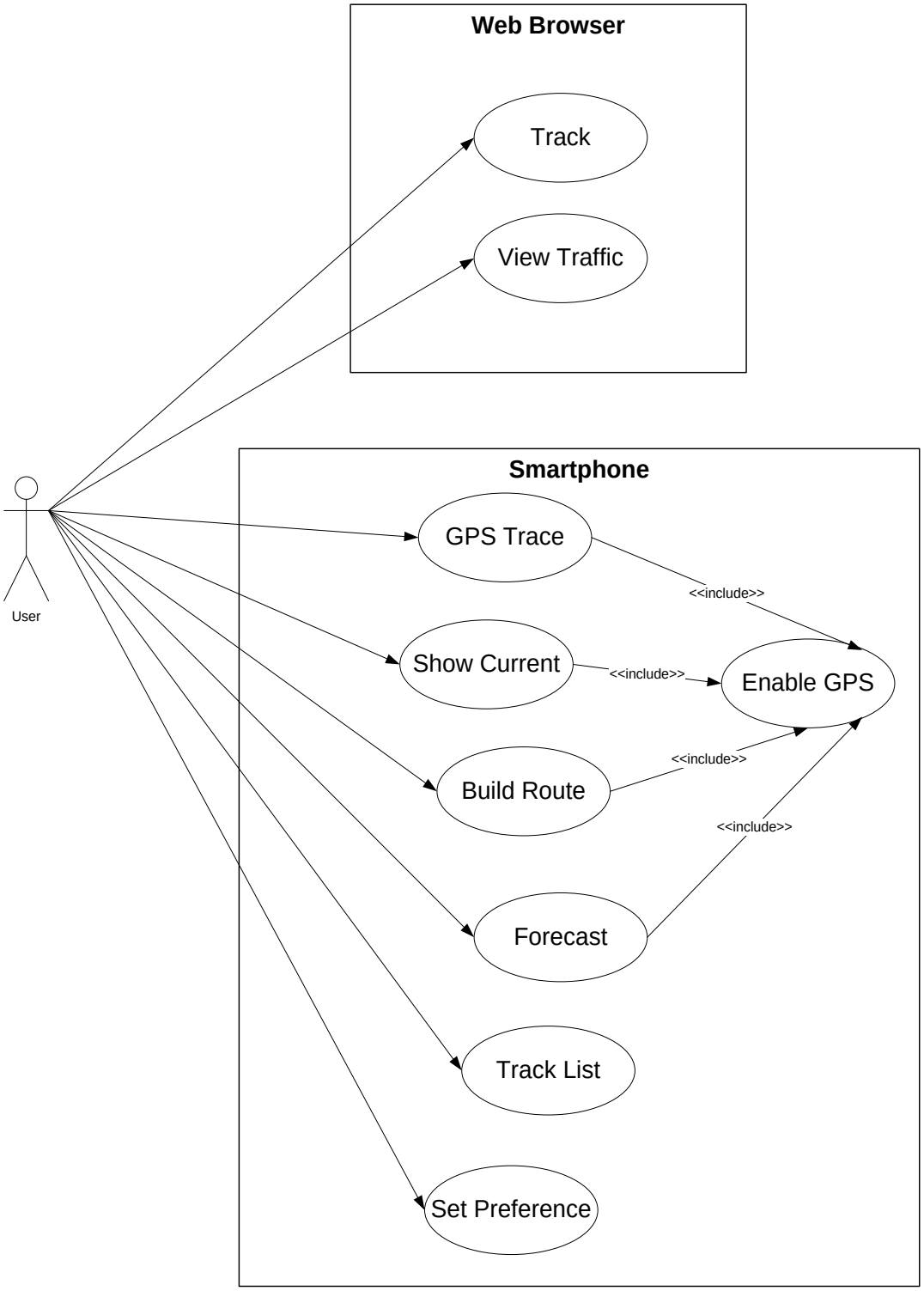
- Phía Browser
  - Theo dõi dữ liệu theo một Device: người dùng đăng nhập vào hệ thống, ứng dụng sẽ hiển thịhành trình của thiết bị.
  - Hiển thị tình trạng giao thông: ứng dụng truy xuất cơ sở dữ liệu để lấy thông tin tình trạng giao thông, và hiển thị lên Web.

3.1.3 Ràng buộc

- Bộ nhớ của smartphone nhỏ

- Ngôn ngữ lập trình: Java
- IDE: Eclipse
- Server: Tomcat
- Hệ điều hành của smart phone: android 2.3 trở về sau

3.1.4 Mô hình usecase



Đặc tả actor

- Yêu cầu xem vị trí hiện tại của mình.
- Yêu cầu dự báo tình trạng giao thông.
- Yêu cầu tìm đường đi.
- Yêu cầu xem lại các đường đi cũ.
- Yêu cầu thu thập dữ liệu.
- Yêu cầu tùy chỉnh ứng dụng.
- Yêu cầu xem tình trạng giao thông qua web browser.
- Yêu cầu theo dõi sự di chuyển của người dùng qua web browser.

Mô tả usecase

Tên UseCase	Show Current
Actor	User
Mục đích	Xem vị trí hiện tại của người dùng
Pre-condition	Smart phone có kết nối wifi hoặc 3G và đã kích hoạt chức năng GPS
Extended	
Include	
Main Flow	-Người dùng nhấn vào Menu -> Menu Show Current -Chọn chế độ hiển thị: Velocity hoặc Density, sau đó nhấn button OK. -Nếu chọn Velocity -Nếu chọn Density

Tên UseCase	Forecast
Actor	User
Mục đích	Xem dự báo tình trạng giao thông quanh vị trí đang đứng
Pre-condition	Smartphone có kết nối Wifi hoặc 3G

Extended	
Include	
Main Flow	-Người dùng nhấn vào Menu -> Menu Forecast -Thông tin dự báo giao thông quanh vị trí hiện tại sẽ được hiển thị lên màn hình.

Tên UseCase	Build Route
Actor	User
Mục đích	Yêu cầu gợi ý đường đi giữa 2 điểm
Pre-condition	Smartphone có kết nối Wifi hoặc 3G
Extended	
Include	
Main Flow	-Người dùng nhấn vào Menu -> Menu Forecast -Người dùng nhập 2 điểm vào BuildRouteform của ứng dụng và nhấn button Search -Đường đi giữa 2 điểm sẽ được thể hiện lên màn hình.

Tên UseCase	TrackList
Actor	User
Mục đích	Xem lại các đường đi đã thu thập
Pre-condition	Các file GPX đã thu thập vẫn còn được lưu trữ trong thẻ nhớ.
Extended	
Include	
Main Flow	-Người dùng nhấn vào Menu -> Menu Track List

	-Ứng dụng sẽ hiển thị các file GPX trong thẻ nhớ đã thu thập được -Người dùng nhấn vào file muốn hiển thị -Hiển thị thông tin track đó lên màn hình
Tên UseCase	GPS Trace
Actor	User
Mục đích	Thu thập dữ liệu giao thông.
Pre-condition	Smart phone có kết nối wifi hoặc 3G và đã kích hoạt chức năng GPS
Extended	
Include	
Main Flow	-Người dùng nhấn vào Menu -> Menu Tracking -Người dùng nhấn button“Start Tracking” để kích hoạt chức năng thu thập dữ liệu GPS và nhấn button “Stop Tracking” để kết thúc.

Tên UseCase	Track
Actor	User
Mục đích	Theo dõi hành trình của thiết bị
Pre-condition	Login
Extended	
Include	Login
Main Flow	-Hiển thị bản đồ -Hiển thị hành trình của thiết bị: vẽ overlay lên bản đồ theo sự di chuyển của thiết bị.

Tên UseCase	Show Traffic
Actor	User
Mục đích	Hiển thị tình trạng giao thông
Pre-condition	
Extended	
Include	
Main Flow	-Hiển thị bản đồ -Hiển thị tình trạng giao thông: Vẽ các lớp Overlay lên các tuyến đường với màu sắc khác nhau, thay đổi theo vận tốc.

3.2 Phân tích yêu cầu chi tiết module thu thập dữ liệu

3.2.1 Giới thiệu tổng quát

Modulethu thập dữ liệu giao thông dựa vào theo dõi tín hiệu GPS của Smartphone.Trong quá trình thập dữ liệu ứng dụng sẽ gửi kết quả lên server và sau khi kết thúc sẽ lưu lại file trên bộ nhớ Smartphone.

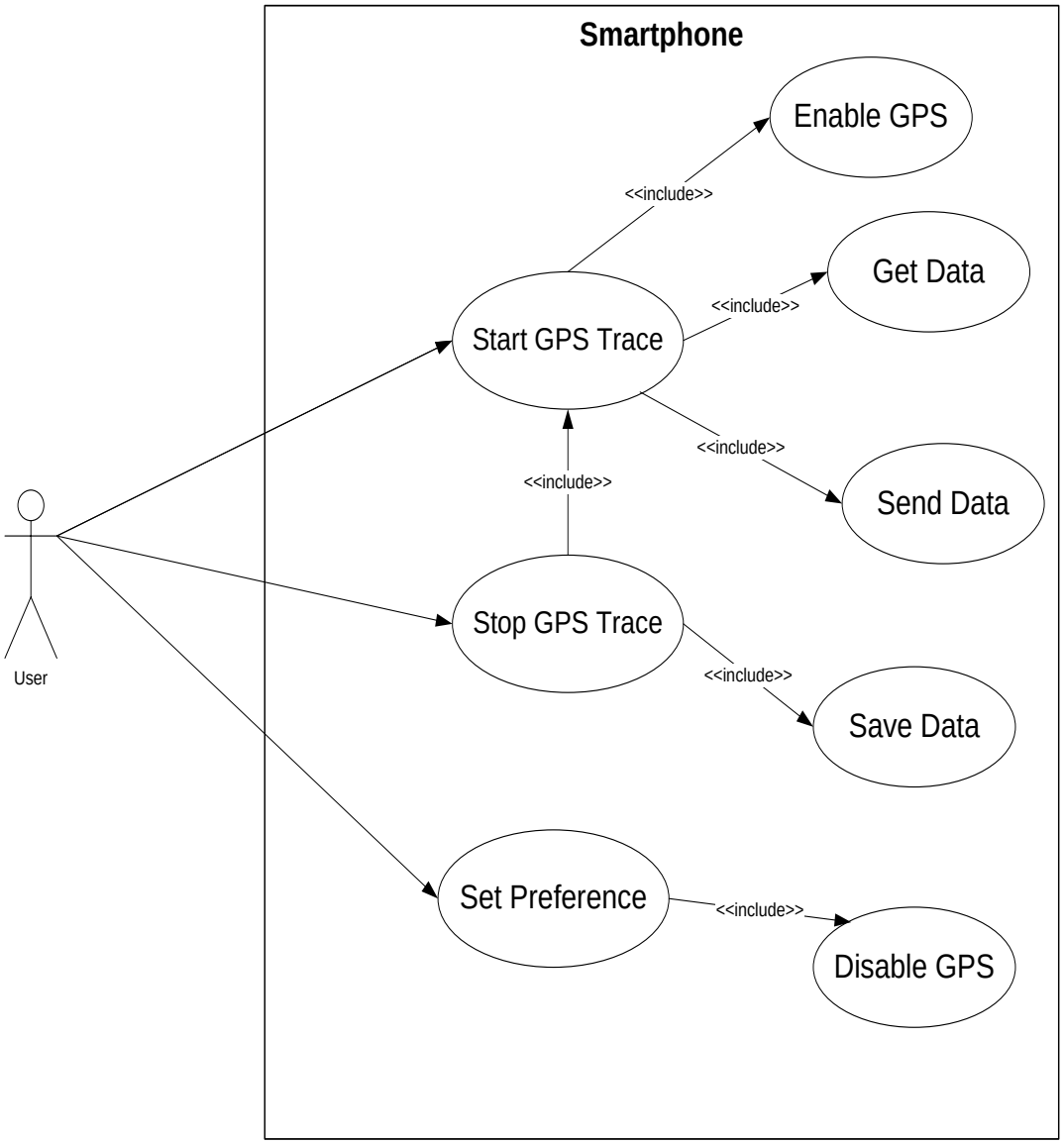
3.2.2 Các chức năng chính

- Thu thập dữ liệu giao thông
  - Xác định tọa độ của người dùng.
  - Xác định thời gian,tốc độ của người dụng tại các điểm đi qua.
- Gửi dữ liệu lên server.
- Ghi dữ liệu ra file và lưu vào thẻ nhớ.

3.2.3 Ràng buộc:

- Bộ nhớ của smart phone nhỏ
- Ngôn ngữ lập trình: Java
- IDE: Eclipse
- Hệ điều hành của smart phone: android 2.3 trở về sau

3.2.4 Mô hình usecase



Đặc tả actor

- Yêu cầu thực hiện theo dõi tín hiệu GPS.
- Yêu cầu dừng theo dõi tín hiệu GPS.
- Yêu cầu tùy chỉnh ứng dụng

Mô tả usecase

Tên UseCase	Start Trace
Actor	User

Mục đích	Thu thập dữ liệu giao thông.
Pre-condition	Smart phone có kết nối wifi hoặc 3G và đã kích hoạt chức năng GPS
Extended	
Include	
Main Flow	-Người dùng nhấn vào Menu -> Menu Tracking -Người dùng nhấn button“Start Tracking” -Form Tracking tự ẩn, chuyển về màn hình chính của ứng dụng.

Tên UseCase	Stop Trace
Actor	User
Mục đích	Ngừng thu thập dữ liệu giao thông.
Pre-condition	Ứng dụng đang thu thập dữ liệu giao thông
Extended	
Include	
Main Flow	-Người dùng nhấn vào Menu -> Menu Tracking -Người dùng nhấn button“Stop Tracking” -Form Tracking tự ẩn, chuyển về màn hình chính của ứng dụng.

Tên usecase	Enable GPS
Actor	User
Mục đích	Kích hoạt chức năng update tín hiệu GPS
Pre-condition	Start GPS trace
Extended	
Include	

Main flow	Sau khi người dùng kích hoạt chức năng sử dụng tín hiệu GPS,hệ thống kích hoạt chức năng cập nhật tín hiệu GPS theo sự thay đổi vị trí và thời gian.
Tên usecase	Get Data
Actor	User
Mục đích	Lấy thông tin tín hiệu GPS gồm kinh độ,vĩ độ,thời gian và tốc độ tạo thời điểm đó.
Pre-condition	Start GPS trace
Extended	
Include	
Main flow	Khi vị trí người dùng thay đổi,hệ thống sẽ nhận được tín hiệu về vị trí hiện tại và hệ thống sẽ tính toán thời gian và tốc độ tại điểm đó.
Tên usecase	SendData
Actor	User
Mục đích	Gửi thông tin thu thập được lên server
Pre-condition	Start GPS trace
Extended	
Include	
Main flow	Sau khi người dùng kích hoạt chức năng theo dõi tín hiệu GPS,hệ thống sẽ tự động thu thập dữ liệu và ửi dữ liệu lên server sau khoảng thời gian nhất định.
Tên usecase	Disable GPS
Actor	User

Mục đích	Tắt update tín hiệu GPS
Pre-condition	Stop GPS trace
Extended	
Include	
Main flow	Sau khi người dùng tắt chức năng thu thập dữ liệu,chức năng update tín hiệu GPS cũng tự động đước tắt.

Tên usecase	Save Data
Actor	User
Mục đích	Lưu dữ liệu đoạn đường đã đi của user
Pre-condition	Stop GPS trace
Extended	
Include	
Main flow	Sau khi người dùng tắt chức năng thu thập dữ liệu,hệ thống sẽ ghi các thông tin đã thu thập vào file gpx và lưu vào bộ nhớ của smart phone.

Tên UseCase	Set Preference
Actor	User
Mục đích	Tùy chỉnh ứng dụng.
Pre-condition	
Extended	
Include	
Main Flow	-Người dùng nhấn vào Menu -> Menu Settings -Nếu nhấn vào “GPS Setting”, ứng dụng chuyển sang màn hình tùy chỉnh GPS. -Nếu nhấn vào “GPS distance logging interval”,

sẽ hiển thị một form để nhập khoảng cách giữa các lần cập nhật. Nhấn OK để lưu tùy chỉnh. Nhấn Cancel để quay trở lại Menu Settings.

-Nếu nhấn vào "GPS time logging interval", sẽ hiển thị một form để nhập thời gian giữa các lần cập nhật. Nhấn OK để lưu tùy chỉnh. Nhấn Cancel để quay trở lại Menu Settings.

-Nếu nhấn vào "Sending data to server interval", sẽ hiển thị một form để nhập thời gian giữa các lần gửi dữ liệu. Nhấn OK để lưu tùy chỉnh. Nhấn Cancel để quay trở lại Menu Settings.

-Nếu nhấn vào "External Storage Directory", sẽ hiển thị một form để nhập nơi lưu trữ các file thu thập được. Nhấn OK để lưu tùy chỉnh. Nhấn Cancel để quay trở lại Menu Settings.

3.3 Phân tích yêu cầu chi tiết module hiển thị dữ liệu qua web browser

3.3.1 Giới thiệu tổng quát

Giao diện website trực quan mô tả module thu thập dữ liệu thông qua smartphone.

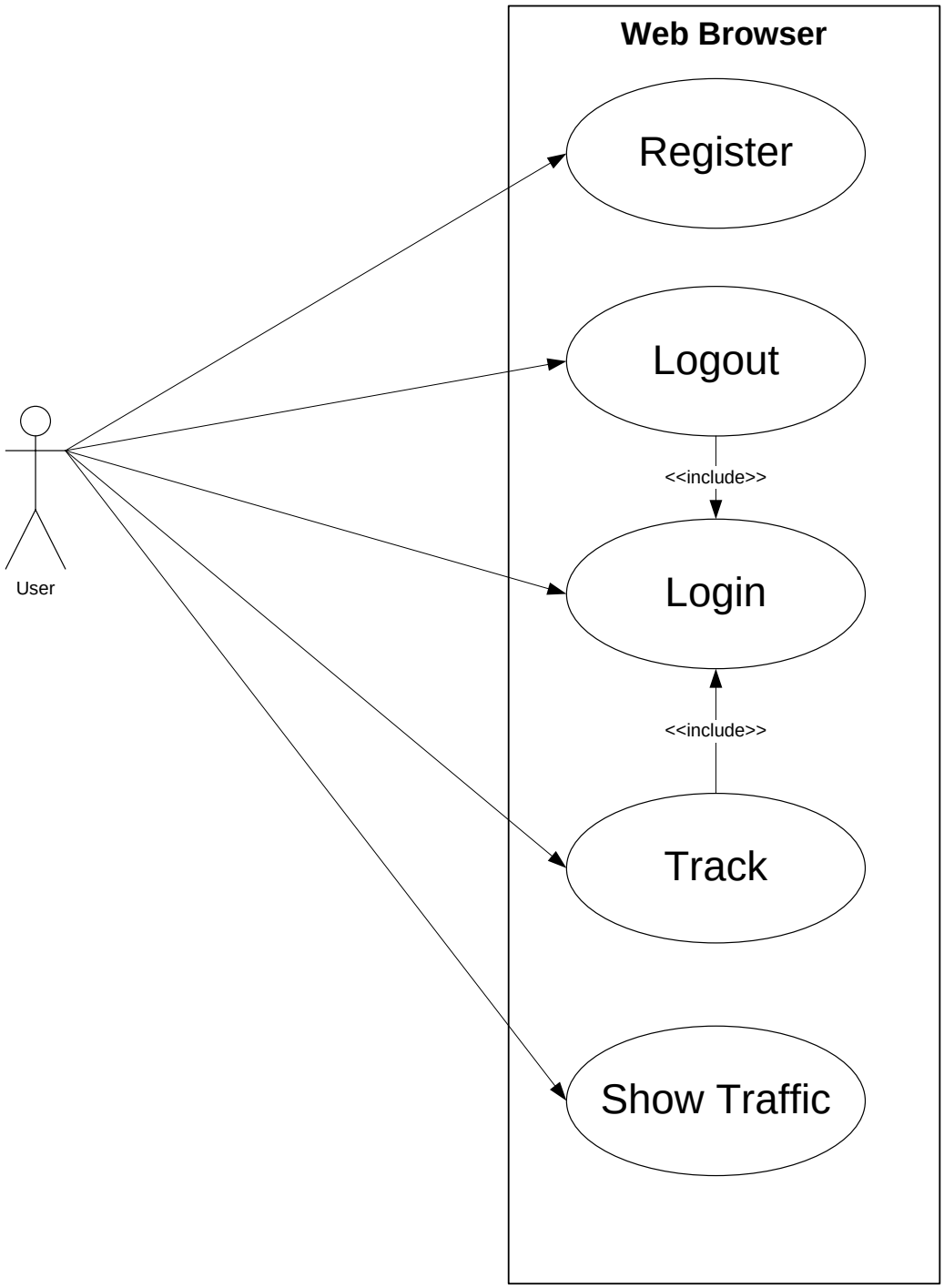
3.3.2 Các chức năng chính

- Đăng ký tài khoản
- Đăng nhập
- Theo dõi sự di chuyển của thiết bị
- Hiển thị tình trạng giao thông

3.3.3 Ràng buộc

- Ngôn ngữ lập trình: Java Servlet, Java Servlet, HTML, CSS.
- IDE: Eclipse
- Server: Tomcat
- Browser: Internet Explorer, Firefox, Chrome.

3.3.4 Mô hình usecase



Đặc tả actor

- Yêu cầu đăng ký tài khoản
- Yêu cầu đăng nhập
- Yêu cầu đăng xuất
- Yêu cầu theo dõi sự di chuyển của thiết bị

- Yêu cầu hiển thị tình trạng giao thông

Mô tả usecase

Tên UseCase	Register
Actor	User
Mục đích	Đăng ký tài khoản
Pre-condition	
Extended	
Include	
Main Flow	-Người dùng điền đầy đủ các thông tin bắt buộc. -Người dùng nhấn nút "Register" -Nếu thành công, chuyển sang trang "Đăng nhập". -Nếu không thành công, người dùng sẽ nhận được thông báo lỗi.

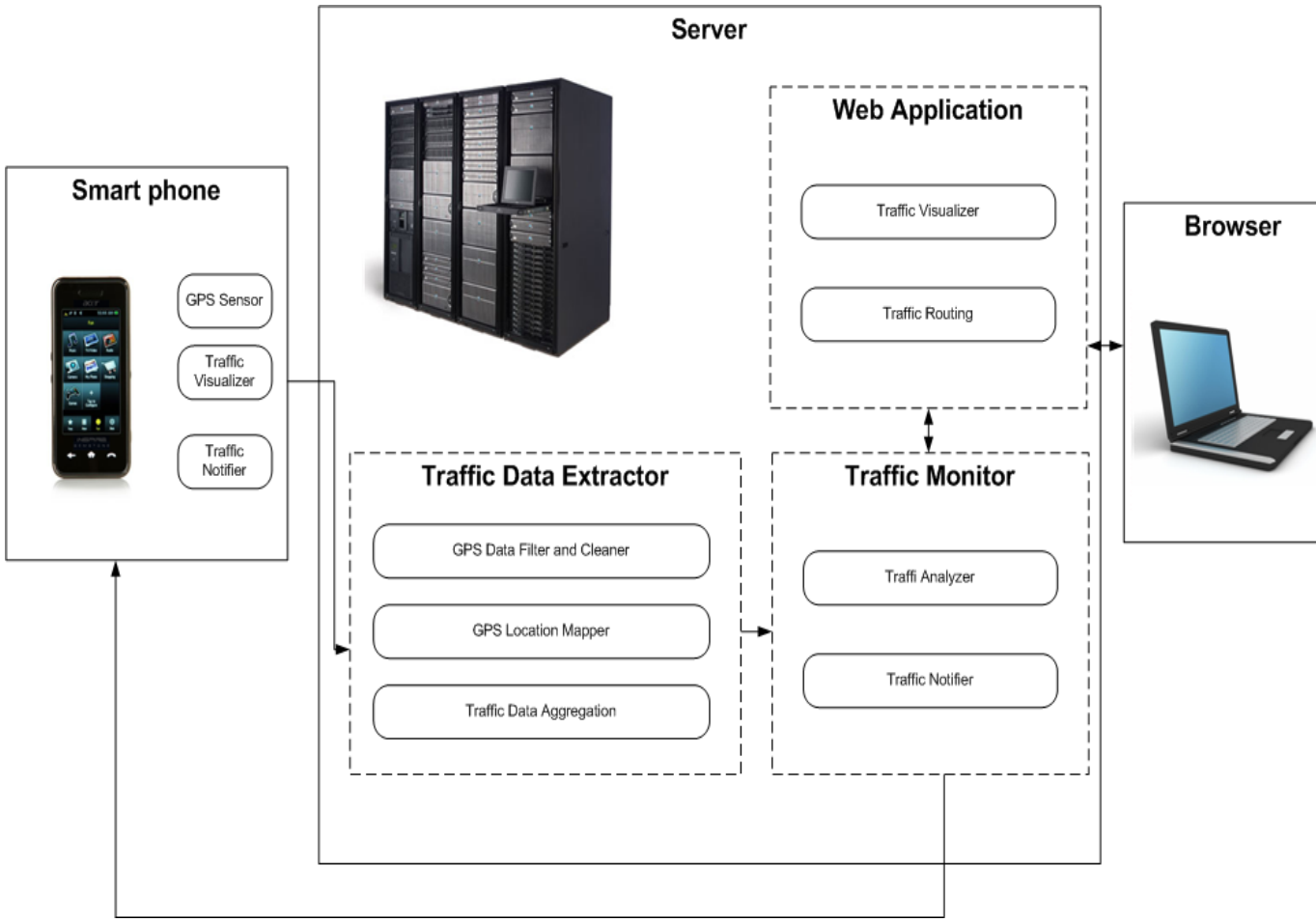
Tên UseCase	Login
Actor	User
Mục đích	Đăng nhập vào hệ thống
Pre-condition	
Extended	
Include	
Main Flow	-Người dùng điền tên tài khoản và mật khẩu. -Người dùng nhấn nút "Login" -Nếu thành công, sẽ chuyển sang trang Track -Nếu không thành công, sẽ nhận thông báo "Invalid Username or Password".

Tên UseCase	Track
Actor	User
Mục đích	Theo dõi hành trình của thiết bị
Pre-condition	Login
Extended	
Include	Login
Main Flow	-Hiển thị bản đồ -Hiển thị hành trình của thiết bị: vẽ overlay lên bản đồ theo sự di chuyển của thiết bị.

Tên UseCase	Show Traffic
Actor	User
Mục đích	Hiển thị tình trạng giao thông
Pre-condition	
Extended	
Include	
Main Flow	-Hiển thị bản đồ -Hiển thị tình trạng giao thông: Vẽ các lớp Overlay lên các tuyến đường với màu sắc khác nhau, thay đổi theo vận tốc.

CHƯƠNG 4: THIẾT KẾ HỆ THỐNG

4.1 Thiết kế cho toàn hệ thống



4.1.1 Smart phone

- GPS Sensor:
  GPS Sensor thực hiện nhiệm vụ gửi yêu cầu và nhân tín hiệu GPS từ vệ tinh.
- Traffic Visualizer
  Traffic Visualizer nhận tín hiệu GPS từ vệ tinh và từ đó phân tích thành dữ liệu bao gồm:
  - Vị trí hiện tại của người dùng.
  - Thời gian hiện tại.
  - Tốc độ di chuyển của người dùng tại vị trí đó.
  - Độ cao tại vị trí người dùng đang di chuyển
  - Độ chính xác của dữ liệu.

  Các thông tin này sẽ được gửi lên server và lưu lại trong thẻ nhớ của smart phone dưới định dạng file GPS

- Traffic Notifier
  Traific Notifier thực hiện chức năng
  - Gửi yêu cầu tìm đường đi hoặc yêu cầu xem hiện trạng giao thông lên server.
  - Nhận response từ server và hiển thị lên đường đi hoặc hiện trạng giao thông cho người dùng.

4.1.2 Server

- Traffic Data Extractor
  - GPS Data Filter and Cleaner
    Dữ liệu GPS có thể bị nhiễu(tốc độ không chính xác,vị trí sai lệch…) và thiếu thông tin nên module này được sử dụng để lọc nhiễu và làm sạch dữ liệu.
  - GPS Location Mapper
    Dữ liệu gửi lên chỉ cung cấp thông tin cho chúng ta về kinh độ vĩ độ,kinh độ. Việc sử dụng module này sẽ giúp chúng ta chuyển dữ liệu GPS thành dữ liệu giao thông tức là có thể xác định chính xác vị trí của người dùng ở đường nào .
  - Traffic Data Aggregation
    Thực hiện lưu dữ liệu vào database của hệ thống.
- Traffic Monitor
  - Traffic Analyzer
    Dựa vào dữ liệu được cung cấp từ các smart phone, module này sẽ thực hiện các chức năng tính toán, dự báo tình trạng giao thông hay tìm đường đi cho user.
  - Traffic Notifier
    Gửi thông tin đã phân tích được ở module Traffic Analyzer cho smart phone hoặc Web Application
- Web Application
  Cho phép người dùng có thể:
  - Đăng kí tài khoảng và đăng nhập tài khoảng.
  - Theo dõi một smart phone đang di chuyển trên đường.
  - Theo dõi cùng lúc nhiều smart phone đang di chuyển trên đường
  - Hiển thị tình trạng giao thông tại thời điểm người dùng yêu cầu

4.1.3 Browser

Hiển thị các thông tin của Web Application.

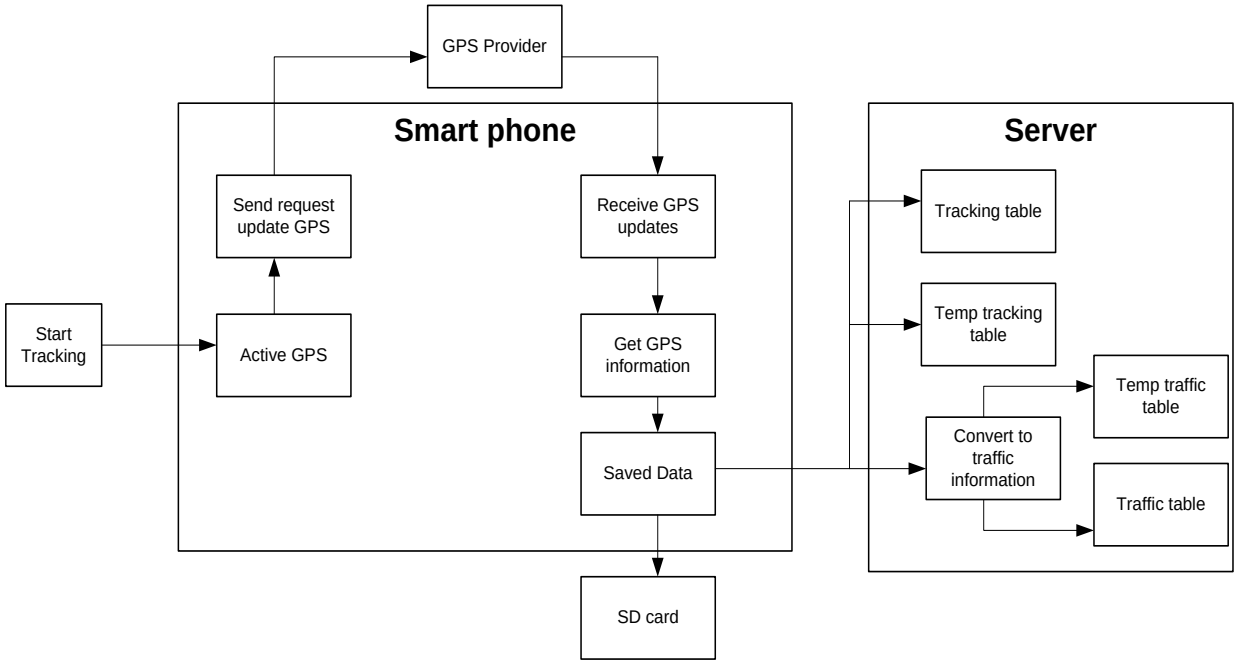
Trong các module trên nhóm chúng tôi chịu trách nhiệm về các module:

- Thu thập dữ liệu GPS(GPS Sensor, Traffic Visualizer của smart phone)
- Lưu dữ liệu vào database hệ thống(Traffic Data Aggregation).

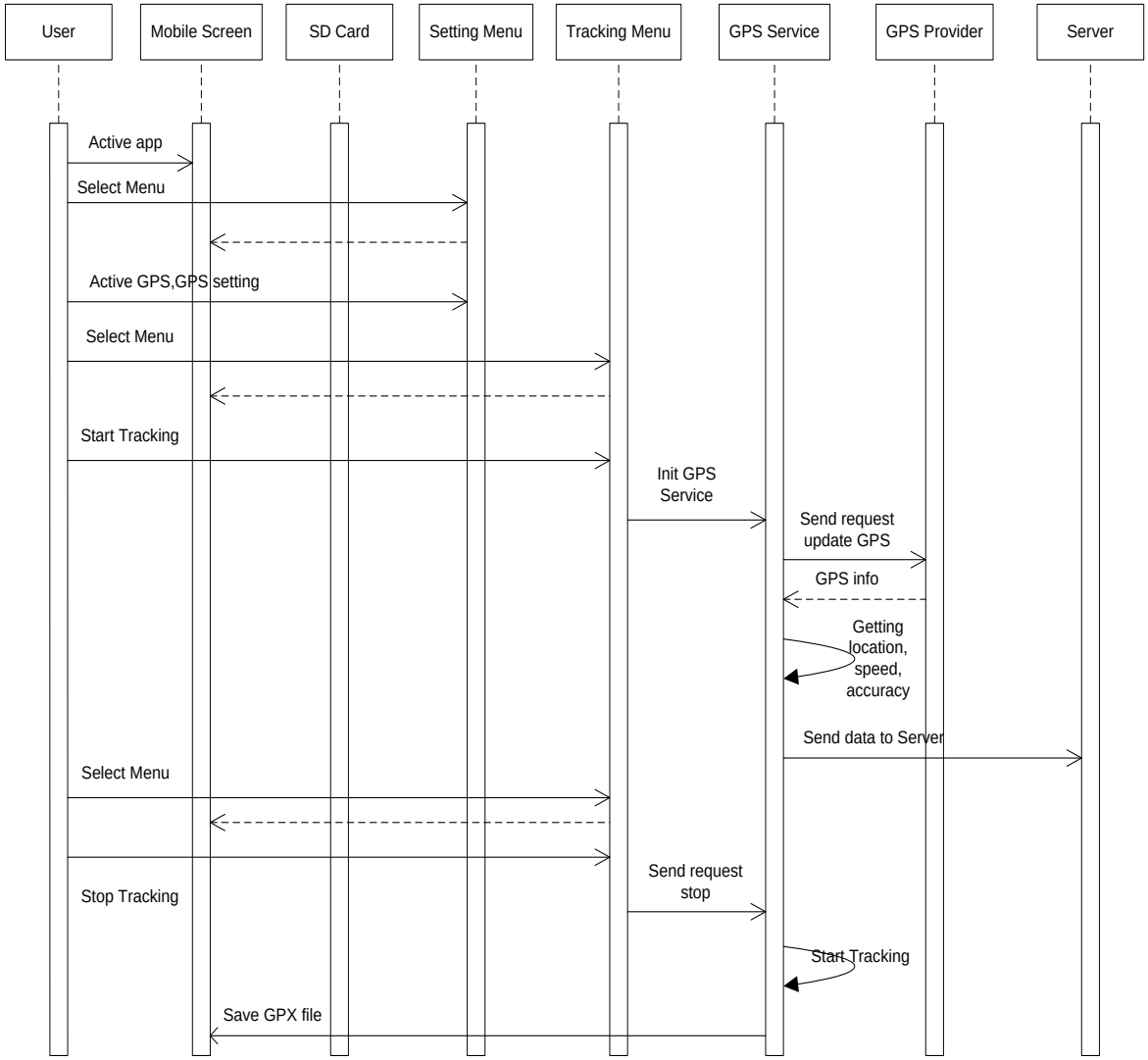
- Hiện thực website theo dõi tình trạng giao thông (Web Application).

4.2 Thiết kế chi tiết chức năng thu thập GPS

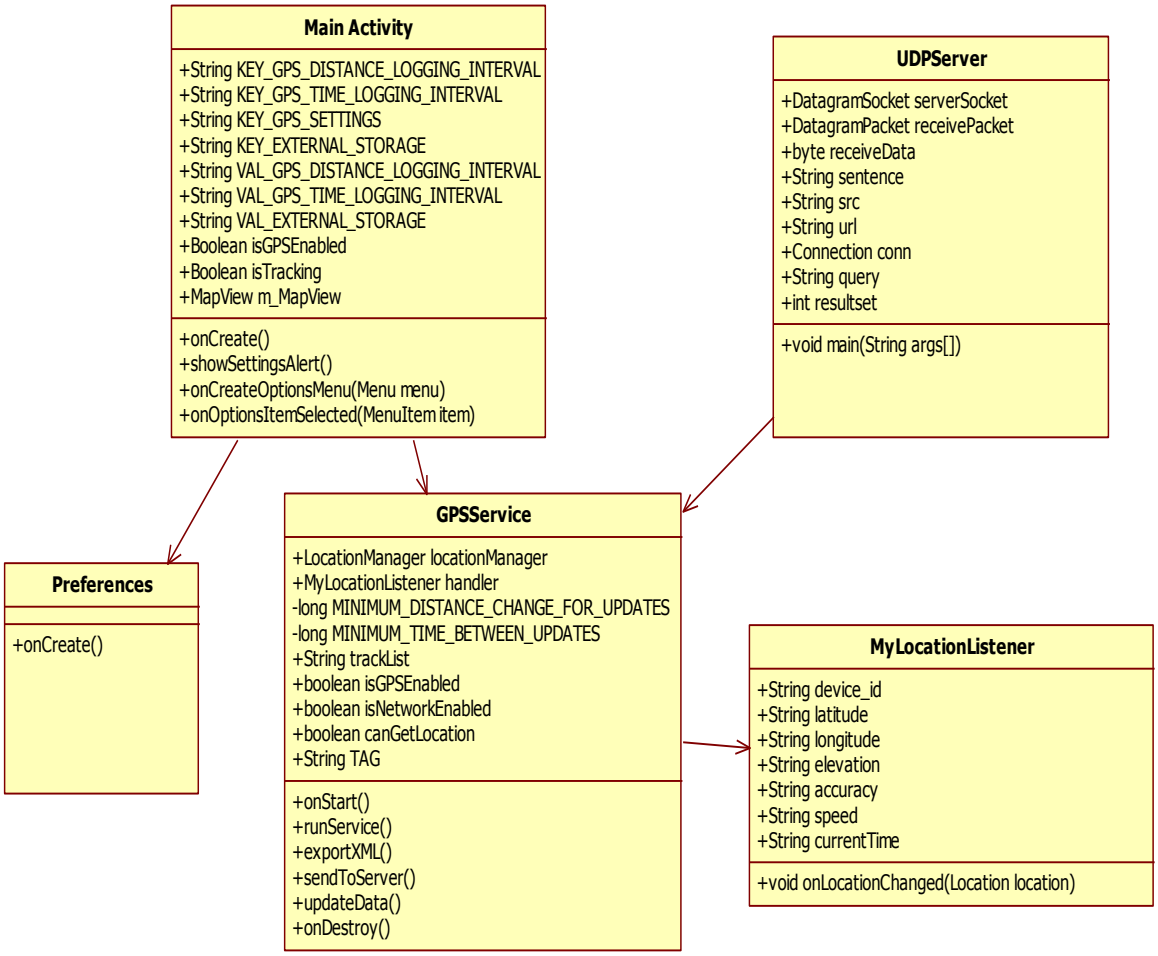
4.2.1 Object Model



4.2.2 Sequence diagram



4.2.3 Class Diagram



Class	GPSService
Thuộc tính	-locationManager: quản lý vị trí của người dùng. -handler: listener nhận tín thông báo thay đổi vị trí. -MINIMUM_DISTANCE_CHANGE_FOR_UPDATE:khoảng cách nhỏ nhất để cập nhật tín hiệu GPS. -MINIMUM_TIME_BETWEEN_FOR_UPDATE: thời gian ngắn nhất để cập nhật tín hiệu GPS. - trackList: list lưu lại các điểm người dùng đã đi qua. - isGPSEnabled: kiểm tra xem tín hiệu GPS đã được bật chưa. - isNetworkEnabled: kiểm tra xem có mạng wifi hoặc 3G không. - TAG: để thông báo cho người dùng biết GPSService đang chạy.

Phương Thức <<onStart()>>	- Đầu vào: 1 biến kiểu intent và 1 biến starteId kiểu int. - Kiểu trả về: void - Precondition: người dùng khởi động chương trình. - Postcondition: hàm sẽ gọi runService() và UpdateData().
Phương Thức <<runService( )>>	- Đầu vào: N/A - Kiểu trả về: void - Precondition: onStart(). - Postcondition: hàm sẽ chọn ra bestProvider, tức là xác định xem lấy dữ liệu giao thông của người dùng theo GPS hay 3G/Wifi tốt hơn, sau đó gọi hàm cập nhật vị trí người dùng.
Phương Thức <<exportXML()>>	- Đầu vào: N/A - Kiểu trả về: void - Precondition: người dùng kết thúc thu thập dữ liệu bằng cách bấm vào button Stop - Postcondition: hàm sẽ đọc dữ liệu từ trackList được cập nhật trong class MyLocationListener(sẽ được mộ tả sau) để ghi vào file GPX và lưu vào thẻ nhớ của smartphone.
Phương Thức <<sendToServer()>>	- Đầu vào: N/A - Kiểu trả về: void - Precondition: người dùng chọn button start Tracking. - Postcondition: hàm sẽ mở kết nối đến server thông qua giao thức UDP và gửi list điểm lưu trong tempList lên Server.
Phương Thức <<updateData()>>	- Đầu vào: N/A - Kiểu trả về: void - Precondition: sendToServer(). - Postcondition: gửi từng gói dữ liệu lên server trong khoảng thời gian xác định.
Phương Thức <<onDestroy()>>	- Đầu vào: N/A - Kiểu trả về: void - Precondition: người dùng chọn button Stop Tracking. - Postcondition: gọi hàm exportXML() ghi file GPX vào thẻ nhớ điện thoại, đồng thời kết thúc quá trình thu thập dữ liệu.

Class	MyLocationListener
Thuộc tính	- device_id: số imei của điện thoại. - latitude: vĩ độ của người dùng. -longitude: kinh độ của người dùng.

	- elevation: độ cao của dữ liệu thu thập được. - accuracy: độ chính xác của dữ liệu thu thập được. - speed: tốc độ của người dùng. - currentTime: thời gian thu thập dữ liệu.
Phương Thức << onLocationChanged(Location location)>>	-Đầu vào: một biến kiểu Location -Kiểu trả về: void -Precondition: khi vị trí người dùng thay đổi - Postcondition: lưu thông tin di chuyển của người dùng vào trackList và tempList để dùng cho những việc khác.

Class	Preferences
Thuộc tính	N/A
Phương Thức <<onCreate()>>	-Đầu vào: N/A -Kiểu trả về: void -Precondition: người dùng nhấn vào button Settings. -Postcondition: bảng settings hiện lên để người dùng tùy chỉnh các thông số GPS_DISTANCE_LOGGING_INTERVAL, GPS_TIME_LOGGING_INTERVAL, EXTERNAL_STORAGE.

Class	MainActivity
Thuộc tính	- KEY_GPS_DISTANCE_LOGGING_INTERVAL: tùy chỉnh khoảng cách giữa các lần cập nhật. - KEY_GPS_TIME_LOGGING_INTERVAL: tùy chỉnh thời gian giữa các lần cập nhật. - KEY_GPS_SETTINGS: thiết lập các chế độ setting của hệ thống thu thập dữ liệu GPS. - String KEY_EXTERNAL_STORAGE: thiết lập bộ nhớ ngoài chứa dữ liệu thu thập được. - VAL_GPS_DISTANCE_LOGGING_INTERVAL: giá trị mặc định của mỗi lần thu thập tín hiệu GPS là 5m. - VAL_GPS_TIME_LOGGING_INTERVAL: giá trị mặc định của mỗi lần thu thập tín hiệu GPS là 2s. - VAL_EXTERNAL_STORAGE: giá trị mặc định của nơi chứa tín hiệu GPSthu thập được /GPSTrackingFiles. - isGPSEnabled: kiểm tra xem GPS đã bật chưa. - isTracking: kiểm tra xem người dùng đã bật chế độ cho phép thu thập dữ liệu chưa. - m_Mapview: biến thuộc kiểu Mapview để hiển thị bản đồ.

Phương Thức <<onCreate()>>	-Đầu vào: biến savedInstanceState kiểu Bundle. -Kiểu trả về: void -Precondition: khi người dùng khởi động chương trình. - Postcondition:khởi tạo layout và các biến cần thiết cho ứng dụng.
Phương Thức <<showSettingsAlert()>>	-Đầu vào: N/A -Kiểu trả về: void -Precondition: khi người dùng khởi động chương trình mà chưa bật GPS, hoặc bấm vào menu Settings, hoặc bấm vào button Cancel một tác vụ nào đó của ứng dụng. - Postcondition: trả về các dialog thông báo cho người dùng được biết.
Phương Thức <<onCreateOptionsMenu(Menu menu)>>	-Đầu vào: một biến kiểu Menu. -Kiểu trả về: boolean -Precondition: N/A - Postcondition: tạo menu Option cho người dùng tùy chỉnh.
Phương Thức <<onOptionsItemSelected(MenuItem item)>>	-Đầu vào: một biến kiểu MenuItem. -Kiểu trả về: boolean -Precondition: N/A - Postcondition: hiển thị cho người dùng biết mình đã chọn những tác vụ nào của ứng dụng, bao gồm chọn Settings, buildRoute, forecast, turnPower, showCurrent, trackList.

Class	UDPServer
Thuộc tính	- serverSocket: biến thuộc kiểu DatagramSocket để mở kết nối UDP đến server thông qua port 6666. - receiveData: chuỗi byte để nhận gói tin. - receivePacket: biến thuộc kiểu DatagramPacket để nhận đóng gói gói tin từ client theo chuỗi byte ở trên. - sentence: chuỗi lưu dữ liệu nhận được. - src: mảng chuỗi lưu các gói tin được tách ra từ gói tin tổng hợp để lấy từng gói dữ liệu riêng lẻ. - url: đường dẫn kết nối đến CSDL của Mysql trên server. - conn: biến tạo kết nối đến CSDL của Mysql trên server. - query: câu lệnh SQL để lưu thông tin của các gói tin nhận được vào bảng dữ liệu GPSTest. - resultset: statement để Update table mỗi lần thêm dữ liệu vào bảng GPSTest.

Phương Thức <<void main(String args[])>>	-Đầu vào: N/A -Kiểu trả về: void -Precondition: N/A -Postcondition: khởi động tác vụ nhận dữ liệu từ client ghi xuống DB trên server.

4.2.4 Nhận xét đánh giá thiết kế

Phần smartphone

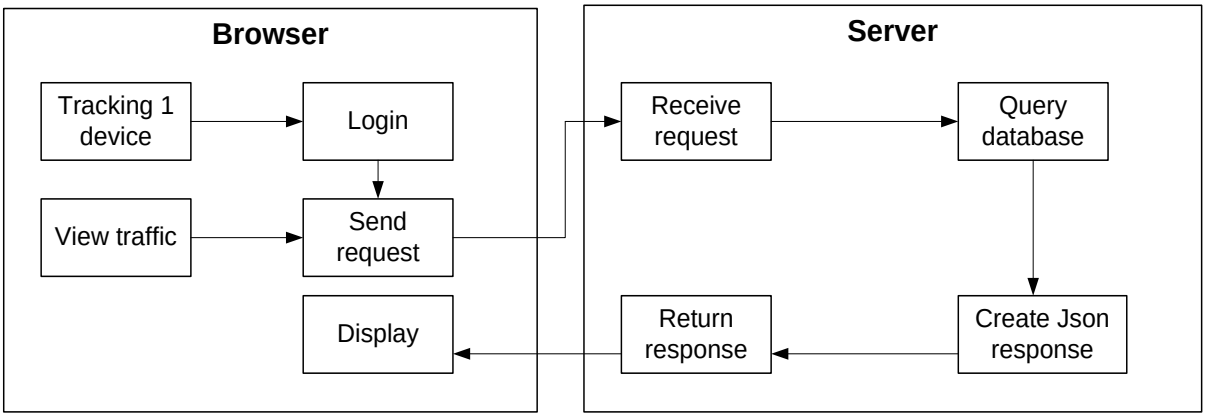
- Ưu điểm:
  - Cho phép người dùng làm việc khác khi đang sử dụng ứng dụng. Ví dụ như họ có thể gọi điện nhắn tin tra cứu thông tin mà không cần phải tắt ứng dụng.
  - Cho phép người dùng tùy chỉnh các thông số mặc định của chương trình theo ý thích như: tùy chỉnh khoảng cách, thời gian giữa các lần cập nhật vị trí.
  - Dữ liệu tracking được lưu trữ mặc định ở folder gợi nhớ tên thuận tiện cho việc tìm kiếm: /GPSTrackingFiles. Người dùng cũng có thể dễ dàng thay đổi nơi lưu trữ thông qua button Settings.
- Nhược điểm:
  - Chưa giải quyết được vấn đề hao tốn năng lượng pin khi cập nhật vị trí người dùng thông qua GPS.

Phần Server

- Ưu điểm:
  Sử dụng giao thức UDP giúp truyền dữ liệu nhanh và hiệu quả, có thể đáp ứng các yêu cầu nhỏ với lượng người dùng lớnvì không cần phải tốn chi phí tạo kết nối giữa client và server, không cần phải ghi nhớ trạng thái gửi/nhận. Ngoài việc lưu dữ liệu vào bản chính,chúng tôi thực hiện lưu dữ liệu vào bản tạm nhằm đáp ứng yêu cầu xem tình trạng giao thông tốt hơn,giảm được chi phí truy vấn cơ sở dữ liệu của hệ thống.Dữ liệu trong bảng tạm sẽ được xóa hàng ngày.
- Nhược điểm:
  Do UDP là giao thức đơn giản, truyền dữ liệu không theo định dạng nên đôi khi các gói dữ liệu có thể đến không đúng thứ tự, và không thể đảm bảo gói dữ liệu được gửi đến đích.Bên cạnh đó, việc bảo mật khi sử dụng giao thức UDP là rất thấp.Cách giải quyết vấn đề này là dùng các kĩ thuật mã hóa dữ liệu trước khi gửi.

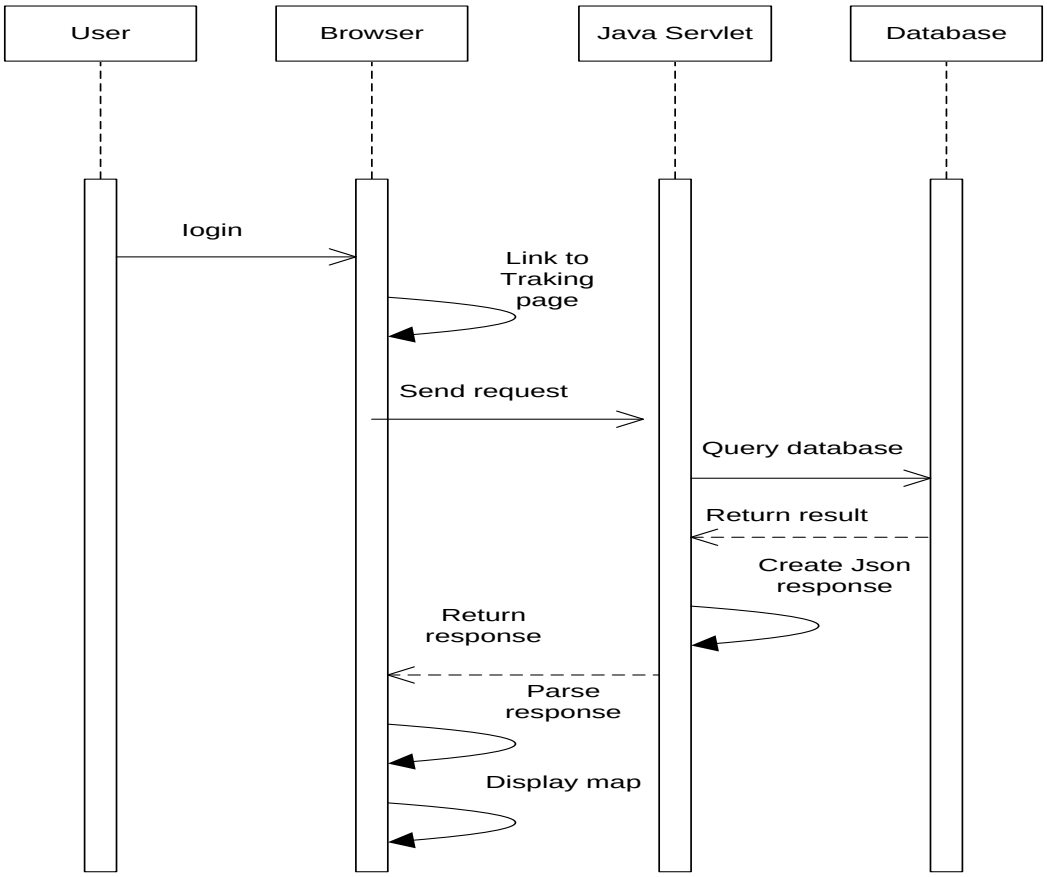
4.3 Thiết kế và hiện thực Tracking website

4.3.1 Object Model

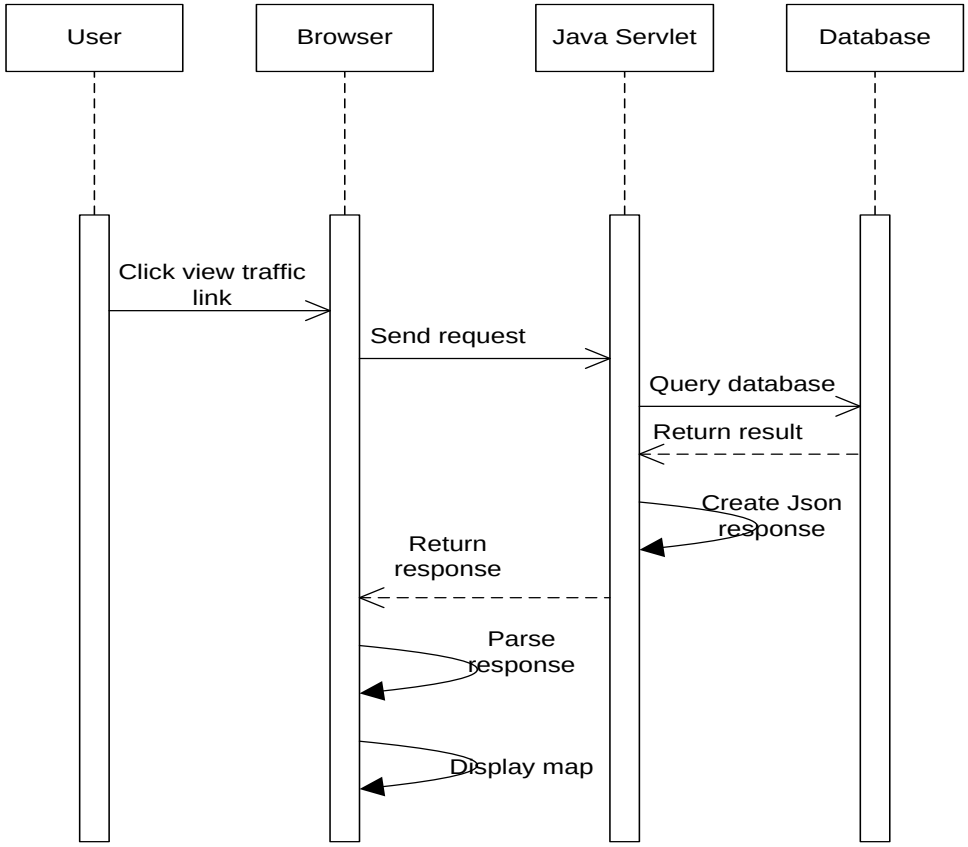


4.3.2 Sequence Diagram

Tracking

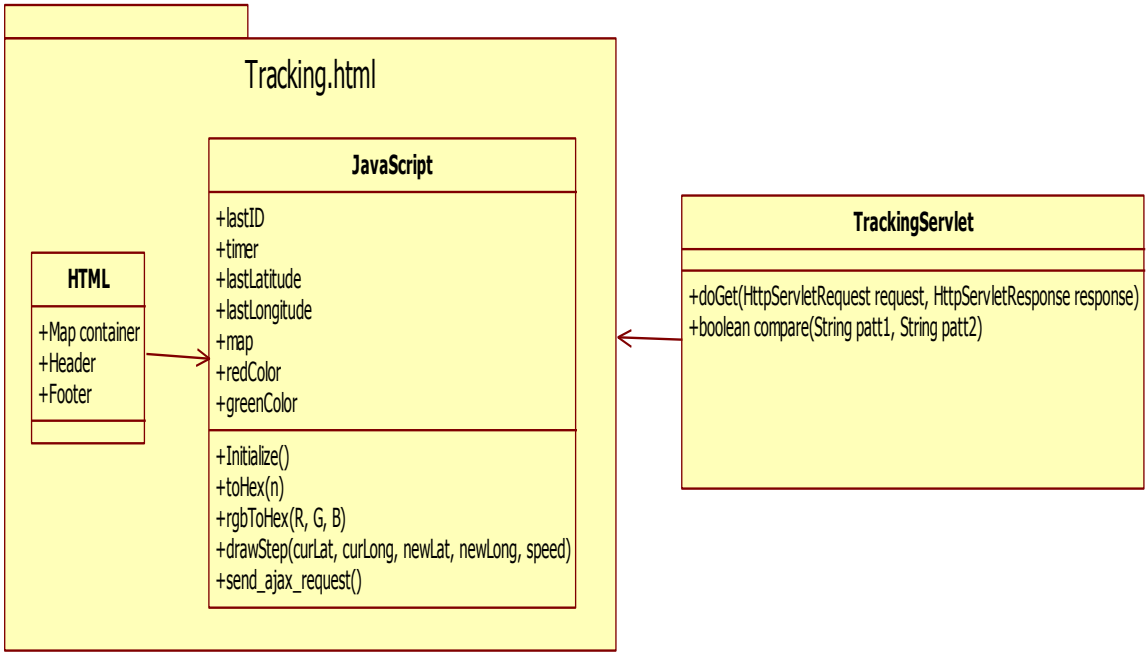


View Traffic



4.3.3 Class Diagram

Tracking



File	HTML
Map container	Khung hiển thị bản đồ
Header	Nagivation chứa link tới các trang login hoặc log out,view traffic và tracking.
Footer	Chứa các thông tin về nhóm thực hiện

File	JavaScript
Variable	-lastID: ID của dòng dữ liệu cuối cùng được truy vấn từ database hệ thống -timer:dùng để thực hiên thao tác gửi yêu cầu lên server sau một khoảng thời gian xác định -lastLatitude, lastLongitude:tọa độ cuối cùng được vẽ lên bản đồ -map :bản đồ -redColor,green Color:được xác định thông qua vận tốc để vẽ màu đường.

Method << initialize()>>	-Đầu vào: N/A -Kiểu trả về: N/A -Precondition: user load trang web tracking -Postcondition: • Load bản đồ vào khung hiển thị bản đồ • Config các thông số cho map bao gồm: mức zoom ,center cho map • Gọi hàm send_ajax_request() để gửi yêu cầu lên server
Method << toHex(n)>>	-Đầu vào: số n dưới dạng String -Kiểu trả về: số kiểu hex -Precondition:hàmrgbToHex(R,G,B) được kích hoạt -Postcondition: • Chuyển n sang kiểu số • Chuyển n sang kiểu hex và trả về
Method << rgbToHex(R,G,B)>>	-Đầu vào: chỉ số màu R,G,B tương ứng với 3 màu đỏ,xanh lá cây và xanh dương -Kiểu trả về: chuỗi kiểu hex(để xác định màu) -Precondition: hàm drawStep() được kích hoạt -Postcondition: • Chuyển các chỉ số R,G,B sang kiểu hex • Nối các chuỗi nhận được sau khi chuyển các chỉ số R,G,B sang kiểu hex và trả về
Method << drawStep(curLat, curLong, newLat, newLong,	-Đầu vào: • curLat,curLong:tọa độ điểm bắt đầu • newLat,newLong:tọa độ điểm kết thúc

speed)>>	• speed:tốc độ
	
	-Kiểu trả về: N/A
	
	-Precondition:hàm send_ajax_request() được kích hoạt và kết quả trả về từ server khác []
	
	-Postcondition:
	• Tính toán greenColor và redColor dựa vào vận tốc theo công thức:
	
	`redColor = ((1.0 - parseFloat(speed)*3.6/40.0)*255);`
	`greenColor = ((parseFloat(speed)*3.6/40.0)*255);`
	
	40:tốc độ tối đa mà thiết bị có thể di chuyển được trên đường,được tính theo km/h
	3.6:vì speed đang tính theo m/s nên ta nhân với 3.6 để chuyển sang km/h
	255:số bit dùng để vẽ màu là 8,nên ta chia cho 255 để xác định tỉ lệ.
	
	• Tạo step từ 2 tọa độ
	• Tạo Polyline:xác định các thông số của Polyline
	o Step:nhận vào step được tạo ở trên
	o strokeColor:màu của polyline được xác định bởi hàmrgbToHex() với các thông số được truyền vào là greenColor và redColor
	o strokeWeight:độ rộng của line
	• Vẽ polyline lên bản đồ

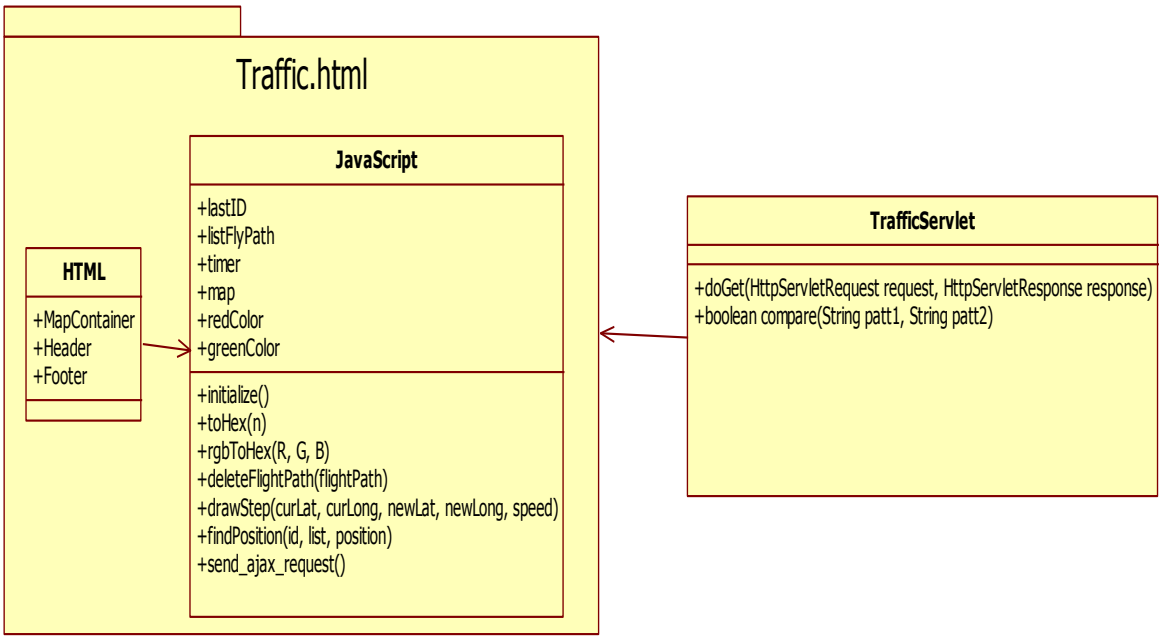
Method <<send_ajax_request()>>	-Đầu vào: N/A -Kiểu trả về: N/A -Precondition:được kích hoạt khi hàm initialize() được kích hoạt -Postcondition: • Load bản đồ mới • Gán các biến lastID, lastLatitude,

	lastLongitude về giá trị “” • Tạo timer để tự động gửi yêu cầu lên server sau một khoảng thời gian xác định • Sử dụng kỹ thuật ajax để gửi yêu cầu lên server • Nhận response từ server Trong trường hợp response trả về khác [] o Parse response ra thành list o Duyệt theo list vẽ ra các step, mỗi step gồm 2 phần tử của list o Cập nhật các biến lasted, lastLatitude, lastLongitude o Xác định lại center cho map.

File	TrackingServlet
Method << doGet(HttpServletRequest request, HttpServletResponse response)>>	-Đầu vào:request từ client -Kiểu trả về: JSONArray -Precondition:client yêu cầu tracking -Postcondition: • Lấy thông số lastID được gửi lên từ client • Tạo kết nối đến database • Lấy username từ session và từ đó lấy ra số imei của user đó thông qua truy vấn database. • Tạo câu query o Nếu lastID = “”,nôi dung câu query là truy vấn các tuple có imei có giá trị bằng và imei và imei tìm được ở trên đồng thời currentTime của tuple phải trùng với thời gian của hệ thống đến mức phút o Nếu lastID khác “” ,nội

	dung câu query là truy vấn các tuple có imei có giá trị bằng và imei tìm được ở trên đồng thời id lớn hơn lastID • Thực thi câu query • Nhận kết quả query và tạo JSONArray.Đối với các kết quả trả về sau khi thực thi query,tiến hành so sánh với thời gian của hệ thống bằng hàm compare() và đối với kết quả trả về True mới được lưu vài JSONArray.Các thông tin của một JSONObject có gồm longitude, latitude, speed, id. • Trả response về cho phía client
Method << compare(String patt1, String patt2>>	-Đầu vào:hai string có định dạng thời gian chuẩn -Kiểu trả về: boolean -Precondition: -Postcondition:so sánh hai chuỗi thời gian và trả về kết quả true nếu patt1 là thời gian trước patt2,và ngược lại.

View Traffic



File	HTML
Map container	Khung hiển thị bản đồ
Header	Nagivation chứa link tới các trang login hoặc log out,view traffic và tracking.
Footer	Chứa các thông tin về nhóm thực hiện

File	JavaScript
Variable	-lastID: ID của dòng dữ liệu cuối cùng được truy vấn từ database hệ thống -timer:dùng để thực hiên thao tác gửi yêu cầu lên server sau một khoảng thời gian xác định - listFlyPath:list các Polyline được vẽ lên bản đồ -map :bản đồ -redColor,green Color:được xác định thông qua vận tốc để vẽ màu đường.
Method	-Đầu vào:

<< initialize()>>	-Kiểu trả về: -Precondition: user load trang web tracking -Postcondition: • Load bản đồ vào khung hiển thị bản đồ • Config các thông số cho map bao gồm: mức zoom ,center cho map • Gọi hàm send_ajax_request() để gửi yêu cầu lên server
Method << toHex(n)>>	-Đầu vào: số n dưới dạng String -Kiểu trả về: số kiểu hex -Precondition:hàmrgbToHex(R,G,B) được kích hoạt -Postcondition: • Chuyển n sang kiểu số • Chuyển n sang kiểu hex và trả về
Method << rgbToHex(R,G,B)>>	-Đầu vào: chỉ số màu R,G,B tương ứng với 3 màu đỏ,xanh lá cây và xanh dương -Kiểu trả về: chuỗi kiểu hex(để xác định màu) -Precondition: hàm drawStep() được kích hoạt -Postcondition: • Chuyển các chỉ số R,G,B sang kiểu hex • Nối các chuỗi nhận được sau khi chuyển các chỉ số R,G,B sang kiểu hex và trả về

Method << deleteFlightPath(flightPath)>>	-Đầu vào: • flightPath:là một Polyline -Kiểu trả về: -Precondition:hàmdrawStep () được kích hoạt -Postcondition: Duyệt theo listFlyPath và kiểm tra nếu phần tử nào của list có step trùng với step của flightPath thì xóa Polyline đó khỏi bản đồ và listFlyPath

Method
<< drawStep(curLat, curLong, newLat, newLong, speed)>>
	-Đầu vào:
	• curLat,curLong:tọa độ điểm bắt đầu
	• newLat,newLong:tọa độ điểm kết thúc
	• speed:tốc độ
	-Kiểu trả về:
	-Precondition:hàm send_ajax_request() được kích hoạt và kết quả trả về từ server khác []
	-Postcondition:
	• Tính toán greenColor và redColor dựa vào vận tốc theo công thức:
	`redColor = ((1.0 - parseFloat(speed)*3.6/40.0)*25`
	`greenColor = ((parseFloat(speed)*3.6/40.0)*255);`
	40:tốc độ tối đa mà thiết bị có thể di chuyển được trên đường,được tính theo km/h
	3.6:vì speed đang tính theo m/s nên ta nhân với 3.6 để chuyển sang km/h

	255:số bit dùng để vẽ màu là 8,nên ta chia cho 255 để xác định tỉ lệ.
	
	• Tạo step từ 2 tọa độ
	• Tạo Polyline:xác định các thông số của Polyline
	  o Step:nhận vào step được tạo ở trên
	  o strokeColor:màu của polyline được xác định bởi hàmrgbToHex() với các thông số được truyền vào là greenColor và redColor
	  o strokeWeight:độ rộng của line
	• Xóa các Polyline có cùng step với Polyline mới bằng hàm deleteFlightPath(flightPath) với flightPath là Polyline mới được tạo ra
	• Thêm Polyline mới được tạo ra vào listFlyPath
	• Vẽ polyline lên bản đồ

Method << findPosition(id,list,position)>>	-Đầu vào: • id:là một id có dạng String • list:list các JSONObject • position:vị trí -Kiểu trả về: boolean -Precondition:được kích hoạt khi hàmsend_ajax_request() được gọi -Postcondition: Duyệt các phần tử của list từ phần tử đầu tiên tới phần tử có vị trí position nếu có phần tử nào có nodeID bằng id thì trả về true,nêu không có sẽ trả về false.

Method <<send_ajax_request()>>	-Đầu vào: N/A -Kiểu trả về: N/A -Precondition:được kích hoạt khi hàm initialize() được kích hoạt. -Postcondition: • Load bản đồ mới • Gán biến lastID về giá trị “” • Tạo timer để tự động gửi yêu cầu lên server sau một khoảng thời gian xác định • Sử dụng kỹ thuật ajax để gửi yêu cầu lên server,thông tin gửi lên gồm có lastID. • Nhận response từ server Trong trường hợp response trả về khác [] o Parse response ra thành list o Duyệt theo list .Kiểm tra từng phần tử có nodeID trùng với các phần tử trước đó không bằng hàm findPosition(id,list,position). Trong đó: -id:là nodeID của phần tử. -list:là list các phần tử -position là vị trí của phần tử trong list. Nếu kết quả trả về false thì ta tiến hành duyệt list từ phần tử sau phần tử hiện tại.Nếu phần tử nào có cùng nodeID với phần tử hiện tại, chúng ta cộng speed của phần tử đó vào biến sum và cuối cùng lấy sum chia cho

	số phần tử để tính tốc độ trung bình.Vẽ lên bản đồ đoạn đường với hàm drawStep(). o Cập nhật các biến lasted, o Xác định lại center cho map.

File	TrafficServlet
Method << doGet(HttpServletRequest request, HttpServletResponse response)>>	-Đầu vào:request từ client -Kiểu trả về: JSONArray -Precondition:client yêu cầu tracking -Postcondition: • Lấy thông số lastID được gửi lên từ client • Tạo kết nối đến database • Tạo câu query o Nếu lastID = “”,nôi dung câu query là truy vấn các tuple có currentTime của tuple phải trùng với thời gian của hệ thống đến mức phút o Nếu lastID khác “” ,nội dung câu query là truy vấn các tuple có có id lớn hơn lastID • Thực thi câu query • Nhận kết quả query và tạo JSONArray.Đối với các kết quả trả về sau khi thực thi query,tiến hành so sánh với thời gian của hệ thống bằng hàm compare() và đối với kết quả trả về True mới được lưu vài JSONArray.Các thông tin của một JSONObject có gồm

	longitude, latitude, speed, id. • Trả response về cho phía client
Method << compare(String patt1, String patt2>>	-Đầu vào:hai string có định dạng thời gian chuẩn -Kiểu trả về: boolean -Precondition: -Postcondition:so sánh hai chuỗi thời gian và trả về kết quả true nếu patt1 là thời gian trước patt2,và ngược lại.

4.3.4 Nhận xét đánh giá thiết kế

Tracking

Kỹ thuật web

- Sử dụng kĩ thuật timer giúp cho trang web tự động gửi yêu cầu lên server .Khoảng cách giữa các lần gửi yêu cầu lên server là 5 giây, đảm bảo cho dữ liệu được cập nhật thường xuyên phản ánh đúng vị trí hiện tại của thiết bị.
- Sử dụng kỹ thuật ajax để gửi yêu cầu lên server và nhận về gói JSON chưa thông tin đường đi của thiết bị.Việc này giúp trang web không phải load lại từ đầu.
- Thông tin gửi lên server có lastID nhằm hạn chế thao tác truy vấn server giúp server hoạt động tốt hơn.

Kỹ thuật server

- Truy vấn database:
  Đối với việc truy vấn database để trả về kết quả tracking của thiết bị chúng ta có thể sử dụng cách so sánh số imei của thiết bị và thời gian được lưu trong các tuple trùng với thời gian hiện tại của hệ thống nhưng cách này gây tốn nhiều chi phí cho việc truy vấn database và dữ liệu có thể bị thiếu.Với việc gửi lên lastID tức là id của tuple được trWuy vấn cuối cùng trước đó thì ta chỉ viêc truy vấn các tuple có id lớn hơn lastID và có cùng imei với imei của thiết bị.
- Response trả về là một JSONArray giúp bên web dễ dàng xử lý thông tin.

View Traffic

Kỹ thuật web

- Sử dụng kĩ thuật timer giúp cho trang web tự động gửi yêu cầu lên server .Khoảng cách giữa các lần gửi yêu cầu lên server là 2 phút, đảm bảo cho dữ liệu được cập nhật thường xuyên phản ánh đúng hiện trạng của hệ thống giao thông.Khoảng cách trên của chúng tôi là vừa đủ tránh tình trạng server bị truy xuất liên tục.
- Sử dụng kỹ thuật ajax để gửi yêu cầu lên server và nhận về gói JSON chưa thông tin đường đi của thiết bị.Việc này giúp trang web không phải load lại từ đầu.
- Thông tin gửi lên server có lastID nhằm hạn chế thao tác truy vấn server giúp server hoạt động tốt hơn.
- Kỹ thuật vẽ: vì tình trạng giao thông thay đổi nên ta phải cập nhật lại màu sắc của các đường đã vẽ.Việc vẽ chồng lên sẽ không phản ánh đúng màu sắc của đường làm hiển thị không đúng tốc độ lưu thông trên đoạn đường đó.Vì vậy chúng tôi sử dụng listFlyPath để lưu các đường đã vẽ nhằm xóa các đường đó khi tốc độ trên đoạn đường đó thay đổi.Sau khi xóa đường cũ thực hiện vẽ đường mới với màu sắc mới.
- Việc tính toán tốc độ trên từng đoạn đường sẽ dựa vào kết quả trả về từ server,tốc độ vẽ lên sẽ là tốc độ trung bình của tất cả các thiết bị lưu thông trên đoạn đường đó vào cùng thời điểm.

Kỹ thuật server

- Truy vấn database:
  Đối với việc truy vấn database để trả về kết quả tracking của thiết bị chúng ta có thể sử dụng cách so sánh thời gian được lưu trong các tuple trùng với thời gian hiện tại của hệ thống nhưng cách này gây tốn nhiều chi phí cho việc truy vấn database và dữ liệu có thể bị thiếu.Với việc gửi lên lastID tức là id của tuple được truy vấn cuối cùng trước đó thì ta chỉ viêc truy vấn các tuple có id lớn hơn lastID.
- Response trả về là một JSONArray giúp bên web dễ dàng xử lý thông tin.

CHƯƠNG 5: HIỆN THỰC HỆ THỐNG

Hiện thực chi tiết hệ thống được nhóm chúng tôi đính kèm trong mã nguồn kèm theo ở dĩa CD.

Sau đây là một số giải pháp hiện thực các phương thức truyền gửi dữ liệu từ client lên server, lưu trữ chúng vào database server và truy vấn database để hiển thị lên website.

5.1 Phía Client

- Dùng chuỗi String để lưu trữ dữ liệu GPS vào thẻ nhớ điện thoại, đồng thời gửi chúng lên server thông qua giao thức UDP. Lí do dùng String thay vì List hay Array bởi vì string có dung lượng nhẹ hơn, khi truyền nhận không tốn nhiều chi phí.

```
oneNode = device_id + " " + latitude + " " + longitude + " "
        + elevation + " " + currentTime + " " + accuracy + " "
        + speed + " ";
// trackList = trackList + oneNode;

String newLine = System.getProperty("line.separator");
onePoint = "\t  <trkpt lat=\"" + latitude + "\" lon=\"" + longitude
        + "\">" + newLine + "\t\t<ele>" + elevation + "</ele>"
        + newLine + "\t\t<time>" + currentTime + "</time" + newLine
        + "\t\t<extensions>" + newLine + "\t\t  <ogt10:accuracy>"
        + accuracy + "</ogt10:accuracy>" + newLine
        + "\t\t  <speed>" + speed + "</speed>" + newLine
        + "\t\t</extensions>" + newLine + "\t  <trkpt>" + newLine;

trackList += onePoint;
tempList = tempList + oneNode;
```

- Thiết lập thời gian mặc định giữa các lần gửi dữ liệu lên server là 300s. Chúng tôi chọn giá trị này để gói dữ liệu có được vừa đủ dùng cho việc dự báo kẹt xe, đồng thờigiúp làm giảm chi phí tương tác giữa client và server.

```
public void updateData() {
    delay = Integer.parseInt(PreferenceManager.getDefaultSharedPreferences(
            this.getApplicationContext()).getString(
            MainActivity.KEY_SERVER_ADDRESS_INTERVAL,
            MainActivity.VAL_SERVER_ADDRESS_INTERVAL)) * 1000;
    timer = new Timer();
    timer.schedule(new TimerTask() {
        @Override
        public void run() {
            sendToServer();
        }
    }, 0, delay);
}
```

5.2 Phía Server

- Dữ liệu từ điện thoại gửi đến server public "www.vre.cse.hcmut.edu.vn"sẽ được lưutrữ thông qua JDBC Mysql trên server "172.28.10.96".

```
String url = "jdbc:mysql://172.28.10.19:3306/test1";
conn = DriverManager.getConnection(url, "minhtri", "12345678");
for (int i = 0; i<count ; i +=7 )
{
String query = "insert into TrackingData (IMEI, longitude, latitude, elevation, currentTime, accuracy, speed) " +
        "values ("
                + src[i] + ", "
                + src[i+1] + ", "
                + src[i+2] + ", "
                + src[i+3] + ", "
                + "'" + src[i+4] + "', "
                + src[i+5] + ", "
                + src[i+6] + ");"
                ;
try
{
    System.out.println(query);
    Statement st = conn.createStatement();
    int rs = st.executeUpdate(query);
```

- Các cấu trúc bảng hỗ trợ việc thao tác trên server

```
mysql> desc GPSdata;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| deviceid  | varchar(15) | YES  |     | NULL    |       |
| latitude  | varchar(25) | YES  |     | NULL    |       |
| longitude | varchar(25) | YES  |     | NULL    |       |
| speed     | varchar(15) | YES  |     | NULL    |       |
| satellite | varchar(5)  | YES  |     | NULL    |       |
| boollock  | varchar(5)  | YES  |     | NULL    |       |
| trktime   | varchar(25) | YES  |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
7 rows in set (0.00 sec)
```

```
mysql> desc TrackingData;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| IMEI        | varchar(15) | YES  |     | NULL    |       |
| longitude   | varchar(25) | YES  |     | NULL    |       |
| latitude    | varchar(25) | YES  |     | NULL    |       |
| elevation   | varchar(20) | YES  |     | NULL    |       |
| currentTime | varchar(25) | YES  |     | NULL    |       |
| accuracy    | varchar(20) | YES  |     | NULL    |       |
| speed       | varchar(20) | YES  |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
7 rows in set (0.00 sec)
```

```
mysql> desc tempGPS;
+-------------+-------------+------+-----+---------+----------------+
| Field       | Type        | Null | Key | Default | Extra          |
+-------------+-------------+------+-----+---------+----------------+
| ID          | int(11)     | NO   | PRI | NULL    | auto_increment |
| imei        | varchar(15) | YES  |     | NULL    |                |
| latitude    | varchar(20) | YES  |     | NULL    |                |
| longitude   | varchar(20) | YES  |     | NULL    |                |
| elevation   | varchar(20) | YES  |     | NULL    |                |
| currentTime | varchar(25) | YES  |     | NULL    |                |
| accuracy    | varchar(20) | YES  |     | NULL    |                |
| speed       | varchar(20) | YES  |     | NULL    |                |
+-------------+-------------+------+-----+---------+----------------+
8 rows in set (0.01 sec)
```

```
mysql> desc TrafficData;
+----------------+-------------+------+-----+---------+-------+
| Field          | Type        | Null | Key | Default | Extra |
+----------------+-------------+------+-----+---------+-------+
| IDNode         | int(8)      | YES  |     | NULL    |       |
| startLatitude  | float(10,6) | YES  |     | NULL    |       |
| startLongitude | float(10,6) | YES  |     | NULL    |       |
| destLatitude   | float(10,6) | YES  |     | NULL    |       |
| destLongitude  | float(10,6) | YES  |     | NULL    |       |
| speed          | varchar(20) | YES  |     | NULL    |       |
| currentTime    | varchar(25) | YES  |     | NULL    |       |
+----------------+-------------+------+-----+---------+-------+
7 rows in set (0.00 sec)
```

```
mysql> desc tempTrafficData;
+----------------+-------------+------+-----+---------+----------------+
| Field          | Type        | Null | Key | Default | Extra          |
+----------------+-------------+------+-----+---------+----------------+
| ID             | int(11)     | NO   | PRI | NULL    | auto_increment |
| IDNode         | int(8)      | YES  |     | NULL    |                |
| startLatitude  | float(10,6) | YES  |     | NULL    |                |
| startLongitude | float(10,6) | YES  |     | NULL    |                |
| destLatitude   | float(10,6) | YES  |     | NULL    |                |
| destLongitude  | float(10,6) | YES  |     | NULL    |                |
| speed          | varchar(20) | YES  |     | NULL    |                |
| currentTime    | varchar(25) | YES  |     | NULL    |                |
+----------------+-------------+------+-----+---------+----------------+
8 rows in set (0.01 sec)
```

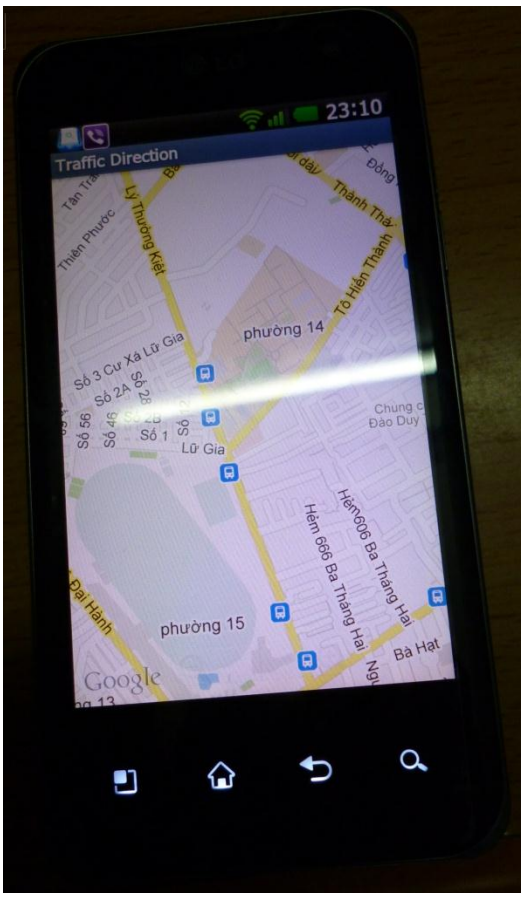
5.3 Phía Web Server

Các cấu trúc bảng hỗ trợ việc thao tác trên web server

```
mysql> desc GPSUser;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| USERNAME  | varchar(25) | NO   | PRI | NULL    |       |
| PASSWORD  | varchar(25) | YES  |     | NULL    |       |
| IMEI      | varchar(15) | YES  |     | NULL    |       |
| EMAIL     | varchar(50) | YES  |     | NULL    |       |
| TELEPHONE | varchar(15) | YES  |     | NULL    |       |
| ADDRESS   | varchar(50) | YES  |     | NULL    |       |
| PRIVS     | varchar(10) | YES  |     | user    |       |
+-----------+-------------+------+-----+---------+-------+
7 rows in set (0.02 sec)
```

5.4 Ứng dụng minh họa

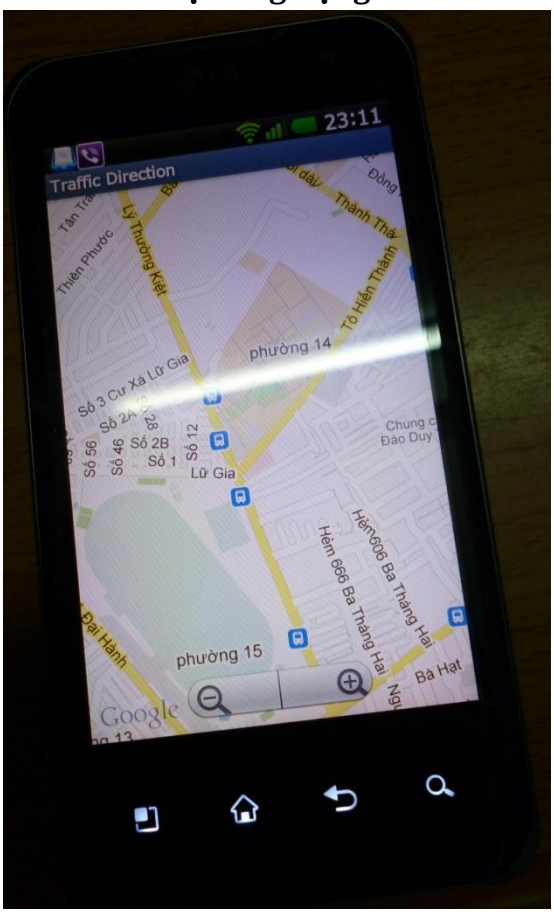
5.4.1 Giao diện mặc định của ứng dụng



Sửdụng Google Maps API đểhiểnthịbảnđồ.

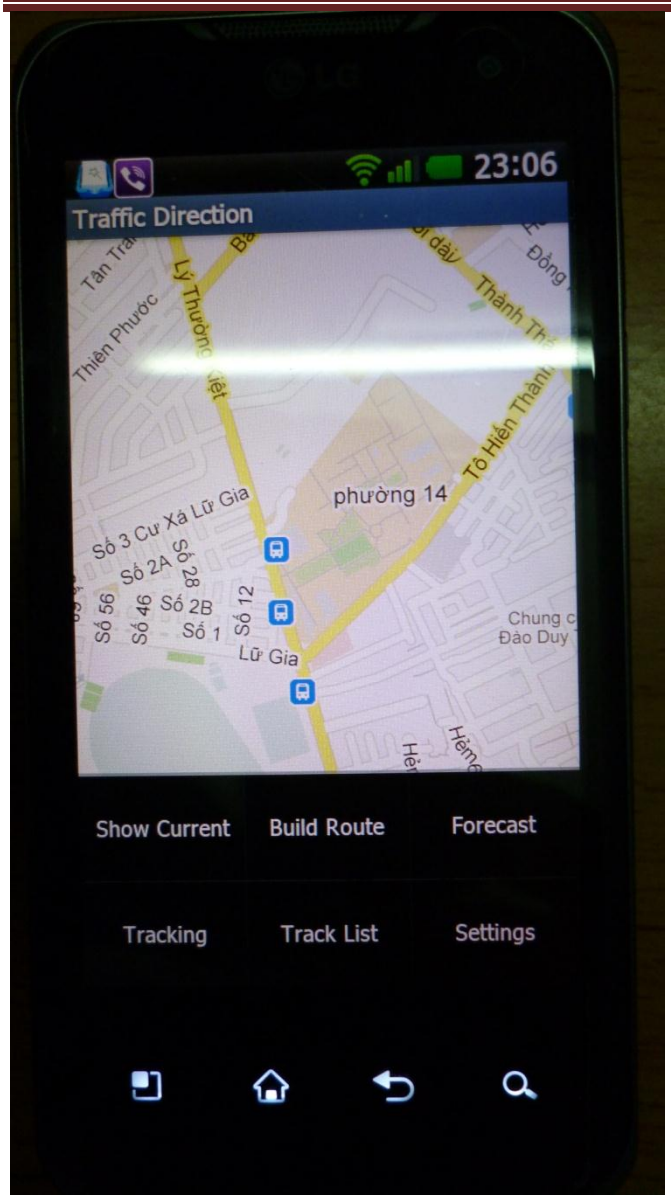
Màn hình mặc định của ứng dụng hiển thị bản đồ khu vực xung quanh Đại học Bách khoa thành phố Hồ Chí Minh.

5.4.2 Giao diện ứng dụng với zoom control



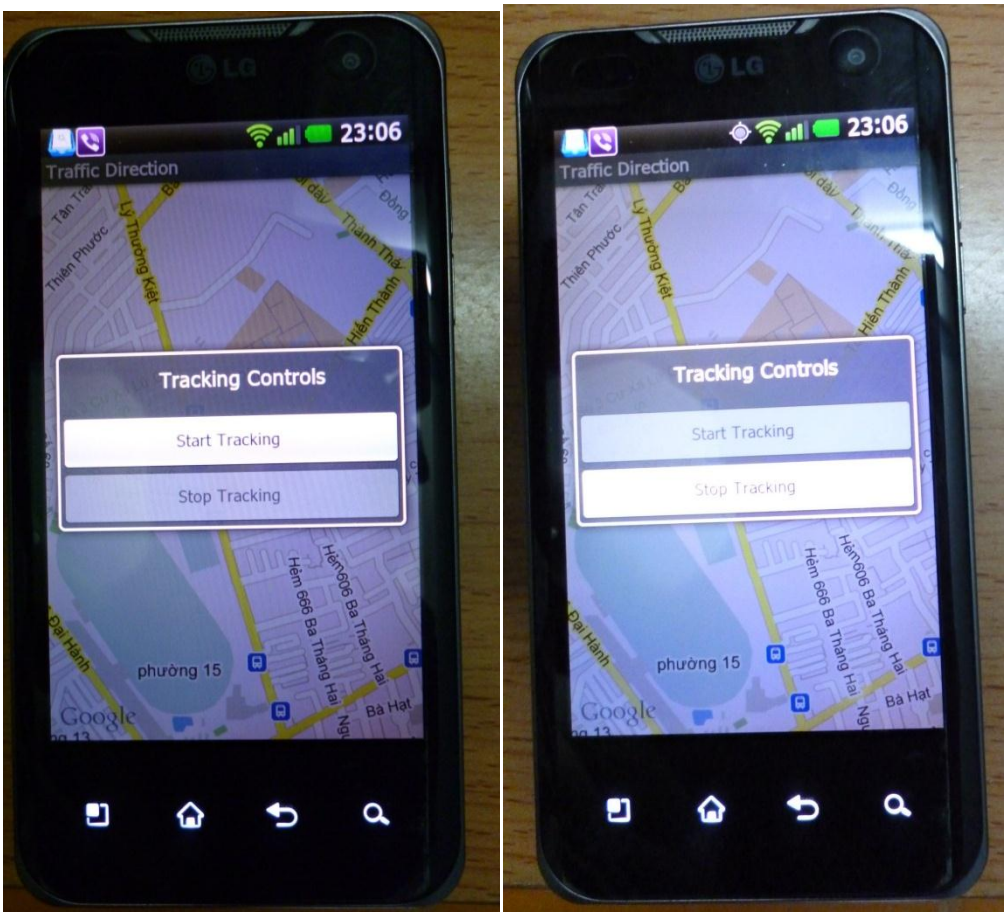
Khi người dùng tương tác vào màn hình ứng dụng, người dùng có thể tùy chỉnh phóng to hoặc thu nhỏ bản đồ.

5.4.3 Giao diện Menu của ứng dụng



Khingườidùngnhấn Menu, ứngdụngsẽhiểnthịcác Menu đểngườitùy chọn tác vụ.

5.4.4 Giao diện chức năng Tracking



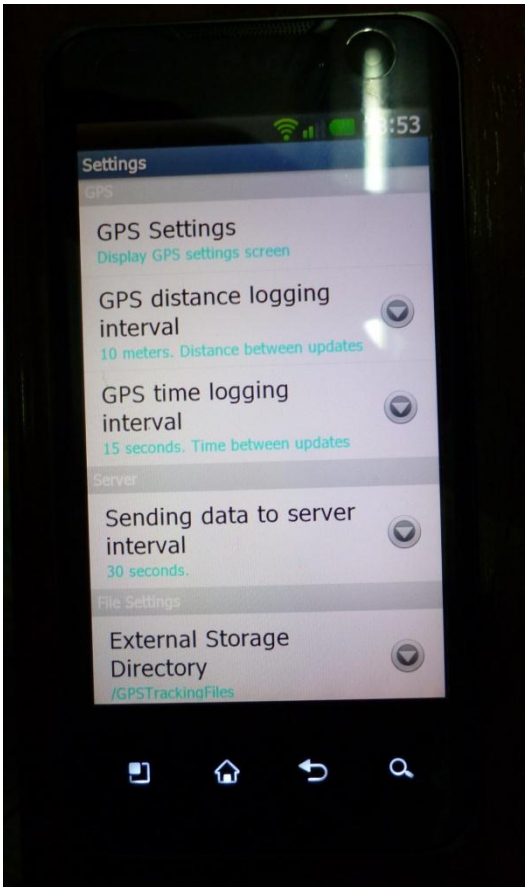
Chứcnăng Tracking – Thu thậpdữliệungườidùng.

Khingườidùngấnvàonút "Start Tracking", ứngdụngsẽkíchhoạtdịchvụchạyngầm, thuthậpdữliệungườidùng, baogồmsố IMEI củathiếtbị, thông tin cácvịtríthuthậpđược và gửi lên server trong khoảng thời gian mặc định là 30s.Chạy ngầm ở đây có nghĩa là người dùng vẫn có thể sử dụng các chức năng khác của điện thoại như nhắn tin, gọi điện khi đang chạy ứng dụng.

Khingườidùngấnvàonút "Stop Tracking", ứngdụngsẽđóngdịchvụngầm, đồngthời, sẽghicácdữliệuđãthuthậpxuống file GPX, lưutrữtrongthẻnhớcủathiếtbị.

Khiứngdụngkhôngthuthậpdữliệu, ngườidùngchỉcóthểấnnút "Start Tracking", nút "Stop Tracking" biẩn vàngượclại.

5.4.5 Giao diện Settings của ứng dụng



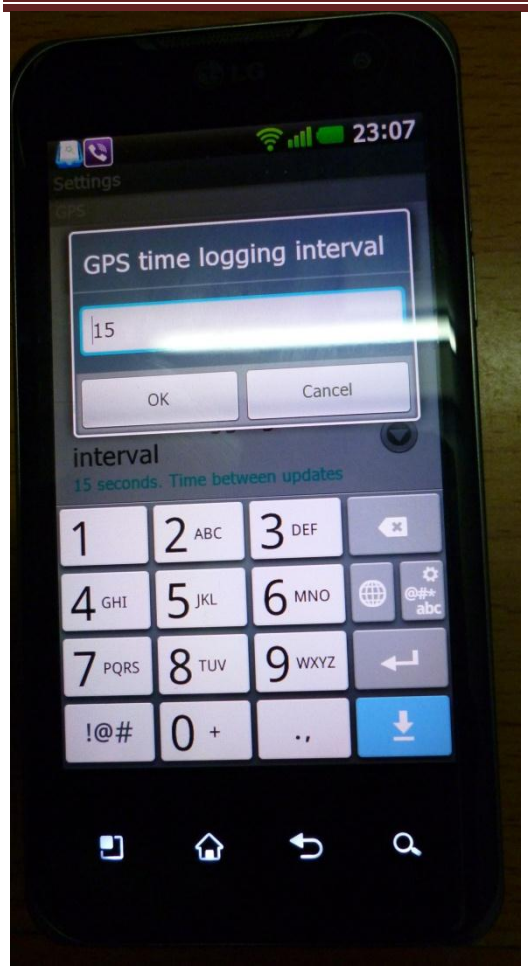
Chứcnăngtùychỉnhứngdụng

Ngườidùngcóthểtùychỉnhlạiứngdụngtheosởthíchngườidùng

GPS Settings: chuyển sang mànhìnhtùychỉnh GPS.

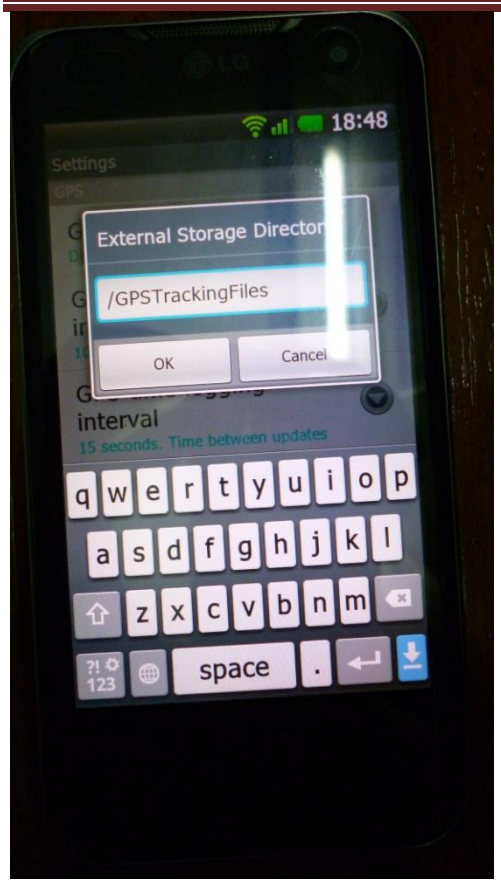
GPS distance logging interval: tùychỉnhkhoảngcáchgiữacáclầncậpnhật.

GPS time logging interval: tùychỉnhthờigiangiữacáclầncậpnhật.



Sending data to server interval: tùychỉnhthờigiangiữacáclầngửidữliệulên Server.

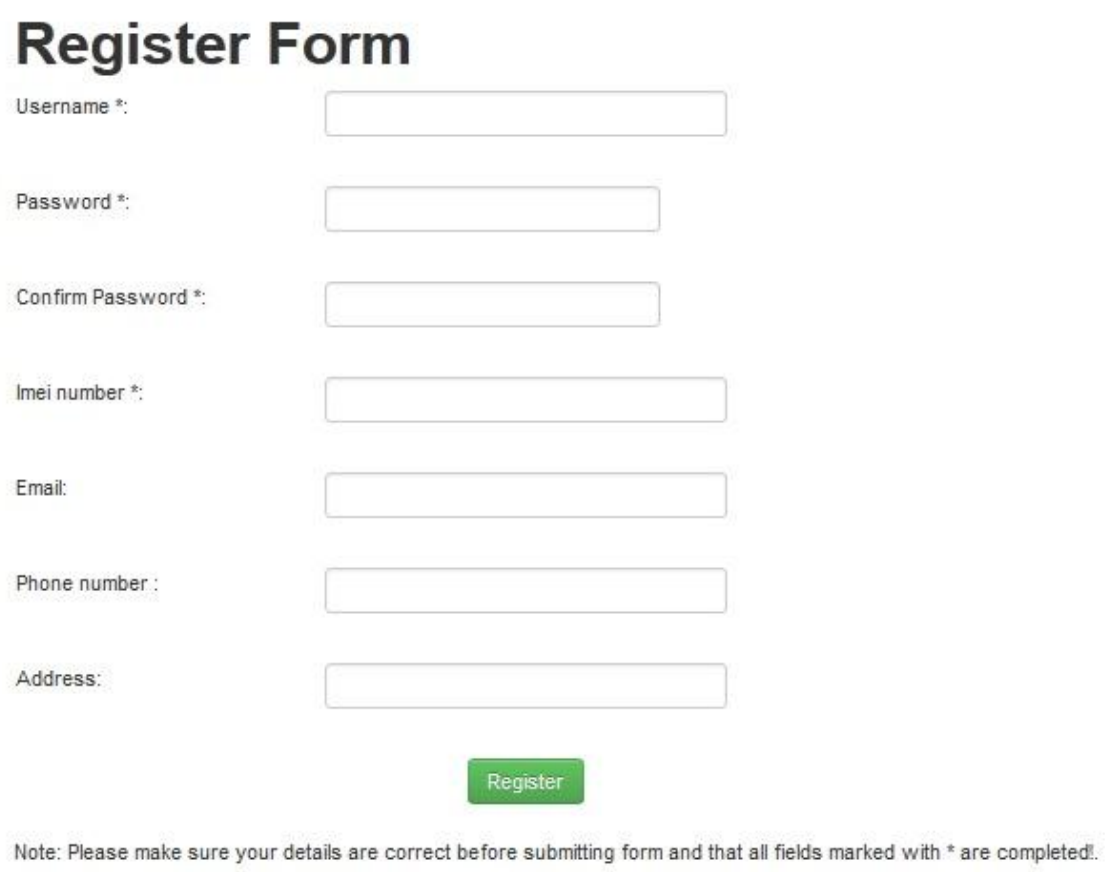
External Storage Directory: tùychọnnơilưu file GPX trongthẻ nhớ.



5.4.6 Giao diện web server

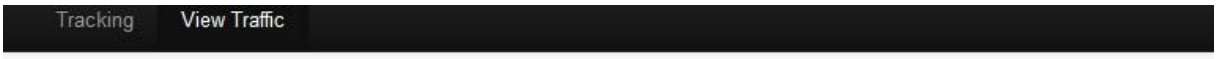
Chức năng Register

Cho phép người dùng đăng kí tài khoản trên website để thuận tiện hơn trong việc truy cập web những lần kế tiếp.

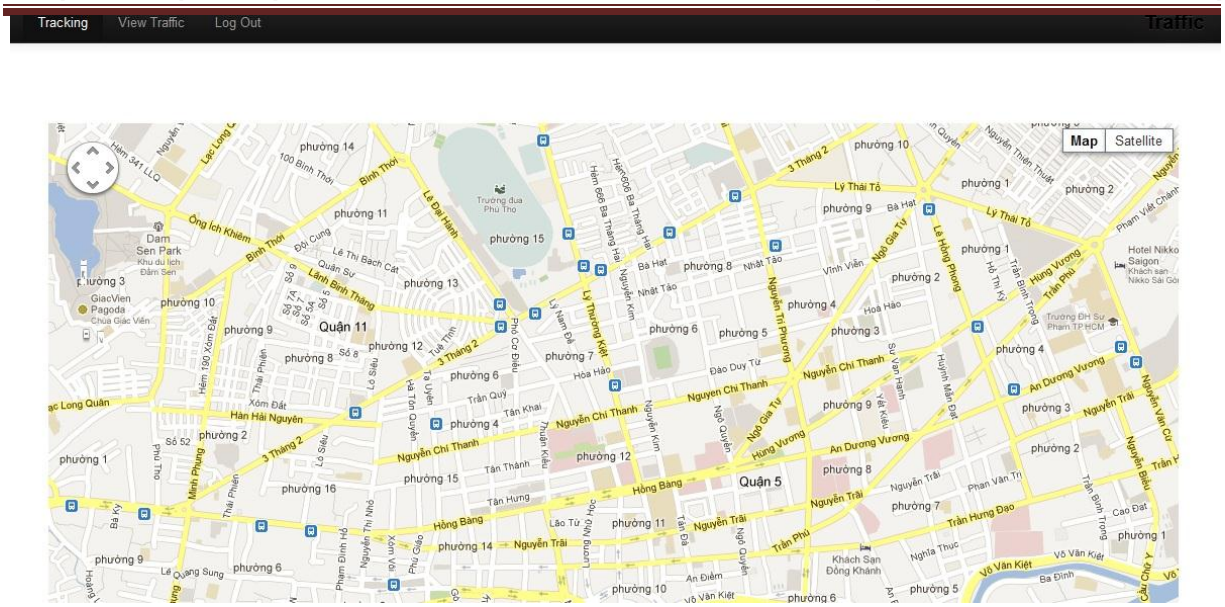


Chức năng Login

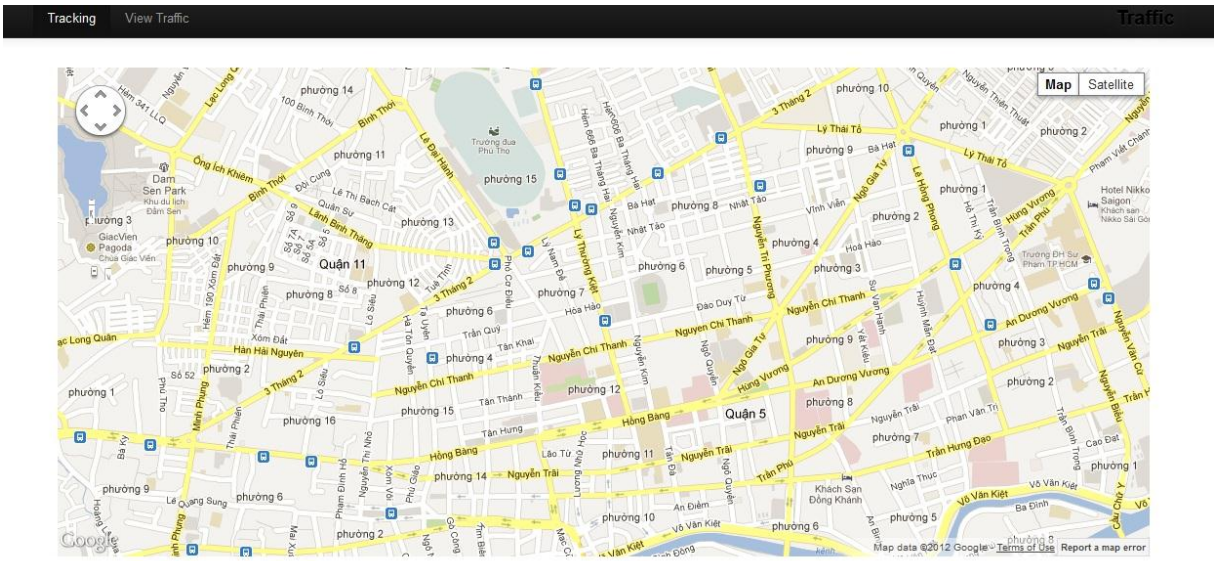
Sau khi register thì website trả về giao diện log in để người dùng đăng nhập tài khoản vừa mới khởi tạo.



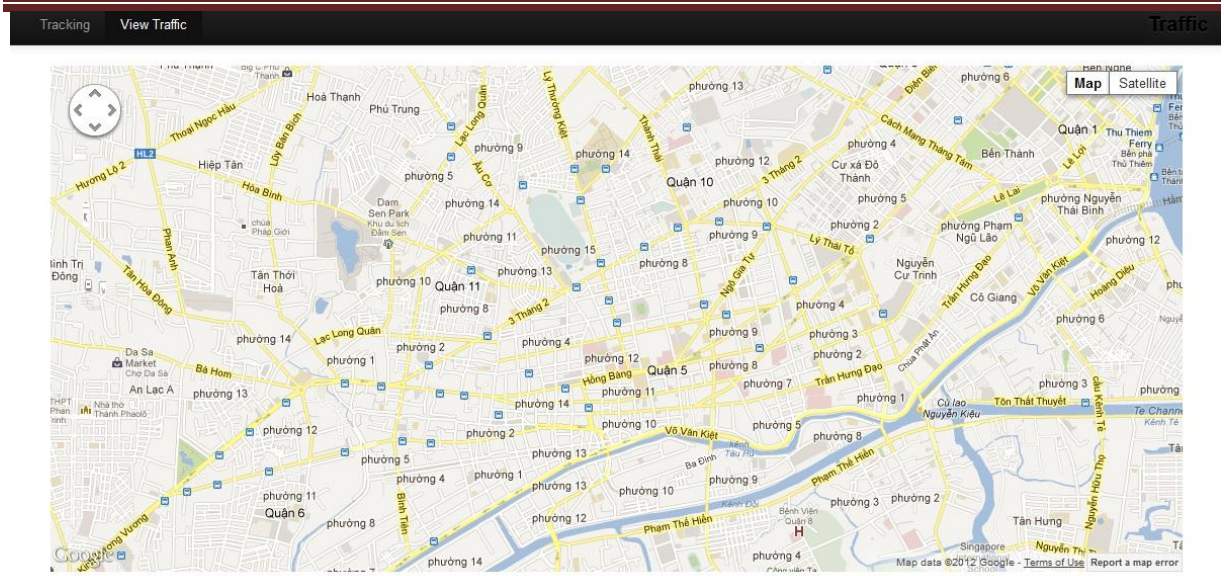
Chức năng Log Out



Giao diện Tracking



Giao diện View Traffic


Tracking
View Traffic
Traffic
Map
Satellite
Quận 1
Quận 10
Quận 11
Quận 5
Quận 6
Bến Thành
Tân Hưng
Map data ©2012 Google - Terms of Use
Report a map error

5.5Đánh giá ứng dụng

5.5.1 Ưu điểm

- Thử nghiệm thành công công nghệ.
- Thu thập được tín hiệu GPS,xác định được số imei, độ cao, thời gian, độ chính xác, vận tốc và tọa độ tại điểm người dùng đi qua.
- Tạo ra được file GPX và lưu thành công vào thẻ nhớ.
- Gửi thành công dữ liệu lên server và lưu vào database.
- Xây dựng thành công ứng dụng web trên server để biểu diễn dữ liệu thu thập được một cách trực quan.

5.5.2 Khuyết điểm

- Tín hiệu GPS đôi khi trả về chưa chính xác.
- Chưa giải quyết được vấn đề hao tốn năng lượng pin khi sử dụng GPS.

5.6 Những khó khăn trong việc thực hiện ứng dụng

- Khi nguồn dữ quá lớn thì server không đáp ứng được.

CHƯƠNG 6: TỔNG KẾT

6.1 Các công việc đã thực hiện

- Thu thập tín hiệu GPS, xác định được số imei, độ cao, thời gian, độ chính xác, vận tốc và tọa độ tại điểm người dùng đi qua.
- Tạo ra được file GPX và lưu thành công vào thẻ nhớ điện thoại cũng như gửi dữ liệu lên server thông qua giao thức UDP.
- Config mysql-server, kết nối jdbc đến mysql để lưu trữ dữ liệu trên server.
- Config tomcat-server, hiện thực website hiển thị đường đi của một hay nhiều người dùng khác nhau để mô tả cho ứng dụng thu thập dữ liệu.
- Import file CSV lưu trữ dữ liệu xe buýt vào database server.

6.2 Hạn chế

- Do nhóm chưa có kinh nghiệm làm việc trên server nên mất không ít thời gian để tìm hiểu.
- Chưa xử lí được vấn đề hao tốn năng lượng pin khi thu thập dữ liệu thông qua GPS.

6.3 Các kinh nghiệm đạt được

- Có khả năng lập trình Android để phát triển ứng dụng trên smartphone.
- Hiểu rõ hơn về giao thức UDP tương tác truyền nhận dữ liệu giữa client và server.
- Phát triển kĩ năng làm việc trên server, sử dụng các câu lệnh cơ bản cũng như config những chương trình cần thiết: mysql-server, tomcat-server.
- Có khả năng sử dụng nhiều loại câu truy vấn mysql khác nhau.
- Có khả năng lập trình website chạy trên tomcat-server dùng java servlet, ajax, html, css.

6.4 Tính ứng dụng của đề tài

Thu thập dữ liệu giao thông để dự báo ùn tắc giao thông là bài toán cấp thiết và quan trọng ở Việt Nam. Qua trình bày ở các chương trên, nhóm chúng tôi đã đưa ra hướng tiếp cận cho bài toán thu thập dữ liệu giao thông nhằm giúp cho việc dự báo tình trạng kẹt xe chính xác hơn. Nhờ đó, giúp người dùng có thể đến được địa điểm mong muốn mà không đi qua các nơi kẹt xe, giúp tiết kiệm chi phí, sức khỏe cho bản thân họ cũng như lưu thông tổng thể trong thành phố được tốt hơn.

Tuy nhiên, do vấn đề về năng lượng pin chưa được giải quyết nên khả năng áp dụng ở Vietnam (nơi phương tiện đi lại chủ yếu là xe máy) là không cao. Nhưng nếu áp dụng với các nước phát triển như ThaiLand, Singapore (nơi mà xe oto là phương tiện đi lại chủ yếu, có thể sạc pin khi lưu thông) thì tình hình sẽ rất khả quan.

PHỤ LỤC

GPS	Hệ thống định vị toàn cầu
Server	Máy chủ
Client	Máy khách
Smartphone	Điện thoại thông minh
IMEI	Số nhận dạng thiết bị di động trên toàn thế giới
Database	Cơ sở dữ liệu
Admin	Người quản trị
User	Người dùng
Camera	Máy quay
3G	Công nghệ truyền thông thế hệ thứ ba
Wifi	Mạng không dây
Upload	Tải lên
Download	Tài xuống
Tracking	Sự theo dõi
Module	Môđun
Open source	Mã nguồn mở
Java	Ngôn ngữ lập trình Java
Application framework	
Intergrated browser	Trình duyệt Web tích hợp
API	Giao diện lập trình ứng dụng