**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG**

****

**KHÓA LUẬN TỐT NGHIỆP**

**XÂY DỰNG HỆ THỐNG QUẢN LÝ XE KHÁCH**

**Giảng viên hướng dẫn : ThS. Hồ Hải**

**Sinh viên thực hiện : Nguyễn Hoàng Nam – 10520287**

**Nguyễn Ngọc Hưng – 10520192**

**Lớp : MMT & TT 05**

**Khóa : 2010 – 2015**

**TP. HỒ CHÍ MINH, THÁNG 6 NĂM 2015**

DANH SÁCH HỘI ĐỒNG BẢO VỆ KHÓA LUẬN

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

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

**LỜI MỞ ĐẦU**

Ngày nay, việc đi xe khách đã quen thuộc với tất cả mọi người, mọi người sử dụng xe khách là phương tiện cho nhưng chuyến đi xa như đi du lịch, sinh viên về thăm nhà sau nhiều ngày đi học xa nhà …. Số lượng nhà xe tăng nhiều, hầu như xe khách đã có trạm ở tất cả tỉnh thành trong nước. Đi xe khách giúp bạn có cảm giác an toàn hơn đi xe máy, có thể làm việc khi đang đi trên xe nên tiết kiệm được thời gian hơn, các xe bus chất lượng cao còn có cả ghế nằm, wifi miễn phí … cho bạn cảm giác thoải mái. Khi đi xa chỉ việc lên xe nằm ngủ một giấc hay lên mạng lướt web, đọc báo, chơi game và đợi xe tới nơi. Bạn sẽ không lo lắng về việc bị lạc đường, hay chán nản, mệt mỏi khi phải ngồi lâu trên yên xe máy. Bạn sẽ không lo lắng về việc bị lạc đường, hay chán nản, mệt mỏi khi phải ngồi lâu trên yên xe máy.

Hiện tại trên các xe khách chất lượng cao đều có wifi và việc sở hữu một thiết bị thông minh chạy hệ điều hành android khá dễ dàng với giá thành hiện giờ, vì vậy nhà quản lý dễ dàng áp dụng hệ thống quản lý xe thông minh thay cho quản lý thủ công như hiện nay. Hiện nay, trên những tuyến đường xa, ban quản lý phải liên lạc với xe qua điện thoại, không theo dõi được tình trạng của chuyến xe. Lơ xe thì thỉnh thoảng lại quên đón và cho hành khách xuống đúng chỗ. Hành khách thì cũng không có cách nào phản hồi lại chất lượng dịch vụ ngoài việc gọi điện thoại. Hệ thống này được sử dụng cho ban quản lý nhà xe, trên xe và hành khách. Ngoài ra hệ thống này có thể giúp hành khách tiện lợi hơn trong việc đặt vé online thông qua di động cũng như xác định điểm dừng chính xác hơn.

Chính việc muốn nâng cao chất lượng xe khách lên, nhóm chúng em đã quyết định phát triển hệ thống quản lý xe khách. Hệ thống được xây dựng trên nền tảng php trên Web Server và Android cho thiết bị di động vì sự phổ biến của các thiết bị di động, đặc biệt là các smart phone sử dụng hệ điều hành Android. Ngoài ra, Android cũng là hệ điều hành được Google hỗ trợ các dịch vụ Google Maps và các dịch vụ tìm kiếm nhiều nhất so với các hệ điều hành khác. Vì thế nhóm em đã chọn phát triển ứng dụng trên hệ điều hành Android.

Trong đề tài nhóm sử dụng bộ thư viện Google Map API Version 2 và 3. Đây là bộ thư viện giúp ứng dụng có thể load maps, đánh dấu, chỉ đường.... Bên cạnh đó nhóm cũng tạo một web service bằng ngôn ngữ PHP để quản lý việc đặt vé, xác định điểm dừng của hành khách, báo cáo các vi phạm của xe .... Do ứng dụng chạy trên hệ điều hành Android nên nhóm sử dụng ngôn ngữ lập trình Java và bộ công công cụ lập trình Android Studio để xây dựng ứng dụng. Những thành phần này sẽ được đề cập chi tiết trong phần sau.

**LỜI CẢM ƠN**

Đầu tiên, chúng em xin bày tỏ lòng biết ơn chân thành đến thầy Hồ Hải, người đã tận tình hướng dẫn, tạo mọi điều kiện thuận lợi để nhóm hoàn thành tốt đề tài. Thầy đã định hướng cho chúng em từ cách đặt vấn đề, phương pháp nghiên cứu khoa học cho đến những công việc cụ thể nhất.

Chúng em xin gửi lời cảm ơn đến gia đình, bạn bè, những người luôn sát cánh, động viên chúng em trên bước đường học tập cũng như trong cuộc sống.

Xin chân thành biết ơn sự tận tình dạy dỗ và sự giúp đỡ của tất cả quý thầy cô tại trường Đại học Công NghệThông Tin, đặc biệt là các thầy cô trong khoa Mạng máy tính và truyền thông. Tất cả các kiến thức mà nhà trường và quý thầy cô đã truyền đạt là hành trang to lớn để chúng em mang theo trên con đường học tập, làm việc và nghiên cứu cũng như trong quá trình hoàn thiện nhân cách của mình.

TP. Hồ Chí Minh, ngày tháng năm 2015

Sinh viên thực hiện khóa luận

Nguyễn Hoàng Nam

Nguyễn Ngọc Hưng

**ĐỀ CƯƠNG CHI TIẾT**

|  |  |
| --- | --- |
| **TÊN ĐỀ TÀI: XÂY DỰNG HỆ THỐNG QUẢN LÝ XE KHÁCH** | |
| **Cán bộ hướng dẫn:** ThS. Hồ Hải | |
| **Thời gian thực hiện:**Từ ngày 19/01/2015 đến ngày 22/06/2015 | |
| **Sinh viên thực hiện:**  Nguyễn Hoàng Nam - 10520287  Nguyễn Ngọc Hưng - 10520192 | |
| **Nội dung đề tài:**   1. **Mục tiêu đề tài :**   Hiện nay, những chuyến đi đường dài của mọi người còn những khó khăn bất cập như sau:   * Phải đặt vé qua điện thoại hoặc tới trực tiếp bến để mua vé. * Không biết được thời gian xe gần tới bến để chuẩn bị. * Không biết được vị trí hiện tại của xe. * Tài xế phải nhớ những trạm đón, trả khách và đường đi. * Việc liên hệ với hành khách khó khăn. * Khó kiểm soát hành khách trên xe.   Xây dựng hệ thống nhằm tạo sự thuận tiện cho cả quản lý, tài xế, hành khách. Quản lý dễ dàng quản lý được thông tin sai phạm của tài xế một cách tự động. Tài xe không còn vất vả phải nhớ từng trạm đón trả khách. Khi gần đến trạm, hệ thống sẽ thông báo cho tài xế có bao nhiêu khách lên, có bao nhiêu khách xuống. Hành khách sẽ tiện lợi hơn trong việc đặt vé, chủ động hơn trong thời gian khi biết được khoảng cách hiện tại của xe đối với mình, khi gần đến trạm xuống hệ thống sẽ tự báo cho hành khách biết.   1. **Ý nghĩa đề tài**   Với những mục tiêu cụ thể đã đặt ra như trên nhóm em hi vọng với hệ thống này sẽ giúp cho việc sử dụng và quản lý xe khách trở nên dễ dàng hơn. Sẽ không gặp các trường hợp như :   * Xe quên dừng đón khách. * Hành khách không biết vị trí hiện tại của xe dẫn đến việc trễ xe. * Quản lý không biết vi phạm tốc độ của xe để xử lý.  1. **Phạm vi nghiên cứu**  * Đối với các kỹ thuật lập trình trên hệ điều hành Android: * Nghiên cứu các khái niệm, kỹ thuật tạo các Activiti và Service có thể hoạt động trên nhiều phiên bản Android, không đi sau vào kiến trúc hạ tầng hay kernel của Android. * Nghiên cứu việc liên kết cơ sở dữ liệu MySQL của Web Server. * Nghiên cứu việc kết nối, truyền và lấy dữ liệu từ Server. * Nghiên cứu thiết kế giao diện trên Android. * Đối với Google Map API * Nghiên cứu Google API hai phiên bản dành cho Android và trên WEB. * Nghiên cứu cách đăng ký và sử dụng các dịch vụ trên Google API. * Nghiên cứu xác định vị tris hiện tại thông qua GPS hoặc mạng. * Nghiên cứu việc vẽ đường đi qua nhiều điểm trên Google Map. * Nghiên cứu cách đánh dấu các vị trí trên bản độ. * Đối với PHP, MySQL * Nghiên cứu cách tạo Web Server , tạo giao diện để thao tác với cơ sở dữ liệu. * Nghiên cứu cách giao tiếp, trao đổi dữ liệu với Android. * Nghiên cứu về Codeigniter framework. * Nghiên cứu về Bootstrap framework. * Đối với dữ liệu các trạm dừng xe * Dựa theo các lộ trình của trạm xe bus trong thành phố Hồ Chí Minh.  1. **Phương pháp thực hiện**   Nghiên cứu dựa trên các các tài liệu về lập trình Android, lập trình Web, khảo sát và phân tích hiện trạng thực tế để xây dựng hệ thống.   1. **Kết quả mong đợi**   Xây dựng hệ thống quán lý xe khách gồm 3 phần dành cho hành khách, lái xe và quản lý. Chức năng của từng phần :   * + Web Server :     - Nhận đặt vé xe của hành khách.     - Lưu lỗi vi phạm của xe.     - Cập nhập dữ liệu hành khách cho xe (tên, địa chỉ trạm đón và xuống của hành khách …).     - Nhận ý kiến của hành khách.   + Xe :     - Xem các trạm dừng có khách lên hoặc xuống trên bản đồ.     - Thông báo khi gần tới trạm có khách lên hoặc xuống.     - Nhắc nhở khi xe vi phạm các lỗi quy định.     - Gửi lỗi vi phạm về cho quản lý.     - Đăng nhập theo xe để lấy thông tin cần thiết.   + Hành khách :     - Đăng ký vé xe.     - Nhận thông báo khi xe gần tới trạm.     - Gửi ý kiến phản hồi cho nhà xe.     - Xem tất cả đơn hàng đã đặt.     - Xác định được vị trí hiện tại của xe. | |
| **Kế hoạch thực hiện:**   |  |  | | --- | --- | | PHÂN CÔNG CÔNG VIỆC | | | TÊN SINH VIÊN | CÔNG VIỆC | | Nguyễn Hoàng Nam | Nghiên cứu và phát triển phần mềm trên hệ điều hành Android.   * Ứng dụng cho hành khách. * Ứng dụng cho tài xế. | | Nguyễn Ngọc Hưng | Nghiên cứu và phát triển ứng dụng web sử dụng ngôn ngữ php.   * Ứng dụng Web dành cho quản lý. * Web Server. | | |
| **Xác nhận của CBHD**  (Ký tên và ghi rõ họ tên) | **TP. HCM, ngày….tháng …..năm…..**  **Sinh viên**  (Ký tên và ghi rõ họ tên) |

**NHẬN XÉT**

(Của giáo viên hướng dẫn)

TP. Hồ Chí Minh, ngày tháng năm 2015.

Giảng viên hướng dẫn

NHẬN XÉT

(Của giáo viên phản biện)

TP. Hồ Chí Minh, ngày tháng năm 2015.

Giáo viên phản biện

MỤC LỤC

[Chương 1 GIỚI THIỆU CHUNG 1](#_Toc422749370)

[1.1 Tên đề tài khóa luận 1](#_Toc422749371)

[1.2 Mục tiêu đề tài 1](#_Toc422749372)

[1.3 Ý nghĩa đề tài 1](#_Toc422749373)

[1.4 Đối tượng, phạm vi và phương pháp nghiên cứu đề tài 1](#_Toc422749374)

[1.4.1 Đối tượng nghiên cứu 1](#_Toc422749375)

[1.4.2 Phạm vi nghiên cứu 2](#_Toc422749376)

[1.4.2.1 Đối với các kỹ thuật lập trình trên hệ điều hành Android: 2](#_Toc422749377)

[1.4.2.2 Đối với Google Map API 2](#_Toc422749378)

[1.4.2.3 Đối với PHP, MySQL 2](#_Toc422749379)

[1.4.2.4 Đối với dữ liệu các trạm dừng xe 2](#_Toc422749380)

[1.4.3 Phương pháp nghiên cứu 2](#_Toc422749381)

[1.4.4 Thời gian nghiên cứu 2](#_Toc422749382)

[1.4.5 Kế hoạch thực hiện đề tài 2](#_Toc422749383)

[1.4.6 Quá trình cụ thể thực hiện kế hoạch 3](#_Toc422749384)

[Chương 2 NHỮNG KIẾN THỨC CƠ BẢN VỀ ĐỀ TÀI 4](#_Toc422749385)

[2.1 Hệ điều hành Android 4](#_Toc422749386)

[2.1.1 Tổng quát về Android 4](#_Toc422749387)

[2.1.2 Phát triển ứng dụng trên Android 4](#_Toc422749388)

[2.1.2.1 Cấu trúc của các tầng phần mềm trên Android 4](#_Toc422749390)

[2.1.2.2 Kiến trúc hệ điều hành Android 7](#_Toc422749391)

[2.1.2.3 Các thư viện trên Android 8](#_Toc422749392)

[2.1.2.4 Các thành phần cơ bản của ứng dụng Android 9](#_Toc422749393)

[2.2 Google Maps API 17](#_Toc422749394)

[2.2.1 Khái niệm Google Maps API 17](#_Toc422749395)

[2.2.2 Ích lợi khi sử dụng Google Maps API 17](#_Toc422749396)

[2.2.3 Các chức năng chính trong Google Maps API 18](#_Toc422749397)

[2.2.3.1 Google Maps API Web Services 18](#_Toc422749398)

[2.2.3.2 Google Maps javascript API 20](#_Toc422749399)

[2.3 Kỹ thuật định vị GPS 20](#_Toc422749400)

[2.4 Bộ công cụ lập trình trên Android SDK 21](#_Toc422749401)

[2.5 Ngôn ngữ lập trình PHP 21](#_Toc422749402)

[2.5.1 Giới thiệu ngôn ngữ lập trình php 21](#_Toc422749403)

[2.5.2 Cú pháp trong lập trình PHP 22](#_Toc422749404)

[2.6 Hệ quản trị CSDL MySQL 22](#_Toc422749405)

[2.7 JSON ( Javescript Object Noattion) 23](#_Toc422749406)

[2.8 Ngôn ngữ XML 24](#_Toc422749407)

[2.9 Codeigniter framework 26](#_Toc422749408)

[2.10 Bootstrap framework 27](#_Toc422749409)

[Chương 3 PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 29](#_Toc422749410)

[3.1 Xác định yêu cầu và đối tượng trong hệ thống 29](#_Toc422749411)

[3.1.1 Yêu cầu đối với hệ thống: 29](#_Toc422749412)

[3.1.2 Các đối tượng trong hệ thống: 29](#_Toc422749413)

[3.1.3 Thiết kế hệ thống: 30](#_Toc422749414)

[3.1.4 Thiết bị di động smartphone 30](#_Toc422749415)

[3.1.5 Google map api: 31](#_Toc422749416)

[3.1.6 Web Server: 31](#_Toc422749417)

[3.2 Phân tích và thiết kế hệ thống 31](#_Toc422749418)

[3.2.1 Sơ đồ User case: 31](#_Toc422749419)

[3.2.1.1 Danh sách User case 32](#_Toc422749420)

[3.2.2 Sơ đồ phân rã chức năng. 32](#_Toc422749421)

[3.2.2.1 Trên thiết bị Android 32](#_Toc422749422)

[3.2.2.2 Trên Web Server 32](#_Toc422749423)

[3.2.3 Sơ đồ DFD 33](#_Toc422749424)

[3.2.3.1 Mô hình mức ngữ cảnh 33](#_Toc422749425)

[3.2.3.2 Mô hình DFD 34](#_Toc422749426)

[3.2.3.3 Mô hình DFD đối với hành khách: 35](#_Toc422749427)

[3.2.3.4 Mô hình DFD đối với Tài xế: 36](#_Toc422749428)

[3.2.3.5 Mô hình DFD đối với Quản lý 37](#_Toc422749429)

[3.2.4 Thiết kế cơ sở dữ liêu 38](#_Toc422749430)

[3.2.4.1 Mô hình thực thể ERD 38](#_Toc422749431)

[3.2.4.2 Mô hình dữ liệu 39](#_Toc422749432)

[3.3 Mô tả chức năng 41](#_Toc422749433)

[3.3.1 Mô tả chức năng quản lý 41](#_Toc422749434)

[3.3.1.1 Chức năng đăng nhập vào hệ thống 41](#_Toc422749435)

[3.3.1.2 Chức năng quản lý báo cáo sai phạm 42](#_Toc422749436)

[3.3.1.3 Chức năng quản lý hành khách 42](#_Toc422749437)

[3.3.1.4 Chức năng quản lý chuyến đi 43](#_Toc422749438)

[3.3.1.5 Chức năng điều hướng chuyến đi 44](#_Toc422749439)

[3.3.1.6 Chức năng thêm tài xế 44](#_Toc422749440)

[3.3.1.7 Chức năng sửa tài xế 44](#_Toc422749441)

[3.3.1.8 Chức năng quản lý lộ trình 45](#_Toc422749442)

[3.3.1.9 Chức năng thêm trạm. 45](#_Toc422749443)

[3.3.2 Mô tả chức năng đối với tài xế 46](#_Toc422749444)

[3.3.2.1 Đăng nhập 46](#_Toc422749445)

[3.3.2.2 Kiểm tra chuyến đi 47](#_Toc422749446)

[3.3.2.3 Xem danh sách hành khách 49](#_Toc422749447)

[3.3.2.4 Xem bản đồ 50](#_Toc422749448)

[3.3.2.5 Nhận thông báo và xem danh sách hành khách ở trạm dừng 50](#_Toc422749449)

[3.3.2.6 Cảnh báo vi phạm tốc độ 51](#_Toc422749450)

[3.3.3 Mô tả chức năng đối với hành khách 51](#_Toc422749451)

[3.3.3.1 Đăng ký và đăng nhập 51](#_Toc422749452)

[3.3.3.2 Kiểm tra chuyến đi trong ngày 53](#_Toc422749453)

[3.3.3.3 Đặt vé 56](#_Toc422749454)

[3.3.3.4 Xem danh sách đơn hàng 62](#_Toc422749455)

[Chương 4 QUÁ TRÌNH XÂY DỰNG HỆ THỐNG 64](#_Toc422749456)

[4.1 Cài đặt bộ công cụ Android SDK. 64](#_Toc422749457)

[4.2 Khai báo Google Play Service vào ứng dụng 64](#_Toc422749458)

[4.3 Đăng ký sử dụng API key với Google 64](#_Toc422749459)

[4.4 Xây dựng ứng dụng trên android. 64](#_Toc422749460)

[4.4.1 Chức năng của tài xế: 64](#_Toc422749461)

[4.4.2 Chức năng hành khách. 69](#_Toc422749462)

[4.5 Xây dựng ứng dụng web. 72](#_Toc422749463)

[4.5.1 Chức năng của quản lý 72](#_Toc422749464)

[4.6 Xây dựng nhận thông tin cần xử lý và xử lý trên webserver 72](#_Toc422749465)

[4.6.1 Chức năng của Web Server. 72](#_Toc422749466)

[Chương 5 TỔNG KẾT VÀ HƯỚNG PHÁT TRIỂN 74](#_Toc422749467)

[5.1 Kết quả đạt được 74](#_Toc422749468)

[5.2 Hạn chế 74](#_Toc422749469)

[5.3 Những khó khăn khi xây dựng hệ thống 74](#_Toc422749470)

[5.4 Hướng phát triển đề tài 75](#_Toc422749471)

[DANH MỤC TÀI LIỆU THAM KHẢO 76](#_Toc422749472)

**Danh mục bảng**

[Bảng 1‑1 Quá trình chi tiết thực hiện đề tài 3](#_Toc422749291)

[Bảng 2‑1 Các phiên bản MySQL 23](#_Toc422749292)

[Bảng 3‑1 Danh sách User case 32](#_Toc422749293)

[Bảng 3‑2 đặc tả bảng dữ liệu hành khách 39](#_Toc422749294)

[Bảng 3‑3 Đặt tả bảng dữ liệu nhân viên 40](#_Toc422749295)

[Bảng 3‑4 Đặc tả bảng dữ liệu đơn hàng 40](#_Toc422749296)

[Bảng 3‑5 Đăc tả bảng dữ liệu chi tiết 40](#_Toc422749297)

[Bảng 3‑6 Đặc tả bảng dữ liệu lộ trình 41](#_Toc422749298)

[Bảng 3‑7 Đặc tả bảng dữ liệu xe 41](#_Toc422749299)

[Bảng 3‑8 Đặc tả bảng dữ liệu báo cáo 41](#_Toc422749300)

**Danh mục hình**

[Hình 2‑1Cấu trúc của các tầng phần mềm trên Android 5](#_Toc422749301)

[Hình 2‑2 Khai báo tab <activity> trong Android Manifest.xml 10](#_Toc422749302)

[Hình 2‑3 Lược đồ vòng đời của một Activity 11](#_Toc422749303)

[Hình 2‑4 Sử dụng Intent để trao đổi thông tin giữa hai Activity 13](#_Toc422749304)

[Hình 2‑5 Lược đồ vòng đơi của một Service 14](#_Toc422749305)

[Hình 2‑6 Cấu trúc các Screen trong View 15](#_Toc422749306)

[Hình 2‑7 Logo MySQL 23](#_Toc422749307)

[Hình 2‑8 Cú pháp của một đối tượng Json 24](#_Toc422749308)

[Hình 2‑9 Cú pháp của một mảng Json 24](#_Toc422749309)

[Hình 2‑10 Logo Codeigniter 26](#_Toc422749310)

[Hình 2‑11 Logo Bootstrap 27](#_Toc422749311)

[Hình 3‑1 Mô hình hệ thống 30](#_Toc422749312)

[Hình 3‑2 Sơ đồ User case 31](#_Toc422749313)

[Hình 3‑3 Sơ đồ phân rã chức năng trên thiết bị Android 32](#_Toc422749314)

[Hình 3‑4 Sơ đô phân rã chức năng trên Web Server 33](#_Toc422749315)

[Hình 3‑5 Sơ đồ DFD mức ngữ cảnh 33](#_Toc422749316)

[Hình 3‑6 Mô hình DFD 34](#_Toc422749317)

[Hình 3‑7 Mô hình DFD đối với hanh khách 35](#_Toc422749318)

[Hình 3‑8 Mô hình DFD đối với tài xế 36](#_Toc422749319)

[Hình 3‑9 Mô hình DFD đối với quản lý 37](#_Toc422749320)

[Hình 3‑10 Mô hình thực thể ERD 38](#_Toc422749321)

[Hình 3‑11 Mô hình dữ liệu 39](#_Toc422749322)

[Hình 3‑12 Giao diện đăng nhập của quản lý 42](#_Toc422749323)

[Hình 3‑13 Giao diện quản lý báo cáo sai phạm 42](#_Toc422749324)

[Hình 3‑14 Giao diện quản lý hành khách 43](#_Toc422749325)

[Hình 3‑15 Giao diện quản lý chuyến đi 43](#_Toc422749326)

[Hình 3‑16 Giao diện điều hướng chuyến đi 44](#_Toc422749327)

[Hình 3‑17 Giao diện thêm tài xế 44](#_Toc422749328)

[Hình 3‑18 Giao diện sửa tài xế 45](#_Toc422749329)

[Hình 3‑19 Giao diện quản lý lộ trình 45](#_Toc422749330)

[Hình 3‑20 Giao diện thêm trạm 46](#_Toc422749331)

[Hình 3‑21 Giao diện đăng nhập đối với xe 46](#_Toc422749332)

[Hình 3‑22 Giao diện hiển thị thông tin chuyến đi 47](#_Toc422749333)

[Hình 3‑23 Giao diện thông báo không có chuyến đi 48](#_Toc422749334)

[Hình 3‑24 Giao diện đăng nhập tài khoản quản lý 48](#_Toc422749335)

[Hình 3‑25 Giao diện lấy trạm và gửi lên Server 49](#_Toc422749336)

[Hình 3‑26 Giao diện hiển thị danh sách hành khách theo chuyến đi 49](#_Toc422749337)

[Hình 3‑27 Giao diên thông tin hành khách 50](#_Toc422749338)

[Hình 3‑28 Giao diện coi trên bản đồ chuyến đi 50](#_Toc422749339)

[Hình 3‑29 Giao diên danh sách hành khách lên xuống ở trạm 51](#_Toc422749340)

[Hình 3‑30 Giao diện thông tin hành khách ở trạm 51](#_Toc422749341)

[Hình 3‑31 Giao diện đăng ký tài khoản của hành khách 52](#_Toc422749342)

[Hình 3‑32 Giao diện đăng nhập của hành khách 52](#_Toc422749343)

[Hình 3‑33 Giao diện chuyến đi trong ngày của hành khách 53](#_Toc422749344)

[Hình 3‑34 Giao diện kiểm tra vị trí hiện tại của xe 54](#_Toc422749345)

[Hình 3‑35 Giao diện gửi ý kiến phản hồi cho nhà xe 55](#_Toc422749346)

[Hình 3‑36 Giao diện bản đồ chỉ đường tới trạm đón cho hành khách 55](#_Toc422749347)

[Hình 3‑37 Giao diện thông báo không có chuyến đi 56](#_Toc422749348)

[Hình 3‑38 Giao diện đặt vé 56](#_Toc422749349)

[Hình 3‑39 Giáo diện chọn lộ trình 57](#_Toc422749350)

[Hình 3‑40 Giao diện chọn ngày đi 57](#_Toc422749351)

[Hình 3‑41 Giao diện chọn chuyến đi 58](#_Toc422749352)

[Hình 3‑42 Giao diện bản đồ chọn trạm 59](#_Toc422749353)

[Hình 3‑43 Giao diện chọn trạm đón 59](#_Toc422749354)

[Hình 3‑44 Giáo diện chọn trạm xuống 60](#_Toc422749355)

[Hình 3‑45 Giáo diện chọn ghế 61](#_Toc422749356)

[Hình 3‑46 Giao diện xác nhận chọn ghế 61](#_Toc422749357)

[Hình 3‑47 Giao diện hiện thông tin vé đặt 62](#_Toc422749358)

[Hình 3‑48 Giao diện danh sách đơn hàng 63](#_Toc422749359)

[Hình 3‑49 Giao diện hiện thông tin đơn hàng 63](#_Toc422749360)

[Hình 4‑1 Mô hình mô phỏng chức năng đăng nhập và liệt kê chuyến đi có trong ngày 65](#_Toc422749361)

[Hình 4‑2 Mô hình mô phỏng chức năng thêm lộ trình mới 66](#_Toc422749362)

[Hình 4‑3 Mô hình mô phỏng chức năng thông báo đón trả khách 67](#_Toc422749363)

[Hình 4‑4 Mô hình mô phỏng chức năng tự báo vi phạm tốc độ 68](#_Toc422749364)

[Hình 4‑5 Mô hình chứ năng câp nhập vị trí xe 69](#_Toc422749365)

[Hình 4‑6 Mô hình mô phỏng chức năng thông báo khi xe tới 70](#_Toc422749366)

[Hình 4‑7 Mô hình mô phỏng chức năng chỉ đường tới trạm xe 71](#_Toc422749367)

[Hình 4‑8 Mô hình chứ năng quản lý của quản lý 72](#_Toc422749368)

[Hình 4‑9 Mô hình chứn ăng của Web Server 72](#_Toc422749369)

**Danh mục các từ viết tắt**

|  |  |
| --- | --- |
| **Từ viết tắt** | **Từ đầy đủ** |
| PHP | Hypertext Preprocessor |
| JSON | JavaScript Object Notation |
| CSS | Cascading Style Sheets |
| HTML | HyperText Markup Language |
| API | Application program interface |
| XML | EXtensible Markup Language |

# GIỚI THIỆU CHUNG

## Tên đề tài khóa luận

Xây dựng hệ thống quản lý xe khách.

## Mục tiêu đề tài

Hiện nay, những chuyến đi đường dài của mọi người còn những khó khăn bất cập như sau:

* Phải đặt vé qua điện thoại hoặc tới trực tiếp bến để mua vé.
* Không biết được thời gian xe gần tới bến để chuẩn bị.
* Không biết được vị trí hiện tại của xe.
* Tài xế phải nhớ những trạm đón, trả khách và đường đi.
* Việc liên hệ với hành khách khó khăn.
* Khó kiểm soát hành khách trên xe.

Xây dựng hệ thống nhằm tạo sự thuận tiện cho cả quản lý, tài xế, hành khách. Quản lý dễ dàng quản lý được thông tin sai phạm của tài xế một cách tự động. Tài xe không còn vất vả phải nhớ từng trạm đón trả khách. Khi gần đến trạm, hệ thống sẽ thông báo cho tài xế có bao nhiêu khách lên, có bao nhiêu khách xuống. Hành khách sẽ tiện lợi hơn trong việc đặt vé, chủ động hơn trong thời gian khi biết được khoảng cách hiện tại của xe đối với mình, khi gần đến trạm xuống hệ thống sẽ tự báo cho hành khách biết.

## Ý nghĩa đề tài

Với những mục tiêu cụ thể đã đặt ra như trên nhóm em hi vọng với hệ thống này sẽ giúp cho việc sử dụng và quản lý xe khách trở nên dễ dàng hơn. Sẽ không gặp các trường hợp như :

* Xe quên dừng đón khách.
* Hành khách không biết vị trí hiện tại của xe dẫn đến việc trễ xe.
* Quản lý không biết vi phạm tốc độ của xe để xử lý.

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

### Đối tượng nghiên cứu

* Các kỹ thuật lập trình trên hệ điều hành Android.
* Google Map API version 2 và 3.
* Hệ quản trị cơ sở dữ liệu MySQL.
* Tìm hiểu về Web Server sử dụng ngôn ngữ lập trình php.
* Ngôn ngữ lập trình Java.
* Tìm hiểu về Codeigniter framework và Bootstrap framework.
* Sử dụng Json trong việc truyền dữ liệu.

### 

### Phạm vi nghiên cứu

#### Đối với các kỹ thuật lập trình trên hệ điều hành Android:

* Nghiên cứu các khái niệm, kỹ thuật tạo các Activiti và Service có thể hoạt động trên nhiều phiên bản Android, không đi sau vào kiến trúc hạ tầng hay kernel của Android.
* Nghiên cứu việc liên kết cơ sở dữ liệu MySQL của Web Server.
* Nghiên cứu việc kết nối, truyền và lấy dữ liệu từ Server.
* Nghiên cứu thiết kế giao diện trên Android.

#### Đối với Google Map API

* Nghiên cứu Google API hai phiên bản dành cho Android và trên WEB.
* Nghiên cứu cách đăng ký và sử dụng các dịch vụ trên Google API.
* Nghiên cứu xác định vị tris hiện tại thông qua GPS hoặc mạng.
* Nghiên cứu việc vẽ đường đi qua nhiều điểm trên Google Map.
* Nghiên cứu cách đánh dấu các vị trí trên bản độ.

#### Đối với PHP, MySQL

* Nghiên cứu cách tạo Web Server , tạo giao diện để thao tác với cơ sở dữ liệu.
* Nghiên cứu cách giao tiếp, trao đổi dữ liệu với Android.
* Nghiên cứu về Codeigniter framework.
* Nghiên cứu về Bootstrap framework.

#### Đối với dữ liệu các trạm dừng xe

* Dựa theo các lộ trình của trạm xe bus trong thành phố Hồ Chí Minh.

### Phương pháp nghiên cứu

Nghiên cứu dựa trên các các tài liệu về lập trình Android, lập trình Web, khảo sát và phân tích hiện trạng thực tế để xây dựng hệ thống.

### Thời gian nghiên cứu

* Thời gian bắt đầu : ngày 19 tháng 1 năm 2015.
* Thời gian kết thúc : ngày 21 tháng 6 năm 2015.

### Kế hoạch thực hiện đề tài

Bước 1: Tìm hiểu những khó khăn bất cập trong xã hội hiện nay.

Bước 2: Lựa chọn đề tài từ những bất cập trong quản lý xe khách.

Bước 3: Đưa ra và phân tích các vấn đề cần giải quyết.

Bước 4: Nghiên cứu các kỹ thuật xử lý.

Bước 5: Bắt đầu xây dựng hệ thống.

Bước 6: Tạo dữ liệu cho hệ thống.

Bước 7: Kiểm tra hoạt động và đưa vào nghiệm thu.

### Quá trình cụ thể thực hiện kế hoạch

|  |  |  |
| --- | --- | --- |
| Thời gian | Kế hoạch | Chi tiết |
| Từ 19/1 đến 19/3 | Nghiên cứu, tìm hiểu các kỹ thuật lập trình Android và Web | Tìm hiểu ngôn ngữ Java, PHP. Tạo những ứng dụng đơn giản để thử nghiệm. |
| Từ 20/3 đến 20/4 | Nghiên cứu về Google Map API | Nghiên Cứu Google Map API trên Android và trên Web.  Xác định vị trí hiện tại.  Đánh dấu các điểm trên bản đồ.  Vẽ đường đi giữa 2 hoặc nhiều điểm.  Đo tốc độ , khoảng cách. |
| Từ 21/4 đến 21/5 | Xây dựng ứng dụng | Bắt đầu lập trình trên Android và Web |
| Từ 22/5 đến 30/5 | Thiết kế và thu thập cơ sở dữ liệu | Thiết kế cơ sở dữ liệu  Nhập dữ liệu vào cơ sở dữ liệu |
| Từ 1/6 đến 6/6 | Kết nối, đồng bộ hóa giữa ứng dụng Android với Sererver | Kết nối ứng dụng Android với Server, đồng bộ hóa cở sở dữ liệu. |
| Từ 7/6 đến 15/6 | Kiểm tra lại hệ thống, sửa những lỗi phát sinh | Kiểm tra các ứng dụng và web.  Sửa các lỗi phát sinh. |
| Từ 15/6 đến 20/6 | Hoạt động và nghiệm thu hệ thống | Đưa hệ thống chạy thử và nghiệm thu kết quả thu được |

Bảng 1‑1 Quá trình chi tiết thực hiện đề tài

# NHỮNG KIẾN THỨC CƠ BẢN VỀ ĐỀ TÀI

## Hệ điều hành Android

### Tổng quát về Android

Android là một hệ điều hành dựa trên nền tảng Linux được thiết kế dành cho các thiết bị di độngcó màn hình cảm ứng như điện thoại thông minh và máy tính bảng. Ban đầu, Android được phát triển bởi Tổng công ty Android, với sự hỗ trợ tài chính từ Google và sau này được chính Google mua lại vào năm 2005. Android ra mắt vào năm 2007 cùng với tuyên bố thành lập Liên minh thiết bị cầm tay mở: một hiệp hội gồm các công ty phần cứng, phần mềm, và viễn thông với mục tiêu đẩy mạnh các tiêu chuẩn mởcho các thiết bị di động. Chiếc điện thoại đầu tiên chạy Android được bán vào tháng 10 năm 2008.

Android có mã nguồn mở và Google phát hành mã nguồn. Chính mã nguồn mở cùng với một giấy phép không có nhiều ràng buộc đã cho phép các nhà phát triển thiết bị, mạng di động và các lập trình viên nhiệt huyết được điều chỉnh và phân phối Android một cách tự do. Ngoài ra, Android còn có một cộng đồng lập trình viên đông đảo chuyên viết các ứng dụng để mở rộng chức năng của thiết bị, bằng một loại ngôn ngữ lập trình Java có sửa đổi.

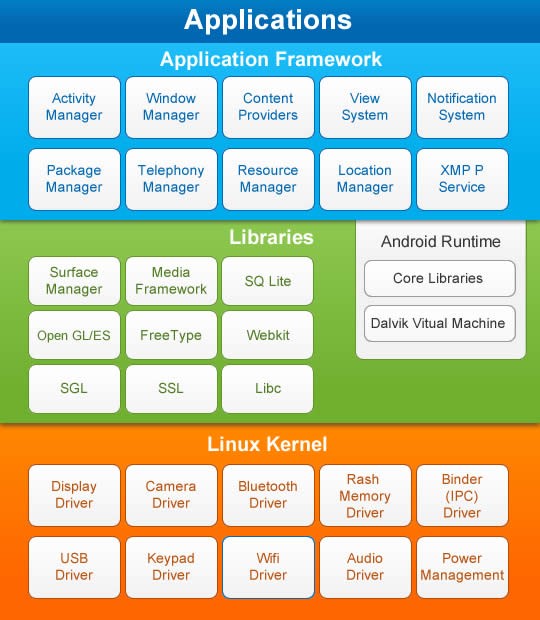
Những yếu tố này đã giúp Android trở thành nền tảng điện thoại thông minh phổ biến nhất thế giới . Android được nhiều công ty lựa chọn khi họ cần một hệ điều hành không nặng nề, có khả năng tinh chỉnh, và giá rẻ chạy trên các thiết bị công nghệ cao thay vì tạo dựng từ đầu. Kết quả là mặc dù được thiết kế để chạy trên điện thoại và máy tính bảng, Android đã xuất hiện trên TV, máy chơi game và các thiết bị điện tử khác. Bản chất mở của Android cũng khích lệ một đội ngũ đông đảo lập trình viên và những người đam mê sử dụng mã nguồn mở để tạo ra những dự án do cộng đồng quản lý.

### Phát triển ứng dụng trên Android



#### Cấu trúc của các tầng phần mềm trên Android

Cấu trúc của Android được chia theo tầng các phần mềm bao gồm tần ứng dụng (application Layer), ứng dụng khung (Application Framework), các thư việc (Libraries), Android Runtime và nhân Linux (Linux Kernel ).



Hình 2‑1Cấu trúc của các tầng phần mềm trên Android

**Linux Kernel: Lớp Nhân Linux**

Toàn bộ HĐH Android được xây dựng bên trên nhân Linux 2.6 với vài kiến trúc thay đổi bởi Google. Lớp này chịu trách nhiệm giao tiếp với các phần cứng và chứa hầu hết các trình điều khiển thiết bị phần cứng. Trình điều khiển thiết bị là những chương trình điều khiển giao tiếp với phần cứng. Ví dụ, chúng ta xem xét về chức năng Bluetooth. Hầu hết các thiết bị đều có phần cứng Bluetooth bên trong. Vì thế nhân của thiết bị đó phải có trình điều khiển Bluetooth để giao tiếp với phần cứng Bluetooth. Nhân Linux còn đóng vai trò như là lớp đối tượng giữa phần cứng và các lớp phần mềm khác. HĐH Android sử dụng Linux cho hầu hết những chức năng cốt lõi của nó như quản lý bộ nhớ, quản lý các tiến trình, liên kết mạng, các thiết lập bảo mật ,v.v…Android được xây dựng trên nền tảng đã được chứng minh và phổ biến, tương thích với nhiều loại phần cứng khác nhau.

**Libraries: Lớp Thư viện**

Lớp này chứa toàn bộ các thư viện chuẩn của Android. Nó cho phép thiết bị có thể thao tác với nhiều loại dữ liệu khác nhau. Các thư viện ở lớp này được viết bằng ngôn ngữ c hoặc là C++ đặc thù riêng cho phần cứng.

Surface Manager: Quản lý hiển thị nội dung màn hình.

Media framework: Nền tảng đa phương tiện, nó cung cấp các chuẩn mã hóa khác nhau, giúp có thể chơi và ghi âm của các loại đa phương tiện khác nhau.

SQLite: Đây là một engine database được sử dụng trong Android cho các mục đích lưu trữ dữ liệu lên thiết bị. Có thể hiểu nó là một tập hợp các thư viện cho phép thao tác với dữ liệu như ghi, xóa, tìm kiếm v.v…

WebKit: Đây là engine về trình duyệt được sử dụng để hiện thị nôi dung HTML.Ví dụ trình duyệt Chrome cài sẵn trên thiết bị sẽ sử dụng các gói thư viện này để hiển thị nôi dung trang web.

OpenGL: Open Graphics Library, là tư viện OpenGL được sử dụng để tạo ra các nội dung hiển thị đồ họa 2D, 3D lên màn hình.

Android Runtime: Bộ thực thi Android này bao gồm máy ảo Dalvik và các thư viện java core.

Davik Viturl Machine: Máy ảo Davik, đây là một loại máy ảo java được sử dụng trong các thiết bị android để thực thi các ứng dụng. Nó được tối ưu hóa cho các xử lý sử dụng nguồn điện và bộ nhớ nhỏ. Không giống như máy ảo Java. Máy ảo Dalvik sẽ thực thi các file .dex chứ không thực thi những file .class. File .dex được build từ file .class tại thời điểm biên dịch, nó hiệu quả hơn ở các môi trường có tài nguyên hạn hẹp. Máy ảo Dalvik cho phép nhiều đối tượng máy ảo được tạo ra một cách đồng thời để quản lý bộ nhớ, bảo mật và hỗ trợ các tiến trình. Tác giả của nó là Dan Bornstein, một nhân viên của Google.

Core Java Libraries: Thư viện Java nền tảng, các thư viện java core này khác với các thư viện Java SE, và Java ME. Tuy nhiên hầu hết nó cung cấp đầy đủ các function được định nghĩa ở các thư viện Java SE.

**Application Framework: Lớp nền tảng ứng dụng**

Đây là những thành phần mà các ứng dụng của chúng ta trực tiếp giao tiếp đến. Những chương trình ở lớp này quản lý những chức năng cơ ản của chiếc điện thoại như :

Quản lý resource, quản lý cuộc gọi, nhắn tin, …Ở khía cạnh lập trình viên, chúng ta hãy xem nó là các công cụ cơ bản giúp chúng xây dựng các ứng dụng.

Những thành phần quan trọng của nền tảng ứng dụng:

* **Activity Manager**: Quản lý vòng đời activity của ứng dụng: Activity được hiểu là một hành động, một sự việc mà người dùng có thể thực hiện trong ứng dụng Android. Activity có nhiệm vụ tạo ra một cửa sổ, để người dùng đặt vào đó một giao diện, và các xử lý liên quan.
* **Content Providers**: Quản lý việc chia sẻ dữ liệu giữa các ứng dụng.  
  Là nơi lưu trữ và cung cấp cách truy cập dữ liệu do các ứng dụng tạo nên
* **Telephony Manager**: Quản lý điện thoại có nhiệm vụ quản lý toàn bộ các cuộc gọi. Nếu ứng dụng của chúng ta muốn truy cập và can thiệp vào các cuộc gọi thì phải sử dụng thành phần này.
* **Location Manager**: Quản lý địa điểm, vị trí sử dụng chức năng định vị toàn cầu GPS
* **Resource Manager**: Quản lý các loại tài nguyên được sử dụng trong ứng dụng.

**Applications: Lớp ứng dụng**

Lớp ứng dụng ở trên cùng trong mô hình kiến thúc phân tầng của HDH Android. Đây là chính là lớp chứa các ứng dụng của chúng ta, nó bao gồm cả những ứng dụng mặc định được cài sẵn trên thiết bị như ứng dụng như trình duyệt web, danh bạ điện thoại, calculator,…

#### Kiến trúc hệ điều hành Android

Kiến trúc của Android khuyến khích khái niệm thàng phần sử dụng lại, cho phép công bố và chia sẻ các Activity, Service, dữ liệu, các ứng dụng khác với quyền truy cập được quản lý bởi khai báo.

Cơ chế đó cho phép người lập trình tạo ra một trình quản lý danh bạ hoặc trình quay số điện thoại mà có các thành phâng người khác có thể tạo mới với giao diện và mở rộng chứ năng thay vì tạo lại chúng.

Những dịch vụ sau là những dịch vụ kiến trúc cơ bản nhất của tất cả các ứng dụng, cung cấp một framework cho mọi phần mềm được xây dựng:

* **Activity Manager**: Điều khiển vòng đời của các Activity bao gồm cả quản lý các tầng Activity.
* **View:** Được sử dụng để tạo lập cái giao diện người dung cho các Activity.
* **Notification Manager** : Cung cấp một cơ chế cố định và quy củ cho việc gửi các thông báo đến người dung
* **Content Provider** : Cho phép ứng dụng chia sẻ dữ liệu giữa các ứng dụng.
* **Resource Manager** : Hỗ trợ các thành phần không thuộc mã nguồn như là chuỗi ký tự, đồ họa được đặt bên ngoài.

#### Các thư viện trên Android

Android cung cấp một số cái APIs cho phát triển ứng dụng. Danh sách các API cơ bản sau được cung cấp bởi tất cả các thiết bị trên nền Android:

* **android.util**: Gói tiện ích chơ bản bao gồm nhiều lớp mức thấp như là các lớp quản lý (List, Stack …), lớp xử lý chuỗi, lớp xữ lý XML
* **android.os**: Gói hệ điều hành cung cấp truy cập đến các dịch vụ cơ bản như là chuyển tin nhắn, thông tin chéo, đồng hồ và gỡ rối.
* **android.graphics**: Cung cấp các lớp đồ họa mức thấp thực hiện các chức năng đồ họa, màu, vẽ cơ bản.
* **android.text**: Công cụ hiển thị và xử lý văn bản.
* **android.database** : Cung cấp các lớp mức thấp bắt buộc cho việc điều khiển cursor khi làm việc với các cơ sở dữ liệu.
* **android.content**: Các giao tiếp lập trình nội dung được dung để quản lý truy cập dữ liệu và xuất văn bản bằng cách cung cấp các dịch vụ thao tác với tài nguyên, Content Provider, và các gói.
* **android.view**: View là lớp giao diện người dung cơ bản nhất. Tất cả giao diện người dung được tạo ra đều phải sử dụng một tập các View để cung cấp cho các thành phần tương tác người dung.
* **android.widget**: Xây dựng dựa trên gói View. Những lớp Widget những thành phần giao diện được tao sẵn được sử dụng để tạo nên giao diện người dùng. Các Widget bao gồm danh sách, nút bấm, hộp nhập, các kiểu trình bày (layout).
* **com.google.android.maps**: Bộ API mức cao cung cấp truy cập đến điều khiển bản đồ sẵn trong android từ ứng dụng được xây dựng. Bao gồm cả lớp MapView cũng như Overlay và MapController để tương tác với bản đồ bên trong ứng dụng.
* **android.app**: Một gói thư viện bậc cao, cung cấp truy cập đến dữ liệu của ứng dụng. Gói ứng dụng cũng bao gồm các lớp Activity và Service là thành phần cơ bản của mọi ứng dụng Android.
* **android.provider**: Để tạo thuận lợi cho người phát triển truy cập đến các Content Provider tiêu chuẩn ( như là dữ liệu danh bạ), gói Cung cấp (Provider) bao gồm các lớp cho phép truy cập đến cơ sở dữ liệu chuẩn trong tất cả các bản phân phối Android.
* **android.telephony**: Các API điện đàm cung cấp khả năng tương tác trực tiếp với tầng điện thoiaj trong các thiết bị, cho phép tạo, nhân, theo dõi các cuộc gọi, tình trạng các cuộc gọi và tin nhắn.
* **android.webkid**: GÓi Webkit cung cấp các API để làm việc với các nội dung Web-based bao gồm một lớp WebView để tạo ra giao diện Web, nhúng trong ứng dụng và một trình quản lý cookie.

Cùng với các API của Android, còn có một tạp các thư viện C/C++ như:

* **OpenGL**: Thư viện dùng để tạo ra các đồ họa 3D dựa vào chuẩn OpenGLES 1.0 API.
* **Freetype**: Hỗ trợ xử lý bitmap và font vector.
* **GGL**: Thử việc cơ bản, dùng để cung cấp các engine đồ họa 2D.
* **Libc**: Thư viện C chuẩn, được tối ưu cho các thiết bị Linux-based.
* **SQLite**: Engine cơ sở dữ liệu quan hệ gọn nhẹ, dùng để lưu trữ dữ liệu của ứng dụng.
* **SSL**: Hỗ trợ sử dụng giao thứ mã hóa Secure Sockets Layer trong bảo mật truyền thông Internet.

Ngoài các thư viên chuẩn của Android, để đáp ứng tiêu chí phù hợp với nhiều thiết bị khác nhau, Android còn có thể có các API phụ thuộc thiết bị như **android.location**, **android.media**, **android.opengl**, **android.hardware**, **android.net.wifi**, **android.bluetooth**, **android.telephony**.

#### Các thành phần cơ bản của ứng dụng Android

##### **Activity**

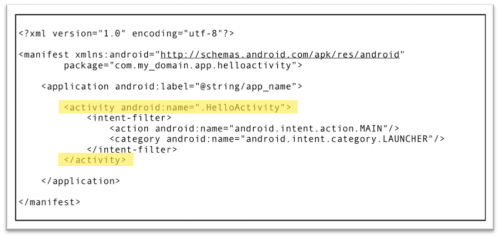
**Định nghĩa Activity**

Là thành phần tối quan trọng của bất kỳ một ứng dụng Android nào. Thuật ngữ Activity chỉ một việc mà người dùng có thể thực hiện trong một ứng dụng Android. Do gần như mọi activity đều tương tác với người dùng, lớp Activity đảm nhận việc tạo ra một cửa sổ (window) để người lập trình đặt lên đó một giao diện UI với setContentView(View). Một activity có thể mang nhiều dạng khác nhau: Một cửa sổ toàn màn hình (full screen window), một cửa sổ floating (với windowsIsFloating) hay nằm lồng bên trong 1 activity khác (với ActivityGroup ).

Có 2 phương thức mà gần như mọi lớp con của Activity đều phải hiện thực:

* **onCreate(Bundle)** - Nơi khởi tạo activity. Quan trọng hơn, đây chính người lập trình gọi setContentView(int) kèm theo layout để thể hiện UI của riêng mình. Đồng thời còn có findViewById(int) giúp gọi các widget (buttons, text boxes, labels,..) để dùng trong UI.
* **onPause()** - Nơi giải quyết sự kiện người dùng rời khỏi activity. Mọi dữ liệu được người dùng tạo ra tới thời điểm này cần phải được lưu vào ContentProvider.

Để có thể sử dụng Context.startActivity(), mọi lớp activity đều phải được khai báo với tag <activity> trong file AndroidManifest.xml.



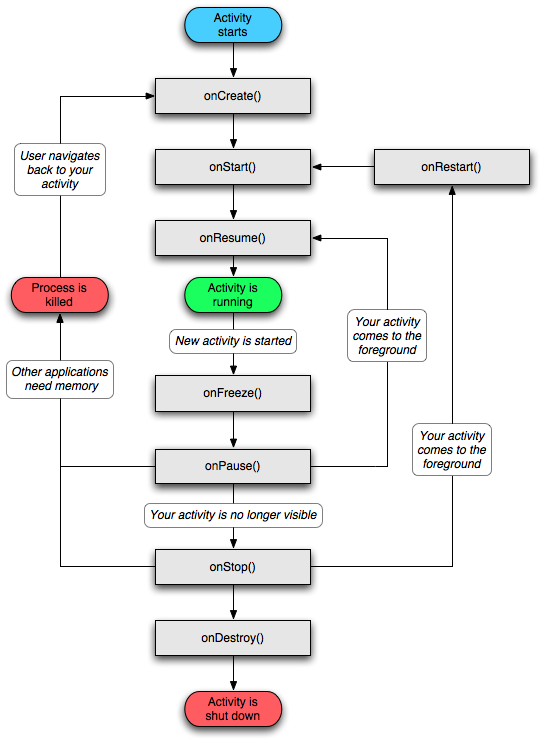
Hình 2‑2 Khai báo tab <activity> trong Android Manifest.xml

**Vòng đời của một Activity**

Các activity được quản lí dưới dạng các activity stack - First-In-Last-Out: Khi một activity mới được khởi tạo, nó sẽ được đưa lên trên cùng stack, các activity khác muốn chạy trên nền (foreground) trở lại thì cần phải chờ tới khi Activity mới này kết thúc.

Một Activity có 4 trạng thái:

* **Active**hay **Running:** Khi một activity đang chạy trên màn hình.
* **Paused:** Khi một activity vẫn đang chạy trên màn hình nhưng đang bị một activity trong suốt (transparent) hay không chiếm toàn màn hình hiển thị phía trên. Tuy vẫn lưu trữ dữ liệu, nhưng các **paused activity** này sẽ bị hệ thống bắt chấm dứt khi đang thiếu bộ nhớ trầm trọng.
* **Stopped:** Khi 1 activity bị che khuất hoàn toàn bởi 1 activity khác. Tuy vẫn lưu trữ dữ liệu, nhưng các stopped activity này sẽ thường xuyên bị hệ thống bắt chấm dứt để dành chỗ cho các tiến trình khác.
* **Killed hay Shut down:** Khi 1 activity đang paused hay stopped, hệ thống sẽ xóa activity ấy ra khỏi bộ nhớ.



Hình 2‑3 Lược đồ vòng đời của một Activity

Dựa vào lược đồ trên, thấy được có 3 vòng lặp quan trọng sau:

* **Vòng đời toàn diện (Entire Lifetime):** Diễn ra từ lần gọi onCreate(Bundle) đầu tiên và kéo dài tới lần gọi onDestroy() cuối cùng.
* **Vòng đời thấy được (Visible Lifetime):** Diễn ra từ khi gọi onStart() và kéo dài tới khi gọionStop(). Ở vòng đời này, activity được hiển thị trên màn hinh nhưng có thế không tương tác với người dùng ở trên nền. Các phương thức onStart(0 và onStop() có thể được gọi nhiều lần.
* **Vòng đời trên nền (Foreground Lifetime):** Diễn ra từ khi gọi onResume(0 và kéo dài tới khi gọi onPause(). Ở vòng đời này, activity nằm trên mọi activity khác và tương tác được với người dùng. 1 activity có thể liên tục thay đổi giữa 2 trạng thái paused và resumed, chẳng hạn khi thiết bị sleep hay 1 intent mới được đưa tới.

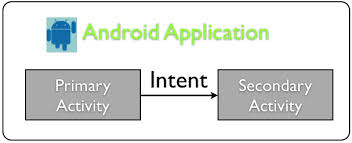
##### **Intent**

**Định nghĩa Intent**

Intent là một mô tả trừu tượng của một hành động được thực thi. Nó đại diện cho một hành động đi kèm với một ngữ cảnh xác định. Với Intent thì có nhiều hành động và nhiều component (Một thể hiện của một class java dùng để thực thi các hành động được đặc tả trong Intent) dành cho Intent của Android hơn là so với HTTP verbs (POST, GET) và nguồn tài nguyên (hình ảnh, web page) của giao thức HTTP, tuy nhiên khái niệm vẫn tương tự nhau.

Intent được sử dụng với phương thức startActivity() để mở một Activity, và dùng với broadcastIntent để gởi nó đến bất kì BroadcastReceiver liên quan nào, và dùng với [startService(Intent)](http://developer.android.com/reference/android/content/Context.html#startService%28android.content.Intent%29), bindService(Intent, ServiceConnection, int) để giao tiếp với các Service chạy dưới nền.

Intent cung cấp một chức năng cho phép kết nối hai chương trình khác nhau trong quá trình thực thi (runtime) (Cung cấp khả năng cho phép hai chương trình khác nhau giao tiếp với nhau). Chức năng quan trọng và được sử dụng nhiều nhất của một Intent  là mở một Activity, nơi mà nó có thểđược dùng như một vật kết nối các Activity lại với nhau (Truyền thông tin giữa hai Activity khác nhau).



Hình 2‑4 Sử dụng Intent để trao đổi thông tin giữa hai Activity

**Thành phần chính của Intent bao gồm:**

* **Action:** Xác định hành động sẽ được thực thi, các hành động này có thể là: ACTION\_VIEW, ACTION\_EDIT, ACTION\_MAIN…
* **Data:** Các dữ liệu được sử dụng để hành động (Action) thao tác trên nó, như bản ghi về một người nào đó trong cơ sở dữ liệu chứa các contact của
* Ngoài ra, Intent còn có các thuộc tính phụ sau:
* **Category:** thông tin chi tiết về hành động được thực thi, ví dụ như **CATEGORY\_LAUNCHER** có nghĩa là nó sẽ xuất hiện trong Launcher như ứng dụng có mức level cao (top-level application),  trong khi **CATEGORY\_ALTERNATIVE** chứa thông tin danh sách các hành động thay thế mà người dùng có thể thực thi trên phần dữ liệu mà **Intent** cung cấp.
* **Type:** Chỉ định 1 kiểudữ liệu chính xác (kiểu MIME) được mang bởi intent. Thường thì type được suy ra từ chính dữ liệu. Bằng cách thiết lập thuộc tính type, bạn có thể vô hiệu hóa sự phỏng đoán kiểu dữ liệu và chỉ định rỏ một kiểu chính xác.
* **Component:** Chỉ định rõ tên của lớp thành phần (Một thể hiện của một class java dùng để thực thi các hành động được đặc tả trong **Intent**) sử dụng cho Intent . Thông thường thì nó được xác định bằng cách tìm kiếm thông tin trong **Intent**(Các thông tin như Action, data/type, và category) và nối chúngvới các component (Một thể hiện của một Class java dùng để thực thi các hành động được đặc tả trong **Intent**) có thể đáp ứng được các yêu cầu sử lý của **Intent**.
* **Extracts:** là một đối tượng **Bundle** dùng để chứa các thông tin kèm theo được dùng để cung cấp thông tin cần thiết cho component. Ví dụ: Nếu có một Intent đặc tả hành động send email thì những thông tin cần chứa trong đối tượng **Bundle** là subject, body…

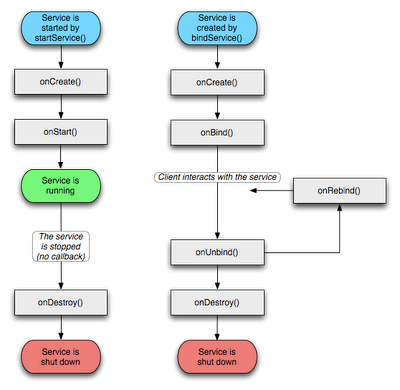
##### **Service**

**Định nghĩa Service**

Một service là một thành phần của ứng dụng, thể hiện mong muốn ứng  dụng thực hiện các hành động trong khi không tương tác với người dùng hoặc cung cấp chức năng cho các ứng dụng khác sử dụng. Nói một cách đơn giản, service là các tác vụ (task) chạy ngầm dưới hệ thống nhằm thực hiện một nhiệm vụ nào đó.  Mỗi class Service phải chứa thẻ <service> được khai báo trong file **AndroidManifext.xml**. Services có thể được bắt đầu bởi Context.startService() và Context.bindservice()

Cũng như các đối tượng của ứng dụng khác, services chạy trên luồng (Thread) của tiến trình chính. Có nghĩa là nếu service của bạn dự định chạy các hành vi có cường độ lớn hoặc các hành vi có thể gây nghẽn mạch, nó sẽ tự sinh ra luồng (Thread) của chính nó để làm thực hiện các tác vụ được đặc tả trong Service.

**Vòng đời của một Service**

****

Hình 2‑5 Lược đồ vòng đơi của một Service

##### **Content Provider**

Là nơi lưu trữ và cung cấp cách truy cập dữ liệu do các ứng dụng tạo nên. Đây là cách duy nhất mà các ứng dụng có thể chia sẻ dữ liệu của nhau.

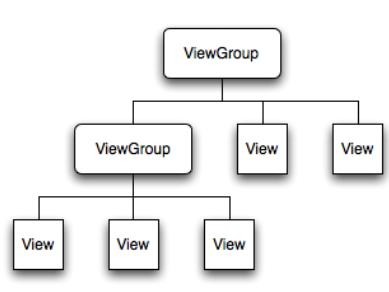
Android cung cấp sẵn content providers cho 1 số kiểu dữ liệu thông dụng như âm thanh, video, thông tin sổ điện thoại,... (Xem thêm trong gói android.provider). Người lập trình cũng có thể tự tạo ra các class con (subclass) của Content Provider để lưu trữ kiểu dữ liệu của riêng mình.

##### **View**

View là thành phần cơ bản để xây dựng giao diện người dùng cho một ứng dụng Android. View là 1 lớp căn bản của widgets (widgets được dùng để tạo các nút nhấn, text fields, …).

Tất cả mọi widget đều có chung các thuộc tính cơ bản như là cách trình bày vị trí, background, kích thước, lề,… Tất cả những thuộc tính chung này được thể hiện hết ở trong đối tượng View.

Trong Android Platform, các screen luôn được bố trí theo một kiểu cấu trúc phân cấp như hình dưới. Một màn hình là một tập hợp các Layout và các widget được bố trí có thứ tự. Để thể hiện một màn hình thì trong hàm onCreate của mỗi Activity cần phải được gọi một hàm là setContentView(R.layout.main); hàm này sẽ load giao diện từ file XML lên để phân tích thành mã bytecode.



Hình 2‑6 Cấu trúc các Screen trong View

##### **BroadcastReceiver**

Là lớp nhận về các intents được gửi bởi sendBroadcast(). Có thể tạo instance cho lớp này bằng 2 cách: hoặcvới Context.registerReceiver() hay thông qua tag <receiver> trong file AndroidManifest.xml.

Có hai lớp broadcast quan trọng cần xét đến:

* Broadcasts bình thường (Normal broadcasts): Được gửi bởi  context.sendBroadcast tới các receivers hoàn toàn không theo 1 trật tự nào, dẫn tới đôi khi tồn tại cùng 1 lúc nhiều receivers khác nhau. Tuy nhanh nhưng không thể trao đổi kết quả nhận về giữa cac1 receivers với nhau.
* Broadcasts có thứ tự (Ordered broadcasts) - Được gửi bởi Context.sendOrderedBroadcast() tới từng receiver một. Như vậy, kết quả nhận về của 1 receiver có thể trở thành input của 1 receiver khác, hay có thể ngay lập tức dừng việc broadcast tới các receiver khác. Trật tự chạy các receivers được quyết định bởi thuộc tính android:priority trong intent-filter tương ứng.

##### **File Manifest**

**File Manifest** là nơi ứng dụng của bạn sẽ trình bày với hệ thống rằng *“có những gì trong ứng dụng này”*, như:

* Đăng ký các *user permission*, tạm hiểu là các quyền hạn mà người dùng xin phép để sử dụng các tính năng của hệ thống, như là quyền được truy cập Interner, được đọc danh bạ (contacts), được ghi file ra thẻ nhớ ngoài,…
* Chỉ định *Minimum Required SDK*, là hệ điều hành thấp nhất của Android mà ứng dụng của ta hỗ trợ, thông số này được tạo ra khi chúng ta tạo một ứng dụng mới ở Bài 2, khi đó chúng ta đã chỉ ra *Minimum Requied SDK* và thông số chúng ta chỉ định đó sẽ được lưu vào thuộc tính này của *file Manifest*.
* Chỉ định các tính năng phần cứng và phần mềm mà ứng dụng của ta muốn sử dụng đến, như camera, bluetooth, hay ngay cả tính năng multitouch.
* Một số các thư viện bên ngoài (bên ngoài ở đây có thể hiểu là các thư viện không có sẵn trong các API Framework của chúng ta) mà ứng dụng của ta có sử dụng đến, như thư viện Google Maps, Google Play,…

##### **Resources**

Một ứng dụng không thể thiếu các **resource**, đó là các hình ảnh, font chữ, màu sắc, âm thanh… dùng để tăng thêm sự phong phú và tính thẩm mỹ cho ứng dụng.

Thông thường khái niệm resource trong Android được hiểu như là một cái gì đó tách biệt với source code, source code là nơi chứa đựng java code và các logic của ứng dụng, trong khi resource là những gì còn lại của ứng dụng không phải là source code.

Resource được chứa đựng trong thư mục **res/** (bạn có thể nhìn vào Package Explorer để thấy thư mục Resource này), trong thư mục này còn có rất nhiều thư mục khác giúp nhóm các resource theo loại và theo các tính năng mà resource đó hỗ trợ.

Cụ thể từng loại và tính năng của resource được nói rõ hơn ở bài sau, ở đây ta chỉ nhắc đến 2 loại resource lớn:

* **Default resources**: là tất cả các resource mà chúng ta không cần quan tâm đến việc hiển thị khác nhau như thế nào trong các thiết bị, resource này sẽ hiển thị default như những gì nó được cung cấp ở tất cả các thiết bị.
* **Alternative resources**: là các resosurce sẽ phụ thuộc vào cụ thể thiết bị mà ta đã chỉ định, chẳng hạn ta muốn ứng dụng chạy trên thiết bị có ngôn ngữ mặc định là tiếng Việt sẽ phải load các text tiếng Việt, hình ảnh cờ Việt,.. còn nếu ứng dụng chạy trên thiết bị có ngôn ngữ mặc định là tiếng Anh sẽ load các text tiếng Anh, hình ảnh cờ Anh,…

## Google Maps API

### Khái niệm Google Maps API

API Google Maps là một bộ sưu tập API cho phép nhà phát triển chồng dữ liệu của riêng họ trên một bản đồ Google tùy chỉnh. Nhà phát triển có thể tạo các ứng dụng lôi cuốn trên web và di động với nền tảng lập bản đồ mạnh mẽ của Google, bao gồm hình ảnh vệ tinh, chế độ xem phố, cấu hình độ cao, chỉ đường lái xe, bản đồ theo kiểu, nhân khẩu học, phân tích và một cơ sở dữ liệu mở rộng về địa điểm. Với độ phủ sóng toàn cầu chính xác nhất thế giới và một cộng đồng lập bản đồ đang hoạt động và tạo các bản cập nhật hằng ngày, người dùng của bạn sẽ được hưởng lợi từ một dịch vụ liên tục được cải thiện.

### Ích lợi khi sử dụng Google Maps API

Google Maps là dịch vụ lập bản đồ trực tuyến được sử dụng rộng rãi nhất trên thế giới, với hơn 800.000 trang web sử dụng API Google Maps và hơn 250 triệu người dùng đang hoạt động chỉ tính riêng trên các thiết bị di động. Với API Google Maps, nhà phát triển có thể cung cấp trải nghiệm quen thuộc này cho người dùng của riêng họ.

Danh mục toàn cầu của ảnh chụp trên không với độ phân giải cao được Google Earth cung cấp cũng có sẵn trong API Google Maps. API Google Maps bao gồm quyền truy cập vào hình ảnh 45 độ ở hơn 120 thành phố trên toàn thế giới, vì vậy người dùng có thể khám phá dữ liệu với một góc nhìn độc đáo trong một khu vực cụ thể.

Điện thoại thông minh đã cách mạng hóa khả năng tiêu thụ và chia sẻ thông tin và phục vụ cho một cơ sở người dùng di động trở nên quan trọng hơn bao giờ hết. Với API Google Maps, nhà phát triển có thể xây dựng ứng dụng dựa trên vị trí có thể được sử dụng thông qua một trình duyệt web, thiết bị di động hoặc các ứng dụng được xây dựng tùy chỉnh.

### Các chức năng chính trong Google Maps API

#### Google Maps API Web Services

Google Maps API Web Services là một tập các giao diện HTTP cung cấp thông tin địa lý cho ứng dụng.

Google Maps API Web Services bao gồm:

* Geocoding API.
* Directions API.
* Distance Matrix API.
* Elevation API.
* Place API.
* **Google Geocoding API**

Mã hóa địa lý là tiến trình chuyển địa chỉ (ví dụ: số 2, đường Nguyễn Đình Chiểu, Vĩnh Thọ, Nha Trang) sang tọa độ địa lý (ví dụ: kinh độ 12.268491,vĩ độ 109.202148) để đánh dấu trên bảng đồ. Google Geocoding API cung cấp một phương thức trực tiếp để truy cập bộ mã hóa địa lý thông qua giao thức HTTP. Ngoài ra dịch vụ Google Geocoding API còn cho phép giải mã từ tọa độ sang địa chỉ.

Google Geocoding API giới hạn ở 2500 truy vấn mỗi ngày. Nếu trả phí con số này tăng lên 100000. Các giới hạn này được đặt ra để ngăn chặn sự lạm dụng dịch vụ.

Nếu vượt quá giới hạn này ứng dụng sẽ không thể truy cập được sau 24 giờ, nếu tiếp tục vượt ở 24 giờ tiếp theo thì dịch dụ Geocoding API có thể sẽ chặn ứng dụng.

Sử dụng Geocoding API cần phải kết hợp với một bản đồ Google Map dùng để hiển thị các địa điểm một cách trực quan trên bản đồ.

* **Google Direction API**

Google Direction API là dịch vụ tính toán đường đi giữa hai điểm sử dụng HTTP request. Thông tin đầu vào của điểm đầu, điểm kết thúc, điểm trung gian có thể ở dạng tên xác định địa điểm hay giá trị kinh độ và vĩ độ. Direction API có thể trả về nhiều hướng đi khác nhau sử dụng một tập các điểm trung gian.

Dịch vụ này được thiết kế để tính toán đường đi giữa những vị trí xác định rõ thông tin trên bản đồ Google Map. Vì quá trình tính toán đường đi mất thời gian và phức tạp (phụ thuộc vào khoảng cách hai điểm) nên bất cứ khi nào có thể nên thực hiện công việc tính toán trước và lưu trữ tạm thời kết quả nhận được.

Có thể truyền vào một số tham số giá trị địa chỉ hoặc kinh độ - vĩ độ. Nếu truyềnvào địa chỉ, hệ thống Google Direction sẽ chuyển nó sang giá trị kinh độ - vĩ độ để tính toán.

Giới hạn sử dụng: Google Direction API cho phép 2500 truy vấn mỗi ngày. Mỗi lần tính toán hướng đi có thể sử dụng tối đa 9 điểm trung gian. Nếu có nhu cầu cao hơn, có thể sử dụng Google Direction API cho doanh nghiệp với tối đa 100000 truy vấn mỗi ngày và 23 điểm trung gian ở mỗi truy vấn.

Ngoài ra mỗi chuỗi URL truy vấn được giới hạn ở 2048 ký tự trước khi mã hóa. Mội số truy vấn tìm đường có thể chứa thông tin các điểm dọc theo đường đi vì vậy cần lưu ý độ dài của nó.

Kết quả trả về của Google Direction API phải được sử dụng kết hợp với bản đồ Goolge Map. Tức là thông tin phải kết hợp với hiển thị đường đi trên bản đồ. Không nên sử dụng thông tin trả về mà không dùng Google Map.

* Google Distance Matrix API

Google Distance Matrix API là một dịch vụ cung cấp khoảng cách và thời gian di chuyển cho một ma trận các điểm gốc và điểm đích. Thông tin trả về phụ thuộc vào đường đi được đề nghị giữa điểm gốc và điểm đích theo tính toán của Google Distance Matrix API, bao gồm các giá trị khoảng cách và thời gian cho mỗi cặp.

Dịch vụ Google Distance Matrix API không cung cấp chi tiết đường đi. Muốn tìm thông tin chi tiết đường đi có thể sử dụng dịch dụ Google Direction API với chỉ một cặp điểm gốc và điểm đích.

Giới hạn sử dụng :

Mỗi truy vấn gởi cho Distance Matrix API bị giới hạn số lượng phần tử, cứ mỗi một điểm gốc đến một điểm đích là một phần tử.

Số lượng phần tử bị giới hạn như sau:

* 100 phần tử cho mỗi truy vấn.
* 100 phần tử cho mỗi 10 giây.
* 2500 phần tử cho mỗi giờ.

Google MAP API trả phí sẽ có giới hạn lớn hơn:

* 625 phần tử cho mỗi truy vấn.
* 1000 phần tử cho mỗi 10 giây.
* 100000 phần tử cho mỗi giờ.

Các Distance Matrix API URL giới hạn ở 2048 ký tự trước khi mã hóa. Trong mỗi chuỗi Distance Matrix API URL sẽ có nhiều thông tin địa điểm được đưa vào vì vậy nên chú ý giới hạn này khi xây dựng chuỗi request URL.

Sử dụng Distance Matrix API cần phải kết hợp với một bản đồ Google Map dùng để hiển thị các địa điểm một cách trực quan trên bản đồ.

* Evelation API:

Evelation là dịch vụ cung cấp về độ cao, độ sâu của tất cả các vị trí trên bề mặt trái đất. Thật ra Google không thể tính toán hết độ cao tất cả các điểm trên trái đất một cách chính xác, mà Google chỉ lấy độ cao trung bình của 4 vị trí có trong cơ sở dữ liệu của Google mà gần nhất điểm cần tính toán. Elevation API có thể được sử dụng để phát triển các ứng dụng hỗ trợ cho việc leo núi, đi xe đạp, hoặc định vị tọa độ một cách chính xác hơn.Hạn chế: Đối với phiên bản miễn phí chỉ được sử dụng 2500 request trong một ngày và mỗi request chỉ được tối đa 512 địa điểm. Với phiên bản thương mại thì sử dụng được 100000 request ngày và mỗi request được 1000000 địa điểm.

* Place API

Place API : là dịch vụ cho phép truy vấn thông tin về địa điểm đó như tên, lịch sử, các địa điểm đặc biệt xung quanh một tọa độ nào đó, cũng như người dùng có thể tạo thêm địa điểm thêm cho Google.

Hạn chế: chỉ được sử dụng 1000 request trong 1 ngày.

#### Google Maps javascript API

API javascript cho phép người lập trình có thể tích hợp Google Map cũng nhưng các chức năng của nó vào ứng dụng web cũng như là desktop application. API này cung cấp nhiều chức năng tương tác với bản đồ và tương tác với dữ liệu của google như lưu hình ảnh, thông tin, nội dung lên google map. API này được Google cung cấp như một dịch vụ miễn phí cho người dùng.

## Kỹ thuật định vị GPS

Hệ thống định vị toàn cầu (Global Positioning System - 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. 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.

Hệ thống định vị toàn cầu của Mỹ là hệ dẫn đường dựa trên một mạng lưới 24 quả vệ tinh được Bộ Quốc phòng Hoa Kỳ đặt trên quỹ đạo không gian.

GPS ban đầu chỉ dành cho các mục đích quân sự, nhưng từ năm 1980 chính phủ Mỹ cho phép sử dụng trong dân sự. GPS hoạt động trong mọi điều kiện thời tiết, mọi nơi trên Trái Đất, 24 giờ một ngày. Không mất phí thuê bao hoặc mất tiền trả cho việc thiết lập sử dụng GPS nhưng phải tốn tiền không rẻ để mua thiết bị thu tín hiệu và phần mềm nhúng trợ.

## Bộ công cụ lập trình trên Android SDK

Bộ công cụ Android SDK cung cấp những thư viện API và những công cụ khác để nhà phát triển có thể xây dựng, kiểm tra hay dedug những ứng dụng trên Android.

Bộ công cụ gồm những thành phần sau:

* Android Studio và công cụ ADT Plugin.
* Bộ công cụ Adroid SDK.
* Bộ công cụ Android Platform.
* Những thành phần để giả lập hệ điều hành Android.

## Ngôn ngữ lập trình PHP

### Giới thiệu ngôn ngữ lập trình php

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

Như chúng ta đã biết , rất nhìu trang web được xây dựng bởi HTML (Hypertext Markup Language) . Đây chỉ là những trang web tĩnh , nghĩa là chúng chỉ chứa đựng 1 nội dung cụ thể với những dòng văn bản đơn thuần,hình ảnh và có thể được hỗ trợ bởi ngôn ngữ JavaScript hoặc Java Apple .Những trang web như vậy người ta thường gọi là client-side. Tuy nhiên, Internet và Intranets đã được sử dụng cho các ứng dụng cần tới cơ sở dữ liệu. Các trang ứng dụng như vậy được gọi là trang web động. Bởi vì nội dung của chúng luôn thay đổi tùy thuộc vào dữ liệu và người sử dụng .PHP là ngôn ngữ làm được những điều như vậy. Bằng cách chạy chương trình PHP trên máy chủ Web Sever, bạn có thể tạo ra ứng dụng có sự tương tác với cơ sở dữ liệu để tạo ra những trang web và đây được gọi là trang web động.

### Cú pháp trong lập trình PHP

PHP chỉ phân tích các đoạn mã nằm trong những dấu giới hạn của nó. Bất cứ mã nào nằm ngoài những dấu giới hạn đều được xuất ra trực tiếp thông qua xử lý bởi PHP. Các dấu giới hạn thường dùng nhất là <?php và ?>, tương ứng với dấu giới hạn mở và đóng. Các dấu giới hạn<script language = ”php”> và </script>.

Cũng đôi khi được sử dụng. Cách viết dấu giới hạn dạng thẻ ngắn cũng có thể được dùng để thông báo bắt đầu đoạn mã PHP, là <? hay <?= (dấu này được sử dụng để in ra (echo) các xâu ký tự hay biến) với thẻ thông báo kết thúc đoạn mã PHPlà ?>. Những thẻ này thường xuyên được sử dụng, tuy nhiên giống với những thẻ kiểu ASP (<% hay <%= và %>), chúng không có tính di động cao bởi có thể bị vô hiệu khi cấu hình PHP. Bởi vậy, việc dùng các thẻ dạng ngăn hay các thẻ kiểu ASP không được khuyến khích. Mục đích của những dấu giới hạn này là ngăn cách mã PHP với những đoạn mã thuộc ngôn ngữ khác, gồm cả HTML.Mọi đoạn mã bên ngoài các dấu này đều bị hệ thống phân tích bỏ qua và được xuất ra một cách trực tiếp.

Các biến được xác định bằng cách thêm vào trước một dấu đô la ($) và không cần xác định trước kiểu dữ liệu.Không giống với tên hàm và lớp, tên biến là trường hợp nhạy cảm. Cả dấu ngoặc kép ("") và ký hiệu đánh dấu văn bản (<<<EOF EOF;) đều có thể dùng để truyền xâu và giá trị biến. PHP coi xuống dòng như một khoảng trắng theo kiểu như một ngôn ngữ dạng tự do (free-form language) (trừ khi nó nằm trong trích dẫn xâu), và các phát biểu được kết thúc bởi một dấu chấm phẩy. PHP có ba kiểu cú pháp chú thích: /\* \*/ cho phép một đoạn chú thích tùy ý, trong khi đó // và # cho phép chú thích trong phạm vi một dòng. Phát biểu echo là một trong những lệnh của PHP cho phép xuất văn bản (vd. ra một trình duyệt web).

Về cú pháp các từ khóa và ngôn ngữ, PHP tương tự hầu hết các ngôn ngữ lập trình bậc cao có cú pháp kiểu C. Các phát biểu điều kiện If (Nếu), vòng lặp for và while, các hàm trả về đều tương tự cú pháp của các ngôn ngữ như C, C++, Java và Perl.

## Hệ quản trị CSDL MySQL

MySQL là hệ quản trị cơ sở dữ liệutự do nguồn mở phổ biến nhất thế giới và được các nhà phát triển rất ưa chuộng trong quá trình phát triển ứng dụng. Vì MySQL là cơ sở dữ liệu tốc độ cao, ổn định và dễ sử dụng, có tính khả chuyển, hoạt động trên nhiều hệ điều hành cung cấp một hệ thống lớn các hàm tiện ích rất mạnh. Với tốc độ và tính bảo mật cao, MySQL rất thích hợp cho các ứng dụng có truy c ập CSDL trên internet. Nó có nhiều phiên bản cho các hệ điều hành khác nhau: phiên bản Win32 cho các hệ điều hành dòng Windows, Linux, Mac OS X, Unix, FreeBSD, NetBSD, Novell NetWare, SGI Irix, Solaris, SunOS,...

MySQL là một trong những ví dụ rất cơ bản về Hệ Quản trị Cơ sở dữ liệu quan hệ sử dụng Ngôn ngữ truy vấn có cấu trúc (SQL).

MySQL được sử dụng cho việc bổ trợ PHP, Perl, và nhiều ngôn ngữ khác, nó làm nơi lưu trữ những thông tin trên các trang web viết bằng PHP hay Perl,...

****

Hình 2‑7 Logo MySQL

Các phiên bản MySQL hiện nay:

|  |  |
| --- | --- |
| Phí bản quyền | Phiên bản |
| Free, Mã nguồn mở | MySQL Community Edition |
| 2000USD/năm | MySQL Standard Edition |
| 5000USD/năm | MySQL Enterprise Edition |
| 1000USD/năm | MySQL Cluster Carrier Grade Edition |

Bảng 2‑1 Các phiên bản MySQL

## JSON ( Javescript Object Noattion)

JSON (JavaScript Object Notation) là cú pháp để lưu trữ và trao đổi dữ liệu văn bản dung lượng nhẹ giống như XML. Nhưng JSON truy xuất nhanh hơn và dễ dàng phân tích hơn so với XML.

JSON sử dụng cú pháp JavaScript để mô tả đối tượng dữ liệu, nhưng JSON là ngôn ngữ và nền tản độc lập. Trình phân tích cú pháp (parsers) JSON và các thư viện JSON tích hợp cho nhiều ngôn ngữ lập trình khác nhau nên JSON được sử dụng một cách rộng rãi để truyền dữ liệu giữa các ứng dụng xây dựng bởi các ngôn ngữ khác nhau.

JSON là một dạng văn bản trơn (tức là không có định dạng màu sắc, cỡ chữ,...). Vì là một dạng cây phân cấp nên người dùng có thể dễ dàng hiểu được cấu trúc của một văn bản JSON. Dữ liệu JSON có thể được truyền đi bằng AJAX và được phân tích cú pháp (parse) bởi JavaScript.

So với các định dạng khác như XML, SOAP... thì JSON được sử dụng dưới dạng mảng và không có các thẻ mở đầu hoặc kết thúc nên cấu trúc ngắn hơn; vì thế JSON nhẹ hơn, dễ đọc, ghi và truyền tải nhanh hơn trong môi trường mạng.

Cú pháp của một đối tượng JSON (JSONObject) là một tập hợp con của cú pháp đối tượng JavaScript:

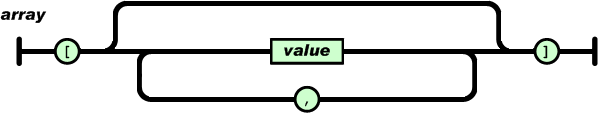
Cú pháp của một đối tượng JSON (JSONObject) là một tập hợp con của cú pháp đối tượng JavaScript:



Hình 2‑8 Cú pháp của một đối tượng Json

* Dữ liệu là các cặp tên/giá trị (“name” : ”value”).
* Dữ liệu được phân cách bằng dấu phẩy (,).
* Mỗi đối tượng nằm trong cặp ngoặc nhọn ‘{}‘.

Một mảng JSON là tập hợp nhiều đối tượng JSON được ngăn cách nhau bởi dấu “,” và tất cả được đặt trong cặp ngoặc vuông ‘[]’.



Hình 2‑9 Cú pháp của một mảng Json

Vì JSON sử dụng cú pháp JavaScript, nên không cần sử dụng một chương trình trung gian để trao đổi giữa JSON và JavaScript. Trong ngôn ngữ lập trình PHP, việc hỗ trợ truy xuất với các đối tượng JSON rất nhanh chóng và dễ dàng.

## Ngôn ngữ XML

XML là ngôn ngữ đánh dấu với mục đích chung do W3C đề nghị, để tạo ra các ngôn ngữ đánh dấu khác. Nó có khả năng mô tả nhiều loại dữ liệu khác nhau. Mục đích chính của XML là đơn giản hóa việc chia sẻ dữ liệu giữa các hệ thống khác nhau, đặc biệt là các hệ thống được kết nối với Internet. Các ngôn ngữ dựa trên XML như: RDF, RSS, MathML, XHTML, SVG, GML và cXML được định nghĩa theo cách thông thường, cho phép các chương trình sửa đổi và kiểm tra hợp lệ bằng các ngôn ngữ này mà không cần có hiểu biết trước về hình thức của chúng.

**Đặt điểm của ngôn ngữ XML :**

XML cung cấp một phương tiện dùng văn bản để mô tả thông tin và áp dụng một cấu trúc kiểu cây cho thông tin đó. Tại mức căn bản, mọi thông tin đều thể hiện dưới dạng text, chen giữa là các thẻ đánh dấu với nhiệm vụ ký hiệu sự phân chia thông tin thành một cấu trúc có thứ bậc của các dữ liệu ký tự, các phần tử dùng để chứa dữ liệu, và các thuộc tính của các phần tử đó. Về mặt đó, XML tương tự với các biểu thức S (S-expression) của ngôn ngữ lập trình LISP ở chỗ chúng đều mô tả các cấu trúc cây mà trong đó mỗi nút có thể có một danh sách tính chất của riêng mình.

Đơn vị cơ sở của XML là các ký tự theo định nghĩa của Universal Character Set. Các ký tự được kết hợp theo các tổ hợp chuỗi hợp lệ để tạo thành một tài liệu XML. Tài liệu này gồm một hoặc nhiều thực thể, mỗi thực thể thường là một phần nào đó của các ký tự thuộc tài liệu, được mã hóa dưới dạng một chuỗi các bit và lưu trữ trong một tệp văn bản.

Sự phổ biến của các phần mềm soạn thảo văn bản đã hỗ trợ việc soạn thảo và bảo trì tài liệu XML một cách nhanh chóng. Trước XML, có rất ít ngôn ngữ mô tả dữ liệu với các đặc điểm đa năng, thân thiện với giao thức Internet, dễ học và dễ tạo. Thực tế, đa số các định dạng trao đổi dữ liệu thời đó đều chuyện dụng, có tính độc quyền, và có định dạng nhị phân khó dùng chung giữa các ứng dụng phần mềm khác nhau hay giữa các hệ nền khác nhau. Việc tạo và bảo trì trên các trình soạn thảo thông dụng lại càng khó khăn.

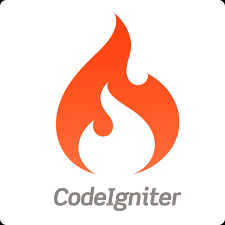
Bằng cách cho phép các tên dữ liệu, cấu trúc thứ bậc được phép, và ý nghĩa của các phần tử và thuộc tính có tính chất mở và có thể được định nghĩa bởi một giản đồtùy biến được, XML cung cấp một cơ sở cú pháp cho việc tạo lập các ngôn ngữ đánh dấu dựa XML theo yêu cầu. Cú pháp chung của các ngôn ngữ đó là cố định - các tài liệu phải tuân theo các quy tắc chung của XML, bảo đảm rằng tất cả các phần mềm hiểu XML ít ra cũng phải có khả năng đọc và hiểu bố cục tương đối của thông tin trong các tài liệu đó. Giản đồ chỉ bổ sung một tập các ràng buộc cho các quy tắc cú pháp. Các giản đồ thường hạn chế tên của phần tử và thuộc tính và các cấu trúc thứ bậc được phép, ví dụ, chỉ cho phép một phần tử tên 'ngày sinh' chứa một phần tử tên 'ngày' và một phần tử có tên 'tháng', mỗi phần tử phải chứa đúng một ký tự. Đây là điểm khác biệt giữa XML và HTML. HTML có một bộ các phần tử và thuộc tính không mềm dẻo, chỉ có một tác dụng và nói chung là không thể dùng cho mục đích khác.

XML không hạn chế về việc nó được sử dụng như thế nào. Mặc dù XML về cơ bản là dạng text, các phần mềm với chức năng trừu tượng hóa nó thành các định dạng khác giàu thông tin hơn đã nhanh chóng xuất hiện, quá trình trừu tượng hóa này được thực hiện chủ yếu qua việc sử dụng các giản đồ định hướng kiểu dữ l iệu và khuôn mẫu lập trình hướng đối tượng. Những phần mềm như vậy có thể coi XML như là dạng text đã được tuần tự hóa chỉ khi nó cần truyền dữ liệu qua mạng.

## Codeigniter framework

**Codeigniter là gì ?**

* Codeigniter (CI) là một framework của ngôn ngữ lập trình PHP.
* CI được phát triển bởi EllisLab, Inc, phiên bản đầu tiên của CI được công bố vào ngày 28 tháng 02 năm 2006. CI được tạo ra với mục đính cho phép người phát triển (developers) sử dụng ngôn ngữ lập trình PHP để xây dựng một ứng dụng web nhanh hơn. CodeIgniter sử dụng mô hình model – view - controller (MVC model) cho các ứng dụng web.



Hình 2‑10 Logo Codeigniter

**Những tính năng của Codeigniter:**

* CodeIgniter rất đơn giản trong việc cài đặt và cấu hình.
* CodeIgniter rất nhẹ (với khoảng hơn 2MB).
* CodeIgniter sử dụng mô hình Model – View – Controller.
* CodeIgniter có định dạng và chuẩn hóa form và dữ liệu đầu vào (validation form).
* CodeIgniter cho phép gửi email, attach file, HTML/Text email, đa giao thức gửi mail (sendmail, SMTP).
* CodeIgniter hỗ trợ Upload file.
* CodeIgniter cung cấp lớp FTP class hỗ trợ cho việc liên quan đến FTP.

**Những điểm nổi bật của Codeigniter:**

* **Được thiết kế theo mô hình MVC (Model-View-Controller):** mô hình MVC giúp tách thành phần hiển thị giao diện (presentation) và xử lý (business logic) của một phần mềm thành các phần độc lập, từ đó giúp cho việc thiết kế Web, xử lý và bảo trì mã nguồn dễ dàng, đồng thời tăng khả năng mở rộng của phần mềm. CodeIgniter vận dụng mô hình này trong thiết kế, giúp tách biệt các tập tin giao diện với các tập tin xử lý dữ liệu, nâng cao khả năng xử lý và bảo trì.
* **Tốc độ nhanh** : CodeIgniter được đánh giá là PHP Framework có tốc độ nhanh nhất hiện nay. Bằng cơ chế lưu nội dung vào bộ đệm (cache), kiểm tra bộ đệm trước khi tiến hành thực hiện yêu cầu, CodeIgniter giảm số lần truy cập và xử lý dữ liệu, từ đó tối ưu hoá tốc độ tải trang.
* **Miễn phí**: CodeIgniter được phát hành dưới giấy phép Apache/BSD mở rộng, cho phép người dùng tự do thay đổi, phát triển và phân phối mã nguồn.
* **Hỗ trợ Search Engine Optimization**: Cấu trúc URL của CodeIgniter rất thân thiện với các Robot tìm kiếm.
* **Hệ thống thư viện phong phú**:CodeIgniter cung cấp những thư viện phục vụ cho những tác vụ thường gặp nhất trong lập trình web, chẳng hạn như truy cập cơ sở dữ liệu, gửi email, kiểm tra dữ liệu, quả lý session, xử lý ảnh,… đến những chức năng nâng cao như XML-RPC, mã hoá, bảo mật,…
* **Bảo mật hệ thống**: cơ chế kiểm tra dữ liệu chặt chẽ, ngăn ngừa XSS và SQL Injection của CodeIgniter giúp giảm thiểu các nguy cơ bảo mật cho hệ thống.

## Bootstrap framework

**Bootstrap là gì ?**

* Bootstrap đuợc phát triển bởi Mark Otto và Jacob Thornton tại Twitter, và xuất bản bộ nguồn mở này vào tháng 8 năm 2011 trên GitHub.
* Bootstrap là một bộ sưu tập miễn phí các công cụ để tạo ra các trang web và các ứng dụng web
* Bootstrap chứa HTML và CSS dựa trên các mẫu thiết kế cho các văn bản, các hình thức, các nút, chuyển hướng và các thành phần khác
* Bootstrap cũng có phần mở rộng JavaScript tùy chọn
* Bootstrap là mã nguồn mở và có sẵn trên GitHub

****

Hình 2‑11 Logo Bootstrap

**Vì sao sử dụng Bootstrap.**

* Được viết bởi những người có óc thẩm mỹ và tài năng trên khắp thế giới. Sự tương thích của trình duyệt với thiết bị đã được kiểm tra nhiều lần nên có thể tin tưởng kết quả mình làm ra và nhiều khi không cần kiểm tra lại. Vì vậy, giúp cho dự án của bạn tiết kiệm được thời gian và tiền bạc.
* Chỉ cần biết sơ qua HTML, CSS, Javascript, Jquery là bạn có thể sử dụng Bootstrap để tạo nên một trang web sang trọng và đầy đủ. Nhưng lại không cần code quá nhiều CSS.
* Với giao diện mặc định là màu xám bạc sang trọng, hỗ trợ các component thông dụng mà các website hiện nay cần có. Vì nó là opensource nên bạn có thể vào mã nguồn của nó để thay đổi theo ý thích của bản thân.
* Do có sử dụng Grid System nên Bootstrap mặc định hỗ trợ Responsive. Bootstrap được viết theo xu hướng Mobile First tức là ưu tiên giao diện trên Mobile trước. Nên việc sử dụng Bootstrap cho website của bạn sẽ phù hợp với tất cả kích thước màn hình. Nhờ đó mà chúng ta không cần xây dựng thêm một trang web riêng biệt cho mobile.
* Đội ngũ phát triển Bootstrap đã bổ sung thêm tính năng Customizer. Giúp cho designer có thể lựa chọn những thuộc tính, component phù hợp với project của họ. Chức năng này giúp ta không cần phải tải toàn bộ mã nguồn về máy.

# PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

## Xác định yêu cầu và đối tượng trong hệ thống

### Yêu cầu đối với hệ thống:

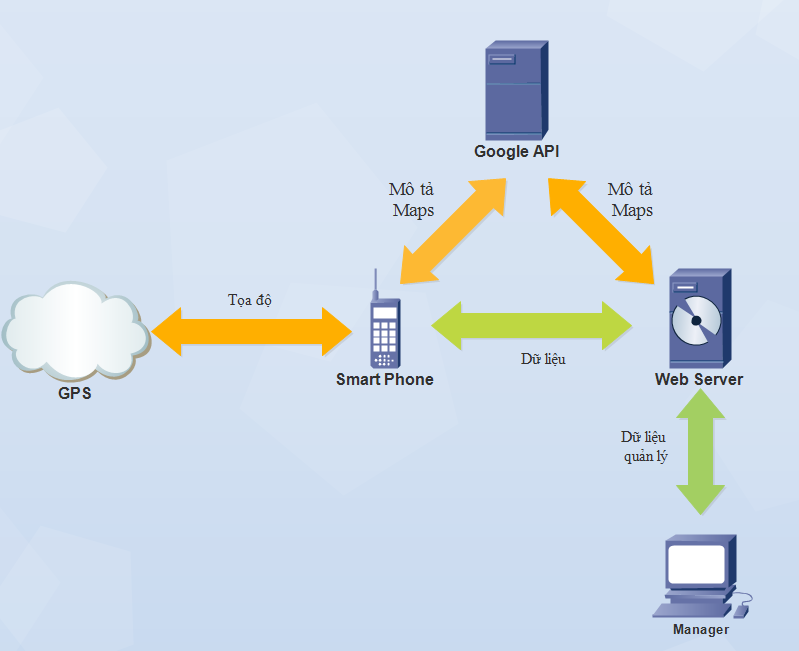
* Đối với hành khách:
* Nhận dữ liệu về thông tin đặt vé của hành khách: ngày giờ, chuyến đi, điểm khởi hành, điểm trả khách ...
* Trả giá trị thông tin chuyến đi cho hành khách: biển số xe, tên tài xế, ngày giờ khởi hành...
* Chức năng định vị, xác định vị trí hiện tại của hành khách.
* Thông báo cho hành khách khi cách điểm trả khách 10km.
* Nhận báo cáo, nội dung góp ý từ hành khách.
* Kiểm tra tốc độ và báo cáo vi phạm tốc độ.
* Đối với lái xe:
* Nhận thông tin về hành khách: thông tin ghế ngồi, tên hành khách, điểm đón khách, điểm trả khách.
* Thông báo điểm trả khách trước 10km.
* Chức năng định vị, xác định vị trí.
* Kiểm tra tốc độ vào nhận báo cáo sai phạm tốc độ.
* Đối với quản lý:
* Nhận thông tin bao cá sai phạm, nội dung góp ý.
* Sắp xếp lịch trình cho tài xế và chuyến đi.
* Đối với web server:
* Làm nơi lưu trữ cơ sở dữ liệu của cả hệ thống.
* Tiếp nhận và gửi trả dữ liệu truy vấn cho các thiết bị đầu cuối.

### Các đối tượng trong hệ thống:

* Hành khách:
* Là người có điện thoại smart phone và có khả năng kết nối internet.
* Là người có yêu cầu muốn giám sát chuyến đi của mình có được an toàn hay không.
* Là người muốn gửi ý kiến góp ý để nâng cao chất lượng của ngành giao thông
* Là người muốn tiện lợi hơn trong việc đón xe và xuống xe.
* Tài xế:
* Là người phải có điện thoại smart phone và có khả năng kết nối internet.
* Là người chịu trách nhiệm đón trả khách đúng vị trí và server trả về.
* Quản lý:
* Là người điều tiết lịch trình đối với các tài xế 1 cách hợp lý.
* Là người nhận các báo cáo từ hành khách và tài xế.

### Thiết kế hệ thống:

Hệ thống quản lý xe khách hoạt động theo mô hình client –server gồm có 3 thành phần chính:



Hình 3‑1 Mô hình hệ thống

* Thiết bị di động đầu cuối.
* Webserver: Là nơi tập trung xử lý các yêu cầu bên phía người dùng, thao tác quản lý cơ sở dữ liệu. Là nơi quản lý sẽ sử dụng để quản lý hệ thống của mình.
* Google Maps API.

### Thiết bị di động smartphone

Sẽ có 2 ứng dụng được tạo trên nền tảng android phục vụ cho từng yêu cầu khác nhau của hành khách, lái xe. Chúng sẽ thực hiện các công việc sau đây:

* Đo tốc độ di chuyển bằng GPS.
* Xác định vị trí hiện tại thông qua kỹ thuật GPS.
* Gởi vị trí hiện tại lên Google API server và yêu cầu tìm kiếm thông tin về địa điểm hiện tại.
* Tính toán khoảng cách với điểm đến cho trước và bật thông báo.
* Gởi truy vấn đến server để lấy thông tin.
* Nhận thông tin từ server và lưu dữ liệu vào bộ đệm trên thiết bị. Cụ thể là lưu trữ bằng file json.

### Google map api:

Là api được Google xậy dựng để hỗ trợ về bản đồ cho người dùng. Trong đề tài này nhóm sử dụng các chức năng sau:

* Xác định vị trí của smartphone dựa trên location mà GPS trả về.
* Vẽ đường đi của hành trình lên google maps trên thiết bị android cũng như webserver.
* Tính khoảng cách của các location được xác định trước.

Vd: Tính khoảng cách đón trả khách để hiển thị thông báo.

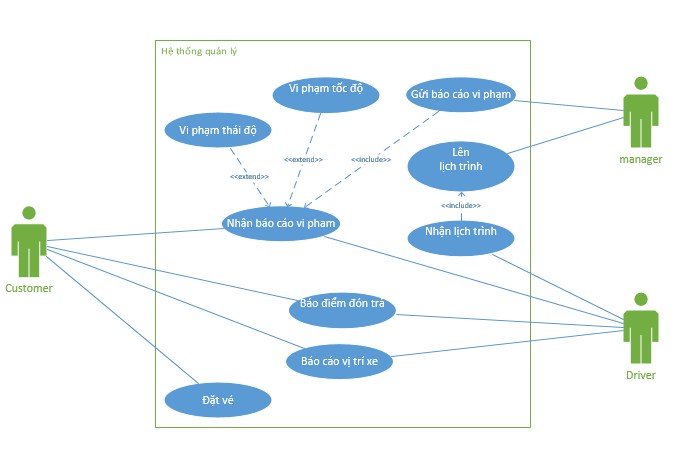
### Web Server:

Các chức năng của Web Server như sau:

* Tiếp nhận và xử lý truy vấn từ thiết bị smartphone.
* Trả kết quả về cho thiết bị smartphone.
* Lưu trữ cơ sở dữ liệu.
* Các chức năng quản lý: quản lý nhân viên, quản lý hành khách, quản lý lộ trình, quản lý trạm...

## Phân tích và thiết kế hệ thống

### Sơ đồ User case:



Hình 3‑2 Sơ đồ User case

#### Danh sách User case

|  |  |  |
| --- | --- | --- |
| STT | Tên User Case | Ý Nghĩa |
| 1 | Đặt vé | Hành khách sử dụng để mua vé xe bằng điện thoại |
| 2 | Bao cáo vi phạm | Hành khách và tài xế gửi những vi phạm về Webserver |
| 3 | Vi phạm thái độ | Những vi phạm về thái độ của nhân viên đối với hành khách |
| 4 | Vi phạm tốc độ | Vi phạm quá tốc độ của tài xế |
| 5 | Nhận báo cao vi phạm | Quản lý nhận những vi phạm được Hành khách và Tài xế gửi về |
| 6 | Lên lich trình | Quản lý lên lịch chạy cho tài xế |
| 7 | Nhận lịch trình | Tài xế nhận lịch trình từ quản lý |
| 8 | Báo điểm đón trả | Hành khách sẽ gửi địa điểm đón trả cho webserver và webserver sẽ trả xuống cho tài xế biết |
| 9 | Báo vị trí xe | Hệ thống sẽ tự động cập nhật vị trí xe và báo cho hành khách lẫn tài xế biết |

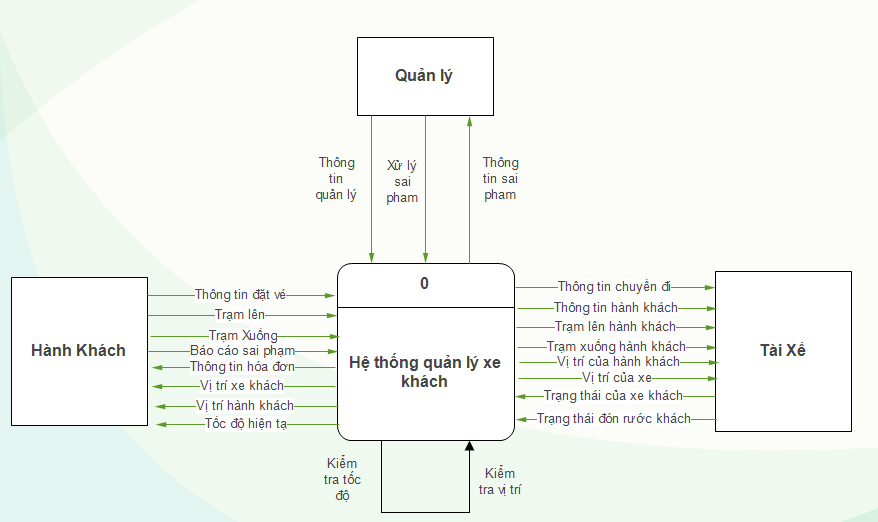
Bảng 3‑1 Danh sách User case

### Sơ đồ phân rã chức năng.

Hình 3‑3 Sơ đồ phân rã chức năng trên thiết bị Android

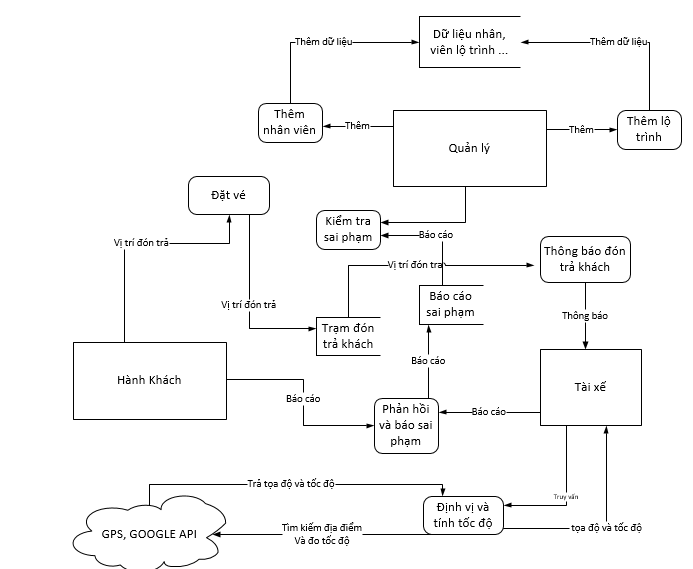
### Sơ đồ DFD

#### Mô hình mức ngữ cảnh



Hình 3‑5 Sơ đồ DFD mức ngữ cảnh

#### Mô hình DFD

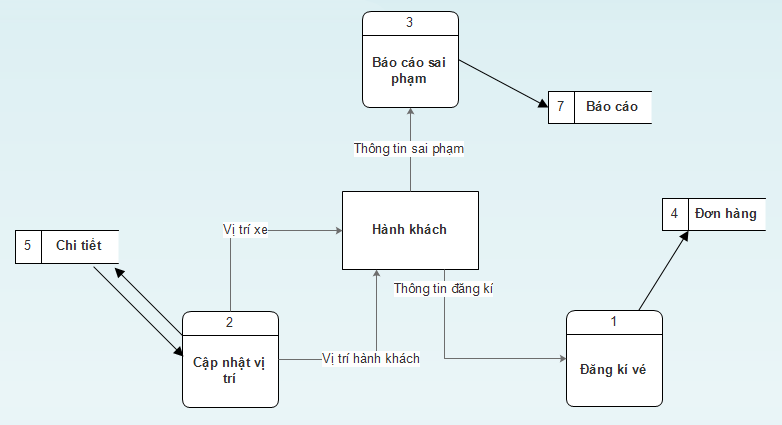


Hình 3‑6 Mô hình DFD

Kho dữ liệu

|  |  |
| --- | --- |
| **STT** | **Tên kho dữ liệu** |
| 1 | Hành khách |
| 2 | Nhân viên |
| 3 | Lộ trình |
| 4 | Đơn hàng |
| 5 | Chi tiết |
| 6 | Xe |
| 7 | Báo cáo |

#### Mô hình DFD đối với hành khách:

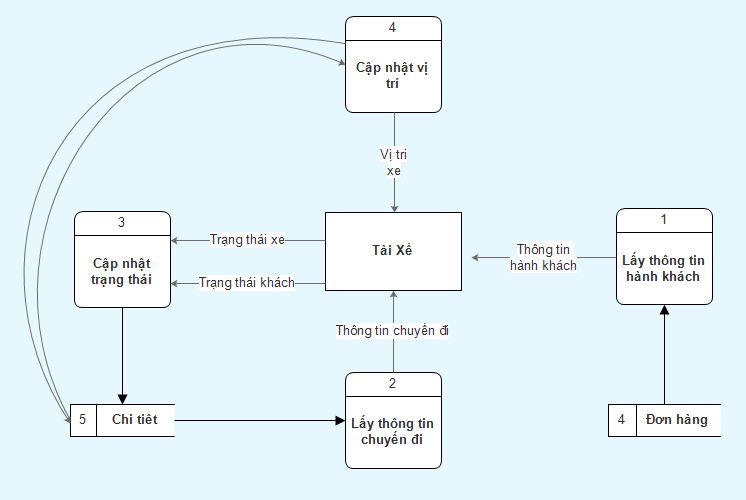


Hình 3‑7 Mô hình DFD đối với hanh khách

Đặc tả mô hình ngữ cảnh đối với hành khách.

* Ô xử lý 1: Đăng kí vé.
* Hành khách gửi thông tin đăng ký gồm: lộ trình, ngày khỏi hành, vị trí ghế, vị trí đón, vị trí trả ...
* Hệ thống sẽ lưu các thông tin trên vào kho Thông tin đăng ký.
* Ô xử lý 2: Cập nhật vị trí.
* Hệ thống sẽ lấy thông tin vị trí xe được lưu trên kho vị trí thông tin và trả về cho hành khách biết xe đang ở đâu và cách mình bao nhiêu kilomet.
* Hệ thống tự động cập nhật vị trí của hành khách và gửi vào kho, đồng thời hiện thị cho hành khách biết mình đang ở đâu.
* Ô xử lý 3: Báo cáo sai phạm.
* Khi Hành khách thấy có dấu hiệu sai phạm ở Tài xế như việc đón trả sai trạm, thái độ không đúng đối với Hành khách, Hành khách có thể gửi thông báo sai phạm về cho hệ thống.
* Thông tin sẽ được lưu lại để chờ xử lý.

#### Mô hình DFD đối với Tài xế:

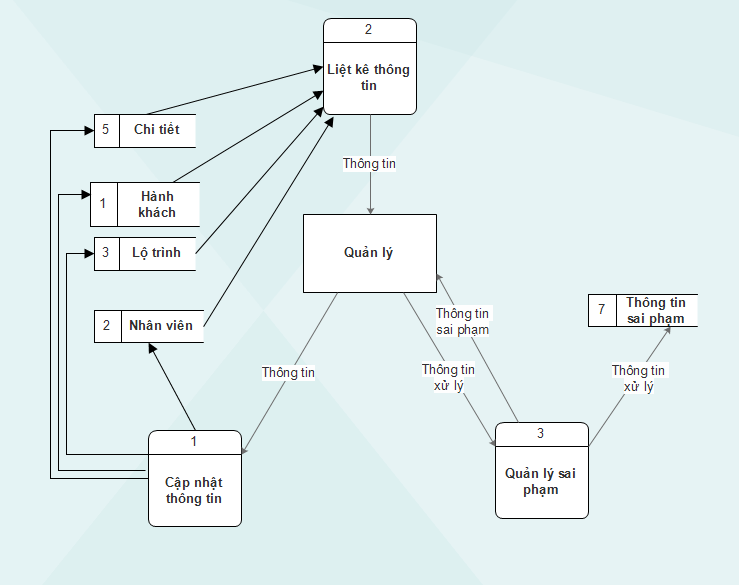


Hình 3‑8 Mô hình DFD đối với tài xế

Đặc tả mô hình ngữ cảnh đối với tài xế

* Process 1: Lấy thông tin chuyến đi.
* Tài xế lấy thông tin chuyến đi từ cơ sở dữ liệu mà quản lý đã cập nhật trước đó.
* Thông tin sẽ ghi chi tiết về lộ trình, chuyến , biển số xe ....
* Ô xử lý 2: Lấy thông tin khách hang.
* Hệ thống sẽ trả về thông tin hành khách như: số điện thoại, tên hành khách, vị trí ghế, nơi đón, nơi trả...
* Ô xử lý 3: Cập nhật trang thái
* Có 4 trạng thái của xe: chưa sẵn sàng, sẵn sàng, đang chay và kết thúc.
* Tài xế có thể cập nhật 2 trạng thái đang chạy và kết thúc cho hệ thông.
* Khi xe rời bến thì sẽ cập nhật trạng thái đang chạy cho hệ thống. Và khi xe kết thúc lộ trình tài xế sẽ cập nhật trạng thái kết thúc cho hệ thống.
* Cập nhật trạng thái đã đón trả khách.

#### Mô hình DFD đối với Quản lý



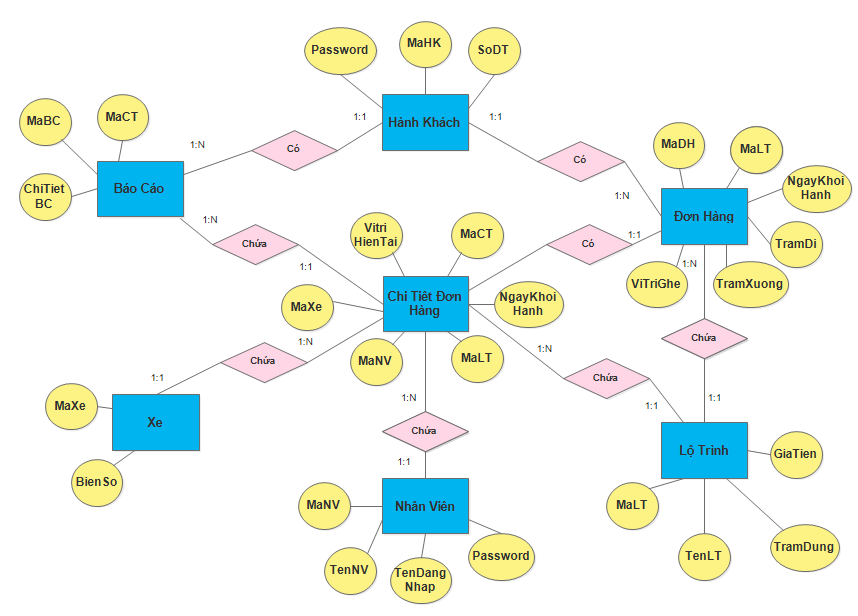
Hình 3‑9 Mô hình DFD đối với quản lý

Đặc tả mô hình ngữ cảnh đối với Quản lý

* Ô xử lý 1: Cập nhật thông tin.
* Quản lý có thể thêm, sửa chữa những thông tin cần thiết cho hệ thống như: thông tin tài xế , thông tin hành khách, thông tin lộ trình, thông tin các chuyến đi.
* Hệ thống sẽ lưu các thông tin trên vào cơ sở dữ liệu.
* Ô xử lý 2: Liệt kê thông tin.
* Quản lý có thể liệt kê tất cả những thông tin liên quan đến tài xế, hành khách, lộ trình, chuyến đi được lưu trong cơ sở dữ liệu.
* Ô xử lý 3:Quản lý sai phạm.
* Quản lý có thể xe tất cả vi phạm của tài xế .
* Quản lý xử lý sai phạm xong sẽ gửi thông tin đã xử lý về cho hệ thống.

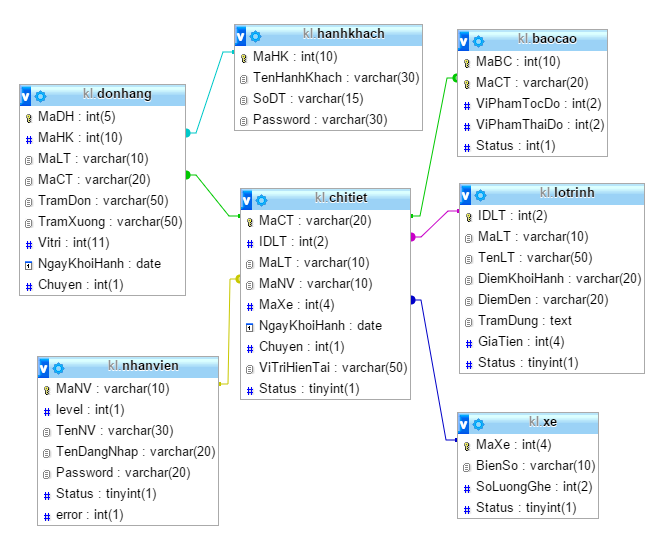
### Thiết kế cơ sở dữ liêu

#### Mô hình thực thể ERD



Hình 3‑10 Mô hình thực thể ERD

#### Mô hình dữ liệu



Hình 3‑11 Mô hình dữ liệu

Đặc tả cơ sở dữ liệu:

* Bảng hành khách:

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên | Kiểu dữ liệu | Ý nghĩa |
| 1 | MaHK | Int(10) | Khóa chính |
| 2 | TenHanhKhach | Varchar(30) | Tên hành khách |
| 3 | SoDT | Varchar(15) | Số điện thoại |
| 4 | Password | Varchar(30) | Mật khẩu. |

Bảng 3‑2 đặc tả bảng dữ liệu hành khách

* Bảng nhân viên

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên | Kiểu dữ liệu | Ý nghĩa |
| 1 | MaNV | Varchar(10) | Khóa chính, mã nhân viên |
| 2 | Level | Int(1) | 1:quản lý 2:tài xế |
| 3 | TenNV | Varchar(30) | Tên nhân viên |
| 4 | TenDangNhap | Varchar(20) | Tên đăng nhập |
| 5 | Password | Varchar(20) | Mật khẩu |
| 6 | Status | Int(1) | Trạng thái |
| 7 | Error | Int(1) | Trạng thái lỗi của nhân viên  0: không có lỗi vi phạm  1: có lỗi vi phạm |

Bảng 3‑3 Đặt tả bảng dữ liệu nhân viên

* Bảng đơn hàng

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên | Kiểu dữ liệu | Ý nghĩa |
| 1 | MaDH | Int(5) | Mã đơn hàng |
| 2 | MaHK | Int(10) | Khóa ngoại , mã hành khách |
| 3 | MaLT | Varchar(10) | Khóa ngoại , mã lộ trình |
| 4 | MaCT | Varchar(20) | Khóa ngoại, mã chi tiết |
| 5 | TramDon | Varchar(50) | Trạm đón khách |
| 6 | TramXuong | Varchar(50) | Trạm trả khách |
| 7 | Vitri | Int(1) | Vị trí ghế ngồi |
| 8 | NgayKhoiHanh | Date | Ngày khởi hành |
| 9 | Chuyen | Int(1) | Chuyến đi trong ngày |

Bảng 3‑4 Đặc tả bảng dữ liệu đơn hàng

* Bảng chi tiết

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên | Kiểu dữ liệu | Ý nghĩa |
| 1 | MaCT | Varchar(20) | Khóa chính, mã chi tiết |
| 2 | IDLT | Int(2) | Khóa ngoại, ID lộ trình |
| 3 | MaLT | Varchar(10) | Mã lộ trình |
| 4 | MaNV | Varchar(10) | Khóa ngoại, mã nhân viên |
| 5 | MaXe | Int(4) | Khóa ngoại, mã xe |
| 6 | NgayKhoiHanh | Date | Ngày khởi hành |
| 7 | Chuyen | Int(1) | Chuyến trong ngày |
| 8 | ViTriHienTai | Varchar(50) | Vị trí hiện tại của xe |
| 9 | Status | Tynyint(1) | Trạng thái của xe  0:chưa sẵn sàng  1:đã săn sàng  2:đang chạy  3:kết thúc |

Bảng 3‑5 Đăc tả bảng dữ liệu chi tiết

* Bảng lộ trình

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên | Kiểu dữ liệu | Ý nghĩa |
| 1 | IDLT | Int(2) | Khóa chính, ID lộ trình |
| 2 | MaLT | Varchar(10) | Mã lộ trình |
| 3 | TenLT | Varchar(50) | Tên lộ trình |
| 4 | DiemKhoiHanh | Varchar(20) | Điểm khởi hành của lộ trình |
| 5 | DiemDen | Varchar(20) | Điểm đến của lộ trình |
| 6 | TramDung | Text | Tập hợp tất cả vị trí của trạm dừng |
| 7 | GiaTien | Int(4) | Giá tiền |
| 8 | Status | Tynyint(1) | 0:lộ trình đã săn sàng  1:lộ trình chưa sẵn sàng |

Bảng 3‑6 Đặc tả bảng dữ liệu lộ trình

* Bảng xe

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên | Kiểu dữ liệu | Ý nghĩa |
| 1 | MaXe | Int(4) | Khóa chính, mã xe |
| 2 | BienSo | Varchar(10) | Biển số xe |
| 3 | SoLuongGhe | Varchar(2) | Số lượng ghế |
| 4 | Status | Tinyint(1) | Trang thái xe  0: xe đang rảnh  1: xe đã có lộ trình chạy |

Bảng 3‑7 Đặc tả bảng dữ liệu xe

* Bảng báo cáo

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên | Kiểu dữ liệu | Ý nghĩa |
| 1 | MaBC | Int(4) | Khóa chính, Mã báo cáo |
| 2 | MaCT | Varchar(20) | Khóa ngoại, mã chi tiết |
| 3 | ViPhamTocDo | Int(2) | Số lần vi phạm tốc độ |
| 4 | ViPhamThaiDo | Int(2) | Số lần vi phạm thái độ |
| 5 | ViPhamTram | Int(2) | Số lần vi phạm trạm |
| 6 | Status | Int(1) | Trạng thái của báo cáo  0: chưa xử lý  1:đã xử lý |

Bảng 3‑8 Đặc tả bảng dữ liệu báo cáo

## Mô tả chức năng

### Mô tả chức năng quản lý

#### Chức năng đăng nhập vào hệ thống

Chức năng này giúp quản lý có thể đăng nhập vào hệ thống.



Hình 3‑12 Giao diện đăng nhập của quản lý

#### Chức năng quản lý báo cáo sai phạm

Chức năng này cung cấp cho quản thông tin chi tiết các vi phạm có trong ngày và xử lý chung.



Hình 3‑13 Giao diện quản lý báo cáo sai phạm

#### Chức năng quản lý hành khách

Chức năng này cung cấp cho quản lý thông tin tất cả các hành khách có trong hệ thống.

Có thể sửa chửa thông tin của khác hàng ngoại trừ mã hành khách.



Hình 3‑14 Giao diện quản lý hành khách

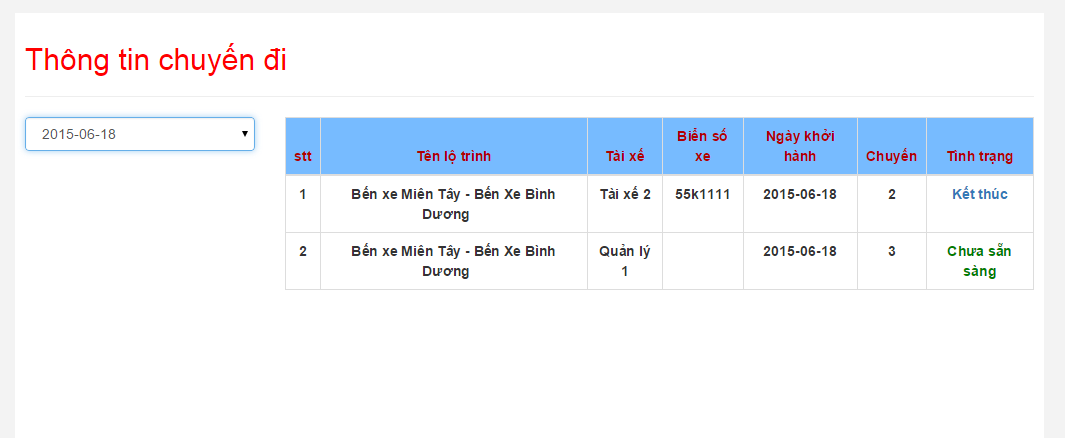
#### Chức năng quản lý chuyến đi

Chức năng này cung cấp cho quản lý biết tình trạng của chuyến đi.

Thông tin gồm: Tên lộ trình, Tên tài xế, Biển số, Ngày khỏi hành, Chuyến, Tình trạng của xe.

Xe có 4 tình trạng:

* Chưa săn sàng: Khi quản lý chưa điều tiết chuyến đi với tài xế nào, xe nào.
* Sẵn sàng: Xe đã được điều tiết.
* Đang chạy: Xe đang chạy.
* Kết thúc: Xe đã tới trạm cuối và kết thúc chuyến đi.



Hình 3‑15 Giao diện quản lý chuyến đi

#### Chức năng điều hướng chuyến đi

Quản lý sử dụng chức năng này để điều tiết chuyến đi với các tài xế và xe.



Hình 3‑16 Giao diện điều hướng chuyến đi

#### Chức năng thêm tài xế

Quản lý sử dụng chức năng này để thêm mới tài xế hoặc xóa bỏ một tài xế nào đó.



Hình 3‑17 Giao diện thêm tài xế

#### Chức năng sửa tài xế

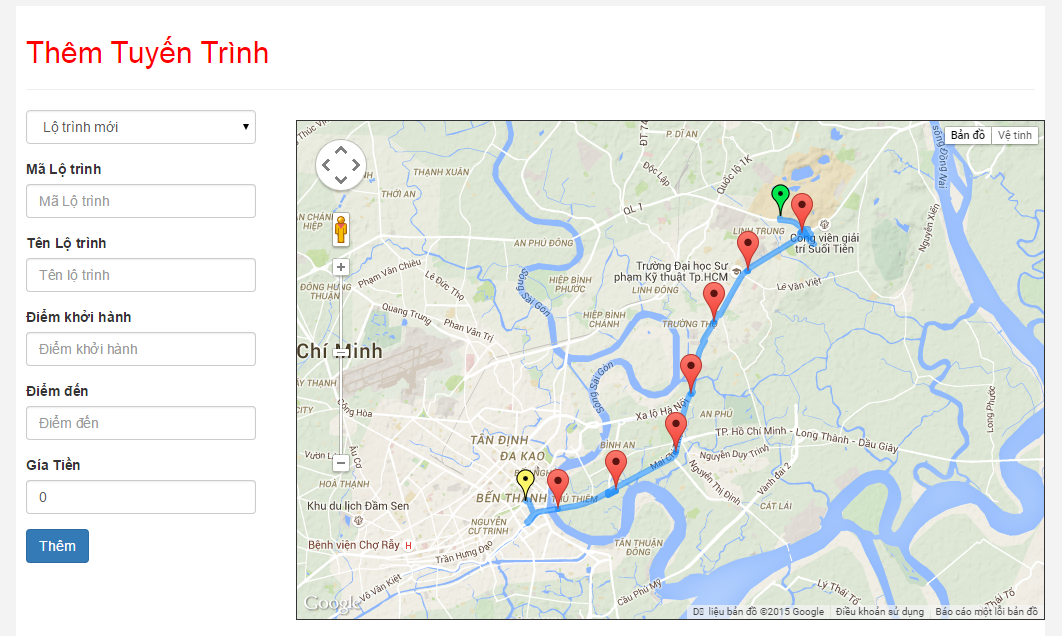
Quản lý sử dụng chức năng này để sửa thông tin tài xế.



Hình 3‑18 Giao diện sửa tài xế

#### Chức năng quản lý lộ trình

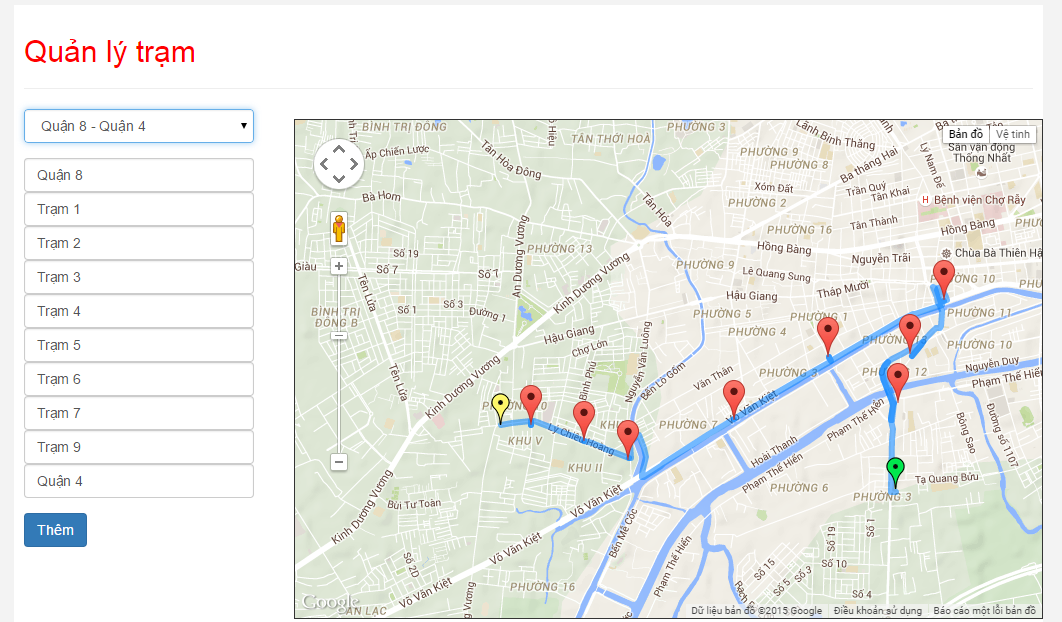
Quản lý có thể sử dụng chức năng này để thêm lộ trình mới hoặc chỉnh sửa lộ trình cũ.



Hình 3‑19 Giao diện quản lý lộ trình

#### Chức năng thêm trạm.

Quản lý sử dụng chức năng này để thêm hoặc thay đổi tên trạm trong từng lộ trình khác nhau.



Hình 3‑20 Giao diện thêm trạm

### Mô tả chức năng đối với tài xế

#### Đăng nhập

Chức năng đăng nhập: mỗi tài xế được cấp một tài khoản riêng. Chức năng đăng nhập cho phép tài xế đăng nhập vào ứng dụng bằng tài khoản của minh và lấy dữ liệu về chuyến đi của mình (thông tin hành khách ).



Hình 3‑21 Giao diện đăng nhập đối với xe

#### Kiểm tra chuyến đi

Sau khi đăng nhập bằng tài khoản của mình:

Nếu có chuyến đi trong ngày sẽ trả về giao diện hiển thị thông tin chuyến đi và các xử lý.



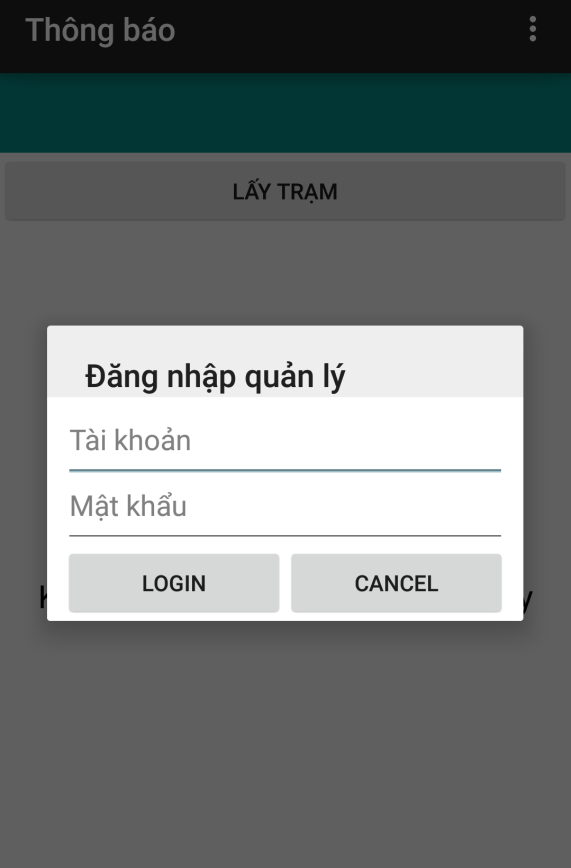
Hình 3‑22 Giao diện hiển thị thông tin chuyến đi

Nếu không có chuyến đi sẽ hiện giao diện thông báo không có chuyến đi và có chức lấy trạm dừng để thêm lộ trình mới.



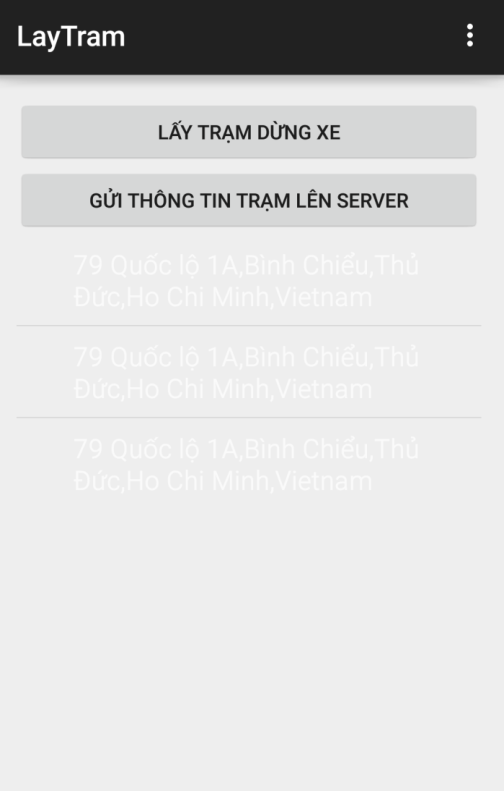
Hình 3‑23 Giao diện thông báo không có chuyến đi

Bấm vào lấy trạm sẽ hiện bảng đăng nhập, yêu cầu nhập tài khoản của quản lý để thực hiện chức năng này



Hình 3‑24 Giao diện đăng nhập tài khoản quản lý

Đăng nhập hoàn tất sẽ hiên ra giao diện lấy trạm và gửi dữ liệu lên Server.



Hình 3‑25 Giao diện lấy trạm và gửi lên Server

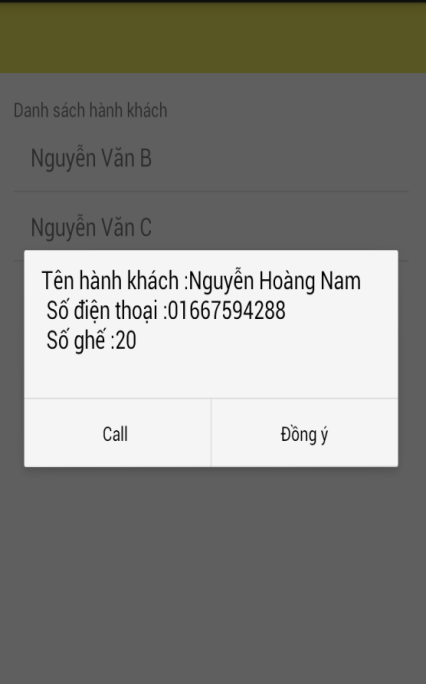
#### Xem danh sách hành khách

Chức năng cho phép xem danh sách hành khách đã đăng ký đi chuyến xe này.



Hình 3‑26 Giao diện hiển thị danh sách hành khách theo chuyến đi

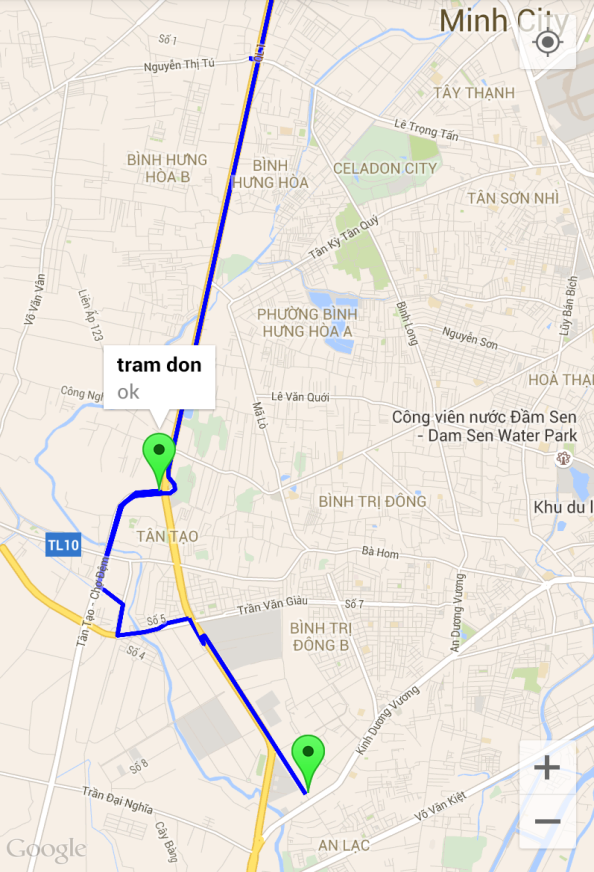
Bấm vào tên hành khách sẽ hiển thị thông tin hành khách cần thiết cho tài xế biết.



Hình 3‑27 Giao diên thông tin hành khách

#### Xem bản đồ

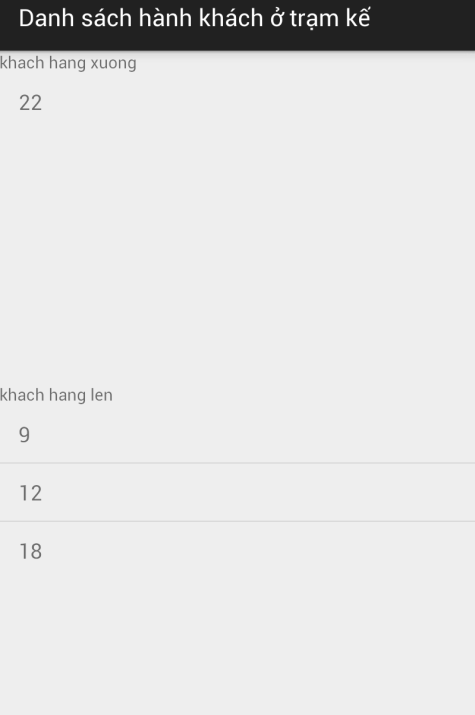
Chức năng cho phép tài xế xem trên bản đồ những trạm nào có khách, hiển thì đường đi của lộ trình, xem được vị trí xe đang chạy.



Hình 3‑28 Giao diện coi trên bản đồ chuyến đi

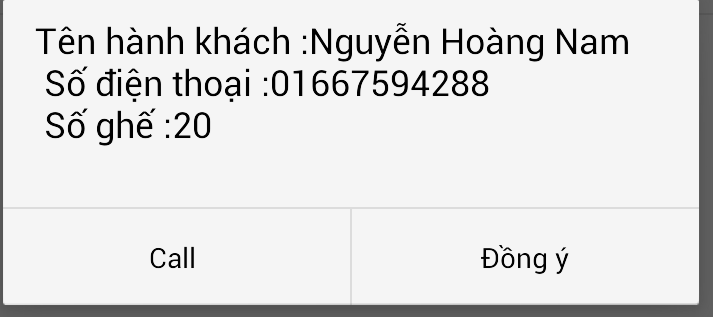
#### Nhận thông báo và xem danh sách hành khách ở trạm dừng

Chức năng này giúp tài xế không quên trạm dừng, khi gần tới một trạm có khách lên hoặc xuống sẽ hiện ra thông báo, và danh sách hành khách lên hoặc xuống ở trạm đó. Danh sách hiện theo số ghế để dễ dàng gọi hành khách lên hoặc xuống trạm.



Hình 3‑29 Giao diên danh sách hành khách lên xuống ở trạm

Bấm vào tên hành khách hiện ra thông tin và có chức năng gọi điện thoại tới số hành khách đã đăng ký.



Hình 3‑30 Giao diện thông tin hành khách ở trạm

#### Cảnh báo vi phạm tốc độ

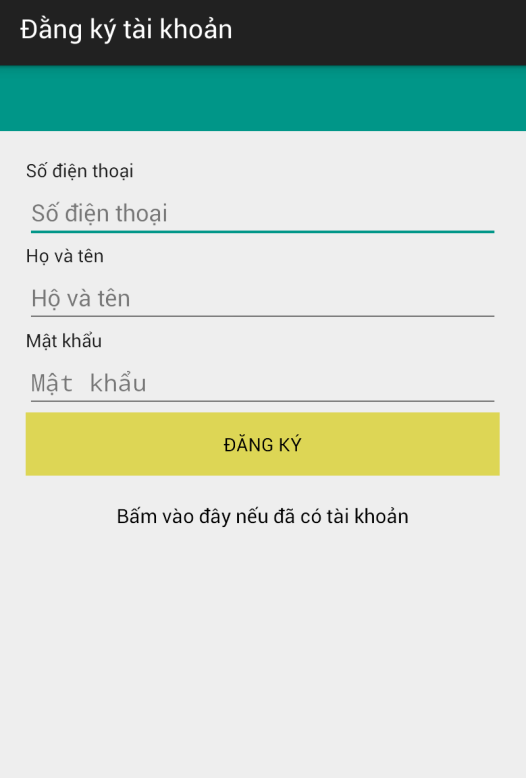
Chức năng này sẽ báo động khi xe chạy quá tốc độ cho phép đồng thời gửi vi phạm về cho quản lý.

### Mô tả chức năng đối với hành khách

#### Đăng ký và đăng nhập

Chức năng cho phép bạn đăng ký tài khoản mới hoặc đang nhập vào ứng dụng nếu như đã có tài khoản.

* Đăng ký : khai báo số điện thoại , họ tên, và mật khẩu để tạo tài khoản mới.



Hình 3‑31 Giao diện đăng ký tài khoản của hành khách

* Đăng nhập : Đăng nhập vào ứng dụng sử dụng số điện thoại và mật khẩu đã đăng ký.



Hình 3‑32 Giao diện đăng nhập của hành khách

* Tự động đăng nhập : nếu trước đó bạn đã đăng nhập một lần, ứng dụng sẽ tự kiểm tra và tự động đăng nhập lại cho bạn. muốn thoát khỏi tài khoản hiện tài chọn đăng xuất.

#### Kiểm tra chuyến đi trong ngày

Sau khi đăng nhập vào ứng dụng nếu không có chuyên đi nào trong ngày sẽ hiển thị giao diện thông báo không có chuyến đi trong ngày. Nếu kiểm tra có chuyến đi đã đăng ký trong ngày sẽ hiển thị thông tin chuyến đi cần thiết cho hành khách.

Nếu có chuyến đi trong ngày:

* Xem được thông tin về chuyến đi.



Hình 3‑33 Giao diện chuyến đi trong ngày của hành khách

* Kiểm tra được vị trí hiện tại của xe khi bấm vào nút cập nhập chuyến đi sẽ biết được : Vị trí hiện tại của xe, khoảng cách từ xe tới trạm đón và trạm xuống của bạn



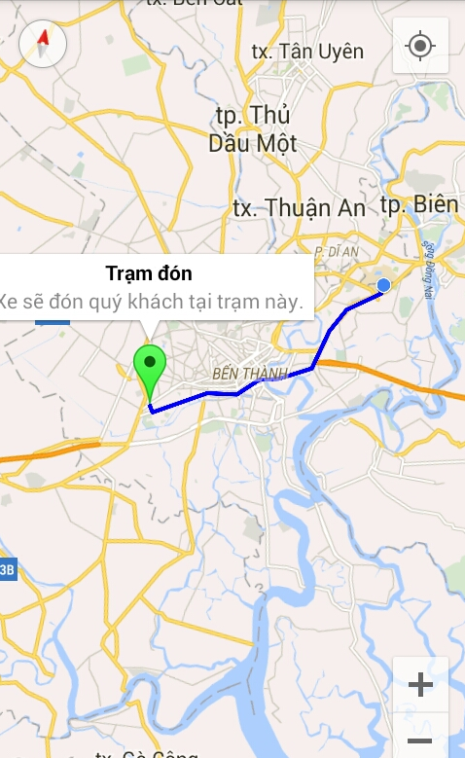
Hình 3‑34 Giao diện kiểm tra vị trí hiện tại của xe

* Nhận thông báo khi xe gần tới: bạn sẽ nhận được thông báo khi xe còn cách trạm đón khoảng 20km để có thể kịp chuẩn bị mọi thứ trước khi lên xe.
* Gửi ý kiến phản hồi: Gửi phản hồi về thái độ của nhân viên, và xe có có dừng đúng trạm không.



Hình 3‑35 Giao diện gửi ý kiến phản hồi cho nhà xe

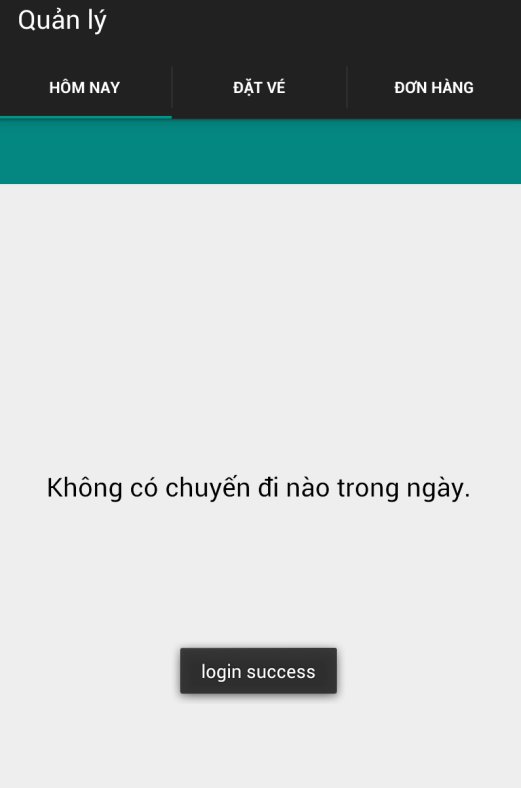
* Coi bản đồ: Hiện bản đồ giúp hành khách coi bản đồ để biệt được vị trí hiện tại, vị trí trạm đón và chỉ đường tới trạm đón.



Hình 3‑36 Giao diện bản đồ chỉ đường tới trạm đón cho hành khách

Nếu không có chuyến đi trong ngày.

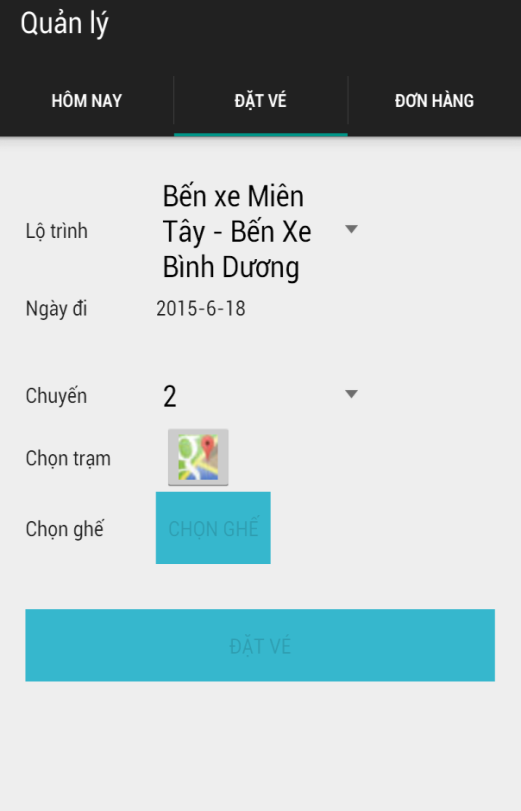
* Hiện thông báo không có chuyến đi trong ngày.



Hình 3‑37 Giao diện thông báo không có chuyến đi

#### Đặt vé

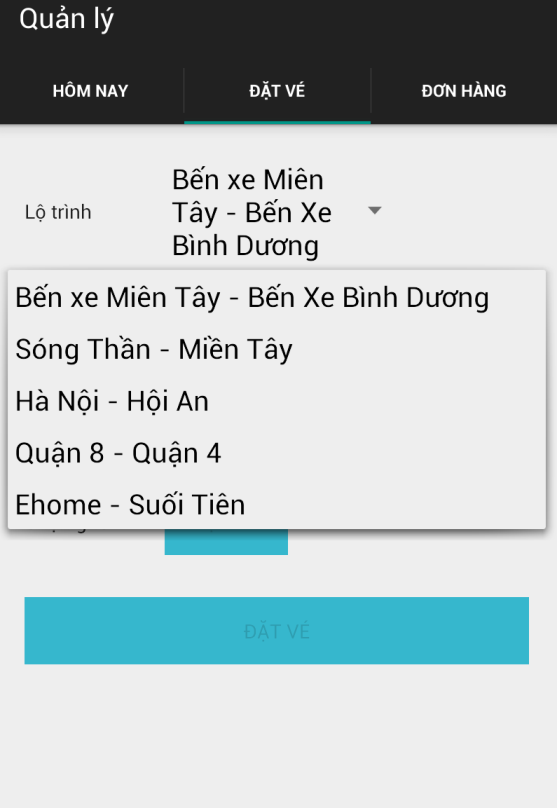
Chức năng cho phép người dung đặt vé theo lộ trình, ngày đi, chuyến đi, chọn trạm đón và xuống và số ghế:



Hình 3‑38 Giao diện đặt vé

* Chọn lộ trình.

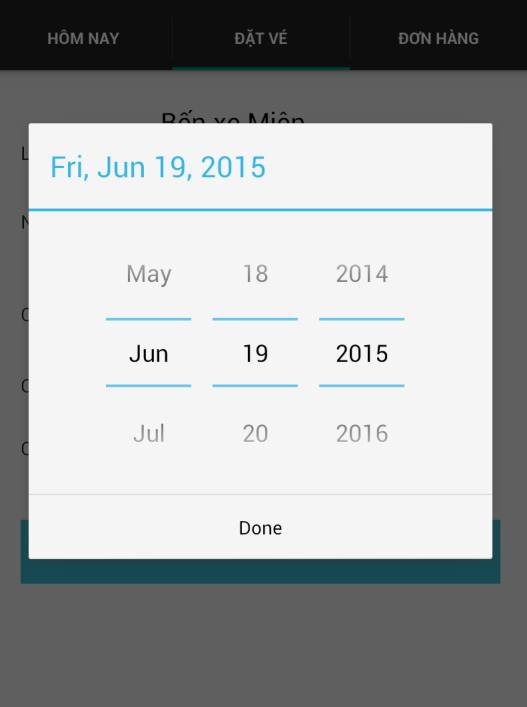
Bấm chọn lộ trình sẽ hiển thị ra danh sách lộ trình cho bạn chọn.



Hình 3‑39 Giáo diện chọn lộ trình

* Chọn ngày đi.

Bấm chọn ngày đi sẽ hiển thị ra cửa sổ để chọn ngày. Chọn ngày bạn muốn đi rùi bấm Done để hoàn tất.



Hình 3‑40 Giao diện chọn ngày đi

* Chọn chuyến đi.

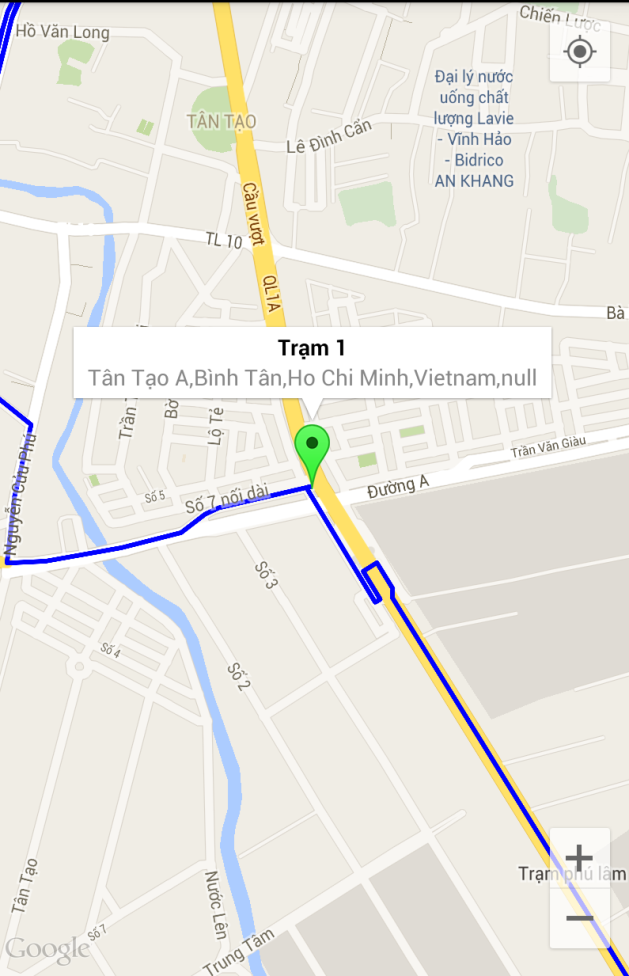
Bấm vào chọn chuyến đi sẽ hiển thị các chuyến đi cho bạn chọn.



Hình 3‑41 Giao diện chọn chuyến đi

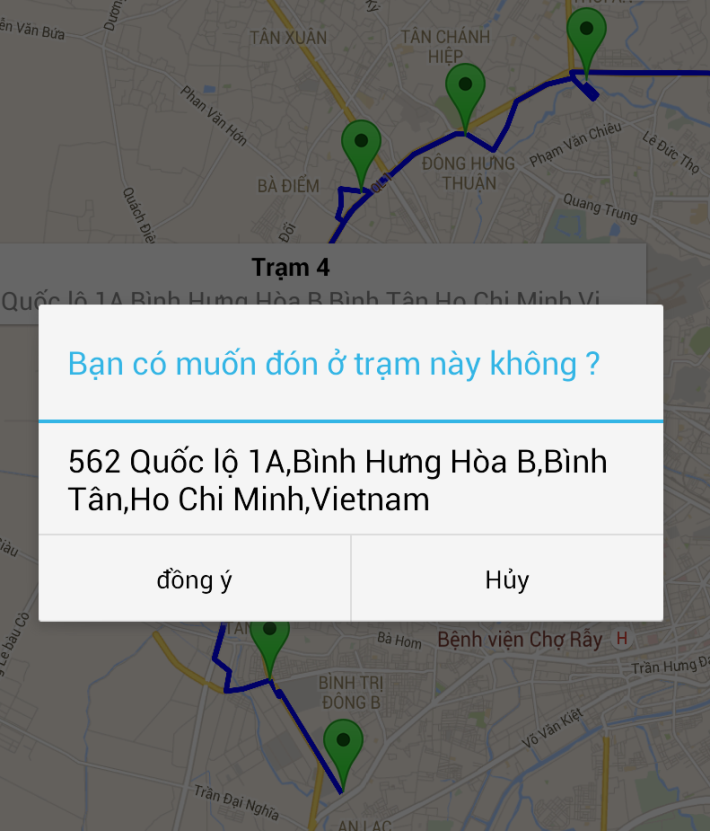
* Chọn trạm.

Khi bấm chọn trạm sẽ ra một bản đồ hiển thị vị trí, tên và địa điểm trạm để bạn dễ dàng nhận biết nơi muốn đón và đến của mình .Việc chọn trạm sẽ theo thứ tự trạm đón và trạm xuống.



Hình 3‑42 Giao diện bản đồ chọn trạm

Chọn trạm đón: bấm giữ vào trạm bạn muốn đặt sẽ xuất hiện thông báo hỏi bạn có để kiểm tra lại bạn có chắc chọn trạm đó không.



Hình 3‑43 Giao diện chọn trạm đón

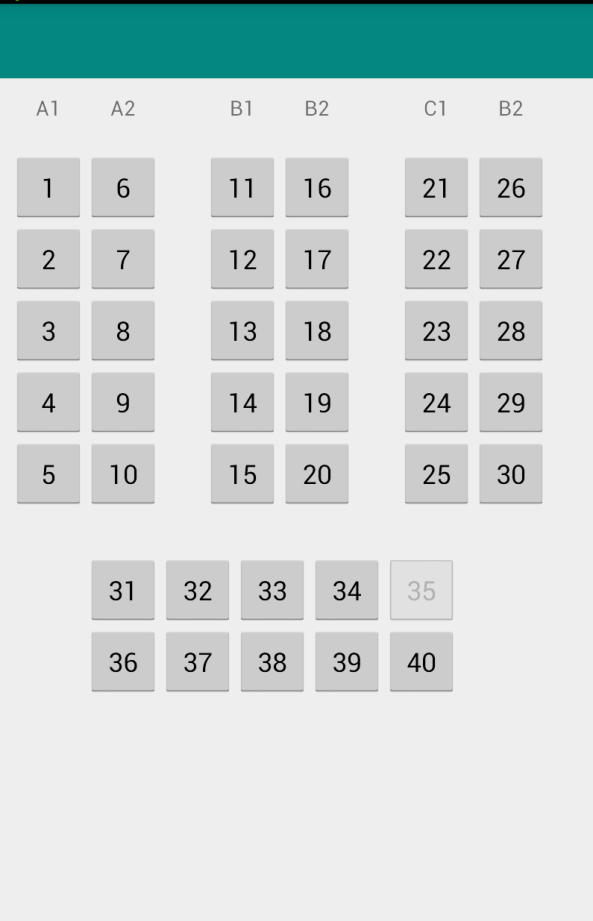
Chọn trạm xuống: Sau khí đã xác nhận chọn trạm đón, màn hình sẽ tự chuyển đến bến cuối của lộ trình để bạn dễ chọn tram xuống hơn. Cũng tương tự như bạn nhấn và giữ để chọn trạm muốn xuống, ứng dụng sẽ hiện thông báo xác nhận lại.



Hình 3‑44 Giáo diện chọn trạm xuống

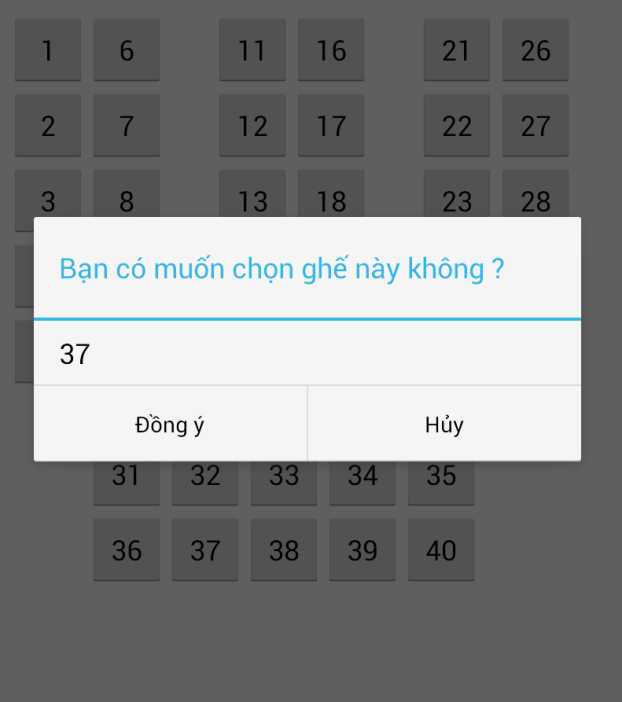
* Chọn ghế.

Hiển thị các ghế bạn có thể đặt, những vé đã đặt rùi sẽ ẩn xuống và không cho chọn. Bấm vào chọn ghế sẽ hiển thị thông báo để chắc rằng bạn chọn ghế đó.



Hình 3‑45 Giáo diện chọn ghế

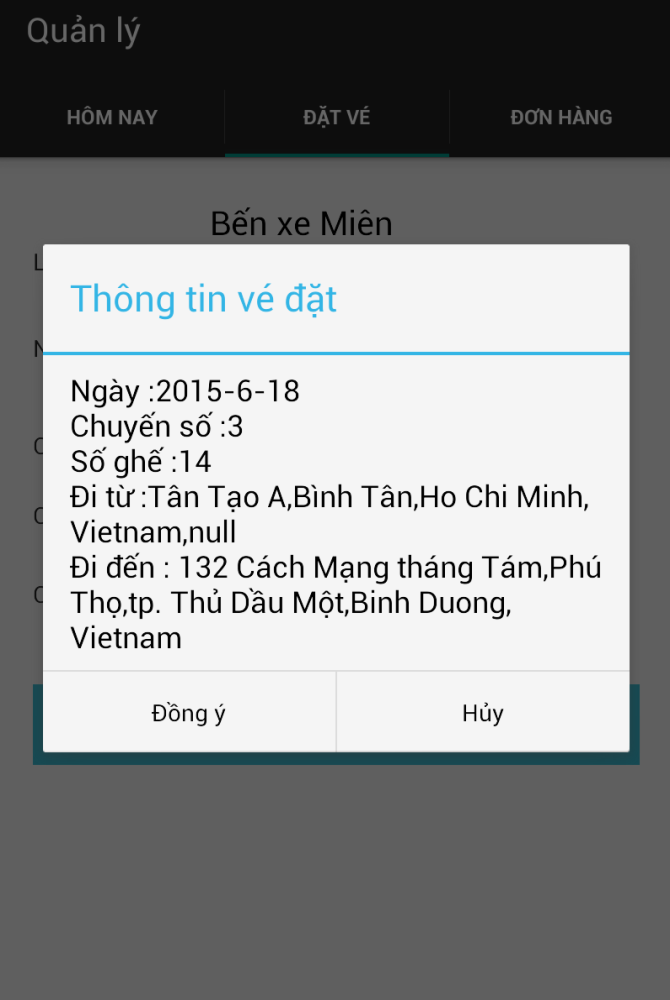
Thông báo hiện ra xác nhận ghế bạn chọn, bấm đồng ý để chấp nhận chọn ghế đã đặt, bấm hủy để chọn lại ghế khác.



Hình 3‑46 Giao diện xác nhận chọn ghế

* Đặt vé.

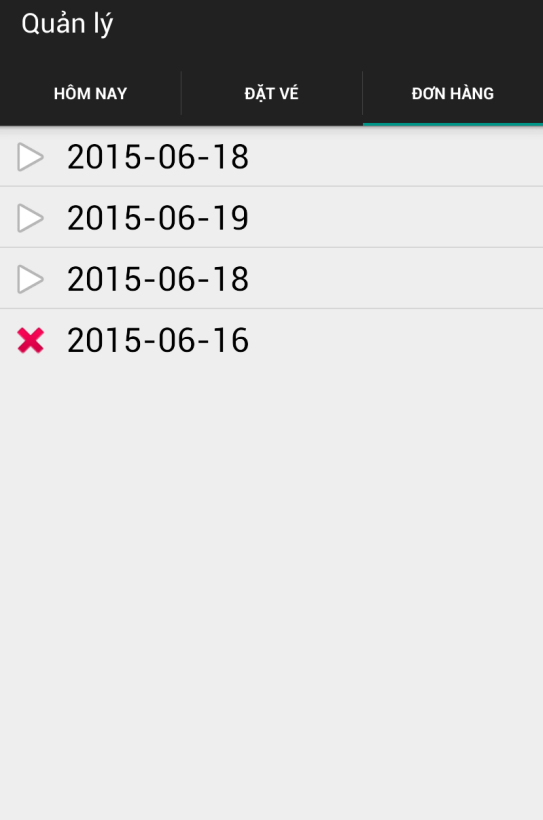
Khi đã hoàn tất chọn thông tin cho chuyến đi bạn muốn đặt bạn sẽ bấm và nút đặt vé sẽ hiện lại thông tin bạn đã chọn trước đó giúp bạn dễ dàng kiểm tra lại. Bấm đồng ý nếu bạn đã chắc chắn và hoàn tất quá trình đăng ký vé.



Hình 3‑47 Giao diện hiện thông tin vé đặt

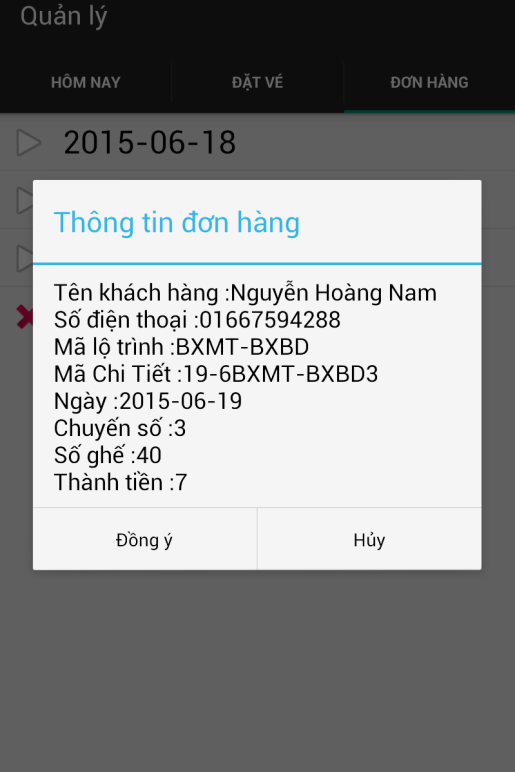
#### Xem danh sách đơn hàng

Cho phép bạn xem tất cả đơn hàng bạn đã đặt, những đơn hàng có dấu x là đã qua ngày, đơn hàng có dấu mũi tên là những chuyến chưa hoặc đang đi.



Hình 3‑48 Giao diện danh sách đơn hàng

Khi chọn một đơn hàng sẽ thấy được thông tin của đơn hàng đó.



Hình 3‑49 Giao diện hiện thông tin đơn hàng

# QUÁ TRÌNH XÂY DỰNG HỆ THỐNG

## Cài đặt bộ công cụ Android SDK.

Để có thể xây dựng một ứng dụng sử dụng Google Maps API. Chúng ta cần cài đặt tối thiểu những gói sau đây:

Bộ Tool SDK:

* + Android SDK Tools.
  + Android SDK Platform-tools.
  + Android SDK Build-tools.
  + SDK Platform.
  + ARM EABI v7a System Image.

Những thư viện hỗ trợ:

* + Android Support Repository.
  + Android Support Library.

Google Play Service:

* + Google Repository.
  + Google Play services.

## Khai báo Google Play Service vào ứng dụng

Sau khi cài đặt bộ công cụ cùng với các gói hỗ trợ. Chúng ta khởi tạo project và khai báo sử dụng Google Play Services trong project đó. Để khai báo chúng ta thêm đoạn sau vào file AndroidManifest.xml

<meta-data android:name=”com.google.android.gms.version”

android:value=”@interger/google\_play\_services\_version”/>

## Đăng ký sử dụng API key với Google

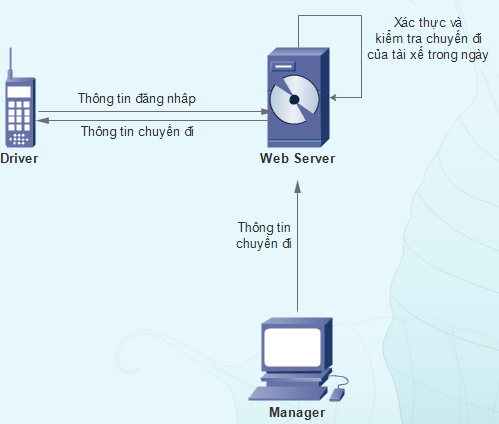
Để có thể sử dụng Google map api ta cần phải khai báo Maps Api key vào ứng dụng. Key này được google cung cấp miễn phí cho người dùng. Google thông qua việc khai báo key để theo dõi và kiểm soát ứng dụng của người dùng.

Với tài khoản Gmail, người dùng có truy cập vào trang phát triển của google <https://console.developers.google.com> để lấy key và sử dụng.

## Xây dựng ứng dụng trên android.

### Chức năng của tài xế:

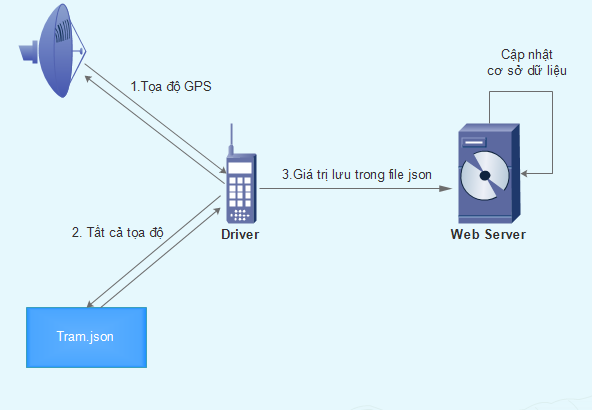
**Đăng nhập và liệt kê chuyến đi có trong ngày.**



Hình 4‑1 Mô hình mô phỏng chức năng đăng nhập và liệt kê chuyến đi có trong ngày

* Tài xế đăng nhập vào hệ thống. Tài xế sẽ gửi tên đăng nhập và mật khẩu.
* xác thực tài khoản tài xế của web server
* Tìm kiếm chuyến đi trong ngày
* Web Server sẽ tìm những chuyến đi trong có trong ngày và trả về cho tài xế biết.
* Nếu không có chuyến đi nào thì tài xế có thể sử dụng chức năng thêm lộ trình mới cho hệ thống.

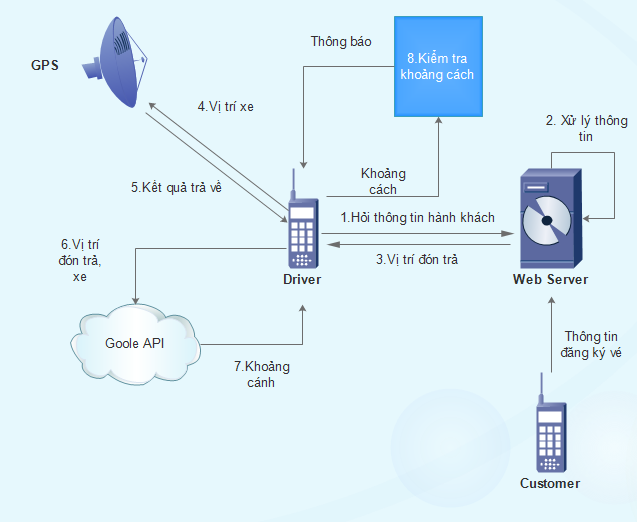
**Chức năng thêm lộ trình mới.**



Hình 4‑2 Mô hình mô phỏng chức năng thêm lộ trình mới

* Khi tài xế không có chuyến đi trong ngày mới có thể thực hiện chức năng này. Tài xế sẽ đi khảo sát từng điểm trên lộ trình mới.
* Lấy tọa độ hiện tại sử dụng GPS: Tài xế sử dụng GPS để lấy tọa độ của các điểm có thể làm trạm dừng trong lộ trình mới.
* Lưu các tọa độ tram vào bộ nhớ máy: Lưu trữ tọa độ của vị trí có thể làm trạm dừng trong một file json được lưu trong điện thoại.
* Gửi file Json chứa tọa độ các trạm lấy được lên Server: Sau khi kết thúc quá trình khảo sát trạm. Tài xế sẽ gửi dữ liệu trong file json cho Web Server
* Server sẽ cập nhập và lưu lại vào cở sở dữ liệu.

**Thông báo đón trả hành khách**

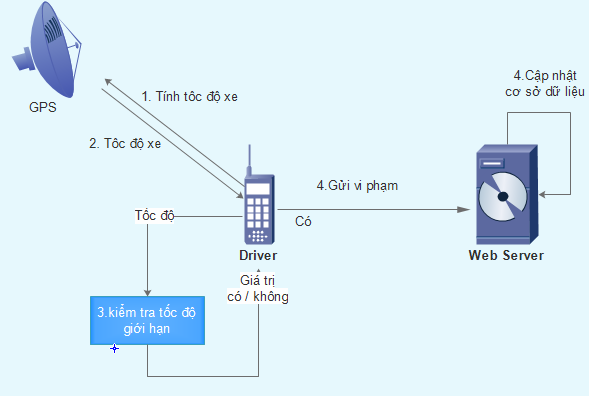


Hình 4‑3 Mô hình mô phỏng chức năng thông báo đón trả khách

Chức năng này được tạo để phục vụ việc đón trả hành khách trở nên thuận tiện hơn. Tài xế không cần phải nhớ các vị trí cần đón trả của hành khách. Khi xe gần tới vị trí của trạm mà có hành khách lên (hoặc xuống) thì hệ thống sẽ tự động thông báo cho tài xế biết.

* Tài xế đăng nhập và yêu cầu thông tin chuyến đi từ WebServer.
* WebServer trả thông tin của chuyến đi trong đó có thông tin trạm đón, trạm trả khách.
* Yêu cầu tọa độ của xe thông qua hệ thống GPS.
* Lấy được vị trí tọa độ . Sau đó thiết bị android của tài xế sẽ gửi giá trị của tọa độ vị trí của xe và tọa độ đón trả của hành khách lên cho Google service.
* Google service sẽ xử lý thông tin và trả về khoảng cách giữa xe và trạm đón khách.
* Thiết bị android của tài xế sẽ tiếp nhận giá trị khoảng cách từ Google service và kiểm tra. Nếu gần tới trạm có hành khách lên (xuống), thiết bị sẽ thông báo trạm nào có hành khách lên (xuống), có bao nhiêu hành khách lên (xuống), số ghế của hành khách ...

**Tự động tính toán tốc độ và báo cáo về cho webserver**

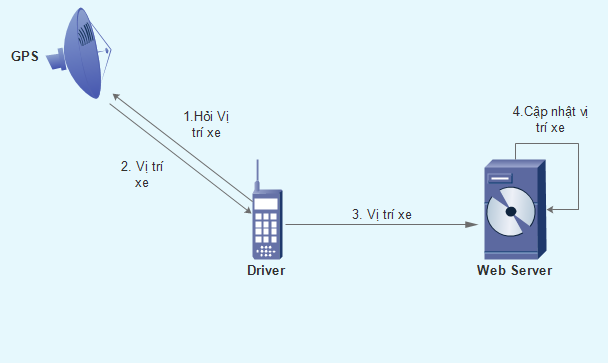


Hình 4‑4 Mô hình mô phỏng chức năng tự báo vi phạm tốc độ

Tự động tính toán tốc độ và báo cáo về cho webserver: chức năng này giúp cho nhà quản lý có thể kiểm soát sai phạm về tốc độ của tài xê. Khi tài xế vượt quá tốc độ cho phép. Thiết bị android của tài xế sẽ tự động gửi thông tin sai phạm về Web Server.

* GPS hiện nay đã hỗ trợ người dùng có thể đo được vận tốc của mình. Vì vậy, thiết bị android dễ dàng lấy được chúng.
* Thiết bị lấy được giá trị tốc độ của xe. Sau đó tự kiểm tra với giới hạn tốc độ.
  + Nếu quá tốc độ cho phép sẽ báo động và gửi lỗi về Server
* Server nhận lỗi và lưu vào cơ sở dữ liệu đợi xử lý.

**Cập nhật vị trí xe**



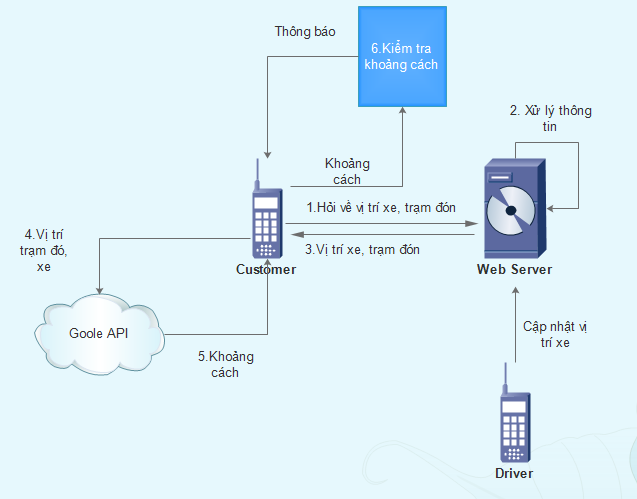
Hình 4‑5 Mô hình chứ năng câp nhập vị trí xe

Chức năng này giúp cho hành khách chủ động hơn trong thời gian của mình khi biết chính xác hiện giờ xe đang ở đâu.

* Lấy tọa độ hiện tại thông qua GPS.
* Gửi tọa độ hiện tại lên cho Server.
* Server cập nhập cơ sở dữ liệu về vị trí hiện tại của xe.

### Chức năng hành khách.

**Tự động nhận thông báo khi xe tới.**

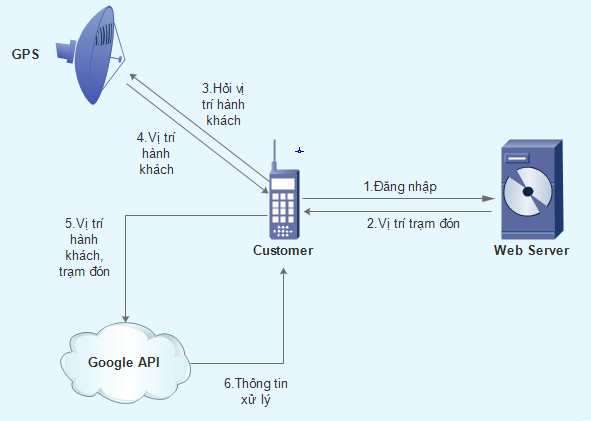


Hình 4‑6 Mô hình mô phỏng chức năng thông báo khi xe tới

Chức năng này giúp cho hành khác chủ động hơn khi biết khoảng cách của xe.

* Hành khách gửi yêu cầu lấy vị trí hiện tại của xe lên Server.
* Server sẽ xử lý thông tin hành khách gửi lên và trả lại vị trí hiện tại của xe, vị trí trạm đón của hành khách.
* Thiết bị android của hành khách sẽ gửi vị trí hiện tại của xe, vị trị trạm đó của hành khách lên cho Google service xử lý. Dữ liệu về khoảng cách giữa hai điểm đó sẽ được trả về.
* Thiết bị android xe lấy dữ liệu trả về và kiểm tra. Nếu xe gần tới sẽ thông báo cho hành khách biết.

**Chỉ đường đến trạm đón xe.**

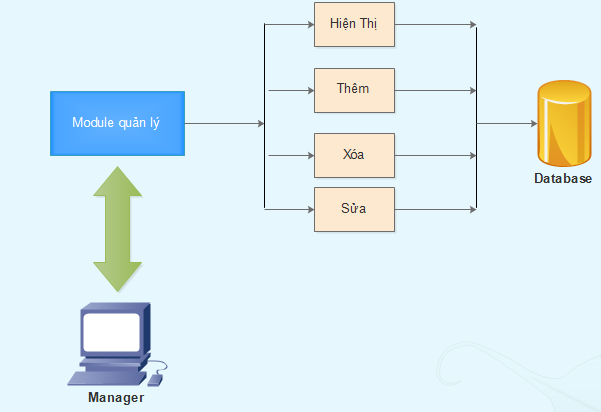


Hình 4‑7 Mô hình mô phỏng chức năng chỉ đường tới trạm xe

* Hành khách đăng nhập gửi thông tin về Web Server. Web Server xử lý thông tin và gửi về trạm đón mà hành khách đã đăng ký.
* Thiết bị android của hành khách sẽ lấy vị trí hiện tại của hành khách thông qua hệ thống GPS.
* Gửi vị trí trạm đón và vị trí hiện tại của hành khách lên Google Service xử lý. Thiết bị android nhận được thông tin xử lý sau đó hiển thị lên bản đồ vẽ chỉ đường cho hành khách.

## Xây dựng ứng dụng web.

### Chức năng của quản lý

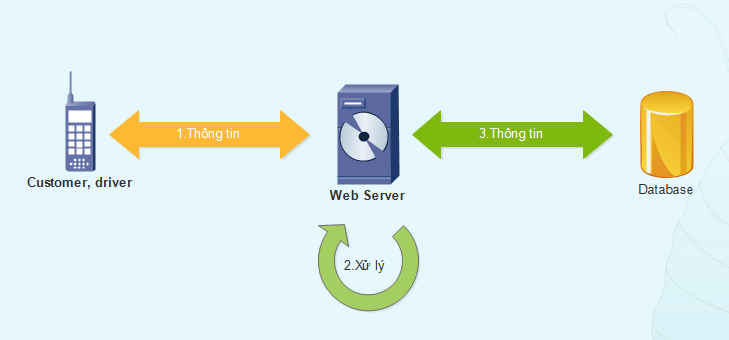


Hình 4‑8 Mô hình chứ năng quản lý của quản lý

Ứng dụng Web của quản lý cung cấp các chức năng để quản lý ( xóa, sửa, thêm) dữ liệu cho cở sở dữ liệu.

## Xây dựng nhận thông tin cần xử lý và xử lý trên webserver

### Chức năng của Web Server.



Hình 4‑9 Mô hình chứn ăng của Web Server

* Webserver sẽ tiếp nhận thông tin yêu cầu từ các thiết bị của hành khách và tài xế.
* Xử lý thông tin, thực thi yêu cầu sau đó trả thông tin đã được xử lý cho thiết bị android của hành khách và tài xế.

# TỔNG KẾT VÀ HƯỚNG PHÁT TRIỂN

## Kết quả đạt được

Sau một thời gian nghiên cứu và thực hiện, nhóm đã xây dựng một hệ thống quản lý xe khách chạy trên nền tảng Android với nhiều tính năng phù hợp với thực tế giúp cho việc đi lại và quản lý xe khách chở nên dễ dàng hơn. Qua quá trình nghiên cứu và phát triển chúng em đã đạt được những kết quả sau:

* Hiểu biết thêm về hệ điều hành Android, kiến trúc của Android, lập trình ứng dụng trên Android.
* Hiểu hơn về lập trình web, tạo server bằng ngôn ngữ lập trình php.
* Hiển biết thêm về Google Apis Add-on và dịch vụ định vị toàn cầu (GPS ) của Google.
* Phát triển được hệ thống có các chức năng cần thiết cho người dùng và quản lý :
  + Đặt vé trên mạng.
  + Thông báo khi xe gần tới trạm dừng.
  + Xem được vị trí hiện tại của xe.
  + Báo cáo vi phạm của xe cho quản lý.
  + Ứng dụng được thiết kế với giao diện trực quan, tiện dụng, tường minh.

## Hạn chế

Vì hạn chế về thời gian và khả năng, nhóm em nhận thấy những hạn chế trong hệ thống như sau:

* Giao diện không bắt mắt, chưa thân thiện với người sử dụng.
* Quá phụ thuộc vào Google Maps api, hệ thống có thể chập chạm khi Google bị lỗi và trả phản hồi chậm.
* Định vi GPS có sai số lớn gây khó khăn trong việc xác định trạm của lộ trình . Ví dụ: nếu lấy địa điểm trạm bên chiều đường bên này dễ trở thành địa điểm bên đường đối diện, gây sai trong việc vẽ lộ trình trên Google Maps.

## Những khó khăn khi xây dựng hệ thống

Trong lúc nghiên cứu và phát triển hệ thống nhóm đã gặp những khó khăn sau:

* Lần đầu tiếp xúc và phát triển ứng dụng trên Android.
* Chương trình giả lập thiết bị Android chưa sát với thực tế.
* Phải kiểm tra tốc độ bằng GPS nên chương trình giả lập không đáp ứng được.
* Khó khăn trong việc chạy thử hệ thống do không có đầy đủ thiết bị.
* Việc lấy tọa độ các trạm bằng GPS có sai số dẫn đến sai lệch về đường đi.
* Do không đi xa được nên chỉ xây dựng các trạm dừng trong thành phố và trên google maps.

## Hướng phát triển đề tài

Đề tài có thể được định hướng phát triển định hướng thêm các chức năng phục vụ cho việc chỉ đường cho tài xế lẫn hành khác, thay đổi lộ trình cho quản lý, cũng như cải thiện giao diện trở nên đẹp và trực quan hơn. Các chứng năng chi tiết như sau:

* Nâng cấp chức năng trả đón khách của tài xế bằng giọng nói. Sẽ có trạm có khách xuống (hoặc lên), trạm không có không có khách xuống (hoặc lên), thiết bị android của tài xế sẽ thông báo bằng giọng nói cho biết trạm tới có bao nhiêu người lên (hoặc xuống) và số ghế của họ.
* Trong nhiều trường hợp, xe khách có thể đi vào những đoạn đường bị ùn tắt. Thiết bị của tài xế có chức năng thông báo kẹt xe và ứng dụng web của quản lý có thể điều chỉnh lại lộ trình mới để cho các chuyến đi sau không phải đi vào đường đã bị kẹt.
* Một khi mà số lượng hành khách quá nhiều mà số lượng xe không đủ đáp ứng , ta có thể cải thiện thuật toán đặt vé trở nên tối ưu và tiện lợi cho hành khách. Cách đặt vé này tương tự với việc đặt vé tàu. Nhưng ưu điểm ở đây là xe nhiều hơn và dễ dàng đặt vé hơn vé tàu. Điều này giảm bớt sự lãng phí, khi đi ít nhưng vẫn phải trả tiền nhiều.

Ví dụ:

* Lộ trình từ Tp Hồ Chí Minh tới Tp Hà nội.
* Hành khách A muốn đi từ Tp Hồ Chí Minh về Đà Nẵng nhưng không kiếm được chuyến nào khác từ chuyến này.
* Hành khách B tương tự hành khách A và muốn đi Quảng Bình tới Hà Nội.
* Hệ thống cho phép hành khách A có thể đặt ghế số 1 từ Hồ Chí Minh đi Đà Nẵng. Hành khách B có thể đặt cùng ghế số 1 từ Quảng Bình ra tiếp Hà Nội. Xe sẽ trả hành khách A tại Đà Nẵng và đón hành khách B tại Quảng Bình để tiếp tục lộ trình.
* Hành khách A và B sẽ chia nhau khoảng tiền từ Đà Nẵng tới Quảng Bình.
* Làm được việc này cần phải có thuật toán tìm kiêm lộ trình tối ưu, để đề xuất lộ trình tốt nhất của từng người.
* Hệ thống phải kết hợp thanh toán trực tuyến trên thiết bị android của hành khách.

# DANH MỤC TÀI LIỆU THAM KHẢO

**Sách tham khảo**

1. G. Allen, *Beginning Android 4*. New York: Apress, 2012.
2. Tyler Anderson, *Build a Web service with PHP*, Stexar Corp, 2006.
3. S. Komatineni, D. MacLean and E. Franchomme, *Pro Android 4*. [New York]: Apress, 2012.
4. R. Richards*, Pro PHP XML and Web services*. Berkeley, CA: Apress, 2006.
5. John Wiley & Sons,*System Analysic and Design*, Inc, 2012.
6. M. Purvis, J. Sambells and C. Turner, *Beginning Google maps applications with PHP and Ajax*. Berkeley: Apress, 2006.
7. C. Pitt, *Pro PHP MVC*. Dordrecht: Springer, 2012.

**Tài liệu online**

1. Android: <http://developer.android.com>
2. Android opera System: <http://android.com>
3. Google API: <http://developers.google.com>
4. My SQL: <http://mysql.com>
5. Stackoverflow: <http://stackoverflow.com>
6. Codeigniter: <http://codeigniter.com>
7. HTML, PHP, CSS: <http://w3schools.com>
8. Bootstrap: <http://getbootstrap.com>