TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

──────── \* ───────

ĐỒ ÁN

**TỐT NGHIỆP ĐẠI HỌC**

NGÀNH CÔNG NGHỆ THÔNG TIN

**TÊN ĐỀ TÀI**

**Xây dựng hệ thống cung cấp dịch vụ hỗ trợ lái xe**

Sinh viên thực hiện : **Nguyễn Trọng Giáp**

Lớp CNTT2.04 – K57

Giáo viên hướng dẫn : **TS**. **Nguyễn Nhất Hải**

HÀ NỘI 12-2017

PHIẾU GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

1. Thông tin về sinh viên

* Họ và tên sinh viên: Nguyễn Trọng Giáp
* Điện thoại liên lạc: 01667019187 Email: [20121604@student.hust.edu.vn](mailto:20121604@student.hust.edu.vn)
* Lớp: CNTT2.04 - K57 Hệ đào tạo: Đại học chính quy
* Đồ án tốt nghiệp được thực hiện tại: Bộ môn Công nghệ phần mềm, Viện Công nghệ thông tin và Truyền thông, Đại học Bách Khoa Hà Nội.
* Thời gian làm ĐATN: Từ 9/2017 đến 12/2017.

1. Mục đích nội dung của ĐATN

* Xây dựng hệ thống cung cấp dịch vụ hỗ trợ lái xe.

1. Các nhiệm vụ cụ thể của ĐATN

* Khảo sát nhu cầu người dùng về hệ thống cung cấp thông tin dịch vụ cho các phương tiện giao thông.
* Khảo sát các ứng dụng, hệ thống trên thị trường có chức năng tương tự.
* Phân tích các chức năng cần có của hệ thống.
* Tìm hiểu các công nghệ và giải pháp cần có của hệ thống.
* Phân tích, thiết kế và xây dựng hệ thống.

1. Lời cam đoan của sinh viên:

* Tôi *Nguyễn Trọng Giáp* cam kết ĐATN là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của *TS.* *Nguyễn Nhất Hải*.
* Các kết quả nêu trong ĐATN là trung thực, không phải là sao chép toàn văn của bất kỳ công trình nào khác.

|  |  |
| --- | --- |
|  | *Hà Nội, ngày 20 tháng 12 năm 2017*  Tác giả ĐATN  *Nguyễn Trọng Giáp* |

1. Xác nhận của giáo viên hướng dẫn về mức độ hoàn thành của ĐATN và cho phép bảo vệ:

|  |  |
| --- | --- |
|  | *Hà Nội, ngày 20 tháng 12 năm 2017*  Giáo viên hướng dẫn  *TS. Nguyễn Nhất Hải* |

TÓM TẮT NỘI DUNG ĐỒ ÁN TỐT NGHIỆP

Cùng với sự bùng nổ về công nghệ thông tin và sự phổ biến của các thiết bị di động thông minh, các ứng dụng trên nền tảng thiết bị di động ngày càng đa dạng, được chú trọng phát triển. Chúng ta có thể dễ dàng kể ra rất nhiều các tiện ích mà chúng mang lại cho con người. Người dùng ngày càng có xu hướng tìm kiếm sự hỗ trợ của các ứng dụng công nghệ thông tin trong các công việc hằng ngày, trong đó được sử dụng nhiều nhất là các ứng dụng tra cứu thông tin. Các ứng dụng này giúp cho chúng ta nâng cao hiệu quả trong công việc cũng như tiết kiệm rất nhiều thời gian.

Hiện nay, trước tình hình khó khăn trong giao thông do vấn đề mật độ dân cư cao, hạ tầng giao thông hạn chế, đặc biệt tại Việt Nam, nhu cầu dễ dàng đi lại và được hỗ trợ trong giao thông là hết sức cần thiết. Các phương tiện giao thông ngày càng nhiều, đi kèm với nó là các dịch vụ liên quan đến các phương tiện này. Trong khi chủ các cửa hàng dịch vụ hay nhà cung cấp không biết cách nào để đưa thông tin dịch vụ của mình đến tài xế, thì ngược lại các lái xe vẫn đang còn loay hoay tìm kiếm thông tin từ những người dân đi đường mỗi khi xe gặp vấn đề, và không phải lúc nào cũng nhận được câu trả lời mong muốn.

Chính vì nhu cầu cấp thiết đó, đồng thời nhận thấy sự phổ biến của thiết bị di động thông minh được sử dụng bởi người tham gia giao thông, để tận dụng được thế mạnh mà nó mang lại, yêu cầu đặt ra cần xây dựng một hệ thống giúp cho nhà cung cấp cũng như tài xế có thể tra cứu và tiếp cận các thông tin một cách dễ dàng.

Mục đích đồ án là xây dựng một hệ thống trợ giúp tra cứu và cung cấp các dịch vụ cho người tham gia giao thông. Từ việc nghiên cứu, phân tích các hệ thống tương tự đã có trên thị trường, hệ thống mà sinh viên đề xuất cố gắng đi vào một thiết kế tổng thể, tích hợp nhiều dịch vụ, đồng thời cung cấp một số dịch vụ hữu ích cho người dùng.

*Nội dung báo cáo được trình bày gồm những phần chính sau:*

**Chương 1:** *Đặt vấn đề.*

Nội dung chương 1 tập trung vào giới thiệu bài toán, tìm hiểu yêu cầu và đưa ra định hướng giải pháp công nghệ cho bài toán.

**Chương 2:** *Công nghệ sử dụng.*

Nội dung chương 2 trình bày tóm tắt về các công nghệ được sử dụng trong hệ thống.

**Chương 3:** *Khảo sát và mô hình hóa yêu cầu.*

Nội dung chương 3 dựa vào các khảo sát nhu cầu người dùng và tiến hành phân tích chức năng của hệ thống.

**Chương 4:** *Phân tích, thiết kế, xây dựng hệ thống và đánh giá kết quả.*

Nội dung chương 4 trình bày các phân tích thiết kế của hệ thống, thiết kế cấu trúc dữ liệu và xây dựng phát triển hệ thống.

**Chương 5:** *Kết luận và hướng phát triển.*

Chương 5 sẽ trình bày tóm tắt các nội dung mà đồ án đã đạt được, đồng thời đánh giá ưu và nhược điểm của hệ thống cùng với định hướng giải pháp trong tương lai.

MỤC LỤC

[PHIẾU GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP 2](#_Toc501533357)

[TÓM TẮT NỘI DUNG ĐỒ ÁN TỐT NGHIỆP 3](#_Toc501533358)

[MỤC LỤC 5](#_Toc501533359)

[DANH MỤC HÌNH VẼ 8](#_Toc501533360)

[DANH MỤC BẢNG 11](#_Toc501533361)

[DANH MỤC TỪ VIẾT TẮT 12](#_Toc501533362)

[CHƯƠNG 1. ĐẶT VẤN ĐỀ 13](#_Toc501533363)

[1.1 Giới thiệu bài toán 13](#_Toc501533364)

[1.2 Tóm tắt một số định hướng giải pháp công nghệ 13](#_Toc501533365)

[1.2.1 Kiến trúc hệ thống 13](#_Toc501533366)

[1.2.2 Hệ thống cơ sở dữ liệu 14](#_Toc501533367)

[1.2.3 Các giải pháp công nghệ liên quan đến bản đồ 14](#_Toc501533368)

[1.2.4 Giải pháp cho tính năng thông báo – notification 14](#_Toc501533369)

[1.2.5 Giải pháp cho tính năng xác thực qua điện thoại 15](#_Toc501533370)

[1.2.6 Giải pháp cho tính năng lưu trữ và quản lý tập tin 15](#_Toc501533371)

[CHƯƠNG 2. NGHIÊN CỨU CÁC GIẢI PHÁP CÔNG NGHỆ 16](#_Toc501533372)

[2.1 Các dịch vụ liên quan đến bản đồ - Google Maps API 16](#_Toc501533373)

[2.1.1 Kết nối với bản đồ Google map - Android API 17](#_Toc501533374)

[2.1.2 Truy xuất các thông tin về địa điểm - Places API 18](#_Toc501533375)

[2.1.3 Định hướng đường đi - Directions API 19](#_Toc501533376)

[2.1.4 Chỉ dẫn đường đi - Roads API 19](#_Toc501533377)

[2.2 Dịch vụ thông báo notification - Google Firebase Cloud Messaging 21](#_Toc501533378)

[2.3 Giải pháp nhắn tin xác thực - Nexmo Verify API 22](#_Toc501533379)

[2.4 Amazon S3 23](#_Toc501533380)

[2.5 SQL và NoSQL 25](#_Toc501533381)

[2.6 Giải pháp lập trình phía server - Kiến trúc MVP 26](#_Toc501533382)

[2.7 Giải pháp thiết kế giao diện - Material Design 27](#_Toc501533383)

[CHƯƠNG 3. KHẢO SÁT VÀ XÁC ĐỊNH YÊU CẦU BÀI TOÁN 29](#_Toc501533384)

[3.1 Khảo sát một số hệ thống hiện có 29](#_Toc501533385)

[3.2 Phân tích yêu cầu của bài toán 31](#_Toc501533392)

[3.2.1 Phân tích người dùng (user) của hệ thống 31](#_Toc501533393)

[3.2.2 Biểu đồ use case tổng quan 31](#_Toc501533394)

[3.2.3 Biểu đồ use case phân rã 33](#_Toc501533395)

[3.2.4 Quy trình nghiệp vụ 35](#_Toc501533396)

[3.3 Phân tích chi tiết yêu cầu người dùng (users) 37](#_Toc501533397)

[3.3.1 Đặc tả use case UC001 – Đăng ký tài khoản 37](#_Toc501533398)

[3.3.2 Đặc tả use case UC002 – Đăng nhập 38](#_Toc501533399)

[3.3.3 Đặc tả use case UC003 – Tìm kiếm dịch vụ 38](#_Toc501533400)

[3.3.4 Đặc tả use case UC004 – Thêm dịch vụ 39](#_Toc501533401)

[3.3.5 Đặc tả use case UC005 – Khóa người dùng 40](#_Toc501533402)

[3.3.6 Đặc tả use case UC006 – Khóa dịch vụ 40](#_Toc501533403)

[3.4 Các yêu cầu khác 41](#_Toc501533404)

[3.4.1 Tính khả thi 41](#_Toc501533405)

[3.4.2 Hiệu suất 41](#_Toc501533406)

[3.4.3 Độ tin cậy 41](#_Toc501533407)

[CHƯƠNG 4. THIẾT KẾ, XÂY DỰNG VÀ ĐÁNH GIÁ HỆ THỐNG 42](#_Toc501533408)

[4.1 Mô tả chung 42](#_Toc501533409)

[4.2 Biểu đồ tương tác 42](#_Toc501533410)

[4.2.1 Biểu đồ tương tác cho use case UC001 – Đăng ký tài khoản 42](#_Toc501533411)

[4.2.2 Biểu đồ tương tác cho use case UC002 – Đăng nhập. 43](#_Toc501533412)

[4.2.3 Biểu đồ tương tác cho use case UC003 – Tìm kiếm dịch vụ. 43](#_Toc501533413)

[4.2.4 Biểu đồ tương tác cho use case UC004 – Thêm dịch vụ. 44](#_Toc501533414)

[4.2.5 Biểu đồ tương tác cho use case UC005 – Khóa người dùng. 44](#_Toc501533415)

[4.2.6 Biểu đồ tương tác cho use case UC006 – Khóa dịch vụ. 45](#_Toc501533416)

[4.3 Biểu đồ cấu trúc – (package/component diagram) 45](#_Toc501533417)

[4.4 Biểu đồ lớp phân tích 47](#_Toc501533418)

[4.4.1 Biểu đồ lớp phân tích package đăng ký – Register 47](#_Toc501533419)

[4.4.2 Biểu đồ lớp phân tích package đăng nhập – Login 47](#_Toc501533420)

[4.4.3 Biểu đồ lớp phân tích package tìm kiếm dịch vụ - Search 48](#_Toc501533421)

[4.4.4 Biểu đồ lớp phân tích package quản lý dịch vụ - ServiceManagement 49](#_Toc501533422)

[4.4.5 Biểu đồ lớp phân tích package quản trị người dùng - UserAdministration 50](#_Toc501533423)

[4.4.6 Biểu đồ lớp phân tích package quản trị dịch vụ - ServiceAdministration 50](#_Toc501533424)

[4.5 Thiết kế lớp 51](#_Toc501533426)

[4.5.1 Biểu đồ lớp chi tiết cho chức năng đăng ký tài khoản 51](#_Toc501533427)

[4.5.2 Biểu đồ lớp chi tiết cho chức năng đăng nhập 51](#_Toc501533428)

[4.5.3 Biểu đồ lớp chi tiết cho chức năng tìm kiếm dịch vụ 52](#_Toc501533429)

[4.5.4 Biểu đồ lớp chi tiết cho chức năng tạo mới/sửa dịch vụ 53](#_Toc501533430)

[4.5.5 Biểu đồ lớp chi tiết cho chức năng quản trị người dùng 53](#_Toc501533431)

[4.5.6 Biểu đồ lớp chi tiết cho chức năng quản trị dịch vụ 54](#_Toc501533432)

[4.6 Thiết kế giao diện 55](#_Toc501533433)

[4.6.1 Giao diện màn hình đăng ký 55](#_Toc501533451)

[4.6.2 Giao diện màn hình đăng nhập 56](#_Toc501533452)

[4.6.3 Giao diện màn hình chi tiết người dùng và dịch vụ 56](#_Toc501533453)

[4.6.4 Giao diện màn hình tìm kiếm dịch vụ 57](#_Toc501533454)

[4.6.5 Giao diện màn hình lịch sử và danh sách đánh dấu 58](#_Toc501533455)

[4.6.6 Giao diệnmàn hình quản lý xe và thông báo 58](#_Toc501533456)

[4.6.7 Giao diện màn hình quản lý dịch vụ 59](#_Toc501533457)

[4.6.8 Giao diện màn hình quản lý dịch vụ 60](#_Toc501533458)

[4.7 Thiết kế mô hình dữ liệu 61](#_Toc501533459)

[4.7.1 Yêu cầu bài toán 61](#_Toc501533460)

[4.7.2 Biểu đồ thực thể liên kết 62](#_Toc501533461)

[4.7.3 Thiết kế cơ sở dữ liệu 63](#_Toc501533462)

[4.7.4 Thiết kế chi tiết bảng 63](#_Toc501533463)

[4.8 Đánh giá kết quả hệ thống 67](#_Toc501533464)

[CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 68](#_Toc501533465)

[5.1 Kết luận 68](#_Toc501533466)

[5.2 Hướng phát triển 68](#_Toc501533467)

[TÀI LIỆU THAM KHẢO 69](#_Toc501533468)

DANH MỤC HÌNH VẼ

[Hình 1: Google Map 16](#_Toc501533469)

[Hình 2: Google Maps APIs Details 17](#_Toc501533470)

[Hình 3: Maps Android API 17](#_Toc501533471)

[Hình 4: Places API 18](#_Toc501533472)

[Hình 5: Sử dụng Directions API 19](#_Toc501533473)

[Hình 6: Sử dụng Roads API kết hợp Directions API 20](#_Toc501533474)

[Hình 7: Firebase Cloud Messaging 21](#_Toc501533475)

[Hình 8: Nexmo APIs 22](#_Toc501533476)

[Hình 9: Xác thực người dùng 23](#_Toc501533477)

[Hình 10: Amazon S3 24](#_Toc501533478)

[Hình 11: Mô hình MVC 26](#_Toc501533479)

[Hình 12: Mô hình MVP 27](#_Toc501533480)

[Hình 13: Material Design - phong cách thiết kế giao diện mới của Google 28](#_Toc501533481)

[Hình 14: Ứng dụng iParking trên Play Store 29](#_Toc501533482)

[Hình 15: Ứng dụng PakMe trên Play Store 29](#_Toc501533483)

[Hình 16: Ứng dụng Free parking trên Play Store 30](#_Toc501533484)

[Hình 17: Use case tổng quan 32](#_Toc501533485)

[Hình 18: Use case phân rã chức năng quản lý xe 33](#_Toc501533486)

[Hình 19: Use case phân rã chức năng quản lý đánh dấu 33](#_Toc501533487)

[Hình 20: Use case phân rã chức năng quản lý lịch sử 34](#_Toc501533488)

[Hình 21: Use case phân rã chức năng quản lý dịch vụ 34](#_Toc501533489)

[Hình 22: Use case phân rã chức năng quản trị người dùng 34](#_Toc501533490)

[Hình 23: Use case phân rã chức năng quản trị dịch vụ 35](#_Toc501533491)

[Hình 24: Quy trình nghiệp vụ của tài xế 35](#_Toc501533492)

[Hình 25: Quy trình nghiệp vụ của nhà cung cấp 36](#_Toc501533493)

[Hình 26: Quy trình nghiệp vụ của quản trị viên 36](#_Toc501533494)

[Hình 27: Biểu đồ tương tác cho use case UC001 – Đăng ký tài khoản 42](#_Toc501533495)

[Hình 28: Biểu đồ tương tác cho use case UC002 – Đăng nhập 43](#_Toc501533496)

[Hình 29: Biểu đồ tương tác cho use case UC003 – Tìm kiếm dịch vụ 43](#_Toc501533497)

[Hình 30: Biểu đồ tương tác cho use case UC004 – Thêm dịch vụ 44](#_Toc501533498)

[Hình 31: Biểu đồ tương tác cho use case UC005 – Khóa người dùng 44](#_Toc501533499)

[Hình 32: Biểu đồ tương tác cho use case UC006 – Khóa dịch vụ 45](#_Toc501533500)

[Hình 33: Biểu đồ các package của client trong hệ thống 45](#_Toc501533501)

[Hình 34: Biểu đồ các package con trong Systems package 46](#_Toc501533502)

[Hình 35: Biểu đồ lớp phân tích cho package đăng ký 47](#_Toc501533503)

[Hình 36: Biểu đồ lớp phân tích cho package đăng nhập 47](#_Toc501533504)

[Hình 37: Biểu đồ lớp phân tích cho package tìm kiếm dịch vụ 48](#_Toc501533505)

[Hình 38: Biểu đồ lớp phân tích cho package quản lý dịch vụ 49](#_Toc501533506)

[Hình 39: Biểu đồ lớp phân tích cho package quản trị người dùng 50](#_Toc501533507)

[Hình 40: Biểu đồ lớp phân tích cho package quản trị dịch vụ 50](#_Toc501533508)

[Hình 41: Biểu đồ lớp chi tiết cho chức năng đăng ký tài khoản 51](#_Toc501533509)

[Hình 42: Biểu đồ lớp chi tiết cho chức năng đăng nhập 51](#_Toc501533510)

[Hình 43: Biểu đồ lớp chi tiết cho chức năng tìm kiếm dịch vụ 52](#_Toc501533511)

[Hình 44: Biểu đồ lớp chi tiết cho chức năng tạo mới/sửa dịch vụ 53](#_Toc501533512)

[Hình 45: Biểu đồ lớp chi tiết cho chức năng quản trị người dùng 53](#_Toc501533513)

[Hình 46: Biểu đồ lớp chi tiết cho chức năng quản trị dịch vụ 54](#_Toc501533514)

[Hình 46: Giao diện các màn hình đăng ký 55](#_Toc501533515)

[Hình 47: Giao diện màn hình đăng nhập 56](#_Toc501533516)

[Hình 48: Giao diện màn hình chi tiết người dùng và dịch vụ 56](#_Toc501533517)

[Hình 49: Giao diện màn hình tìm kiếm dịch vụ 57](#_Toc501533518)

[Hình 50: Giao diên màn hình lịch sử và danh sách đánh dấu 58](#_Toc501533519)

[Hình 51: Giao diên màn hình quản lý xe và thông báo 58](#_Toc501533520)

[Hình 52: Giao diện màn hình quản lý dịch vụ 59](#_Toc501533521)

[Hình 53: Giao diện màn hình quản trị của admin 60](#_Toc501533522)

[Hình 54: Biểu đồ thực thể liên kết 62](#_Toc501533523)

[Hình 55: Biểu đồ cơ sở dữ liệu 63](#_Toc501533524)

DANH MỤC BẢNG

[Bảng 1: SQL và NoSQL 25](#_Toc501533525)

[Bảng 2: Bảng chức năng của các ứng dụng được khảo sát 30](#_Toc501533526)

[Bảng 3: Bảng các use case tổng quan cho từng loại người dùng 32](#_Toc501533527)

[Bảng 4: Đặc tả use case UC001 – Đăng ký tài khoản 37](#_Toc501533528)

[Bảng 5: Đặc tả use case UC002 – Đăng nhập tài khoản 38](#_Toc501533529)

[Bảng 6: Đặc tả use case UC003 – Tìm kiếm dịch vụ 39](#_Toc501533530)

[Bảng 7: Đặc tả use case UC004 – Thêm dịch vụ 39](#_Toc501533531)

[Bảng 8: Đặc tả use case UC005 – Khóa người dùng 40](#_Toc501533532)

[Bảng 9: Đặc tả use case UC006 – Khóa dịch vụ 41](#_Toc501533533)

[Bảng 10: Bảng user 64](#_Toc501533534)

[Bảng 11: Bảng dịch vụ 64](#_Toc501533535)

[Bảng 12: Bảng type 65](#_Toc501533536)

[Bảng 13: Bảng nối service\_type 65](#_Toc501533537)

[Bảng 14: Bảng vehicle 65](#_Toc501533538)

[Bảng 15: Bảng notification 65](#_Toc501533539)

[Bảng 16: Bảng history 66](#_Toc501533540)

[Bảng 17: Bảng tag 66](#_Toc501533541)

[Bảng 18: Bảng activation 66](#_Toc501533542)

DANH MỤC TỪ VIẾT TẮT

|  |  |
| --- | --- |
| SDK | Software Development Kit. |
| API | Application Programming Interface. |
| FCM | Firebase Cloud Messaging. |
| Amazon S3 | Amazon Simple Storage Service |
| MVP | Model - View - Presenter. |
| MVC | Model - View - Controller. |
| GPS | Global Positioning System. |
| SQL | Structured Query Language. |
| RDBMS | Relational Database Management System. |
| CRUD | Create - Read - Update - Delete |
| AWS | Amazon Web Service |

# ĐẶT VẤN ĐỀ

## Giới thiệu bài toán

Tại các thành phố lớn đông dân cư, nhu cầu đi lại ngày càng gia tăng. Chính vì thế các dịch vụ đi kèm đành cho phương tiện đi lại cũng không kém phần quan trọng như đậu xe, sửa xe, xăng dầu, ăn nghỉ cho tài xế lái xe…

Liệu nếu ra khỏi nhà đi chơi xa, xe cộ gặp vấn đề thì quanh đấy có điểm nào sửa xe không? Nếu đi ăn sáng ở một nhà hàng mới mở, liệu quanh đấy có điểm đậu xe không? Mặc dù thời đại hiện nay thông tin nhiều vô số nhưng đa số mọi người vẫn không khỏi lúng túng khi gặp các tình huống này, không biết phải giải quyết thế nào. Nguyên nhân có thể do không biết cách tìm kiếm, hoặc do thông tin quá nhiều mà không biết phải chọn lọc từ nguồn nào mới là chính xác.

Thêm nữa, gần đây đặc biệt được kể đến là chủ trương dọn hè phố ở thành phố Hà Nội đã làm tác động không ít đến các lái xe. Các điểm đỗ xe trái phép bị gỡ bỏ, các tài xế thiếu thông tin về điểm đậu xe mới làm xuất hiện việc đậu tạm trên đường nhỏ bị công an giao thông xử phạt. Những thông tin này đăng đầy trên các trang tin tức hằng ngày cho thấy nhu cầu cấp thiết về hệ thống cung cấp các thông tin cho các tài xế lái xe.

Cùng với đó, các ứng dụng trên thị trường vẫn chưa giải quyết được yêu cầu bài toán đặt ra, và nếu có thì cũng còn thiếu một số chức năng, chưa giải quyết hay cung cấp được các dịch vụ một cách toàn diện.

Từ đó, yêu cầu đặt ra là cần xây dựng một hệ thống tập trung giúp mọi người tìm kiếm thông tin về những địa điểm dành cho các dịch vụ hỗ trợ phương tiện giao thông một cách tiện lợi nhất. Tận dụng lợi thế về sự phổ biến của smartphone để giúp mọi người dễ dàng tiếp cận và biện pháp giúp mọi người có thể liên lạc tới cửa hàng, địa điểm đấy.

## Tóm tắt một số định hướng giải pháp công nghệ

### Kiến trúc hệ thống

Mục tiêu của đồ án tốt nghiệp là xây dựng một ứng dụng trên nền tảng thiết bị di động, trong giai đoạn đầu tập trung vào các thiết bị sử dụng hệ điều hành Android. Hệ thống hoạt động theo mô hình client-server. Trong đó:

* ***Client (frontend***): Là ứng dụng trên thiết bị Android, cung cấp cho cả ba người dùng là lái xe, nhà cung cấp và quản trị viên. Tùy theo vai trò của từng người dùng mà ứng dụng sẽ cung cấp các chức năng tương ứng. Ứng dụng đơn giản, dễ dùng và có thao tác nhanh gọn để có thể giúp các lái xe tiết kiệm thời gian cũng như không phải gặp nguy hiểm nếu đang lái xe.
* ***Server (backend)***: Được xây dựng dựa trên Framework Laravel (php) cung cấp các API để ứng dụng trên thiết bị di động có thể truy cập và lấy thông tin.

### Hệ thống cơ sở dữ liệu

Hệ thống cơ sở dữ liệu được thiết kế bao gồm một phần lưu trữ trên client, sử dụng hệ thống lưu trữ dữ liệu dưới dạng NoSQL, một phần khác, phía server sử dụng kiểu lưu trữ dữ liệu dạng SQL.

Phần phân tích thiết kế cơ sở dữ liệu sẽ được trình bày chi tiết trong chương tiếp theo.

### Các giải pháp công nghệ liên quan đến bản đồ

Ứng dụng trên thiết bị di động được thiết kế sao cho thân thiện và dễ dùng với người dùng, trên đó cho phép người dùng tìm kiếm và hiển thị thông tin trên bản đồ một cách trực quan cũng như thực hiện được tính năng dẫn đường, ước lượng thời gian, ước lượng khoảng cách… Để thực hiện điều đó, ứng dụng sẽ áp dụng, tích hợp một loạt các công cụ, giải pháp liên quan đến tìm kiếm và xử lý thông tin trên bản đồ thực, được cung cấp bởi Google mà cụ thể là nhóm giải pháp của Google Maps API [1], cụ thể:

* Dịch vụ ***Maps Android API***: Có chức năng nhúng bản đồ Google Maps vào ứng dụng. Từ đấy các thông tin của ứng dụng sẽ được hiển thị một cách trực quan trên bản đồ. [4]
* Dịch vụ ***Maps Directions API***: Có chức năng tính toán và trả về đường đi giữa các cặp vị trí. Chúng thường được sử dụng cho mục đích tìm đường đi giữa hai vị trí trên bản đồ.
* Dịch vụ ***Maps Roads API***: Có chức năng tinh chỉnh dữ liệu đưa vào để cho chúng là con đường mà người lái xe có khả năng đi qua nhiều nhất.
* Dịch vụ ***Places API***: Có chức năng tìm kiếm và truy xuất các thông tin bổ sung dựa vào thông tin tọa độ của một địa điểm.

Có thể nói, Google Maps là nền tảng đáng tin cậy và được sử dụng bởi rất nhiều nhà phát triển khi muốn cài đặt các tính năng liên quan đến bản đồ. Chính vì vậy, sinh viên đã quyết định sử dụng nền tảng này trong ứng dụng của mình.

### Giải pháp cho tính năng thông báo – notification

Một ứng dụng liên quan đến thiết bị thông tin di động cần phải có chức năng thông báo, cảnh báo cho người dùng. Trong đồ án nay, hệ thống mà sinh viên xây dựng cũng có chức năng gửi thông báo cần thiết tới người dùng, hệ thống sử dụng dịch vụ gửi thông báo Firebase Cloud Messaging.

*Firebase Cloud Messaging (FCM) là một dịch vụ miễn phí của Google. Thông qua FCM, nhà phát triển ứng dụng có thể gửi thông điệp một cách nhanh chóng, an toàn tới các thiết bị cài đặt ứng dụng của họ.* [2]

### Giải pháp cho tính năng xác thực qua điện thoại

Hệ thống không chỉ cho phép tra cứu các dịch vụ mà còn nhắm tới mục tiêu cung cấp các dịch vụ (có phí hay miễn phí) tới người dùng. Chính vì vậy, yêu cầu của hệ thống đòi hỏi phải thực hiện tốt việc xác thực người dùng. Một giải pháp tốt và hay được các hệ thống hiện nay thường tiến hành là xác thực dựa trên số điện thoại di động cá nhân. Để thực hiện tính năng này, hiện nay có một số giải pháp của bên thứ 3, trong đó giải pháp dùng Nexmo API là một lựa chọn giải pháp tốt, đáp ứng được yêu cầu bài toán.

*Nexmo API là dịch vụ bên thứ ba cung cấp các dịch vụ truyền thông. Cho phép nhà phát triển có thể xây dựng các ứng dụng truyền thông như tin nhắn SMS, đàm thoại, chat… Và đặc biệt trong dịch vụ xác thực người dùng thông qua số điện thoại.* [3]

### Giải pháp cho tính năng lưu trữ và quản lý tập tin

Vì hệ thống cho phép cung cấp các dịch vụ liên quan đến đỗ xe, rửa xe… trong đó yêu cầu cung cấp các hình ảnh, mô tả cụ thể, do đó cần phải có giải pháp tốt để lưu trữ các tập tin, các file hình ảnh. Để thực hiện điều đó, hệ thống sẽ sử dụng giải pháp do AWS (Amazon Web Service) cung cấp, cụ thể là sử dụng dịch vụ Amazon S3 (Simple Storage Service) lưu trữ và quản lý tập tin (thường là dạng dạng ảnh) mà người dùng gửi lên hệ thống.

Nhược điểm của việc lưu trữ file trên host của website là rất tốn băng thông của hệ thống khi chúng ta có nhiều hình ảnh hoặc nhiều lượt truy cập. Việc khó backup nếu dung lượng ngày càng lớn cũng như tốn tài nguyên tính toán CPU của máy chủ vì nó sẽ phải thực hiện việc tìm kiếm, dò file được lưu. Hiện nay, Amazon S3 là giải pháp tốt, hay được dùng cho vấn đề này bởi các đặc điểm ưu trội của nó.

Phần giới thiệu, phân tích chi tiết hơn về các giải pháp công nghệ sẽ được trình bày ở chương tiếp theo.

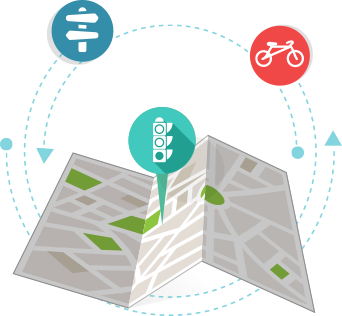
# NGHIÊN CỨU CÁC GIẢI PHÁP CÔNG NGHỆ

Như đã trình bày tóm tắt trong phần phân tích một số định hướng giải pháp công nghệ ở trên, Chương này sẽ tập trung vào phân tích và giải thích rõ các giải pháp đó. Bao gồm:

1. Nhóm các giải pháp liên quan đến dịch vụ bản đồ, sử dụng các dịch vụ của *Google Maps API.*
2. Giải pháp gửi thông báo (notification) đến người dùng cùng với *Google* *Firebase Cloud Messaging (FCM).*
3. Giải pháp xác thực qua tin nhắn điện thoại đi động cùng với *Nexmo API*.
4. Giải pháp lưu trữ và quản lý tập tin cùng với *Amazon S3*.

## Các dịch vụ liên quan đến bản đồ - Google Maps API

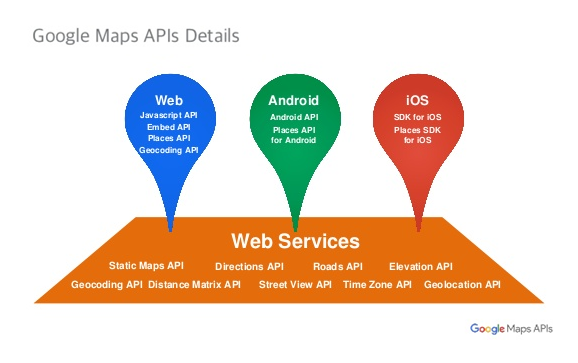
Google Map là dịch vụ bản đồ trực truyến được phát triển bởi Google. Nó cung cấp hình ảnh vệ tinh, bản đồ đường phố… với mục đích thay thế cho các loại bản đồ giấy thông thường. Dịch vụ bản đồ này tỏ ra vượt trội và được ưa chuộng nhất so với những dịch vụ bản đồ hiện tại, số lượng người sử dụng hàng tháng lên tới con số 1 tỷ người dùng trên 200 quốc gia.



Hình : Google Map

*[Nguồn: HUGE IT]*

Vào 6/2005, Google đã cho ra mắt Google Maps API, một dịch vụ miễn phí cho phép các nhà phát triển tích hợp Google Maps vào các trang web của họ.



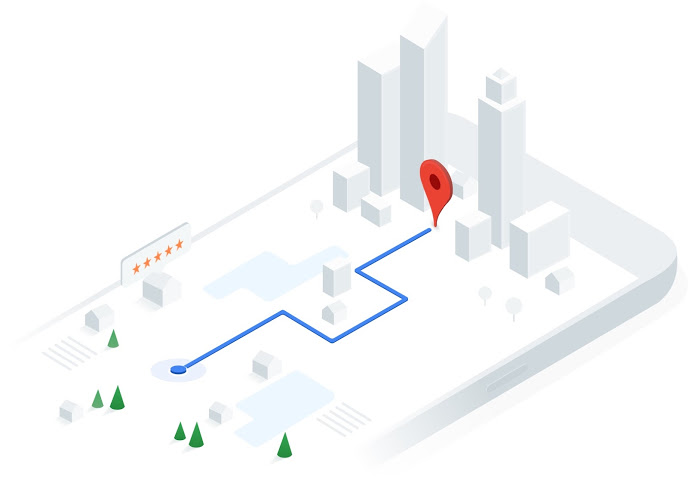
Hình : Google Maps APIs Details

*[Nguồn: Google Maps API]*

Google Maps API cung cấp rất nhiều các thư viện hỗ trợ mạnh mẽ cho nhà phát triển bao gồm các bộ SDK trên ứng dụng (như Place API, Android API, iOS SDK, Javascript API...) cùng với các Web Service API (như Directions API, Roads API...).

### Kết nối với bản đồ Google map - Android API

Google Maps API được Google hỗ trợ trên các nền tảng Web, Android và iOS. Đồ án sử dụng Maps Android API cho ứng dụng bên phía client trên các thiết bị thông tin di động. Google đã cung cấp các tài liệu hướng dẫn khá chi tiết giúp cho các nhà phát triển dễ dàng tìm hiểu và áp dụng.



Hình : Maps Android API

*[Nguồn: Google Maps API]*

*Với Google Maps Android API, chúng ta có thể thêm bản đồ dựa trên dữ liệu Google Maps vào ứng dụng. API tự động xử lý quyền truy cập vào các máy chủ của Google Maps, tải dữ liệu, hiển thị bản đồ và phản hồi từ các thao tác cử chỉ cảm ứng trên bản đồ. Chúng ta cũng có thể sử dụng các cuộc gọi API để thêm các điểm đánh dấu, đa giác và lớp phủ lên bản đồ và thay đổi khung nhìn đến một khu vục bản đồ cụ thể…* [1]

### Truy xuất các thông tin về địa điểm - Places API

Places API là một trong số các dịch vụ Web Service API của Google Maps API, giúp tìm kiếm và truy xuất các thông tin phong phú về các doanh nghiệp cũng các địa điểm nổi tiếng.



Hình : Places API

*[Nguồn: Google Maps API]*

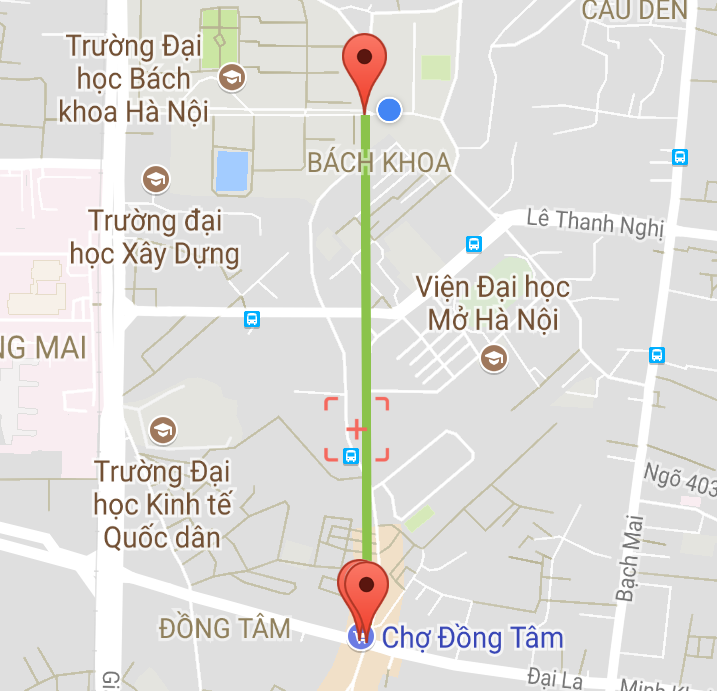
Với Places API chúng ta có thể lấy thông tin địa điểm từ một vị trí (được xác định thông qua) GPS hoặc ngược lại, tìm kiếm vị trí dựa vào thông tin từ một địa điểm (nhà hàng nổi tiếng, chỗ đậu xe, bệnh viện…). Điều này rất hữu ích khi đi kèm với Maps Andrioid API, khi đó chúng ta có thể hiển thị thông tin tại địa điểm chọn trên bản đồ cũng như các dịch vụ có sẵn trên Google Maps.

### Định hướng đường đi - Directions API

Để hiển thị đường đi từ một điểm đến một điểm khác, hệ thống lựa chọn sử dụng dịch vụ Directions API. Giống với Places API, Directions API cũng là một trong số các dịch vụ Web của Google Maps API, nó tính toán đường đi giữa các vị trí và trả về kết quả một tập các địa điểm GPS có thứ tự mà con đường đi qua khi nhận một yêu cầu từ người dùng.

Tập kết quả điểm GPS được trả về này đảm bảo khá chính xác tại các điểm rẽ của con đường. Tuy nhiên, thường kém chính xác hơn đối với con đường có các đoạn bị uốn khúc, khiến cho kết quả mà API trả về vẫn còn khuyết điểm.

Ví dụ: như trương hợp khi mà tìm đường đi từ “Bách Khoa” đến “Chợ Đồng Tâm”, dữ liệu Directions API không bám vào con đường cong “Trần Đại Nghĩa”.



Hình : Sử dụng Directions API

Direction API không trả về dữ liệu chính xác cho đoạn đường cong trên con đường khiến cho nhà phát triển khó có thể vẽ đúng đường đi lên bản đồ bởi nếu nối tập các điểm này lại có thể khiến cắt chéo các ngôi nhà.

Để khắc phục vấn đề này, chúng ta sẽ sử dụng thêm một dịch vụ nữa của Google Maps API là Roads API.

### Chỉ dẫn đường đi - Roads API

*“Roads Maps API xác định tuyến đường một chiếc xe đã đi dọc theo và cung cấp các dữ liệu bổ sung về con đường, chẳng hạn như giới hạn tốc độ.”*[1]

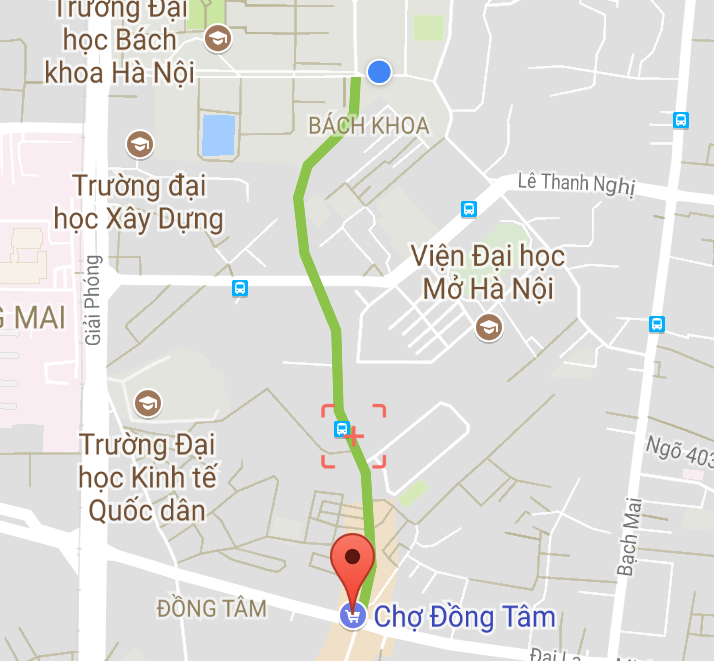
API được cung cấp thông qua một giao diện đơn giản của HTTPS với một số chức năng tra cứu sau:

* ***Snap to roads*:** Với một tập các tọa độ GPS, dịch vụ này trả về tập các tọa độ tương tự nhưng được hiệu chỉnh sao cho tập các điểm này nằm dọc theo con đường phù hợp nhất mà một chiếc xe đi qua. Dịch vụ này có thể tiếp nhận lên đến 100 điểm và còn cung cấp tùy chọn cho phép chúng ta lấy tọa độ con đường theo đường cong nội suy.
* ***Nearest roads*:** Với một tập các tọa độ GPS, dịch vụ này trả về vị trí nằm trên con đường gần nhất cho mỗi điểm. Cũng có thể tiếp nhận lên đến 100 điểm. Không như snap to roads, tập điểm trả về không phải là một đường dẫn liên tục.
* ***Speed limits*:** Dịch vụ này trả về tốc độ giới hạn cho một đoạn đường. Dịch vụ giới hạn chỉ dành gói Google Maps API Premium.

Với vấn đề Direction API không trả về đúng trên các con đường cong như đã nêu ở trong mục [2.1.3], chúng ta có thể sử dụng bằng cách sử dụng kết hợp cả hai dịch vụ này:

* Gọi một API đến Directions API để lấy ra một tập các điểm GPS nằm trên con đường.
* Với tập điểm thu được, ta thêm vào giữa 2 điểm liên tiếp thêm một vài điểm nhằm mục đích tăng mật độ điểm trên con đường.
* Gọi một API đến Roads API với tùy chọn “*Snap to roads***”** để hiệu chỉnh các điểm đã thêm ở bước 2, giúp cho các điểm này *“bám”* vào con đường phù hợp nhất cho đường đi chiếc xe. Như vậy chúng ta sẽ thu về một con đường phù hợp hơn.

Với vấn đề gặp phải ở bài toán tìm đường đi từ “Bách Khoa” đến “Chợ Đồng Tâm”, sau khi áp dụng hướng giải quyết này, chúng ta đã thu được về con đường “cong” như ý muốn.



Hình : Sử dụng Roads API kết hợp Directions API

Tuy nhiên, việc sử dụng kết hợp 2 API này cũng có nhược điểm:

* Dữ liệu phản hồi chậm, bởi kết quả được hiển thị sau 2 bước gọi API lần lượt, và như thế sẽ tạo ra độ trễ khá đáng kể.
* Tại một số trường hợp, Roads API có thể nhận dạng sai con đường, cũng sẽ trả về kết quả không hợp lý. Bởi vậy, khi kết hợp 2 API này sẽ cho ta kết quả hợp lý hơn, nhưng không phải với mọi trường hợp.

## Dịch vụ thông báo notification - Google Firebase Cloud Messaging

*“Firebase Cloud Messaging (FCM) là một giải pháp thông báo đa nền tảng cho phép chúng ta gửi thông báo một cách đáng tin cậy và không mất chi phí.”* [2]

FCM bao gồm hai thành phần chính cho việc gửi và nhận:

* Môi trường tin cậy như Cloud Functions cho Firebase hoặc một ứng dụng server để tạo, và gửi tin nhắn.
* Thiết bị iOs, Android, Web đóng vai trò client để nhận tin nhắn.

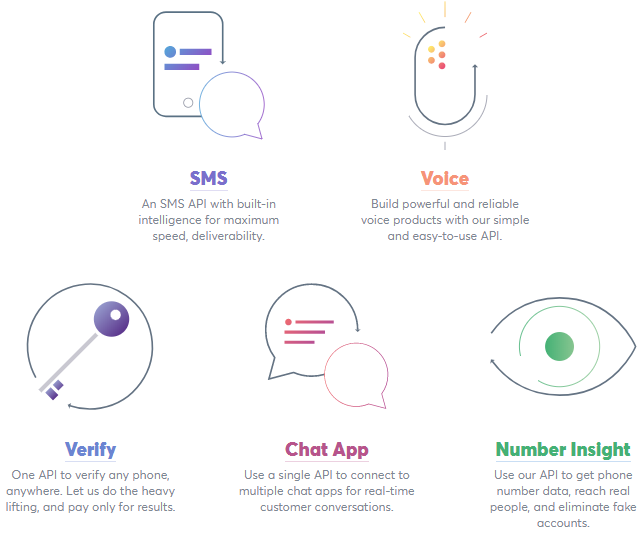
Hình : Firebase Cloud Messaging

Ứng dụng phía server hoặc môi trường server tin cậy gửi gói tin yêu cầu đến các máy chủ FCM qua Admin SDK hoặc HTTP và XMPP API, sau đó các máy chủ FCM sẽ gửi tin nhắn cho các ứng dụng client đang chạy trên thiết bị của người dùng.

Để phân biệt giữa các client với nhau, mỗi FCM khi trả về thông tin cho client sẽ tạo ra một token riêng. Khi muốn gửi thông báo đến một client, ứng dụng server cần đính kèm token trong gói tin gửi cho FCM. Ngoài ra, FCM còn cung cấp gom nhóm client theo từng chủ đề (*topic*). Mỗi client có thể đăng ký nhiều chủ đề cùng một lúc. Khi FCM được lựa chọn gửi theo chủ đề, những client có cùng chủ đề sẽ được gửi thông báo.

## Giải pháp nhắn tin xác thực - Nexmo Verify API

*Nexmo API* là API cung cấp các chức năng truyền thông. Cho phép nhà phát triển có thể xây dựng các ứng dụng truyền thông như tin nhắn SMS, đàm thoại, chat… Nexmo API cung cấp khá nhiều các giải pháp. Hình bên dưới mô tả các giải pháp được Nexmo hỗ trợ.



Hình : Nexmo APIs

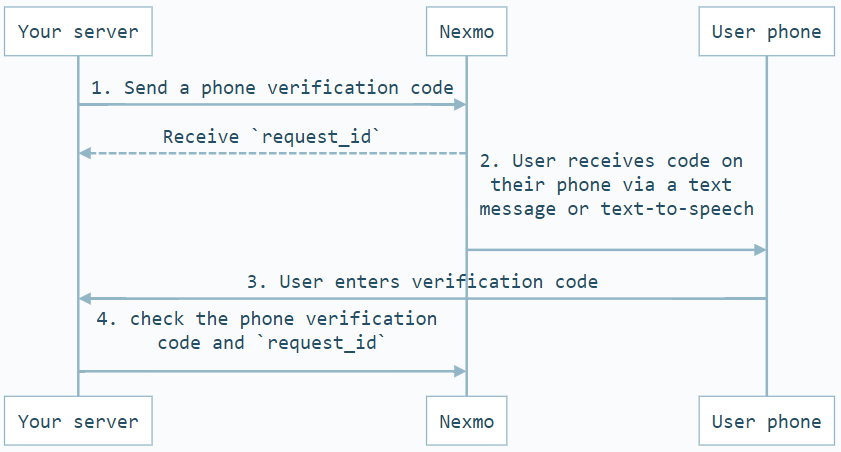
*[Nguồn: Nexmo API]*

Trong số các API được kể ở trên, Verify cũng là một API rất mạnh mẽ và có trả phí của Nexmo. Với API này, nhà phát triển có thể xác thực người dùng ở bất kỳ số điện thoại nào, ở bất kỳ đâu. API này hỗ trợ SDK cho rất nhiều ngôn ngữ.

Verify API cho phép chúng ta gửi một mã PIN bằng tin nhắn SMS và điện thoại để xác thực người dùng theo số điện thoại cụ thể.

*Verify API làm việc như thế nảo?*

* Khi một người gửi yêu cầu xác thực lên server của bạn, server của bạn sử dụng số điện thoại của người dùng gửi gọi một API đến Nexmo để yêu cầu xác thực số điện thoại.
* Nexmo sẽ trả về cho server bạn một id (request\_id) về request của bạn, đồng thời gửi mã PIN đến điện thoại người dùng thông qua tin nhắn hoặc cuộc gọi thoại.
* Khi nhận được mã PIN, người dùng sử dụng nó để gửi xác nhận lên server của bạn.
* Server của bạn sử dụng mã PIN mà người dùng gửi đến và request\_id lúc Nexmo trả về gọi một API đến Nexmo để xác thực xác thực số điện thoại
* Nexmo trả về kết quả cho server, server đối chiếu và cho phép người dùng hoàn thành xác thực.



Hình : Xác thực người dùng

*[Nguồn: Nexmo API]*

Với cơ chế xác thực rất chặt chẽ, Nexmo Verify API là một công cụ tuyệt vời cho việc xác thực thông qua số điện thoại.

Các chức năng Nexmo đều phải trả phí, tuy nhiên khi tạo tài khoản bạn sẽ được tặng 2 Euro cho người dùng mới, cũng khá đủ cho các nhà phát triển muốn kiểm tra qua các API của Nexmo.

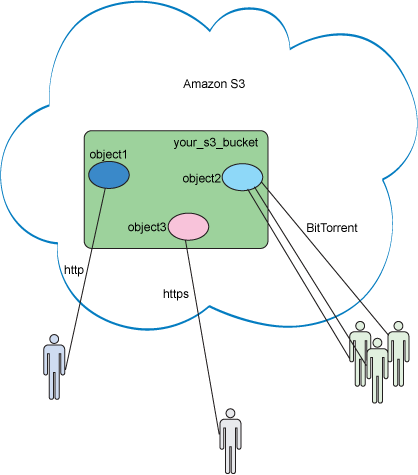
## Amazon S3

*Amazon S3 (Amazon Simple Storage Service)* là dịch vụ đám mây phục vụ cho việc lưu trữ. Chúng ta có thể tải lên các tệp, các tài liệu, các dữ liệu tải về của người dùng hoặc các bản sao lưu.

Amazon S3 cung cấp các giao diện dịch vụ Web cho việc lưu trữ và khôi phục dữ liệu. Dữ liệu có thể được lưu trữ ở bất kỳ loại nào và có thể được truy cập đến từ bất kỳ vị trí nào thông qua Internet. Bạn có thể lưu trữ không giới hạn một lượng lớn các đối tượng trong S3 với kích thước của mỗi đối tượng trong khoảng từ 1 Byte tới 5 GB. Dữ liệu có thể được lưu trữ ở vị trí địa lý mà bạn mong muốn. Dữ liệu bạn tạo ra được Amazon S3 lưu dưới định dạng “*buckets*”, một khái niệm mà Amazon S3 đưa ra, tương tự như khái niệm của thư mục trong hệ thống xử lý hằng ngày ta vẫn dùng.

*Các thành phần Amazon S3:*

* ***Bucket*:** là các thùng chứa cho các đối tượng. Bạn có thể tạo 1 hay nhiều bucket. Với mỗi bucket, bạn có thể điều khiển việc truy xuất đến nó (ai có thể tạo, xóa và xem các object trong bucket). Tuy nhiên, mỗi một tài khoản chỉ tạo được tối đa là 100 buckets, và mỗi bucket có tên là duy nhất.
* ***Object*:** là đối tượng chứa dữ liệu được lưu trữ trong các bucket ở S3. Các đối tượng được coi như là các tệp mà bạn muốn lưu trữ. Mỗi đối tượng được lưu trữ được cấu thành bởi: dữ liệu (data) và dữ liệu thông tin (metadata).
* ***Key*:** Mỗi đối tượng lưu trữ trong một bucket trong S3 được định danh bởi một khóa duy nhất. Nó cũng giống như khái niệm tên tệp trong một thư mục trên hệ thống tệp của bạn. Tên tệp trong một thư mục trên đĩa cứng của bạn phải là duy nhất. Mỗi đối tượng trong một bucket chỉ có một khóa. Tên của bucket và khóa được sử dụng cùng nhau để cung cấp định danh duy nhất cho mỗi đối tượng lưu trữ trong S3.



Hình : Amazon S3

*[Nguồn: VIBLO]*

*Đặc điểm của Amazon S3:*

* ***Tính ổn định***: Nó được thiết kế để chịu được các hỏng hóc và phục hồi hệ thống rất nhanh với thời gian tối thiểu.
* ***Đơn giản****,* ***dễ dùng***: S3 được xây dụng trên các khái niệm đơn giản và cung cấp tính mềm dẻo cao cho việc phát triển các ứng dụng của bạn. Bạn có thể xây dựng các lược đồ lưu trữ phức tạp hơn, nếu cần, bằng cách thêm các hàm vào các thành phần của S3.
* ***Tính mở rộng***: Thiết kế của S3 cung cấp một cấp độ cao về tính mở rộng và cho phép sự điều chỉnh dễ dàng trong dịch vụ khi lượng truy cập vào ứng dụng web của bạn tăng đột biến với lưu lượng khổng lồ.
* ***Rất Rẻ***: Chi phí sử dụng S3 rất cạnh tranh với các giải pháp của công ty và cá nhân khác trên thị trường.

Trong hệ thống, Amazon S3 được sử dụng để lưu trữ và quản lý hình ảnh dịch vụ cũng avatar mà người dùng gửi lên hệ thống.

## SQL và NoSQL

*SQL (Structured Query Language)* hay ngôn ngữ truy vấn mang tính cấu trúc, là một ngôn ngữ được thiết kế để quản lý dữ liệu trong một hệ thống quản lý cơ sở dữ liệu quan hệ (RDBMS).

*NoSQL (tên gốc “NonSQL”, “Not only SQL”)* cung cấp một cơ chế để lưu trữ và truy xuất dữ liệu được mô hình hóa khác với các quan hệ bảng (thường được sử dụng trong RDBMS). NoSQL được phát triển trên Javascript Framework với kiểu dữ liệu là JSON và dạng dữ liệu theo kiểu key và value (1 đặc trưng về dữ liệu trong JSON). NoSQL ra đời như là 1 mảnh vá cho những khuyết điểm và thiếu xót cũng như hạn chế của mô hình dữ liệu quan hệ RDBMS về tốc độ, tính năng, khả năng mở rộng, memory cache…

*So sánh SQL và NoSQL*

|  |  |
| --- | --- |
| SQL | NoSQL |
| SQL databases cung cấp kiểu lưu trữ dữ liệu dưới dạng bảng và các bảng này có quan hệ với nhau. | NoSQL databases lưu trữ dưới dạng JSON dưới dạng field-value từng cặp một. |
| SQL phù hợp với những dự án đã có yêu cầu dữ liệu rõ ràng xác định quan hệ logic có thể được xác định trước. | NoSQL database có thể phù hợp hơn cho các dự án mà các yêu cầu dữ liệu ban đầu rất khó xác định. |
| SQL cung cấp truy vấn JOIN rất mạnh mẽ. Chúng ta có thể lấy dữ liệu liên quan trong nhiều bảng bằng cách sử dụng một câu lệnh SQL. | NoSQL không trang bị JOIN, chúng ta cần phải lấy tất cả các trường của đối tượng, liên kết chúng trong logic chương trình của chúng ta. |
| Để hệ thống SQL hoạt động tốt và nhanh thì việc thiết kế tốt là cực kì quan trong. | NoSQL thường được cho là nhanh hơn SQL. |

Bảng : SQL và NoSQL

Như vậy, với việc dữ liệu có thể được phân tích xác định rõ ràng, cũng như cần đến quan hệ dữ liệu, thì việc lựa chọn SQL là mô hình dữ liệu bên server là một lựa chọn hợp lý. Còn tại client, vì việc truy vấn không quá phức tạp, Android cũng cung cấp Preference Share cho phép chúng ta truy cập dữ liệu dưới dạng key-value nên NoSQL là sẽ lựa chọn tốt.

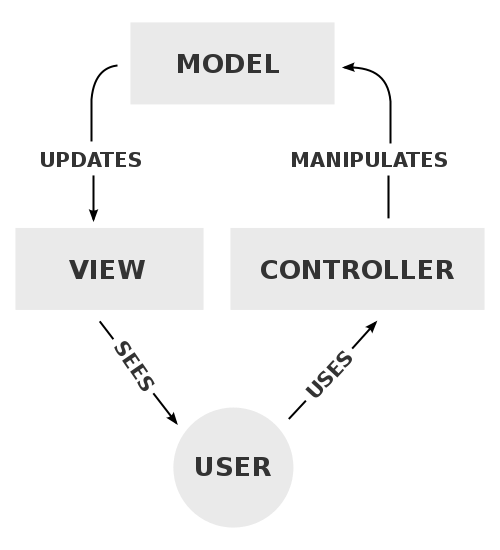
## Giải pháp lập trình phía server - Kiến trúc MVP

Trong nội dung đồ án, hệ thống có xây dựng một server nhằm mục đích cung cấp các thông tin, dữ liệu cần thiết cho ứng dụng (client) cài đặt trên thiết bị di động. Đồ án sử dụng Framework Laravel phục vụ cho việc lập trình phía Server. Framework này hoạt động theo mô hình MVC.

*“MVC (Model-View-Controller) là một mẫu kiến trúc phần mềm để tạo lập giao diện người dùng trên máy tính. MVC chia một ứng dụng thành ba phần tương tác được với nhau để tách biệt giữa cách thức mà thông tin được xử lý nội hàm và phần thông tin được trình bày và tiếp nhận từ phía người dùng.”* [12]

MVC bao gồm 3 thành phần: Model, View, Controller.

* ***Model:*** Nơi lưu trữ giữ liệu và được lấy ra bởi câu lệnh từ bên lớp điều khiển *(Controller)* để hiển thị lên giao diện *(View)*
* ***View:***Nơi xuất đầu ra mới cho người dùng dựa trên những thay đổi của dữ liệu *(model).*
* ***Controller:*** Nơi gửi những câu lệnh sang *Model* để cập nhật trạng thái dữ liệu. Sau đấy, nó có thể gửi trả dữ liệu đấy về *View* được liên kết để thay đổi giao diện tới người dùng.



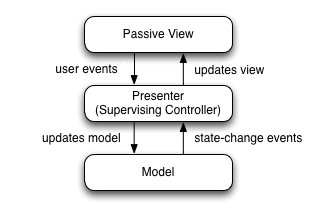
Hình : Mô hình MVC

*[Nguồn: Wikipedia]*

Với MVC, *Model* và *View* đã được tách ra và như vậy các *Model* có thể dễ dàng test bởi vì nó không gắn với bất kỳ thành phần nào. Tuy nhiên *Controller* có một vài vấn đề..

* ***Testability***: *Controller* được gắn rất chặt chẽ đến các API của Andoird mà rất khó để unit test.
* ***Modularity & Flexibility***: Các *Controller* được kết chặt chẽ với các *View*. Nó cũng có thể là một phần mở rộng của View*.* Nếu chúng ta thay đổi *View*, chúng ta phải quay trở lại và thay đổi *Controller*.
* ***Maintenance***: Theo thời gian, càng ngày code của *Controller* càng nhiều, làm cho chúng cồng kềnh và dễ gãy.

MVP (Model-View-Presenter) là một cải tiến của mô hình MVC. Trong MVP, *View* và *Model* có sự tách biệt rõ ràng hơn trong khi *Controller* được thay thế bởi *Presenter.* Đấy cũng là điểm khác biệt nhất giữa hai mô hình này bởi *Presenter* mang gánh vác nhiện vụ “người đứng giữa”. Tất cả xử lý logic đều được đẩy hết cho *Presenter.*



Hình : Mô hình MVP

*[Nguồn: Wikipedia]*

Với thiết kế như vậy, so với MVC thì MVP đã rõ ràng hơn rất nhiều. Chúng ta có thể dễ dàng unit test logic của *Presenter* bởi vì nó không gắn với bất kỳ *View* cụ thể nào cả.

## Giải pháp thiết kế giao diện - Material Design

Material Design là một phong cách thiết kế mới được Google giới thiệu cùng lúc với phiên bản Android 5.0 Lollipop.

Phong cách thiết kế Material Design nhắm đến những đường nét đơn giản, sử dụng nhiều mảng màu đậm nổi bật, các đối tượng đồ họa trong giao diện dường như: “trôi nổi” lên. Ngoài ra, nó còn bao gồm cả những hiệu ứng chuyển động tự nhiên khi các nút, menu hiện diện trên màn hình. Tất cả đều nhằm mang lại cho người dùng trải nghiệm mới mẻ hơn, thú vị hơn và gần giống đời thực hơn. [6]



Hình 13: Material Design - phong cách thiết kế giao diện mới của Google

[Nguồn: MaterialIO]

*Đặc điểm của Material Design*

* Sử dụng các màu nổi bật, thường có một mảng màu chủ đạo nằm ở cạnh trên ứng dụng.
* Các biểu tượng phẳng, đơn giản nhưng dễ hiểu.
* Một số ứng dụng sẽ có một nút tròn to nằm ở góc dưới bên phải, thường có chức năng tạo mới.
* Giao diện phẳng, ít hoặc không có hiệu ứng chuyển màu, có hoặc không có hiệu ứng đổ bóng đen.
* Menu, nút nhấn, chữ viết… có nhiều khoảng cách trắng nên trông thoáng đãng.
* Có các hiệu ứng chuyển động tự nhiên, dễ hiểu, có thể gợi ý cho một tính năng nào đó.

*Lợi ích của Material Design*

Material Design là một ngôn ngữ được Google tạo ra nhằm “*phát triển một hệ thống nền duy nhất cho phép xây dựng các trải nghiệm đồng bộ giữa nhiều loại và kích thước thiết bị khác nhau. Việc thiết kế cho thiết bị di động được đặt lên hàng đầu, nó còn phải chú trọng đến việc nhập liệu bằng cảm ứng, giọng nói, chuột, bàn phím”. [6]*

Google muốn mang lại trải nghiệm thống nhất cho người dùng trong hệ sinh thái của hãng. Người dùng cuối như chúng ta sẽ có cảm giác quen thuộc hơn, dành ít thời gian để học hỏi hơn khi sử dụng cùng một app nhưng trên nhiều máy khác nhau. Các lập trình viên thì có thể đảm bảo rằng cách người dùng trải nghiệm app của họ là như nhau không quan trọng thiết bị đang chạy là gì.

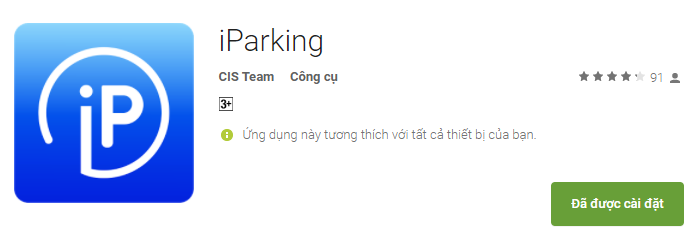
# KHẢO SÁT VÀ XÁC ĐỊNH YÊU CẦU BÀI TOÁN

## Khảo sát một số hệ thống hiện có

Theo như bài toán đặt ra, hệ thống của chúng ta cần có các chức năng tra cứu, tìm kiếm thông tin về dịch vụ phương tiện giao thông dành cho các tài xế, cũng như các chức năng quản lý dịch vụ đấy dành cho nhà cung cấp. Là môi trường tìm kiếm và trao đổi thông tin giữa hai loại người dùng này.

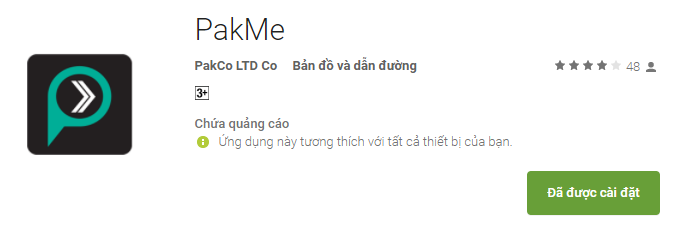
Hiện nay trên thị trường có một số hệ thống/ ứng dụng có chức năng tương tự với yêu cầu bài toán:

* ***iParking:*** Là ứng dụng tìm bãi đỗ xe, cung cấp danh sách các điểm đỗ và số chỗ còn trống dành cho tài xế. Tuy nhiên, ứng dụng này chỉ là ứng dụng thương mại của Công ty Khai thác điểm đỗ xe Hà Nội.



Hình : Ứng dụng iParking trên Play Store

* ***PakMe:*** là ứng dụng giúp người dùng tìm kiếm được điểm đỗ xe thuận tiện và gần nhất. Giúp lái xe tìm được điểm đỗ xe ô tô xung quanh vị trí hiện tại hoặc nơi muốn đi đến. Các lái xe còn có thể tự đề xuất các địa điểm lên ứng dụng.



Hình : Ứng dụng PakMe trên Play Store

* ***Free parking***: (Tên tiếng việt: *Bãi đậu xe miễn phí*) là một ứng dụng di động hữu giúp người dùng tìm được chỗ đỗ xe miễn phí hoặc rẻ nhất, gần người dùng nhất hoặc tại địa chỉ cụ thể! Nhờ có sự đóng góp của hàng nghìn người dùng, tìm các bãi đỗ xe miễn phí trên khắp thế giới và tiết kiệm thời gian và tiền bạc!



Hình 16: Ứng dụng Free parking trên Play Store

|  |  |  |
| --- | --- | --- |
| STT | Tên ứng dụng | Các chức năng |
| 1 | iParking | * Tìm bãi đỗ xe quanh một khu vực * Xem thông tin bãi đỗ * Chỉ đường đi đến bãi đỗ * Quản lý lịch sử đỗ xe * Quản lý xe * Thanh toán qua điện thoại * Quản lý các thẻ thanh toán |
| 2 | PakMe | * Tìm bãi đỗ xe xung quanh một vị trí * Xem các thông tin bãi đỗ * Chỉ đường đi đến bãi đỗ * Nhận thông báo * Phản hồi cho quản trị viên |
| 3 | Free parking | * Đăng nhập * Tìm bãi đỗ xe quanh một vị trí * Báo cáo số chỗ đỗ xe bị thiếu * Báo cáo lỗi liên quan đến bãi đậu xe * Tìm và điều hướng đến một địa điểm * Tự động làm mới các bãi đỗ xe * Giao diện đơn giản và trực quan * Lưu một bãi đậu xe vào danh sách yêu thích * Lưu lưu vị trí bất kỳ trên bản đồ |

Bảng 2: Bảng chức năng của các ứng dụng được khảo sát

Với ứng dụng được khảo sát ở trên, một số ứng dụng có các chức năng cơ bản khá hoàn thiện. Các ứng dụng này đa số là dành cho việc cung cấp các thông tin như điểm đỗ xe cho ô tô. Tuy nhiên, các ứng dụng này chưa có chức năng cho người dùng với vai trò là người cung cấp dịch vụ, hoặc một số trường hợp, vì mục đích thương mại nên chỉ cung cấp các dịch vụ của một công ty.

Qua khảo sát nói trên, hệ thống mà sinh viên mong muốn xây dựng cần có chức năng đầy đủ hơn. Giải pháp đề xuất có thể là tổng hợp các chức năng cơ bản hoàn thiện của các hệ thống trên, đồng thời bổ sung thêm các chức năng mới mà các ứng dụng trên thị trường chưa có.

## Phân tích yêu cầu của bài toán

### Phân tích người dùng (user) của

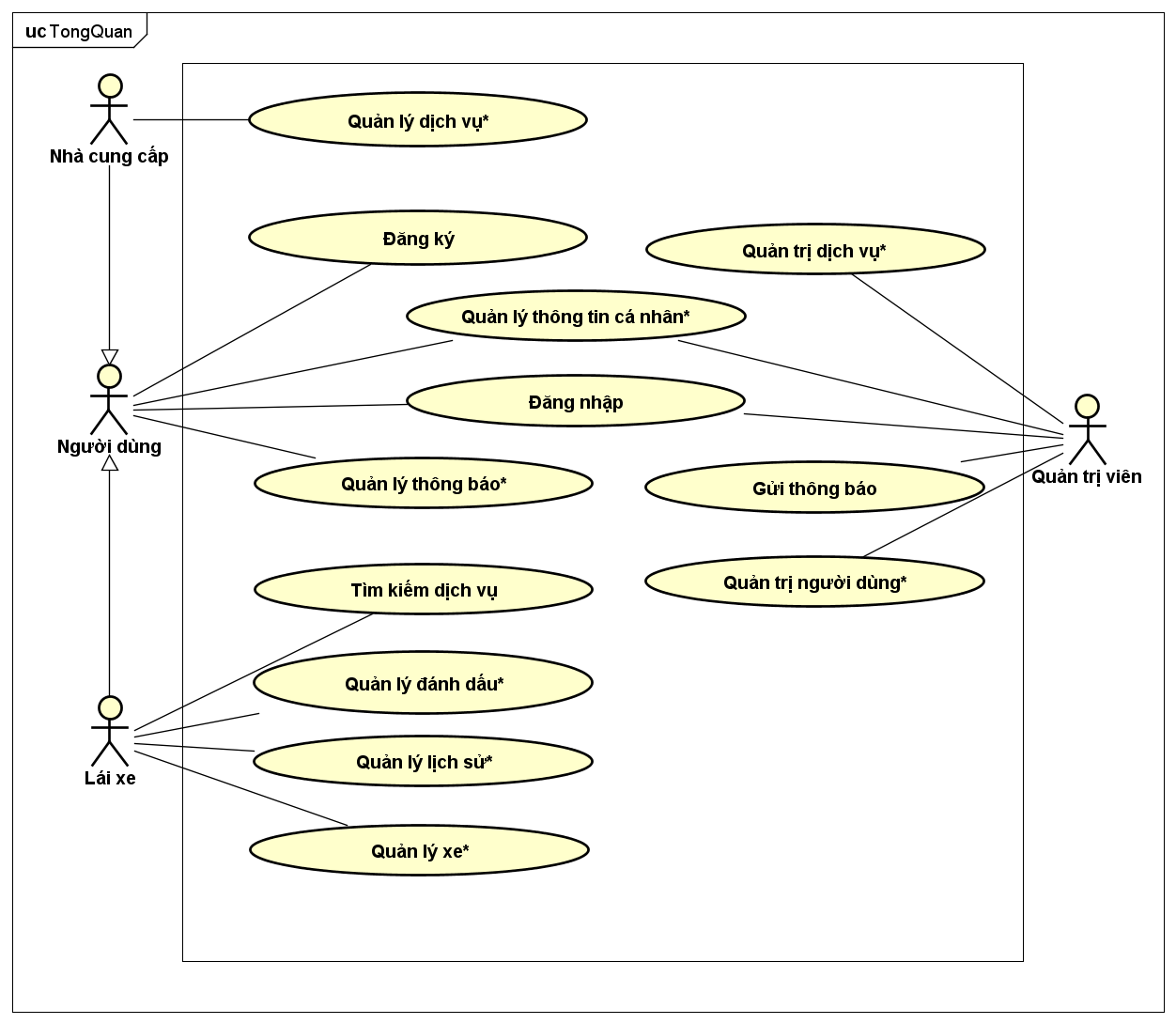
### Biểu đồ use case tổng quan

Hệ thống gồm có 3 tác nhân chính: ***nhà cung cấp***, ***lái xe***, ***quản trị viên***.

|  |  |  |
| --- | --- | --- |
| STT | Tên tác nhân | Các use case |
| 1 | Lái xe | * Đăng ký tài khoản * Đăng nhập hệ thống * Quản lý thông báo (\*) * Cập nhật thông tin cá nhân * Tra cứu, xem thông tin về dịch vụ * Quản lý lịch sử xem dịch vụ (\*) * Quản lý danh sách đánh dấu (\*) * Quản lý phương tiện (\*) |
| 2 | Nhà cung cấp | * Đăng ký tài khoản * Đăng nhập hệ thống * Quản lý thông báo (\*) * Cập nhật thông tin cá nhân * Quản lý danh sách dịch vụ (\*) |
| 3 | Quản trị viên | * Quản trị người dùng (\*) * Quản trị dịch vụ (\*) * Gửi thông báo. * Quản lý thông tin cá nhân |

Bảng 3: Bảng các use case tổng quan cho từng loại người dùng

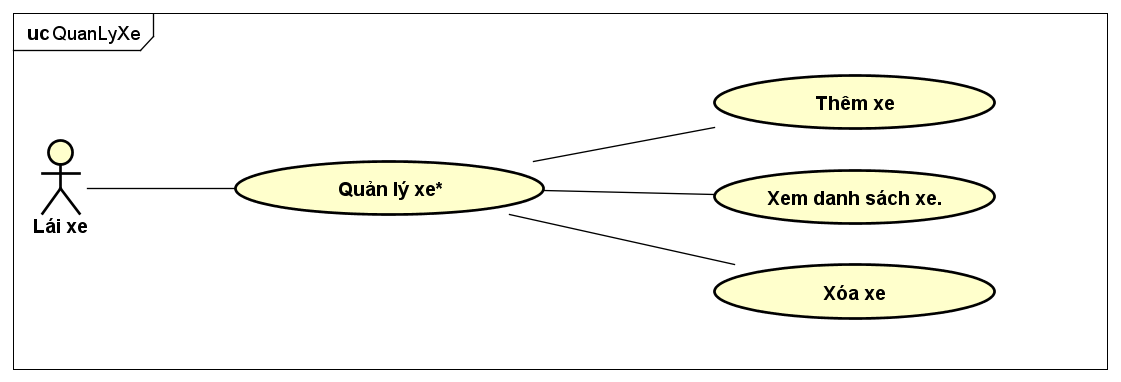
*Lưu ý: Các use case được đánh dấu (\*) là các use case phức, với biểu đồ phân rã ở mục [3.2.3].*



Hình : Use case tổng quan

### Biểu đồ use case phân rã

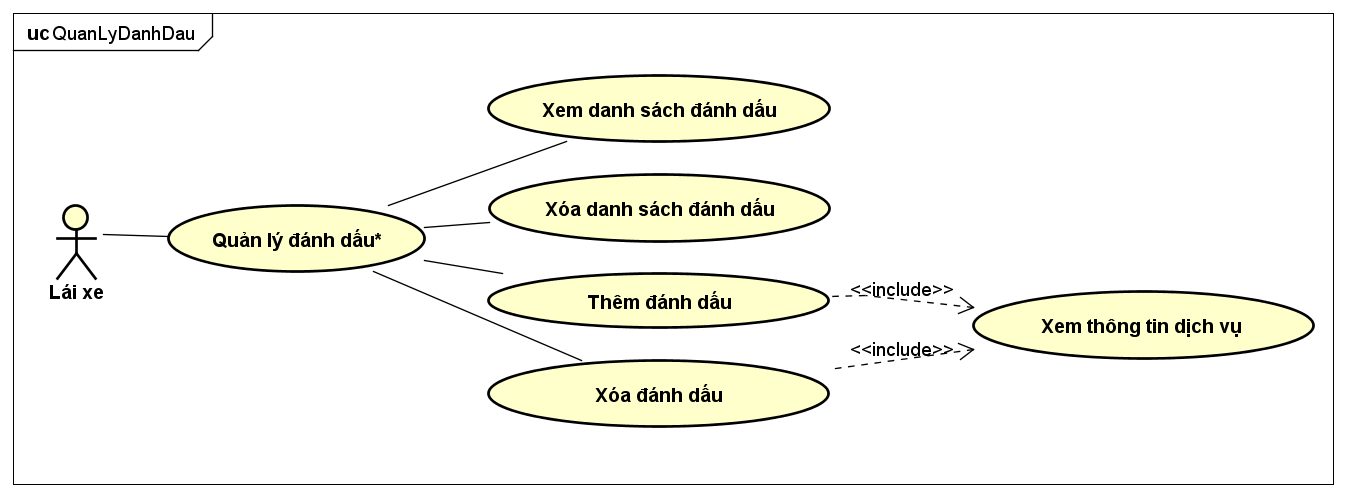
* Biểu đồ use case phân rã cho chức năng ***quản lý xe***



Hình : Use case phân rã chức năng quản lý xe

Với chức năng quản lý xe, lái xe có thể thực hiện các thao tác CRUD (Create – Read -Update - Delete) thông tin phương tiện thông qua hệ thống.

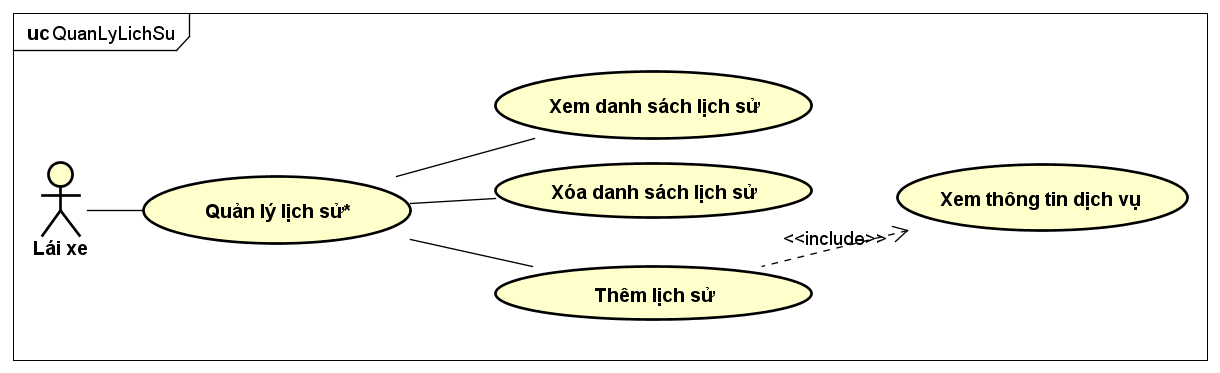
* Biểu đồ use case phân rã cho chức năng ***quản lý đánh dấu*** dùng để đánh dấu các dịch vụ mà lái xe yêu thích/hay dùng



Hình : Use case phân rã chức năng quản lý đánh dấu

Sau khi xem thông tin dịch vụ, lái xe có thể đánh dấu chúng vào danh sách của mình để dễ dàng xem lại. Với chức năng quản lý đánh dấu, lái xe có thể xem danh sách những dịch vụ đã đánh dấu cũng như xóa chúng.

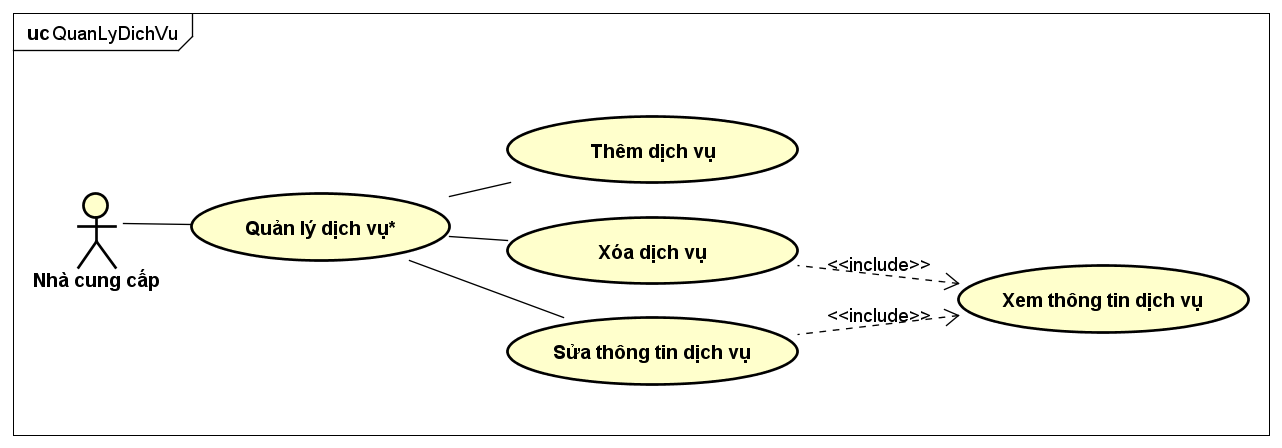
* Biểu đồ use case phân rã cho chức năng ***quản lý lịch sử***



Hình : Use case phân rã chức năng quản lý lịch sử

Với mỗi lần xem thông tin dịch vụ, thông tin đấy sẽ được thêm vào lịch sử của lái xe. Ở chức năng quản lý lịch sử, sau khi xem danh sách lịch sử, lái xe có thể xóa những danh sách đấy.

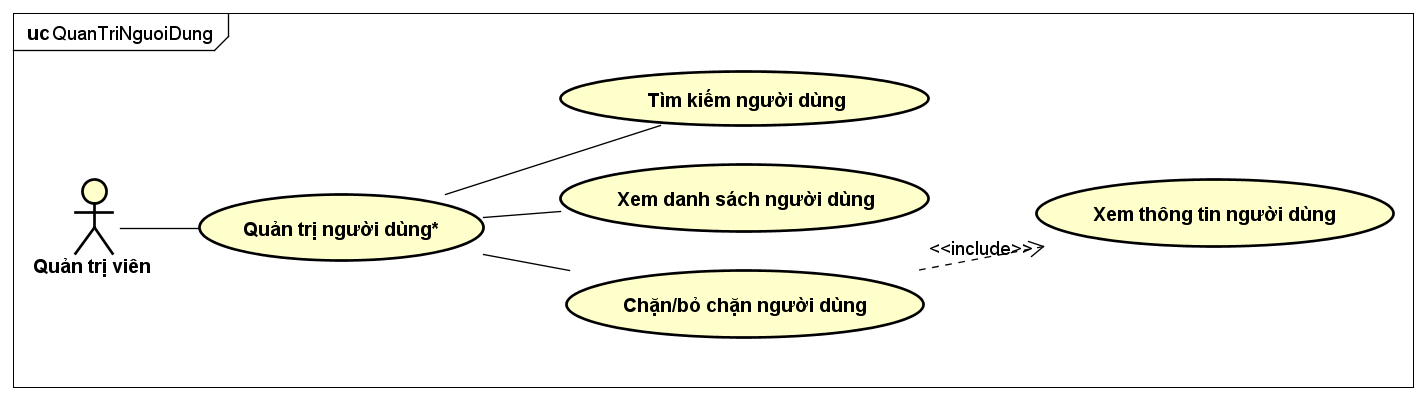
* Biểu đồ use case phân rã cho chức năng ***quản lý dịch vụ***



Hình : Use case phân rã chức năng quản lý dịch vụ

Nhà cung cấp đưa những thông tin dịch vụ của mình lên ứng dụng thông qua chức năng quản lý dịch vụ. Chức năng này cũng bao gồm CRUD.

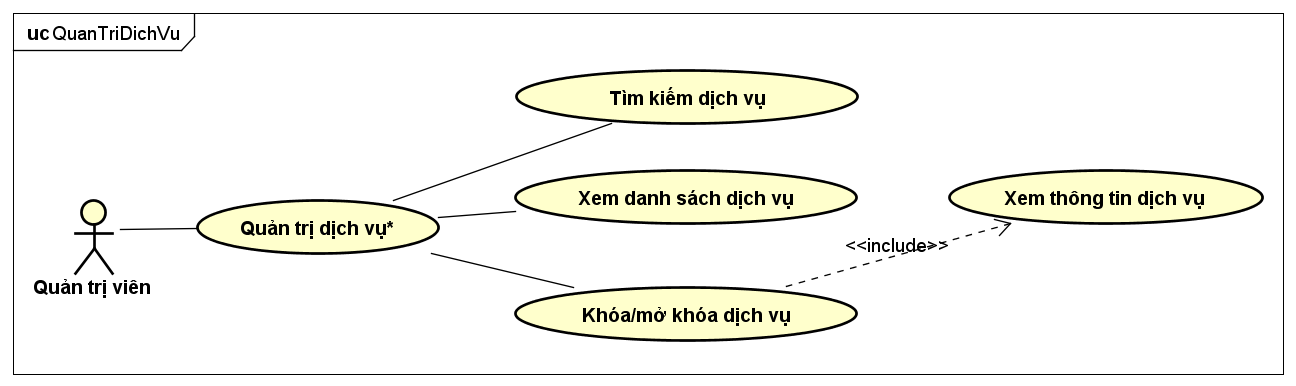
* Biểu đồ use case phân ra cho chức năng ***quản trị người dùng***



Hình : Use case phân rã chức năng quản trị người dùng

Để quản lý người dùng, hệ thống cung cấp chức năng quản trị người dùng cho quản trị viên, bên cạnh các chức năng tìm kiếm cũng như xem thông tin thì quản trị viên còn có thể chặn hoặc bỏ chặn người dùng.

* Biểu đồ use case phân rã cho chức năng ***quản trị dịch vụ***



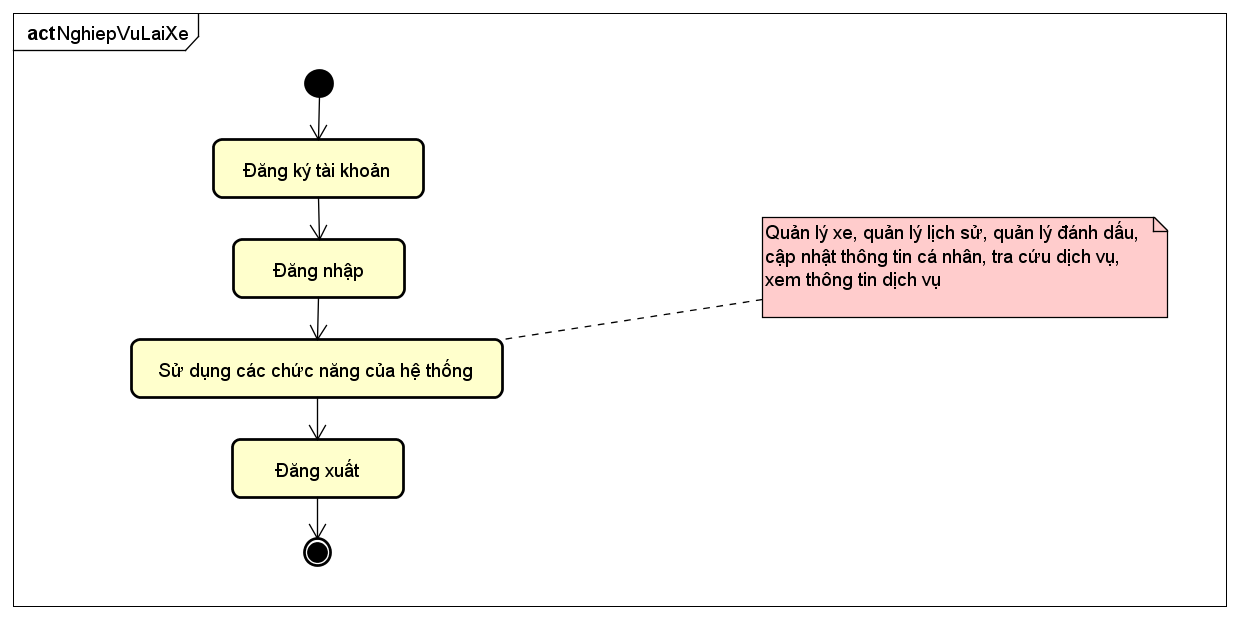
Hình : Use case phân rã chức năng quản trị dịch vụ

Tương tự với các chức năng ở quản trị người dùng, quản trị dịch vụ cũng có các chức năng cho quản trị viên với thông tin dịch vụ.

### Quy trình nghiệp vụ

Trong hệ thống có 3 quy trình nghiệp vụ chính: Quy trình sử dụng ứng dụng của lái xe (tài xế), nhà cung cấp dịch vụ, quản trị viên.

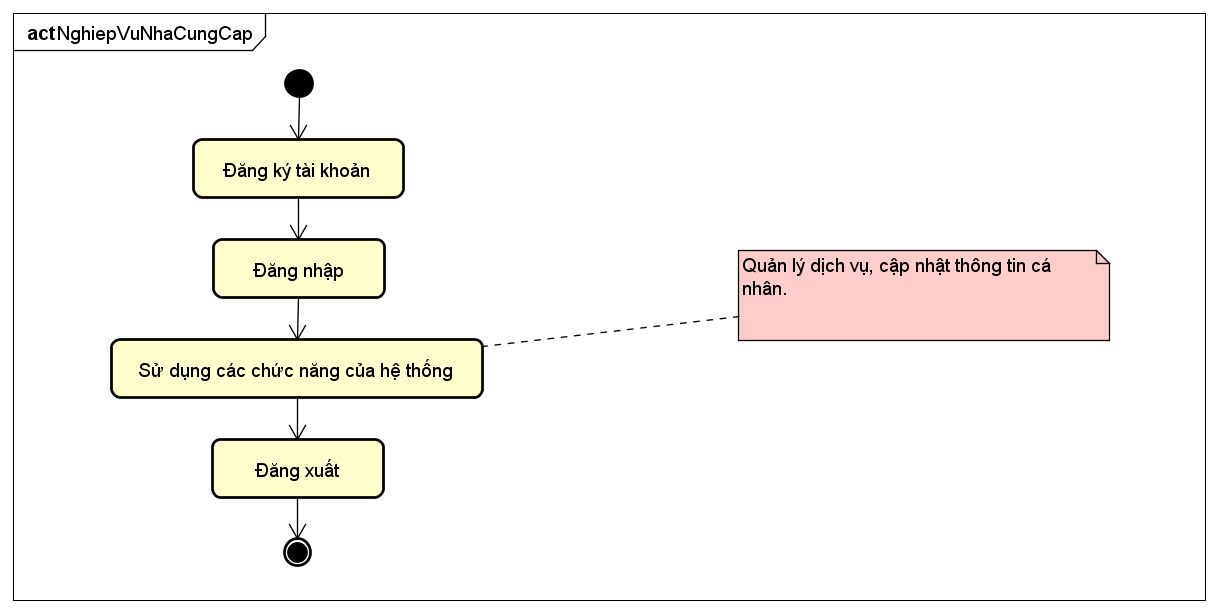
* Quy trình nghiệp vụ của ***tài xế***.



Hình : Quy trình nghiệp vụ của tài xế

Người dùng có thể đăng ký và lựa chọn loại người dùng là tài xế. Sau khi đăng ký và đăng nhập với vai trò tài xế vào hệ thống. Tài xế có thể quản lý xe, lịch sử, đánh dấu cũng như tìm kiếm và xem thông tin các dịch vụ và cập nhật thông tin cá nhân lên hệ thống.

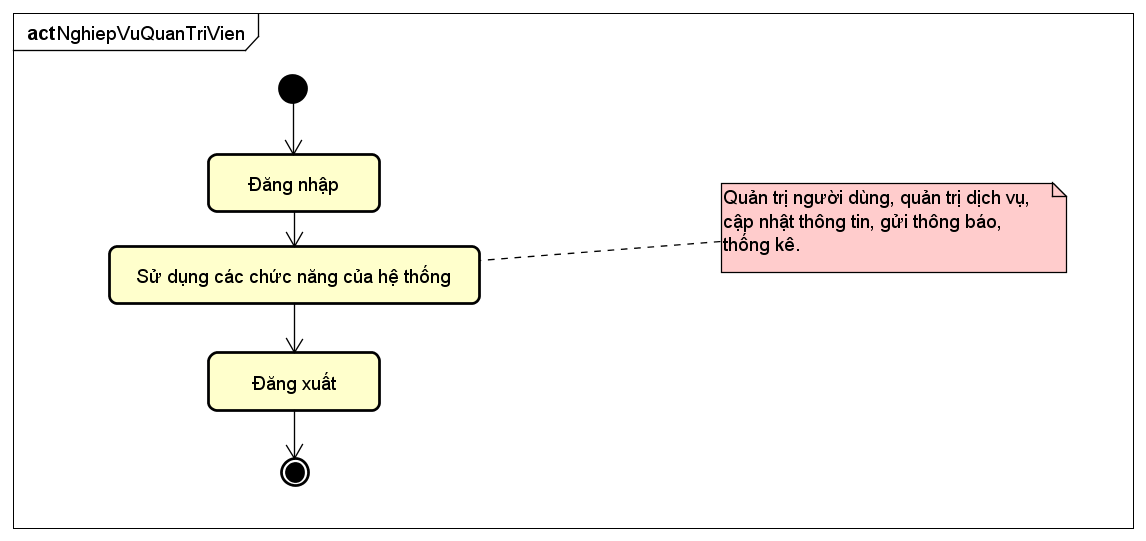
* Quy trình sử dụng ứng dụng của ***nhà cung cấp***.



Hình : Quy trình nghiệp vụ của nhà cung cấp

Nhà cung cấp có thể đăng ký và lựa chọn loại người dùng (nhà cung cấp) cho mình. Sau khi đăng ký và đăng nhập với vai trò tài xế vào hệ thống. Với vai trò đấy, nhà cung cấp có thể quản lý thông tin các dịch vụ do mình tạo ra cũng như có thể cập nhật thông tin cá nhân lên hệ thống.

* Quy trình sử dụng ứng dụng của ***quản trị viên***.



Hình : Quy trình nghiệp vụ của quản trị viên

Sau khi đăng nhập vào hệ thống thành công, quản trị viên có thể thực hiện các chức năng quản trị người dùng, quản trị dịch vụ cũng như xem thống kê về người dùng, dịch vụ.

## Phân tích chi tiết yêu cầu người dùng (users)

### Đặc tả use case UC001 – Đăng ký tài khoản

|  |  |  |  |
| --- | --- | --- | --- |
| **Mã use case** | **UC001** | **Tên use case** | **Đăng ký tài khoản** |
| **Tác nhân** | Người dùng | | |
| **Mô tả** | Use case mô tả quá trình người dùng đăng ký tài khoản trong hệ thống. | | |
| **Tiền điều kiện** | Không | | |
| **Luồng sự kiện chính** | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 1 | Người dùng | Chọn chức năng đăng ký tài khoản. | | 2 | Hệ thống | Hiển thị form lựa chọn loại người dùng. | | 3 | Người dùng | Chọn một một người dùng và gửi yêu cầu. | | 4 | Hệ thống | Hiển thị form nhập số điện thoại. | | 5 | Người dùng | Nhập số điện thoại và gửi yêu cầu. | | 6 | Hệ thống | Kiểm tra thông tin số điện thoại. | | 7 | Hệ thống | Gửi mã PIN đến số điện thoại và hiển thị form nhập mã PIN. | | 8 | Người dùng | Nhập mã PIN vào form và gửi yêu cầu. | | 9 | Hệ thống | Kiểm tra mã PIN được nhập. | | 10 | Hệ thống | Hiển thị các form thông tin tài khoản. | | 11 | Người dùng | Nhập các thông tin tài khoản (\*). | | 12 | Hệ thống | Kiểm trả các thông tin được nhập. | | 13 | Hệ thống | Đi đến màn hình chính của người dùng. | | | |
| **Luồng sự kiện thay thế** | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 6a | Hệ thống | Thông báo lỗi nếu nhập sai số điện thoại hoặc số điện thoại đã được đăng ký. | | 9a | Hệ thống | Thông báo lỗi nếu nhập sai mã PIN. | | 12a | Hệ thống | Thông báo lỗi nếu nhập sai hoặc thiếu thông tin. | | | |
| **Hậu điều kiện** | Không | | |

Bảng : Đặc tả use case UC001 – Đăng ký tài khoản

### Đặc tả use case UC002 – Đăng nhập

|  |  |  |  |
| --- | --- | --- | --- |
| **Mã use case** | **UC002** | **Tên use case** | **Đăng nhập** |
| **Tác nhân** | Người dùng | | |
| **Mô tả** | Use case mô tả quá trình người dùng đăng nhập tài khoản vào hệ thống | | |
| **Tiền điều kiện** | Không | | |
| **Luồng sự kiện chính** | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 1 | Người dùng | Chọn chức năng đăng nhập tài khoản. | | 2 | Hệ thống | Hiển thị form nhập số điện thoại. | | 3 | Người dùng | Nhập số điện thoại và gửi yêu cầu. | | 4 | Hệ thống | Kiểm tra thông tin số điện thoại. | | 5 | Hệ thống | Gửi mã PIN đến số điện thoại và hiển thị form nhập mã PIN. | | 6 | Người dùng | Nhập mã PIN vào form và gửi yêu cầu. | | 7 | Hệ thống | Kiểm tra mã PIN được nhập. | | 8 | Hệ thống | Đi đến màn hình chính của người dùng. | | | |
| **Luồng sự kiện thay thế** | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 4a | Hệ thống | Thông báo lỗi nếu nhập sai số điện thoại hoặc số điện thoại chưa được đăng ký. | | 7a | Hệ thống | Thông báo lỗi nếu nhập sai mã PIN. | | | |
| **Hậu điều kiện** | Không | | |

Bảng : Đặc tả use case UC002 – Đăng nhập tài khoản

### Đặc tả use case UC003 – Tìm kiếm dịch vụ

|  |  |  |  |
| --- | --- | --- | --- |
| **Mã use case** | **UC003** | **Tên use case** | **Tìm kiếm dịch vụ** |
| **Tác nhân** | Lái xe | | |
| **Mô tả** | Use case mô tả quá trình lái xe tìm kiếm thông tin dịch vụ khi tham gia vào hệ thống. | | |
| **Tiền điều kiện** | Lái xe đã đăng nhập vào hệ thống. | | |
| **Luồng sự kiện chính** | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 1 | Lái xe | Vào màn hình tìm kiếm (màn hình chính). | | 2 | Lái xe | Nhập từ khóa địa điểm hoặc di chuyển bản đồ đến địa điểm cần tìm. | | 3 | Hệ thống | Hiển thị danh sách dịch vụ lên bản đồ, hiển thị thông tin địa điểm chọn ở trên bản đồ. | | 4 | Lái xe | Chọn xem danh sách dịch vụ. | | 5 | Hệ thống | Hiển thị danh sách dịch vụ. | | | |
| **Luồng sự kiện thay thế** | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 5a | Hệ thống | Thông báo nếu danh sách dịch vụ trống. | | | |
| **Hậu điều kiện** | Không | | |

Bảng : Đặc tả use case UC003 – Tìm kiếm dịch vụ

### Đặc tả use case UC004 – Thêm dịch vụ

|  |  |  |  |
| --- | --- | --- | --- |
| **Mã use case** | **UC004** | **Tên use case** | **Thêm dịch vụ** |
| **Tác nhân** | Lái xe | | |
| **Mô tả** | Use case mô tả quá trình nhà cung cấp thêm mới thông tin dịch. | | |
| **Tiền điều kiện** | Nhà cung cấp đã đăng nhập vào hệ thống. | | |
| **Luồng sự kiện chính** | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 1 | Nhà cung cấp | Chọn chức năng thêm dịch vụ. | | 2 | Hệ thống | Hiển thị giao diện thêm dịch vụ mới. | | 3 | Nhà cung cấp | Nhập các thông tin dịch vụ (\*). | | 4 | Nhà cung cấp | Gửi yêu cầu thêm. | | 5 | Hệ thống | Cập nhật danh sách dịch vụ. | | | |
| **Luồng sự kiện thay thế** | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 5a | Hệ thống | Thông báo các một trong trường bắt buộc để trống. | | | |
| **Hậu điều kiện** | Không | | |

Bảng : Đặc tả use case UC004 – Thêm dịch vụ

### Đặc tả use case UC005 – Khóa người dùng

|  |  |  |  |
| --- | --- | --- | --- |
| **Mã use case** | **UC005** | **Tên use case** | **Khóa người dùng** |
| **Tác nhân** | Quản trị viên | | |
| **Mô tả** | Use case mô tả quá trình quản trị viên khóa một người dùng. | | |
| **Tiền điều kiện** | Quản trị viên đã đăng nhập vào hệ thống. | | |
| **Luồng sự kiện chính** | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 1 | Quản trị viên | Chọn chức năng quản trị người dùng. | | 2 | Hệ thống | Hiển thị giao diện quản trị người dùng. | | 3 | Quản trị viên | Nhập từ khóa tìm kiếm một người dùng. | | 4 | Quản trị viên | Gửi yêu cầu tìm kiếm | | 5 | Hệ thống | Hiển thị danh sách người dùng. | | 6 | Quản trị viên | Chọn một người dùng, yêu cầu xem thông tin | | 7 | Hệ thống | Hiển thị thông tin người dùng được chọn. | | 8 | Quản trị viên | Gửi yêu cầu khóa người dùng. | | 9 | Hệ thống | Cập nhật trạng thái của người dùng. | | | |
| **Luồng sự kiện thay thế** | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 5a | Hệ thống | Thông báo nếu danh sách người dùng trống. | | | |
| **Hậu điều kiện** | Không | | |

Bảng : Đặc tả use case UC005 – Khóa người dùng

### Đặc tả use case UC006 – Khóa dịch vụ

|  |  |  |  |
| --- | --- | --- | --- |
| **Mã use case** | **UC006** | **Tên use case** | **Khóa dịch vụ** |
| **Tác nhân** | Quản trị viên | | |
| **Mô tả** | Use case mô tả quá trình quản trị viên khóa một dịch vụ. | | |
| **Tiền điều kiện** | Quản trị viên đã đăng nhập vào hệ thống. | | |
| **Luồng sự kiện chính** | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 1 | Quản trị viên | Chọn chức năng quản trị dịch vụ. | | 2 | Hệ thống | Hiển thị giao diện quản trị dịch vụ. | | 3 | Quản trị viên | Nhập từ khóa tìm kiếm một dịch vụ. | | 4 | Quản trị viên | Gửi yêu cầu tìm kiếm | | 5 | Hệ thống | Hiển thị danh sách dịch vụ. | | 6 | Quản trị viên | Chọn một dịch vụ, yêu cầu xem thông tin | | 7 | Hệ thống | Hiển thị thông tin dịch vụ được chọn. | | 8 | Quản trị viên | Gửi yêu cầu khóa dịch vụ. | | 9 | Hệ thống | Cập nhật trạng thái của dịch vụ. | | | |
| **Luồng sự kiện thay thế** | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 5a | Hệ thống | Thông báo nếu danh sách dịch vụ trống. | | | |
| **Hậu điều kiện** | Không | | |

Bảng : Đặc tả use case UC006 – Khóa dịch vụ

## Các yêu cầu khác

### Tính khả thi

* Các chức năng, giao diện dễ sử dụng và thân thiện với người dùng.
* Có phản hồi với người dùng về các vấn đề lỗi đường truyền mạng.

### Hiệu suất

* Hệ thống sẽ hoạt động 24 giờ một ngày, 7 ngày một tuần.
* Thời gian ngưng hoạt động để khắc phục lỗi của hệ thống không lớn hơn 1 giờ.
* Thời gian đáp ứng yêu cầu của người dùng không quá 2s.
* Thông tin hiển thị lên giao diện cho người dùng không quá 3s

### Độ tin cậy

* Hệ thống có tính an toàn, bảo mật cao.
* Dữ liệu chính xác, được cập nhật liên tục

# THIẾT KẾ, XÂY DỰNG VÀ ĐÁNH GIÁ HỆ THỐNG

## Mô tả chung

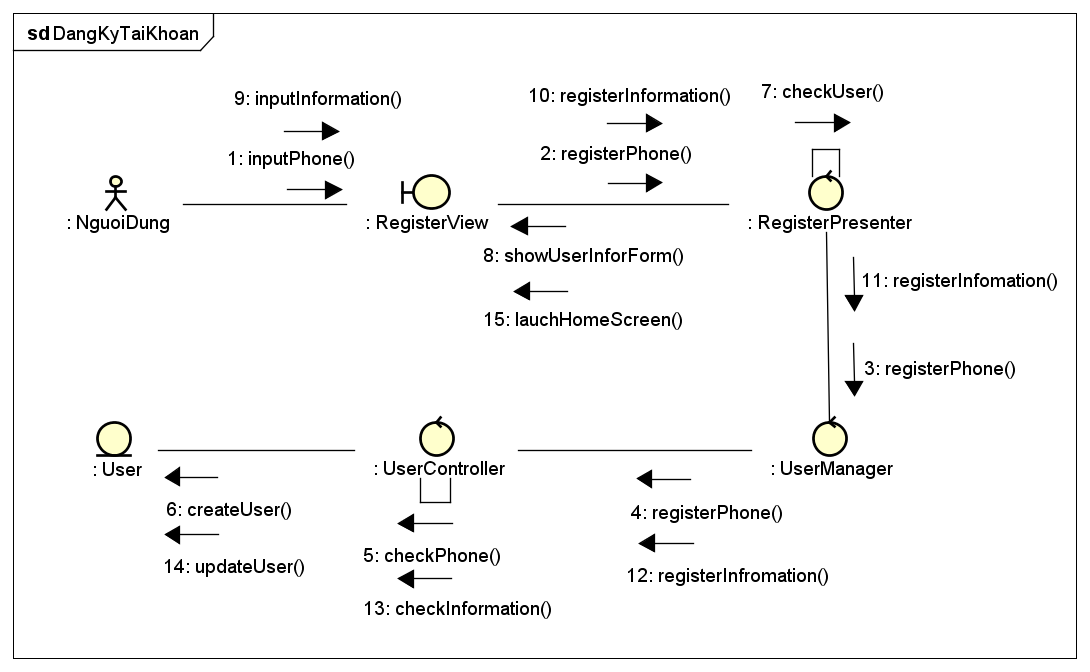
Thiết kế hệ thống tuân theo các phân tích thiết kế người dùng (user cases) đã trình bày ở trên. Nội dung chương dưới đây sẽ tập trung vào trình bày các biểu đồ tương tác, biểu đồ lớp… của hệ thống.

Theo biểu đồ user cases, hệ thống bao gồm rất nhiều chức năng. Do kích thước báo cáo đồ án có hạn, nên trong chương này chỉ tập trung trình bày một số chức năng chính trong hệ thống bao gồm*:*

* *Đăng ký (Register)*
* *Đăng nhập (Login)*
* *Tìm kiếm dịch vụ (Search)*
* *Quản lý dịch vụ (ServiceManagement)*
* *Quản trị người dùng (UserAdministraton)*
* *Quản trị dịch vụ (ServiceAdministration)*

## Biểu đồ tương tác

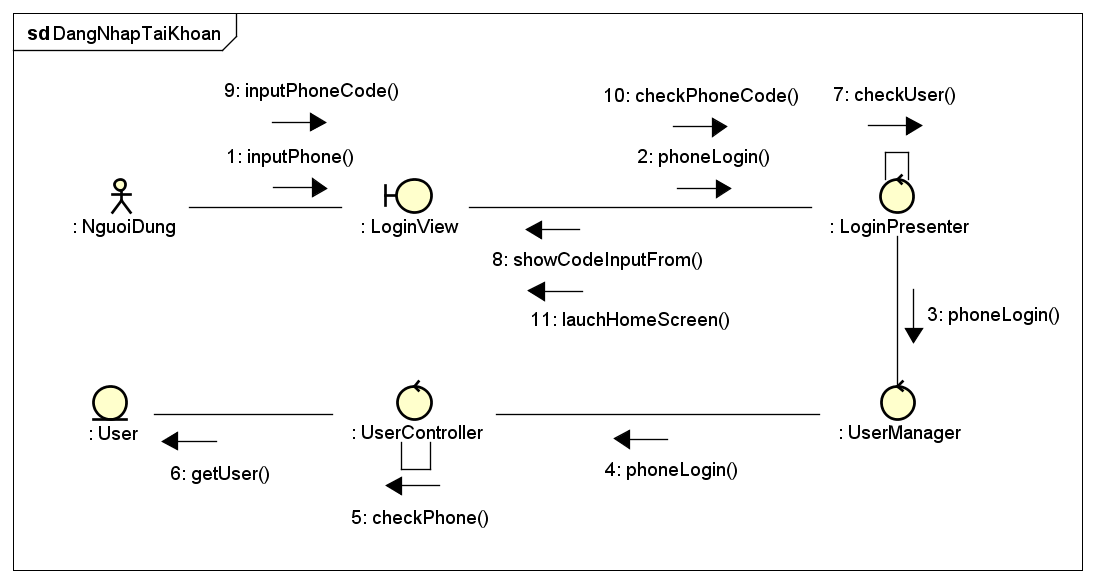
### Biểu đồ tương tác cho use case UC001 – Đăng ký tài khoản



Hình : Biểu đồ tương tác cho use case UC001 – Đăng ký tài khoản

* Lớp biên: *RegisterView*
* Lớp điều khiển: *RegisterPresenter, UserManager, UserController*
* Lớp thực thể: *User*

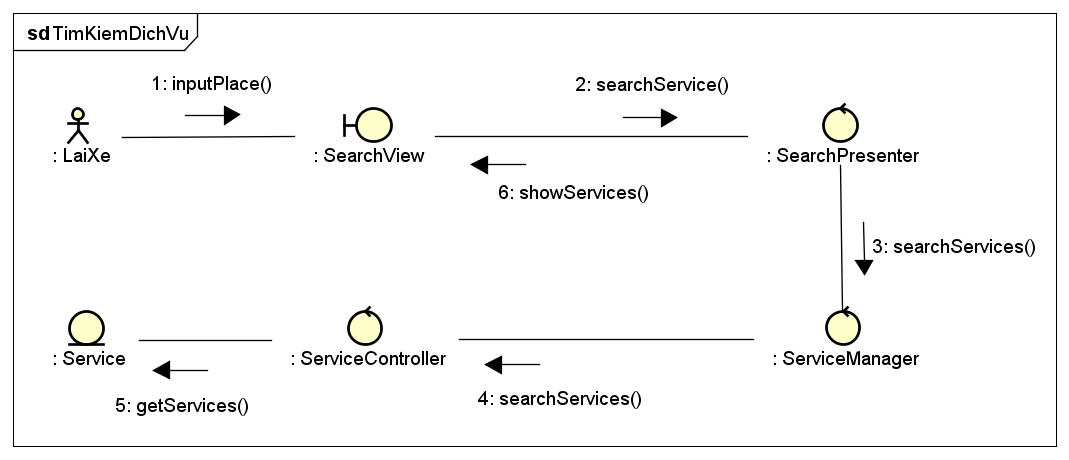
### Biểu đồ tương tác cho use case UC002 – Đăng nhập.



Hình : Biểu đồ tương tác cho use case UC002 – Đăng nhập

* Lớp biên: *LoginView*
* Lớp điều khiển: *LoginPresenter, UserManager, UserController*
* Lớp thực thể: *User*

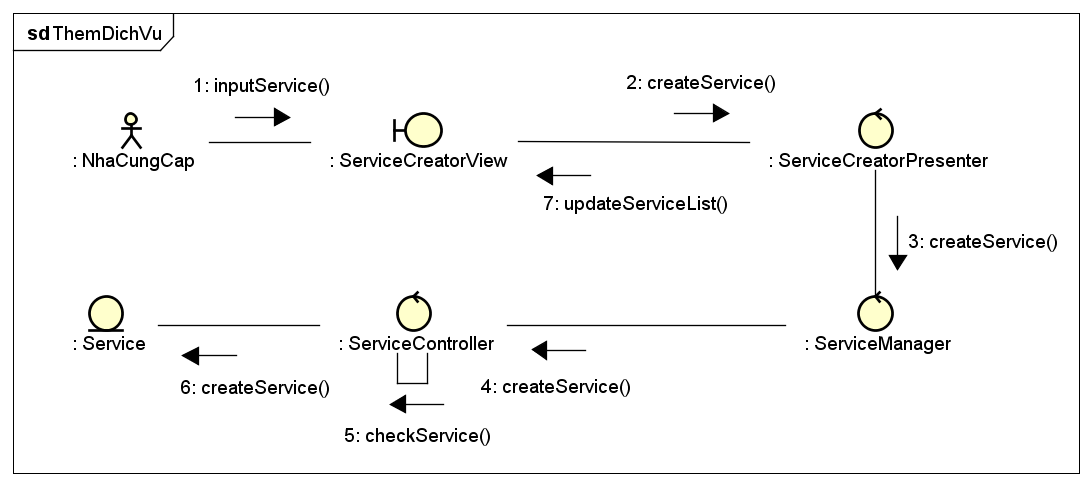
### Biểu đồ tương tác cho use case UC003 – Tìm kiếm dịch vụ.



Hình : Biểu đồ tương tác cho use case UC003 – Tìm kiếm dịch vụ

* Lớp biên: *SearchView*
* Lớp điều khiển: *SearchPresenter, ServiceManager, ServiceController*
* Lớp thực thể: *Service*

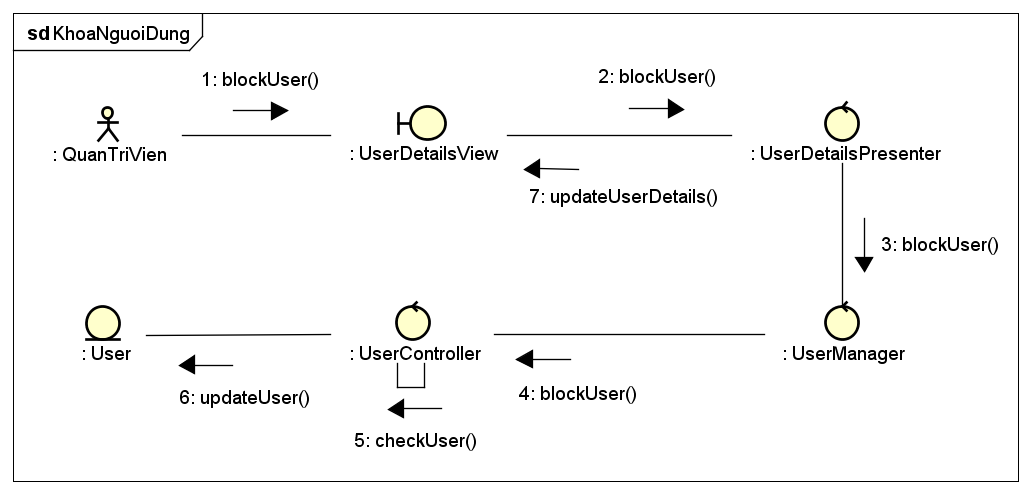
### Biểu đồ tương tác cho use case UC004 – Thêm dịch vụ.



Hình : Biểu đồ tương tác cho use case UC004 – Thêm dịch vụ

* Lớp biên: *ServiceCreatorView*
* Lớp điều khiển: *ServiceCreatorPresenter, ServiceManager, ServiceController*
* Lớp thực thể: *Service*

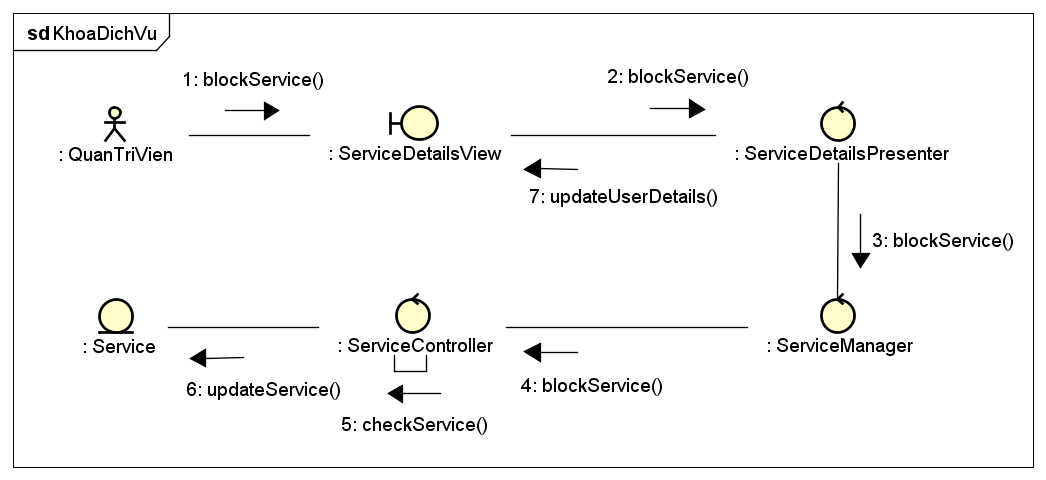
### Biểu đồ tương tác cho use case UC005 – Khóa người dùng.



Hình : Biểu đồ tương tác cho use case UC005 – Khóa người dùng

* Lớp biên: *UserDetailsView*
* Lớp điều khiển: *UserDetailsPresenter, UserManager, UserController*
* Lớp thực thể: *User*

### Biểu đồ tương tác cho use case UC006 – Khóa dịch vụ.



Hình : Biểu đồ tương tác cho use case UC006 – Khóa dịch vụ

* Lớp biên: *ServiceDetailsView*
* Lớp điều khiển: *ServiceDetailsPresenter, ServiceManager, ServiceController*
* Lớp thực thể: *Service*

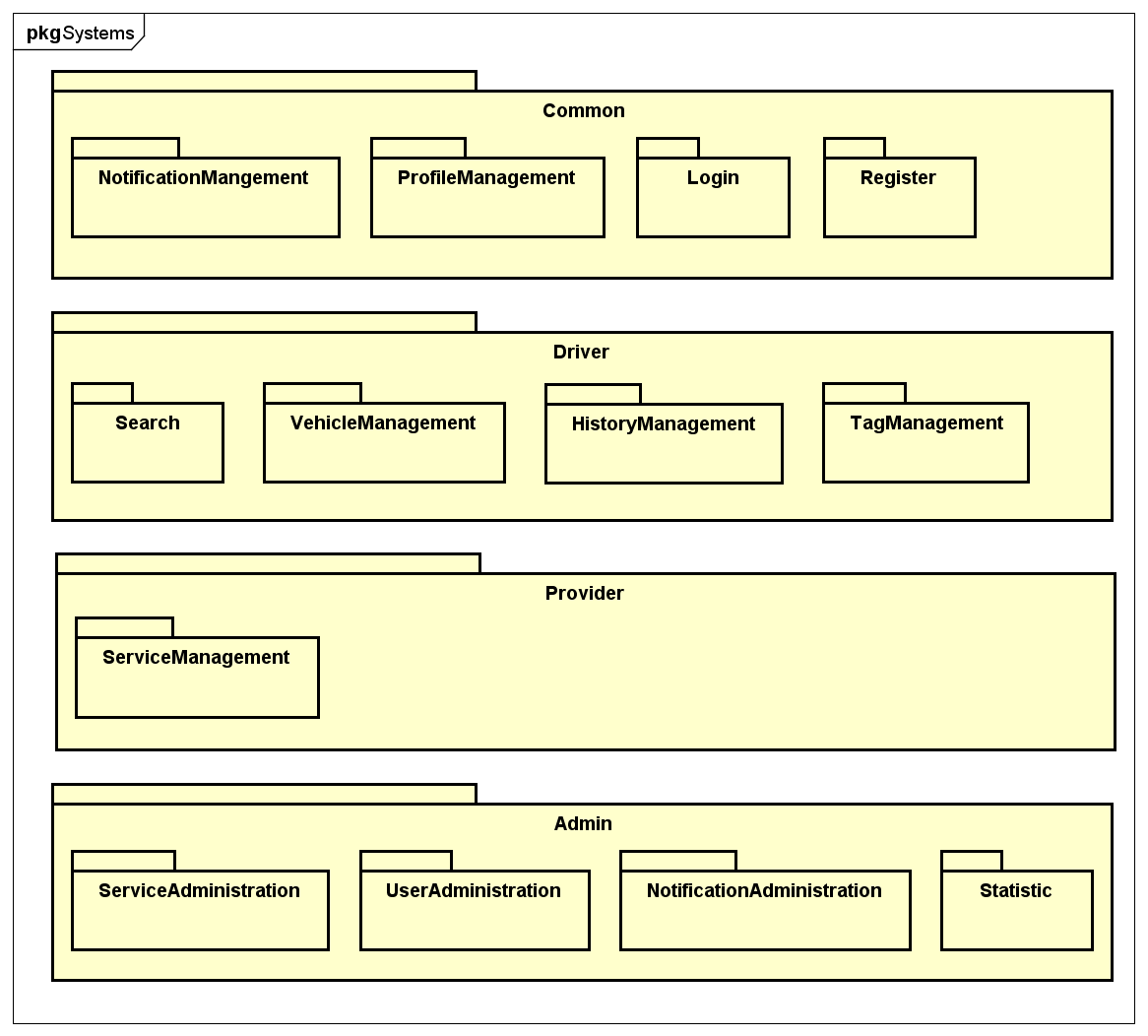
## Biểu đồ cấu trúc – (package/component diagram)

Với client trong hệ thống được chi làm 6 package như sau:



Hình : Biểu đồ các package của client trong hệ thống

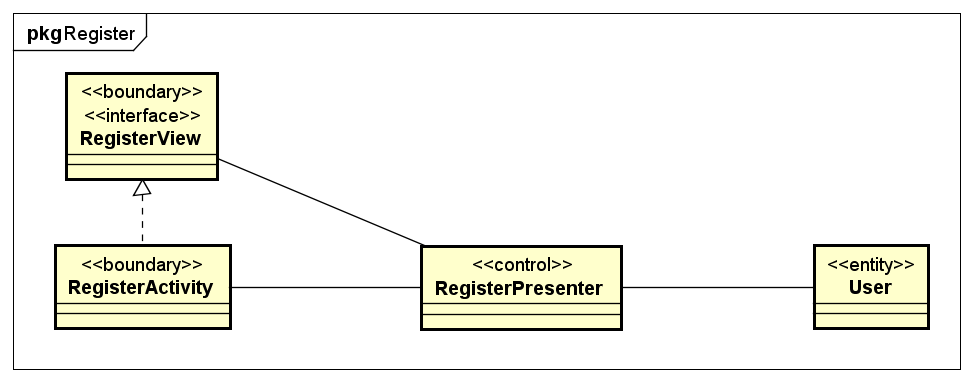
Trong client, package *Main* chứa các package con được phân theo loại người dùng và chức năng của chúng.



Hình : Biểu đồ các package con trong Systems package

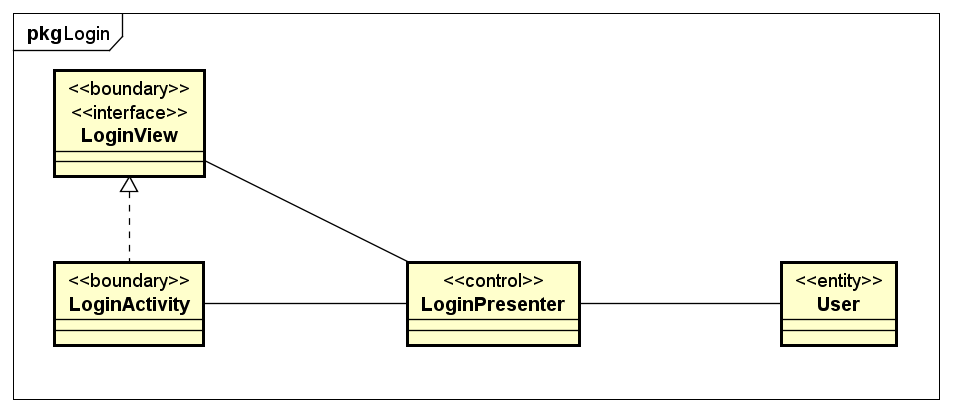
## Biểu đồ lớp phân tích

### Biểu đồ lớp phân tích package đăng ký – Register



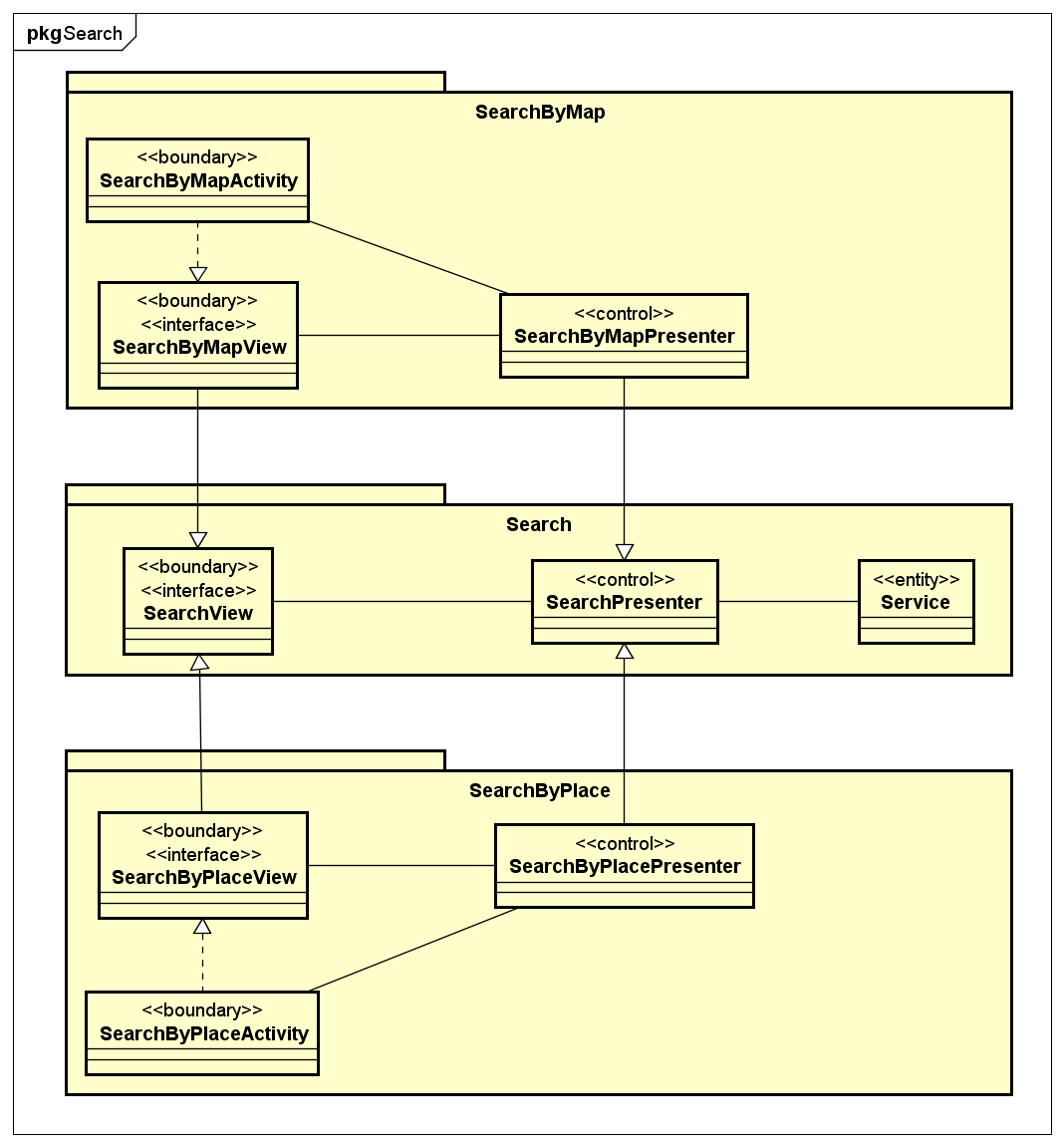
Hình : Biểu đồ lớp phân tích cho package đăng ký

### Biểu đồ lớp phân tích package đăng nhập – Login



Hình : Biểu đồ lớp phân tích cho package đăng nhập

### Biểu đồ lớp phân tích package tìm kiếm dịch vụ - Search



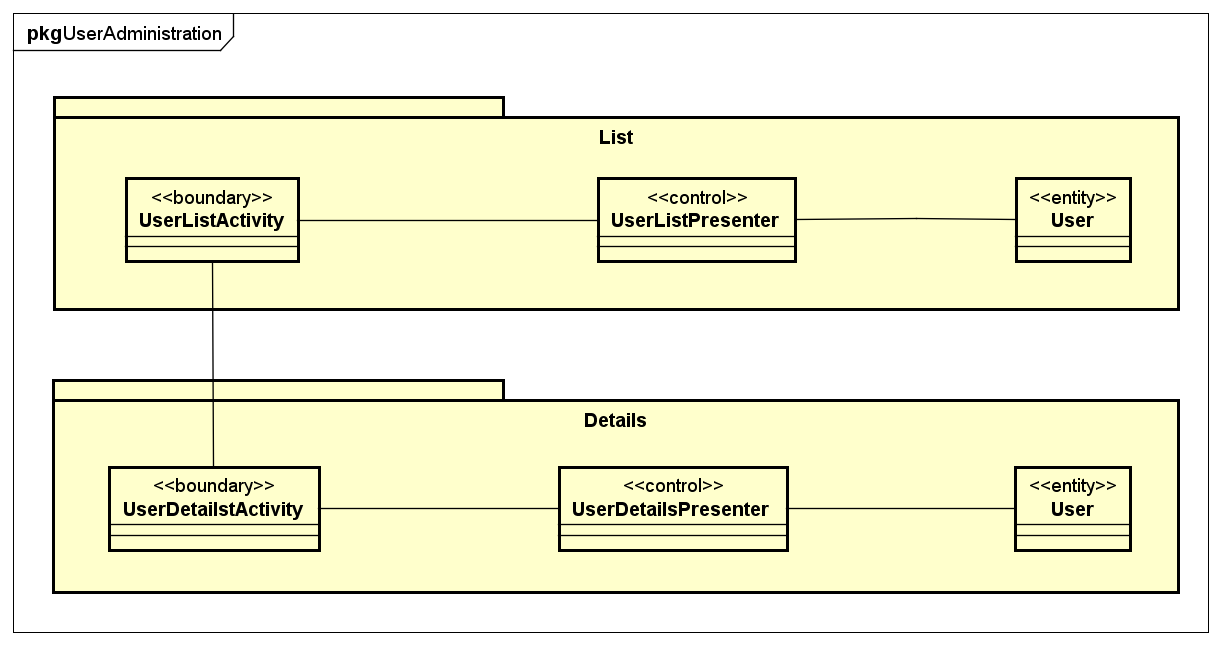
Hình : Biểu đồ lớp phân tích cho package tìm kiếm dịch vụ

### Biểu đồ lớp phân tích package quản lý dịch vụ - ServiceManagement



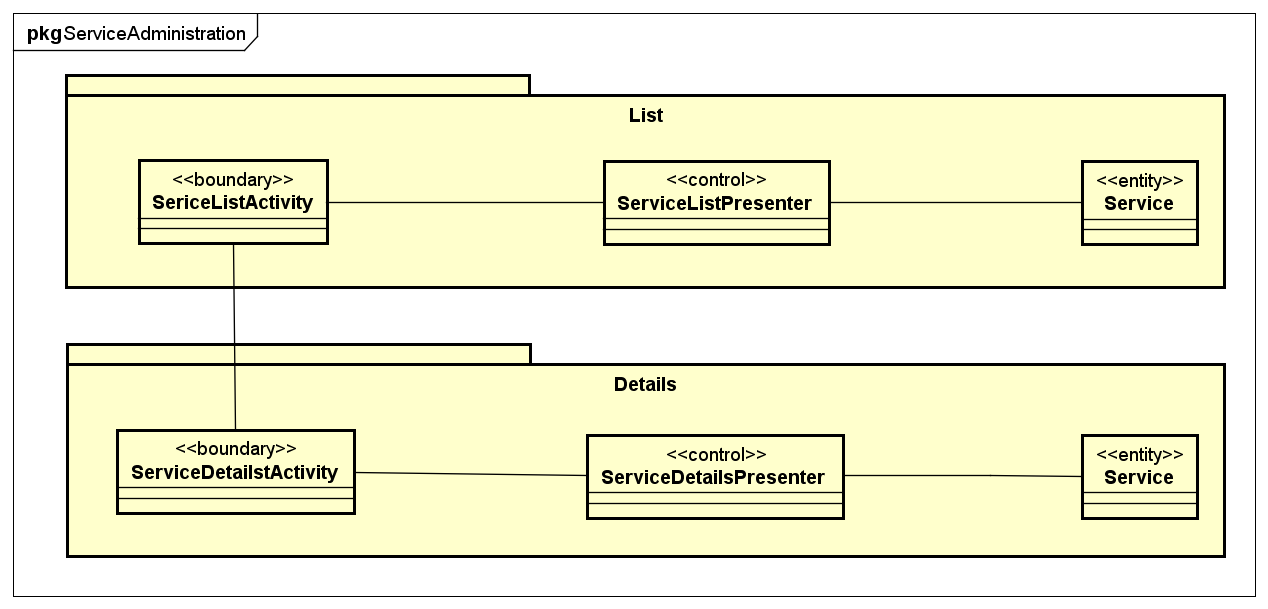
Hình : Biểu đồ lớp phân tích cho package quản lý dịch vụ

### Biểu đồ lớp phân tích package quản trị người dùng - UserAdministration



Hình : Biểu đồ lớp phân tích cho package quản trị người dùng

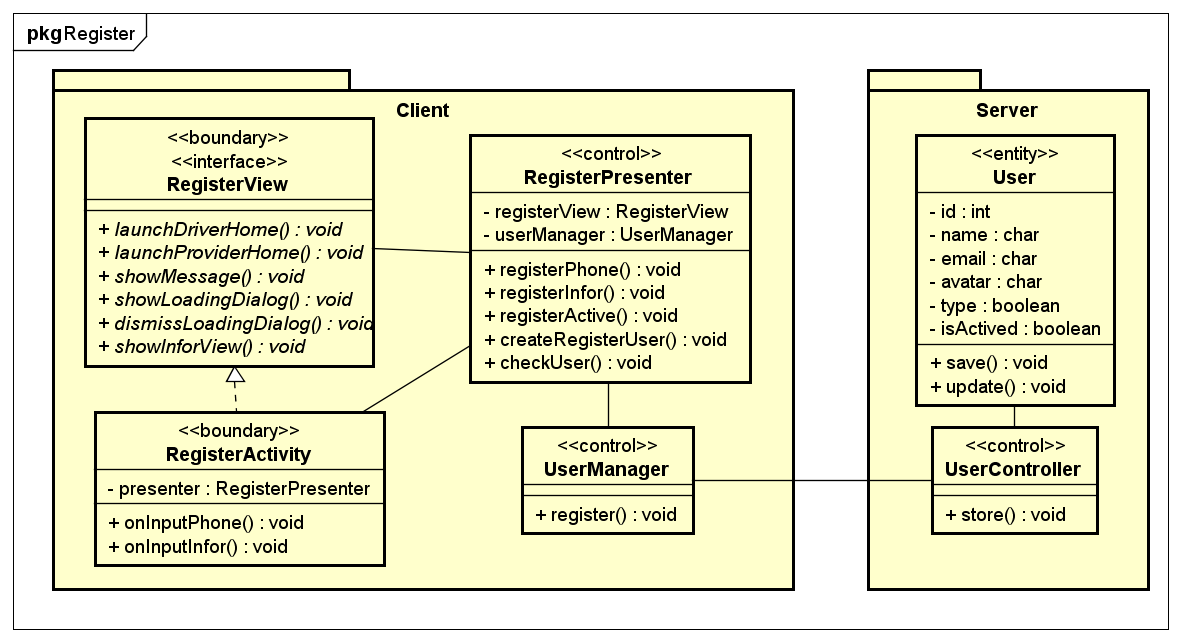
### Biểu đồ lớp phân tích package quản trị dịch vụ - ServiceAdministration



Hình : Biểu đồ lớp phân tích cho package quản trị dịch vụ

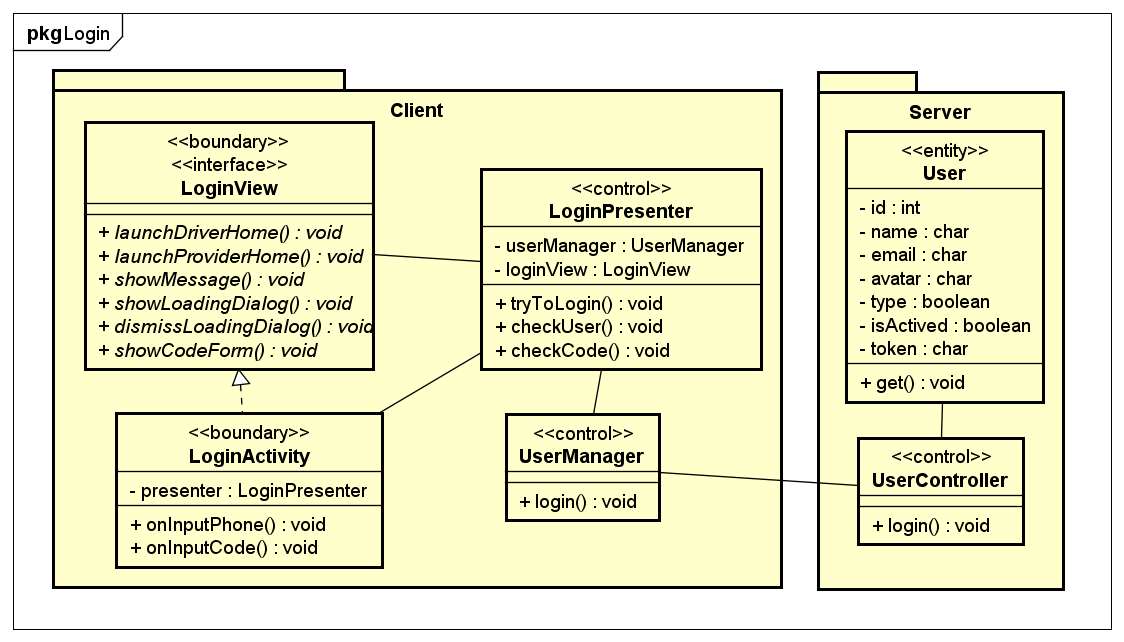
## Thiết kế lớp

### Biểu đồ lớp chi tiết cho chức năng đăng ký tài khoản



Hình : Biểu đồ lớp chi tiết cho chức năng đăng ký tài khoản

### Biểu đồ lớp chi tiết cho chức năng đăng nhập



Hình : Biểu đồ lớp chi tiết cho chức năng đăng nhập

### Biểu đồ lớp chi tiết cho chức năng tìm kiếm dịch vụ



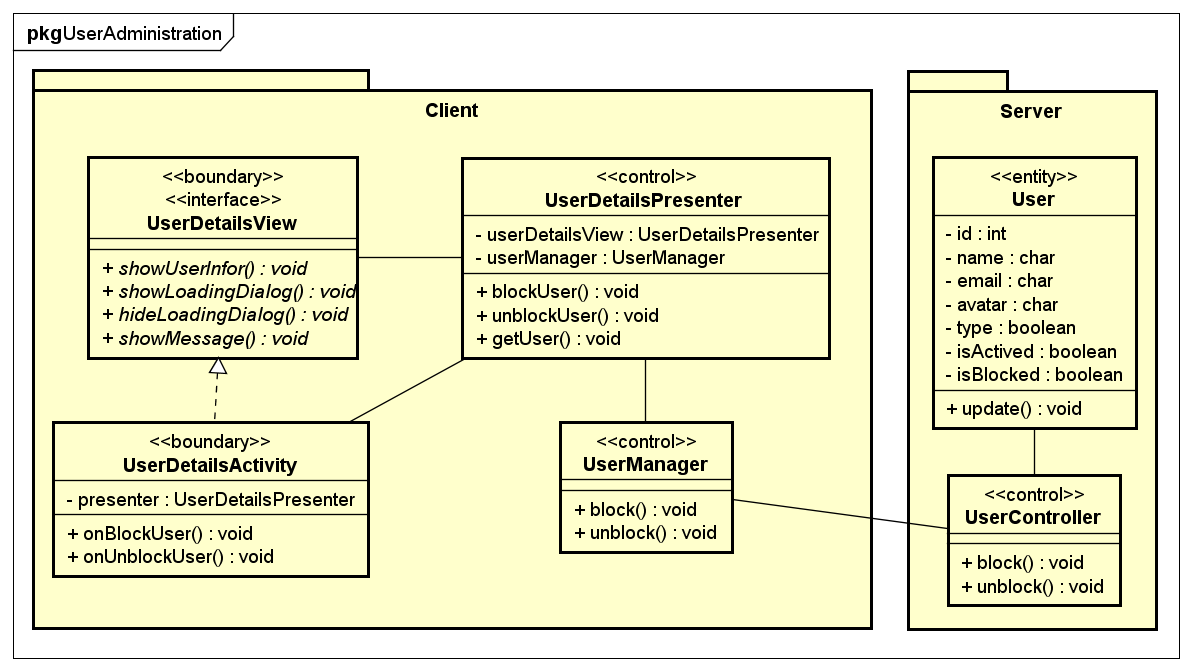
Hình : Biểu đồ lớp chi tiết cho chức năng tìm kiếm dịch vụ

### Biểu đồ lớp chi tiết cho chức năng tạo mới/sửa dịch vụ



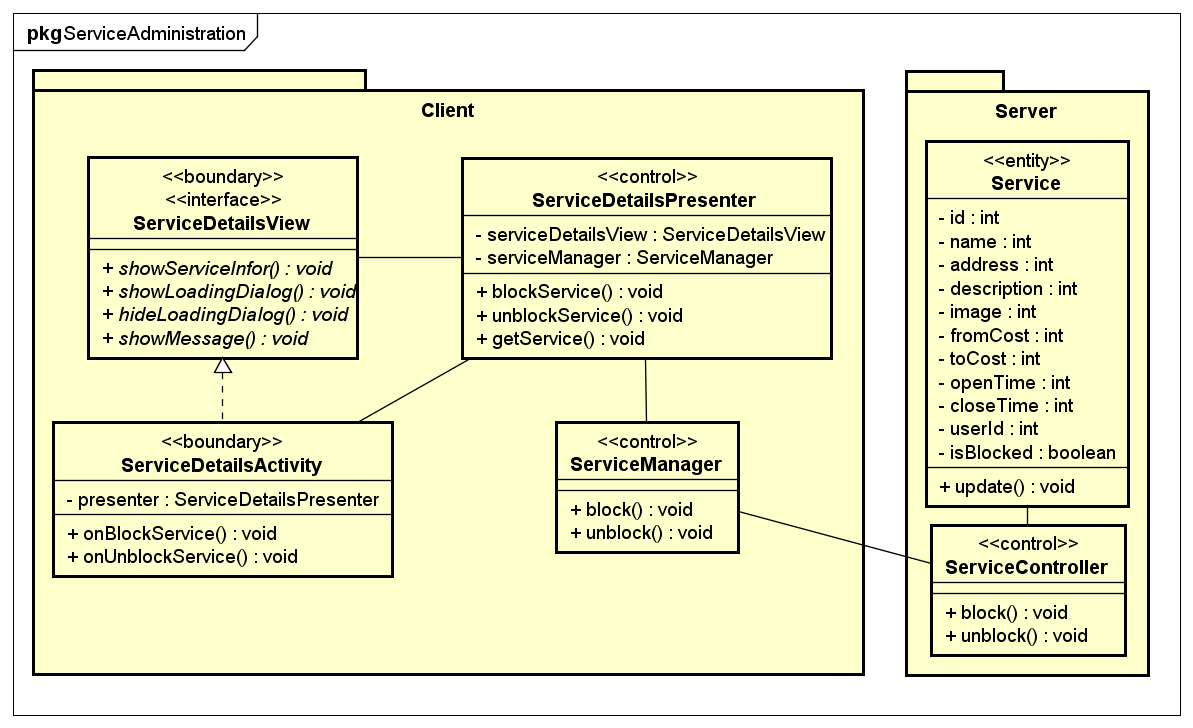
Hình : Biểu đồ lớp chi tiết cho chức năng tạo mới/sửa dịch vụ

### Biểu đồ lớp chi tiết cho chức năng quản trị người dùng



Hình : Biểu đồ lớp chi tiết cho chức năng quản trị người dùng

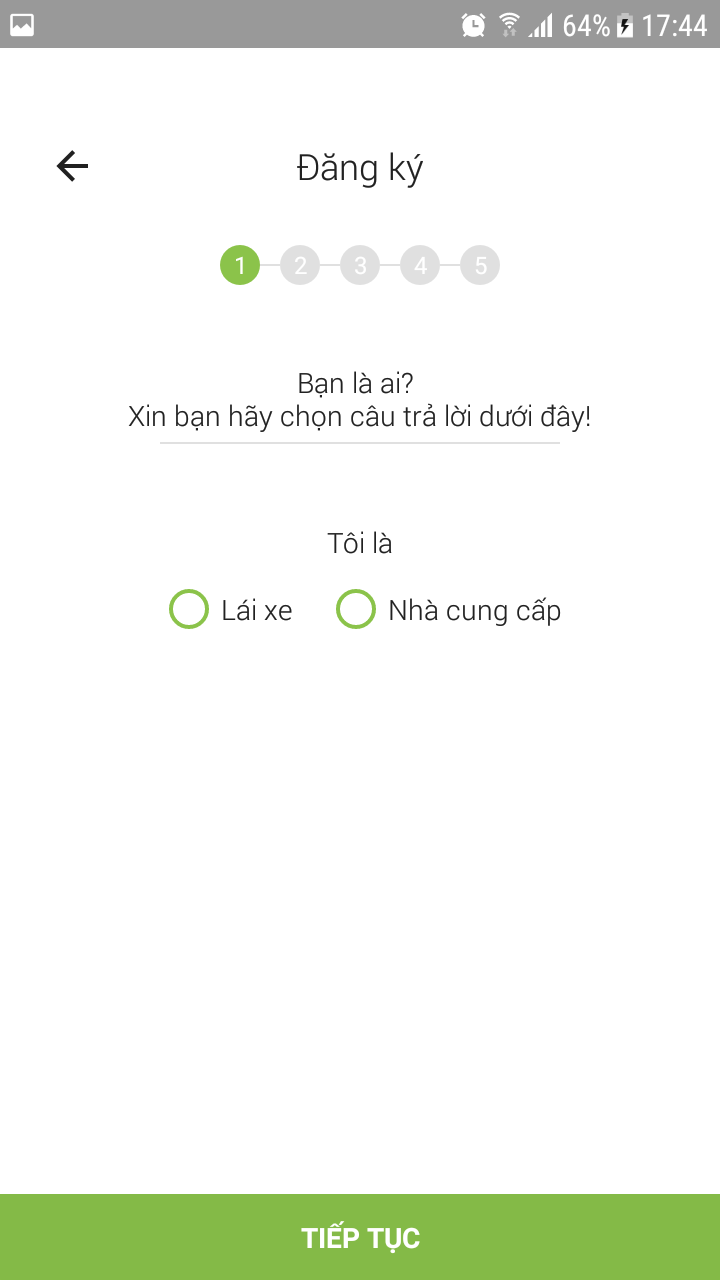
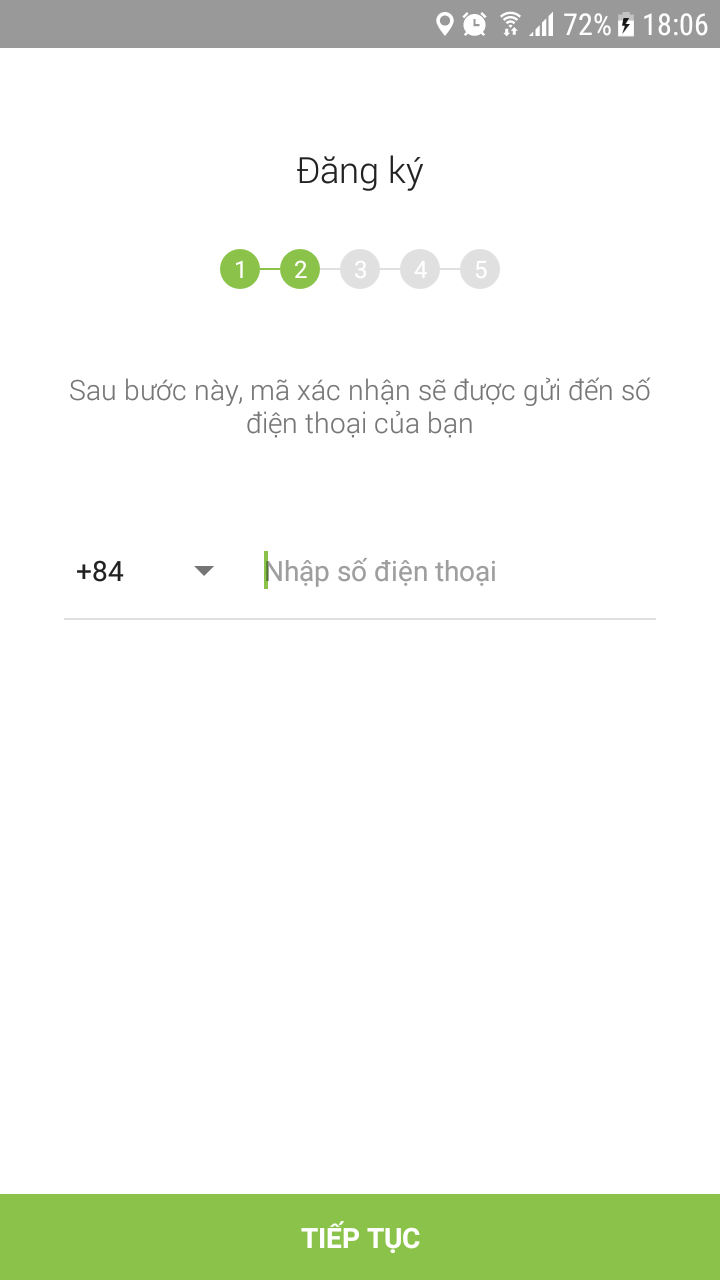
### Biểu đồ lớp chi tiết cho chức năng quản trị dịch vụ

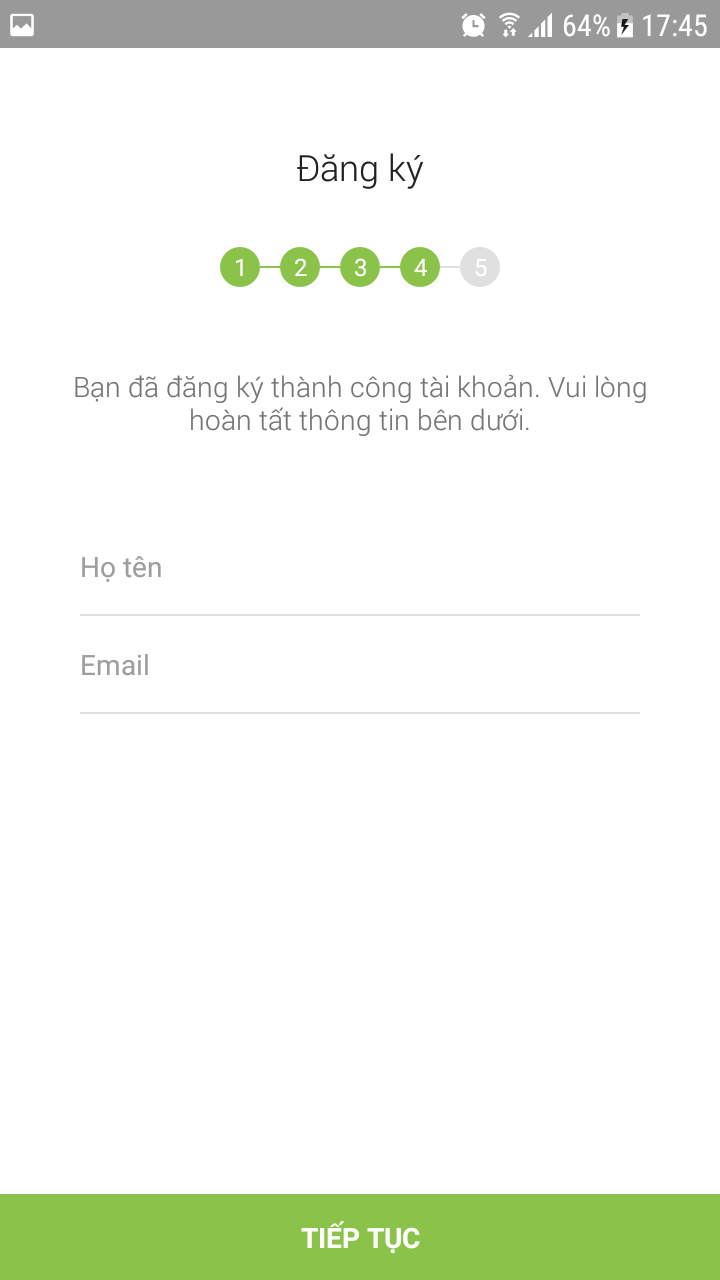


Hình : Biểu đồ lớp chi tiết cho chức năng quản trị dịch vụ

## Thiết kế giao diện

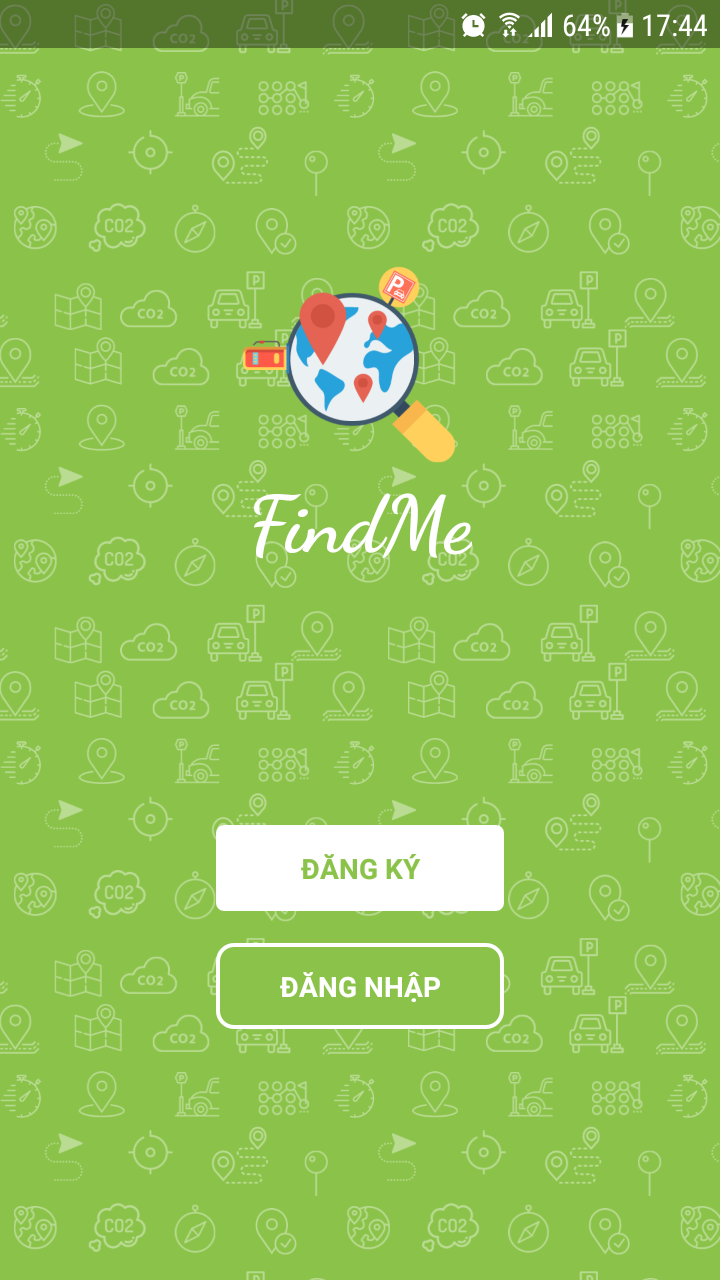
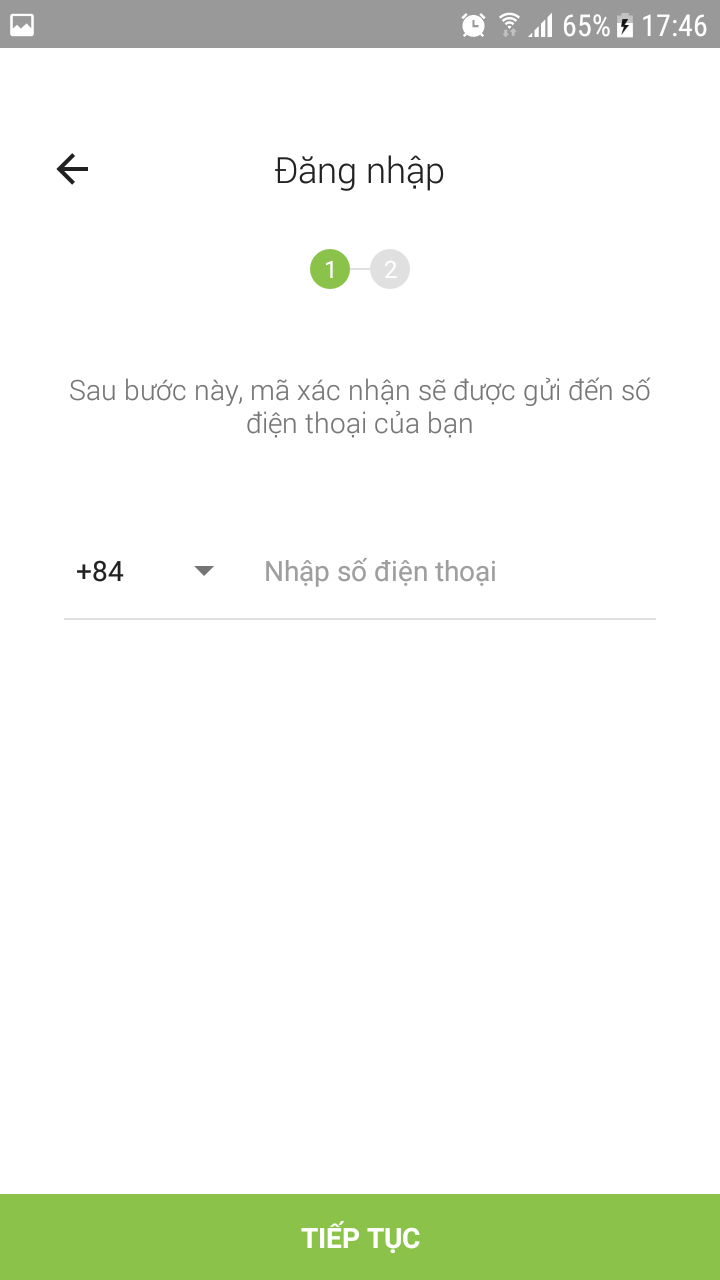
### Giao diện màn hình đăng ký

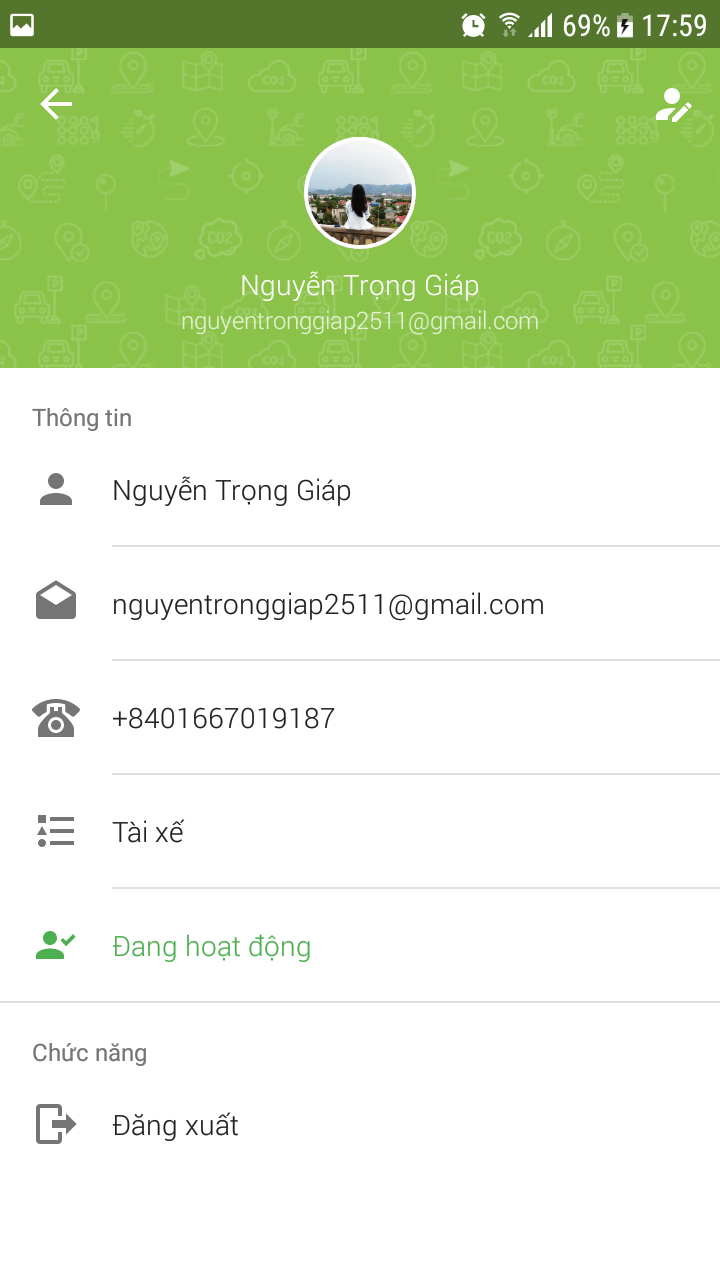
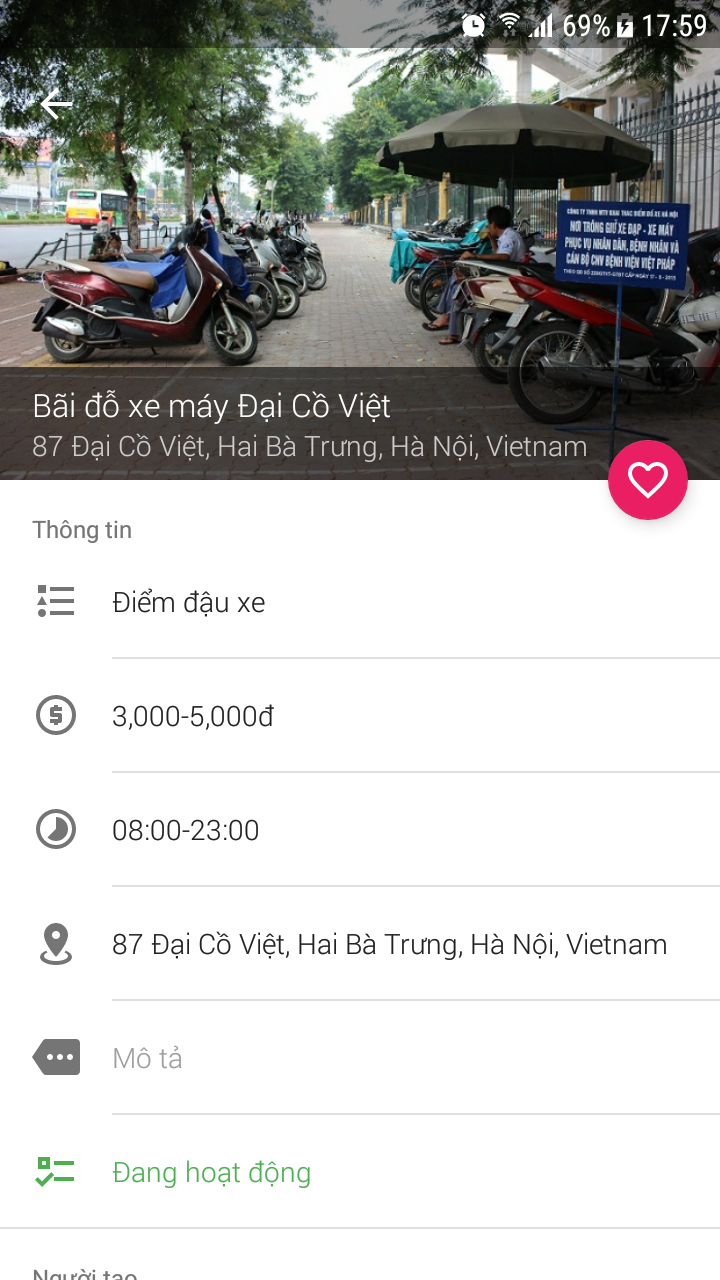
Hình : Giao diện các màn hình đăng ký

### Giao diện màn hình đăng nhập

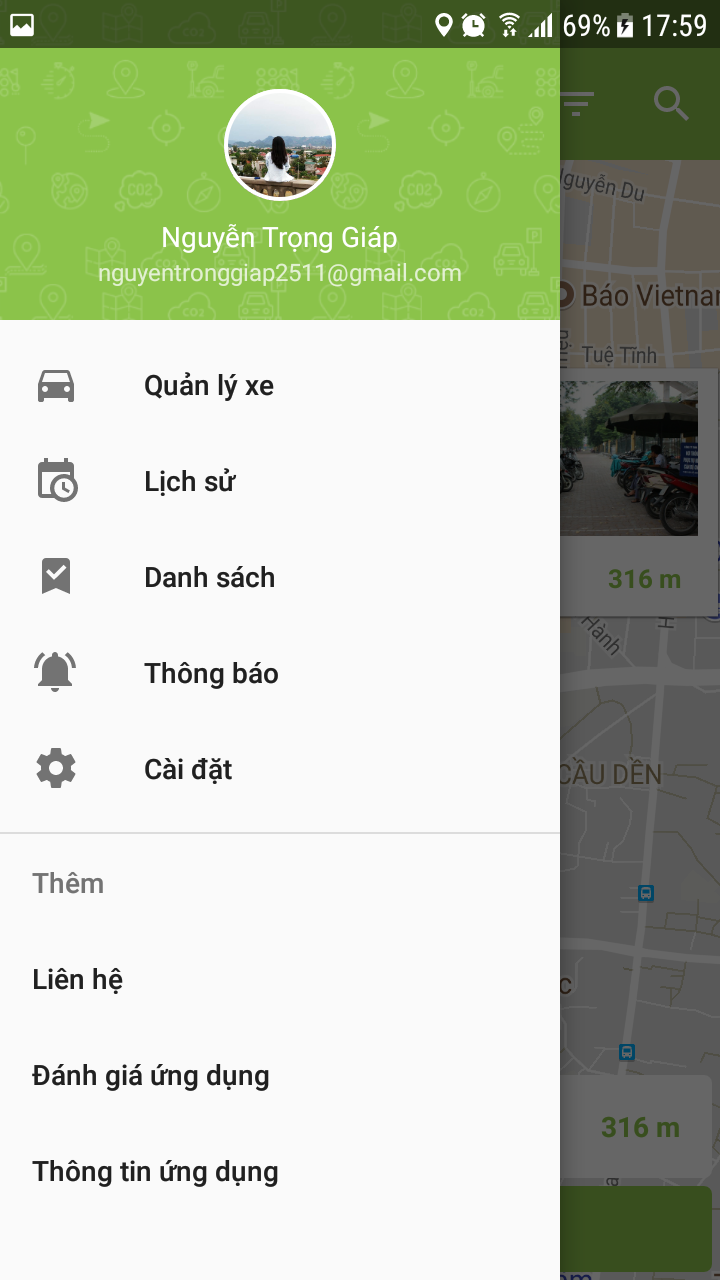
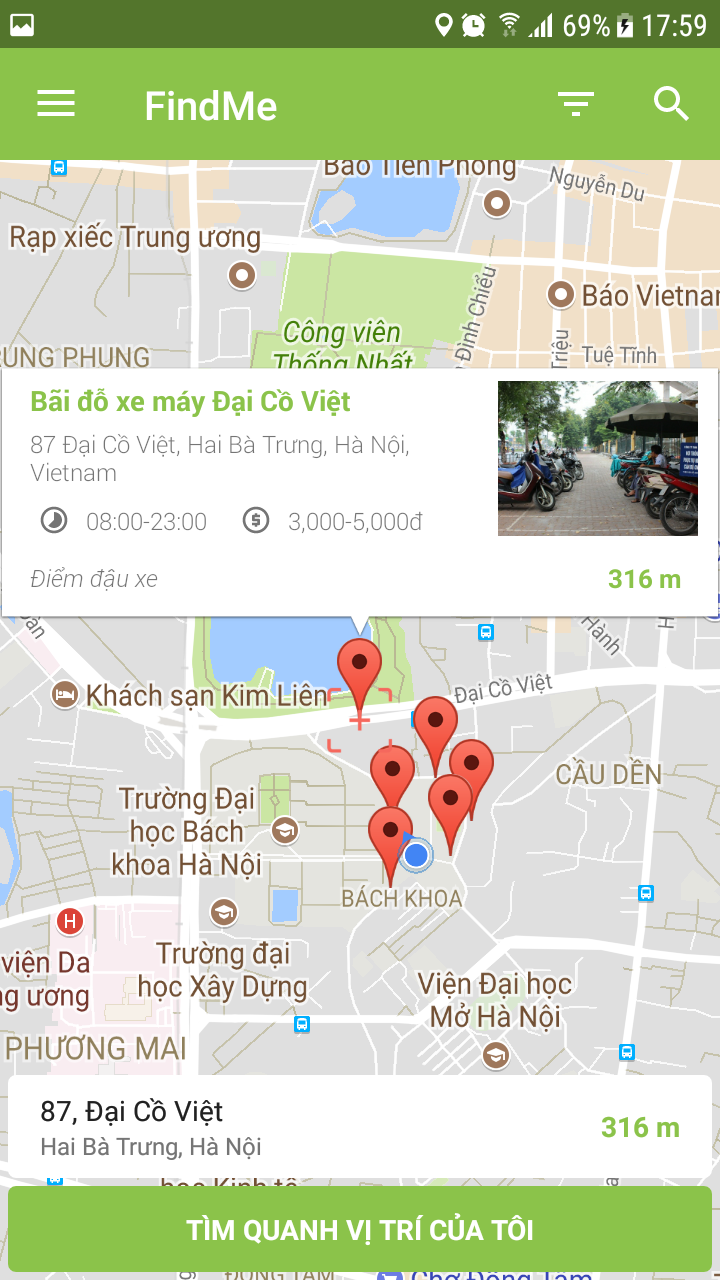
Hình : Giao diện màn hình đăng nhập

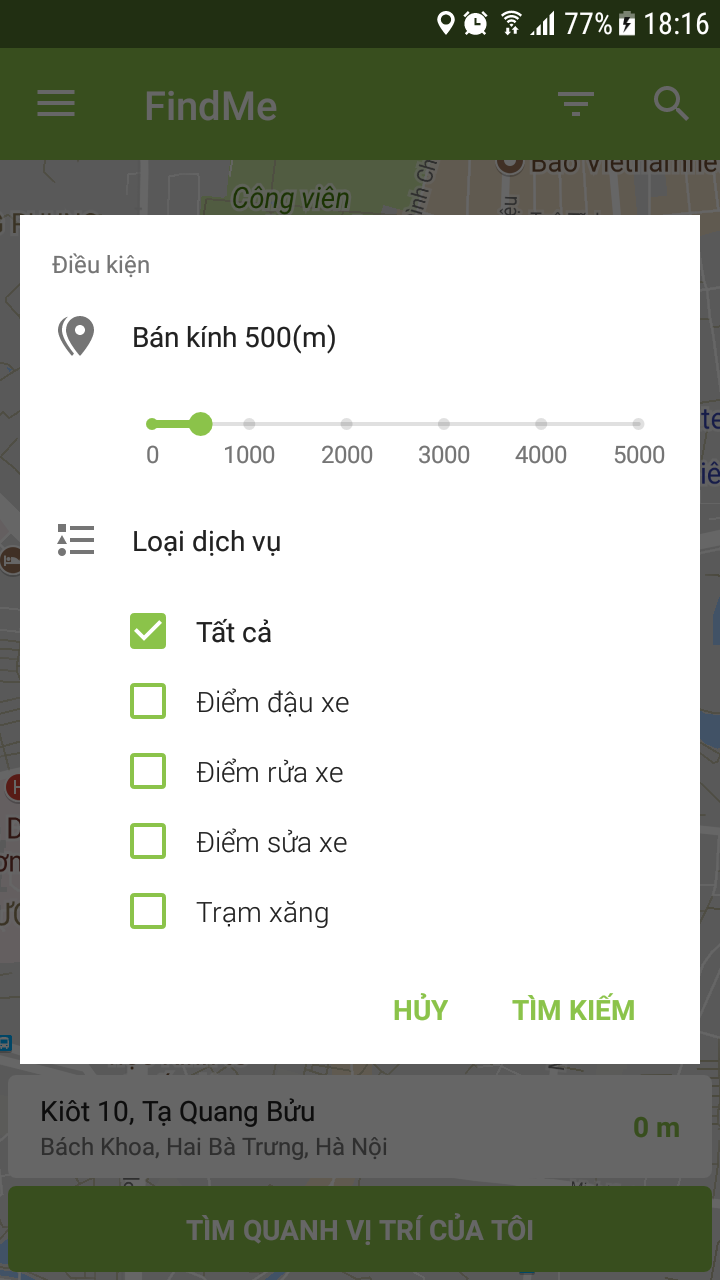
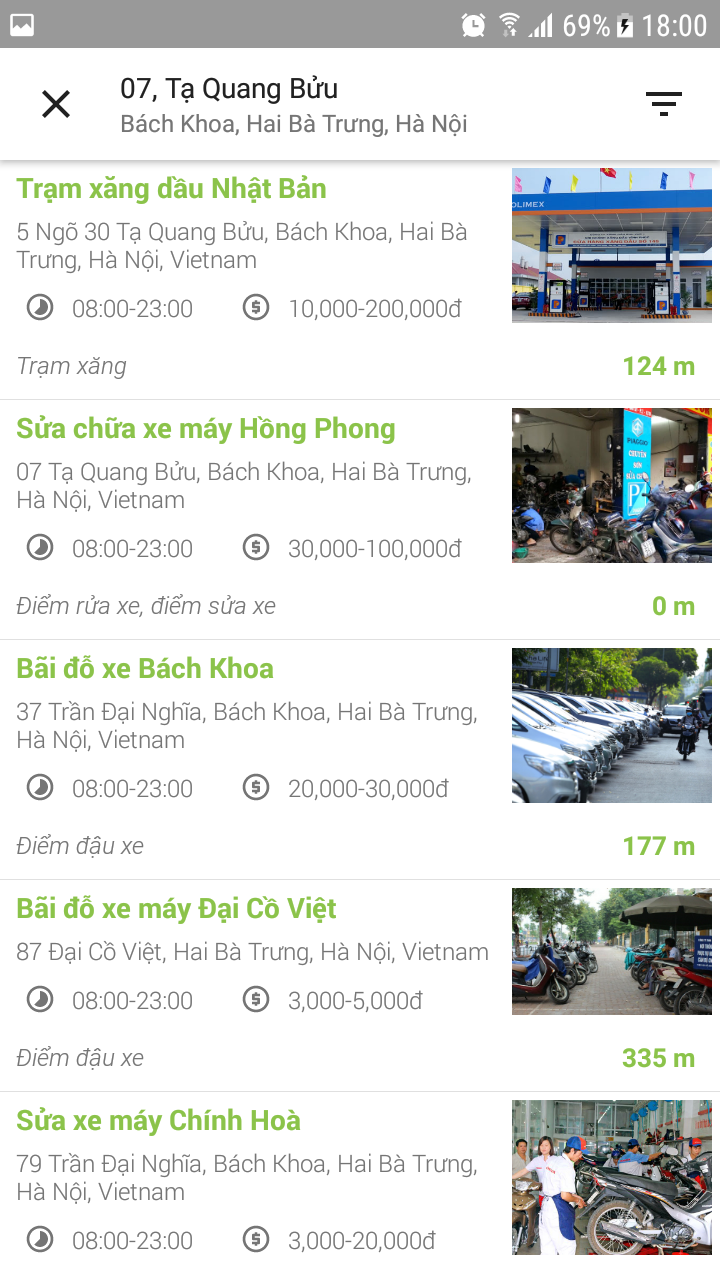
### Giao diện màn hình chi tiết người dùng và dịch vụ

Hình : Giao diện màn hình chi tiết người dùng và dịch vụ

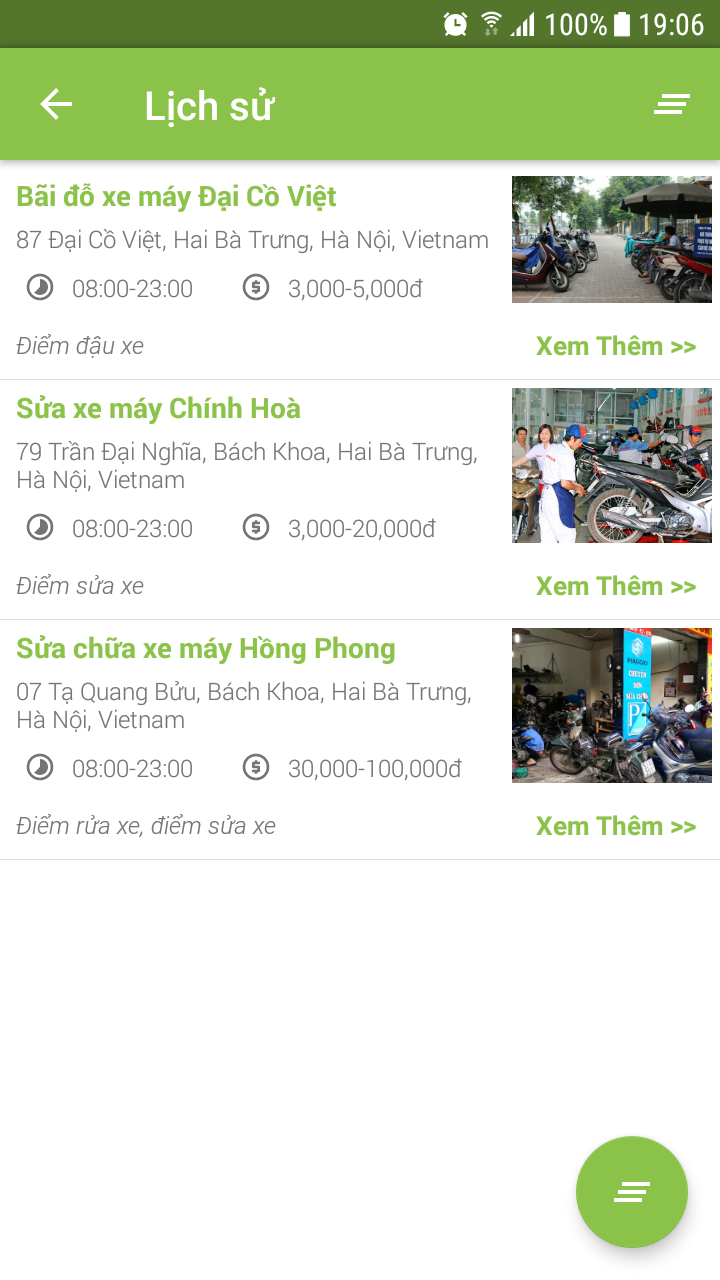
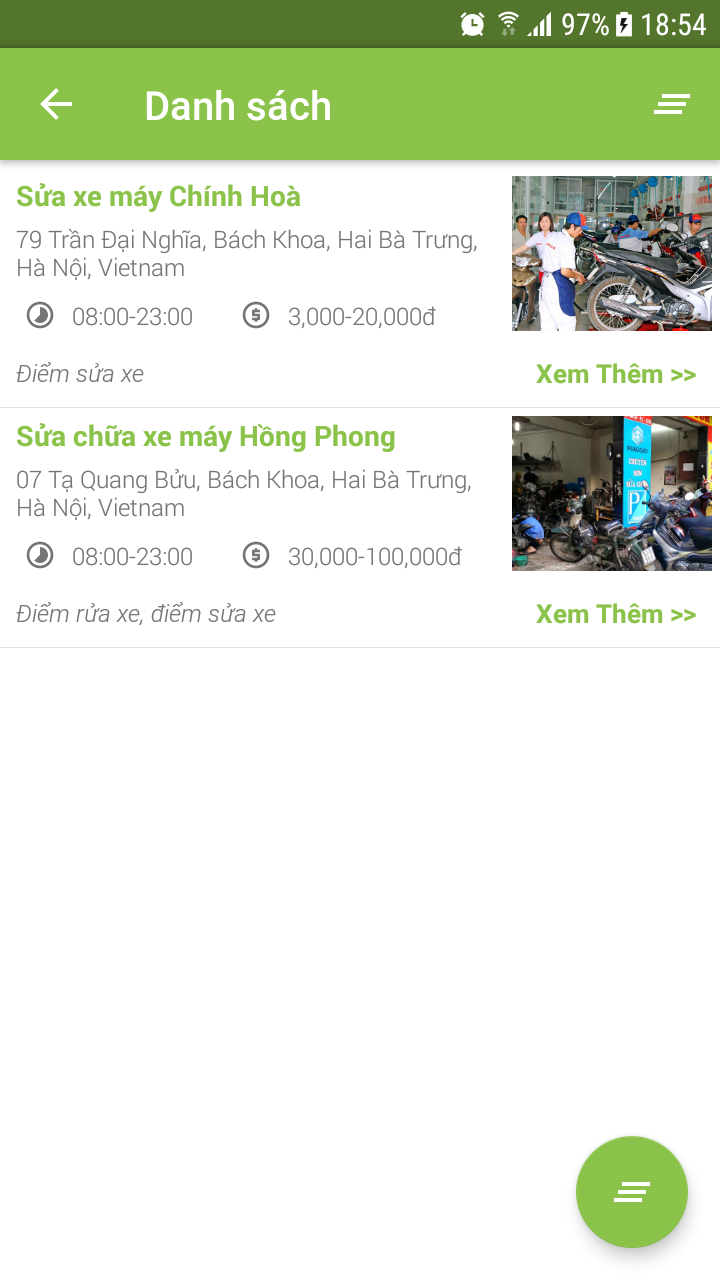
### Giao diện màn hình tìm kiếm dịch vụ

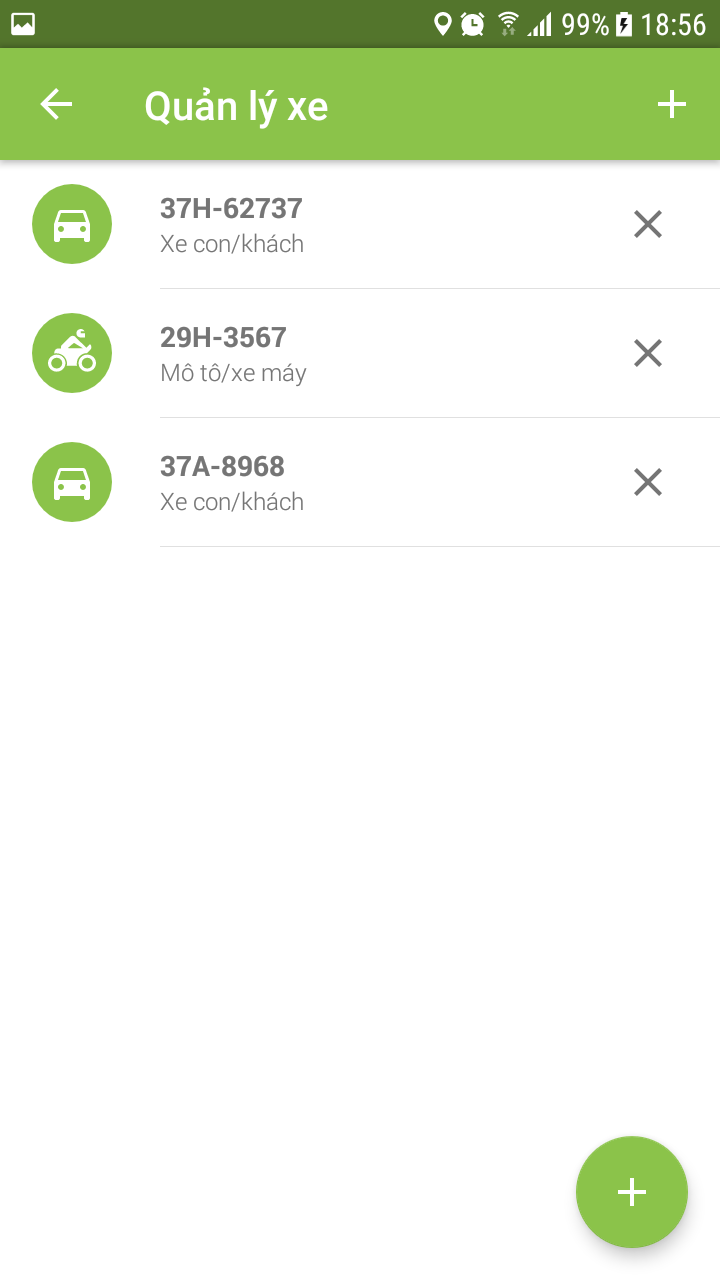
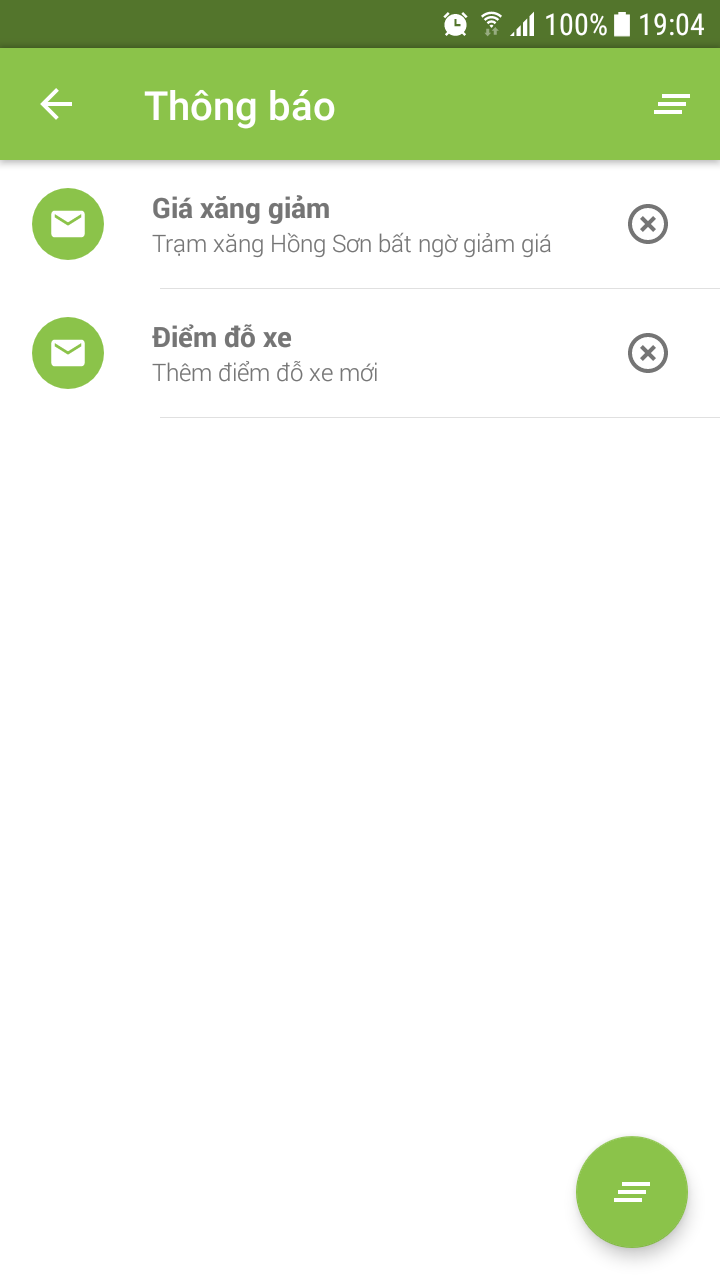
Hình : Giao diện màn hình tìm kiếm dịch vụ

### Giao diện màn hình lịch sử và danh sách đánh dấu

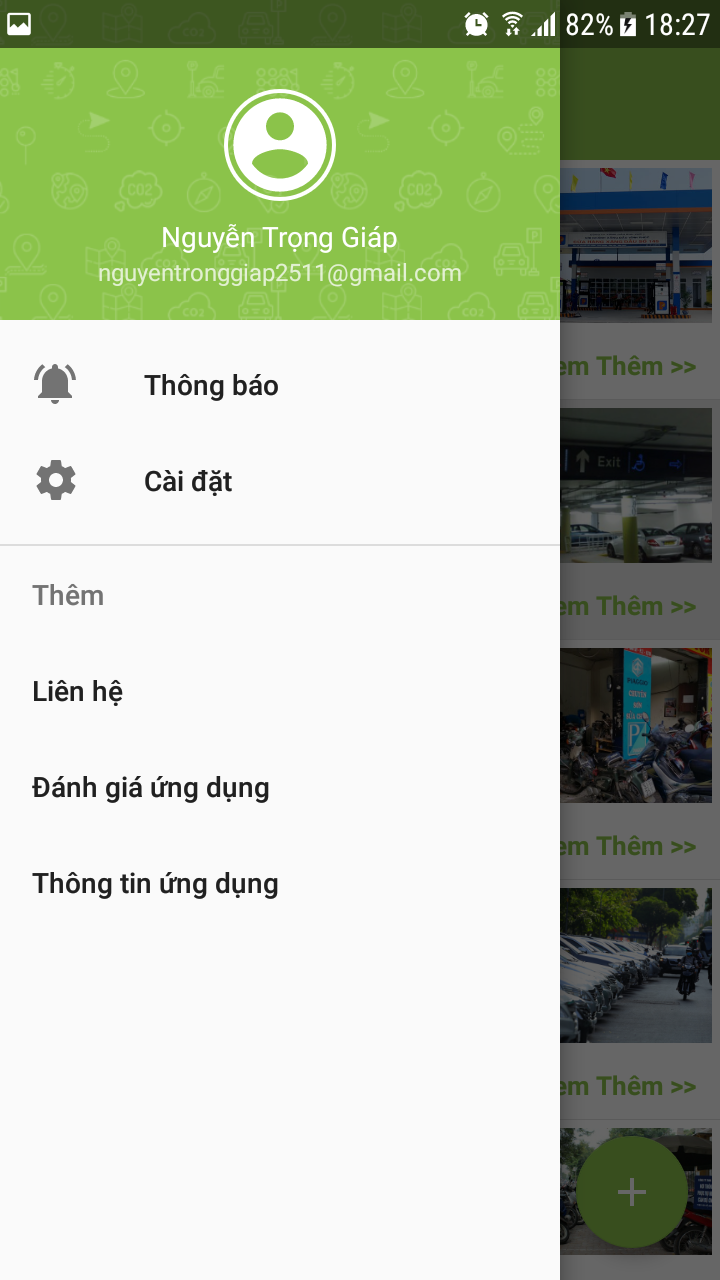
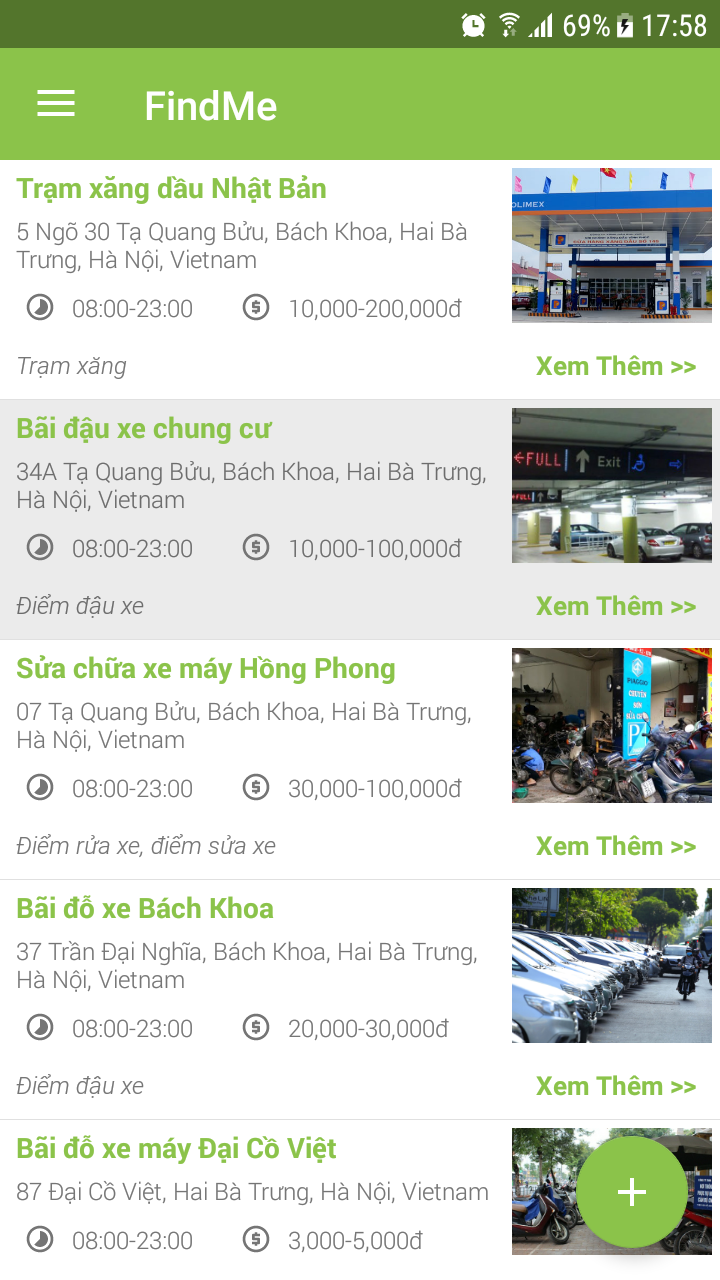
Hình : Giao diên màn hình lịch sử và danh sách đánh dấu

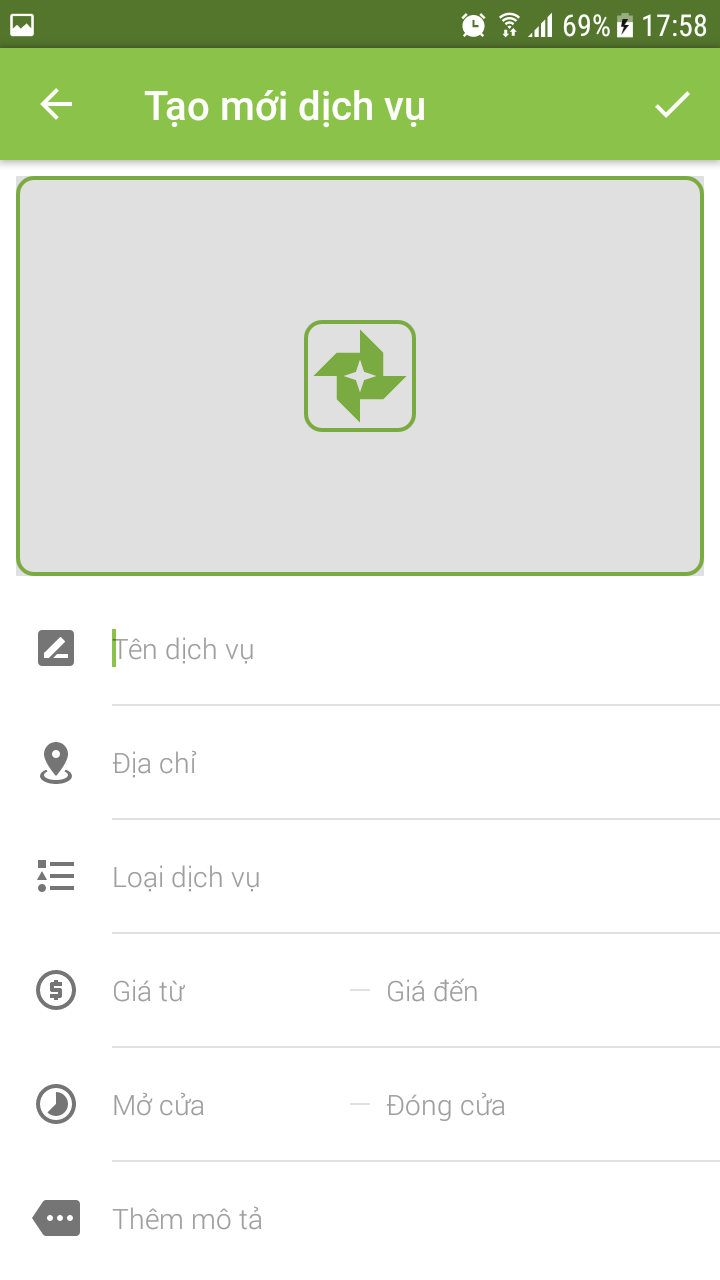
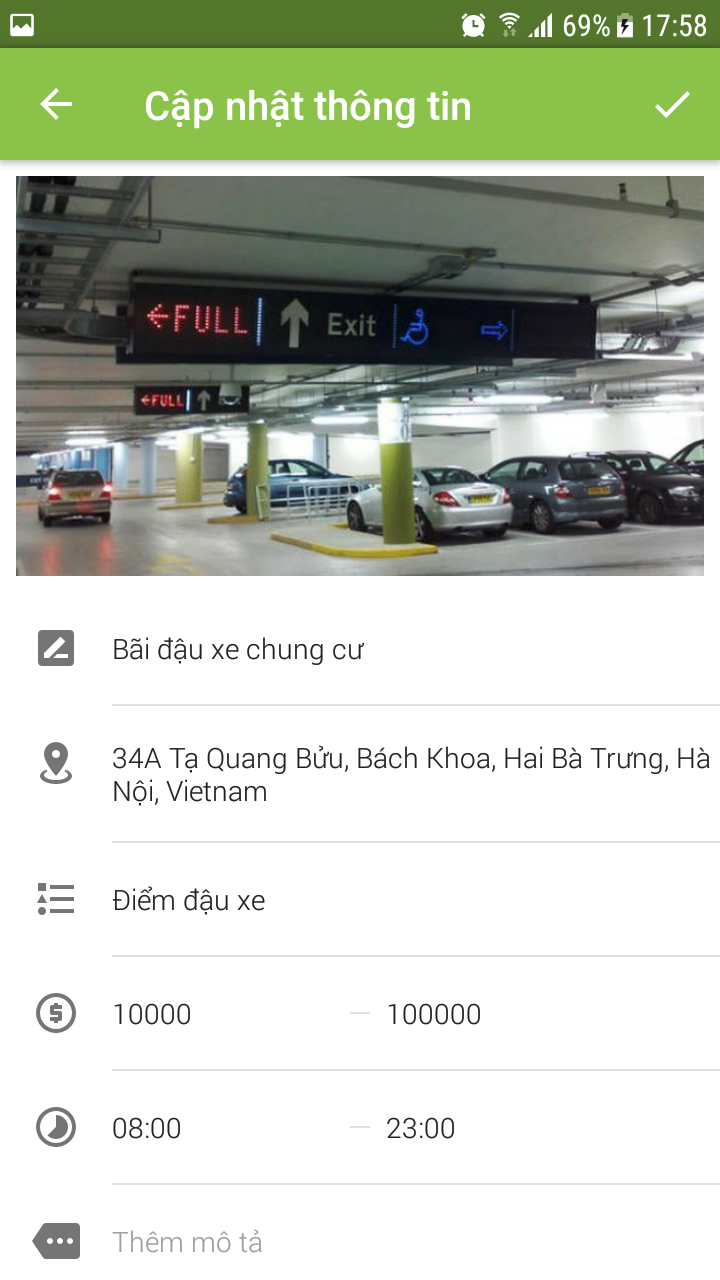
### Giao diệnmàn hình quản lý xe và thông báo

Hình : Giao diên màn hình quản lý xe và thông báo

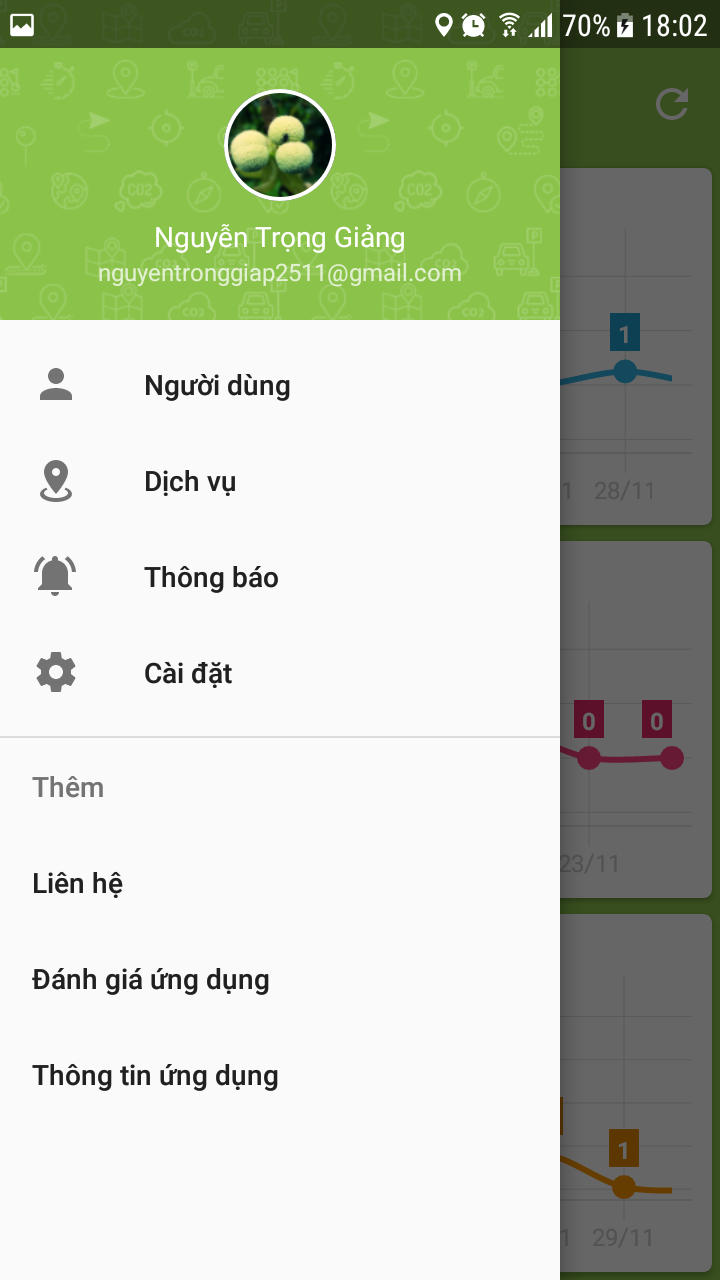
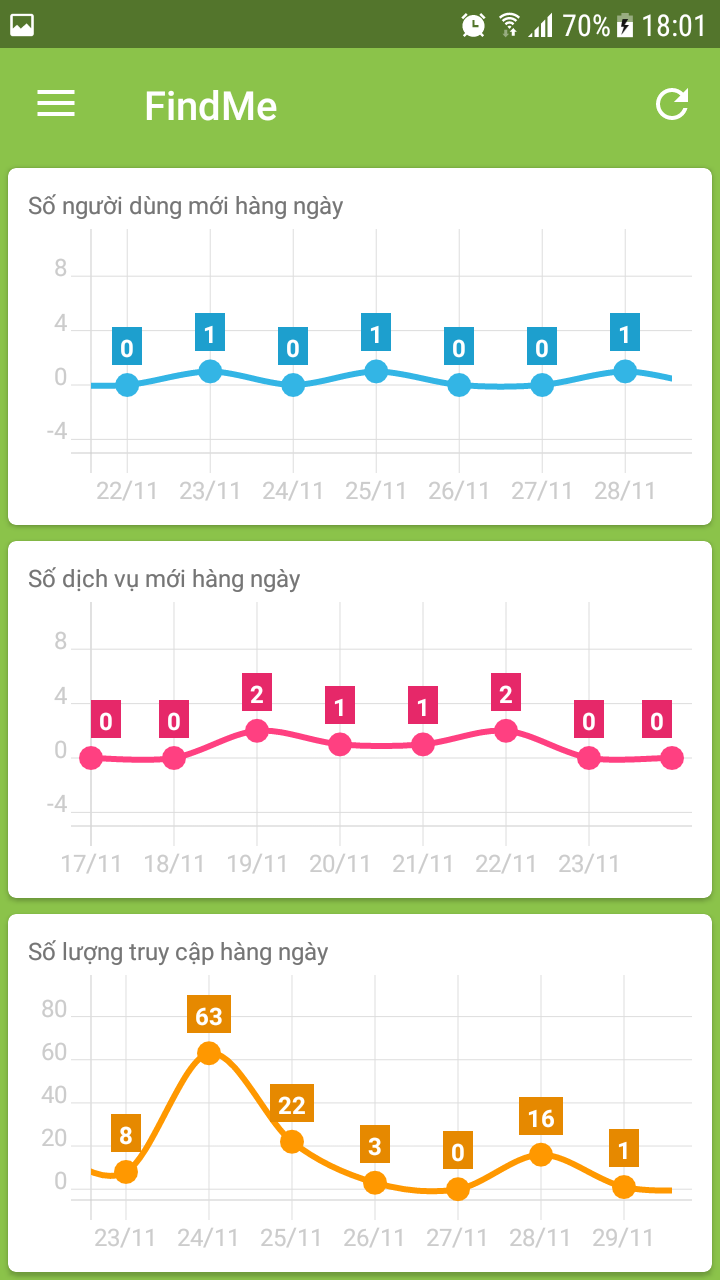
### Giao diện màn hình quản lý dịch vụ

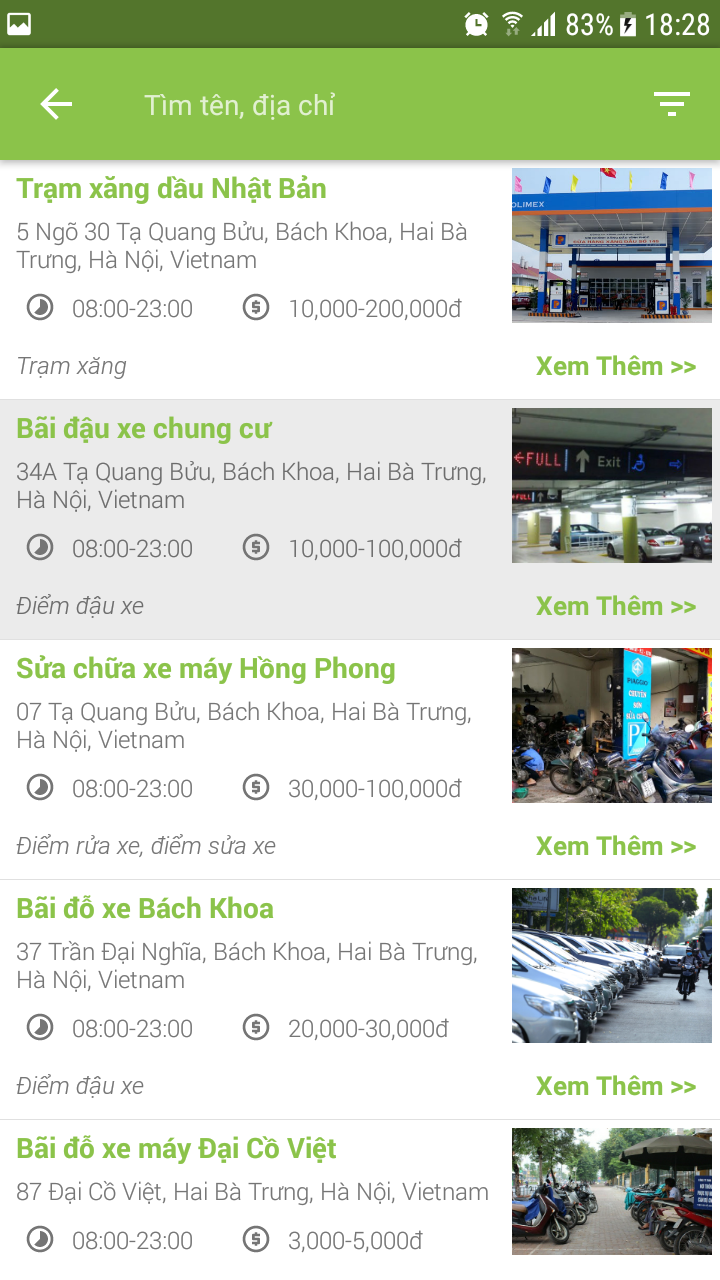
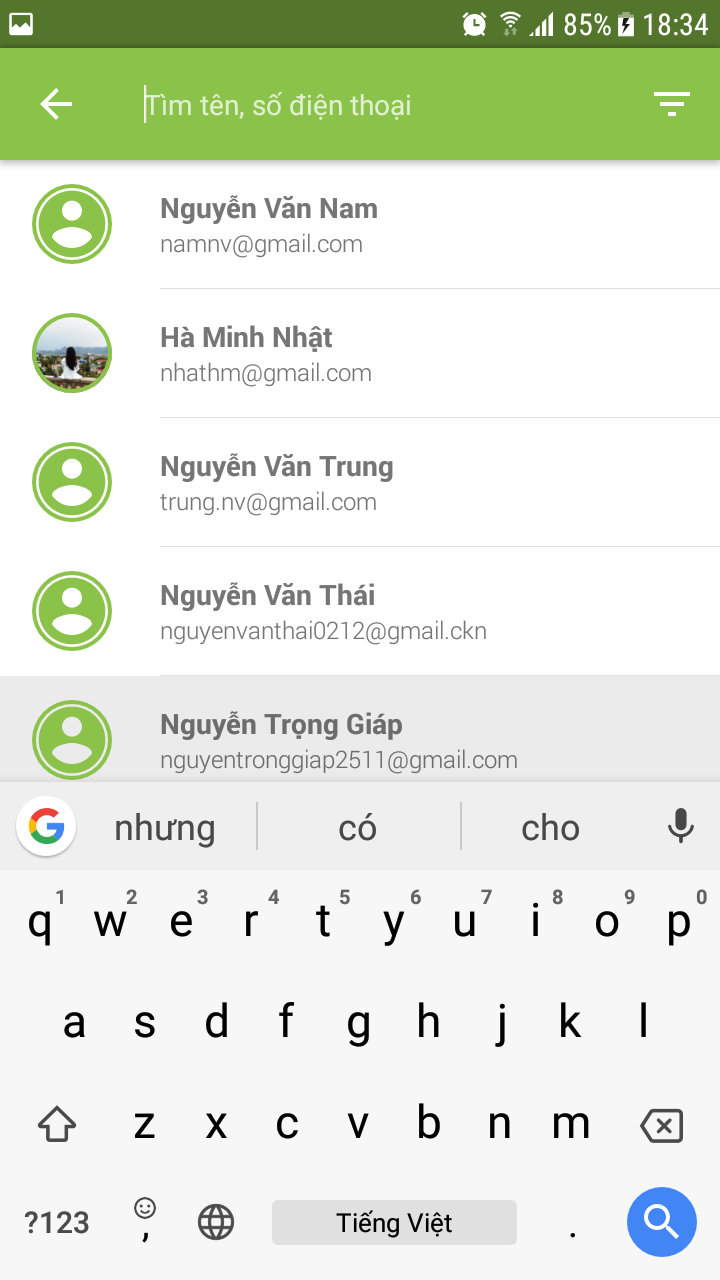
 

Hình : Giao diện màn hình quản lý dịch vụ

### Giao diện màn hình quản lý dịch vụ

 .

Hình : Giao diện màn hình quản trị của admin

## Thiết kế mô hình dữ liệu

### Yêu cầu bài toán

Một người dùng có thể đăng ký làm một trong hai loại người dùng của hệ thống: Lái xe, Nhà cung cấp. Các thông tin của người dùng bao gồm họ tên, số điện thoại, địa chỉ email.

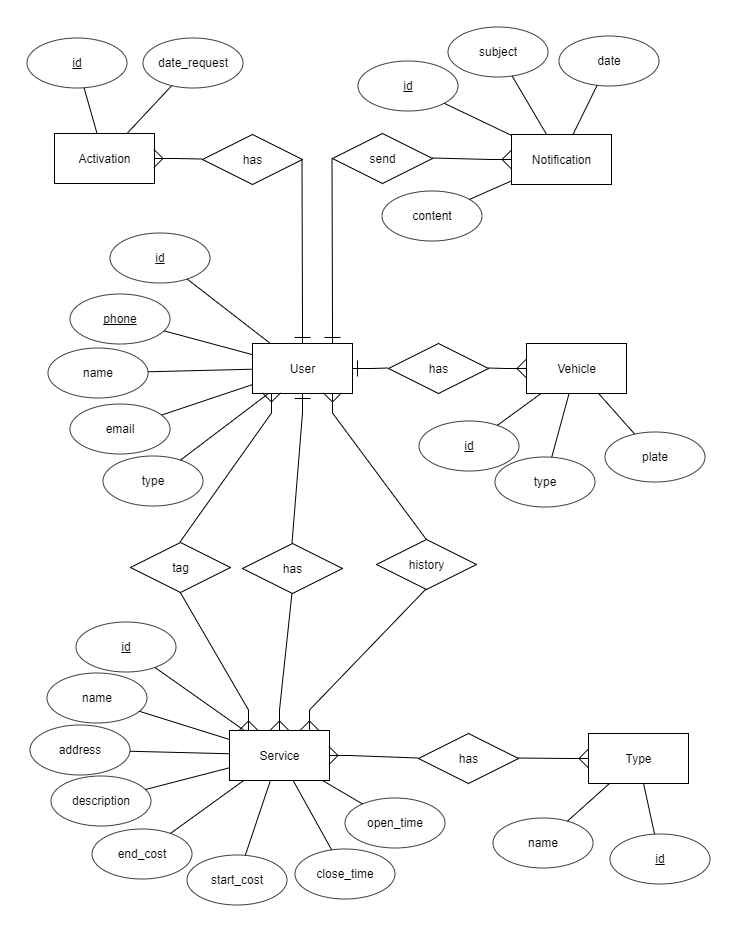
Nhà cung cấp có thể sở hữu nhiều dịch vụ, các thông tin dịch vụ bao gồm: tên dịch vụ, địa chỉ, mô tả, giá bắt đầu, giá kết thúc, thời gian đóng mở cửa, ảnh dịch vụ. Mỗi dịch vụ có thể bao gồm nhiều loại dịch vụ cùng lúc như đỗ xe, trạm xăng, sửa xe…

Lái xe có thể sở hữu nhiều phương tiện giao thông với các thông tin: biển số, và loại phương tiện (xe máy, ô tô). Hệ thống tự động lưu lịch sử cho lái xe với mỗi lần truy cập đến dịch vụ. Ngoài ra còn có chức năng đánh dấu dịch vụ để cho vào danh sách ghi chú.

Mỗi người dùng khi đăng nhập vào hệ thống sẽ được tính là một lần kích hoạt sử dụng.

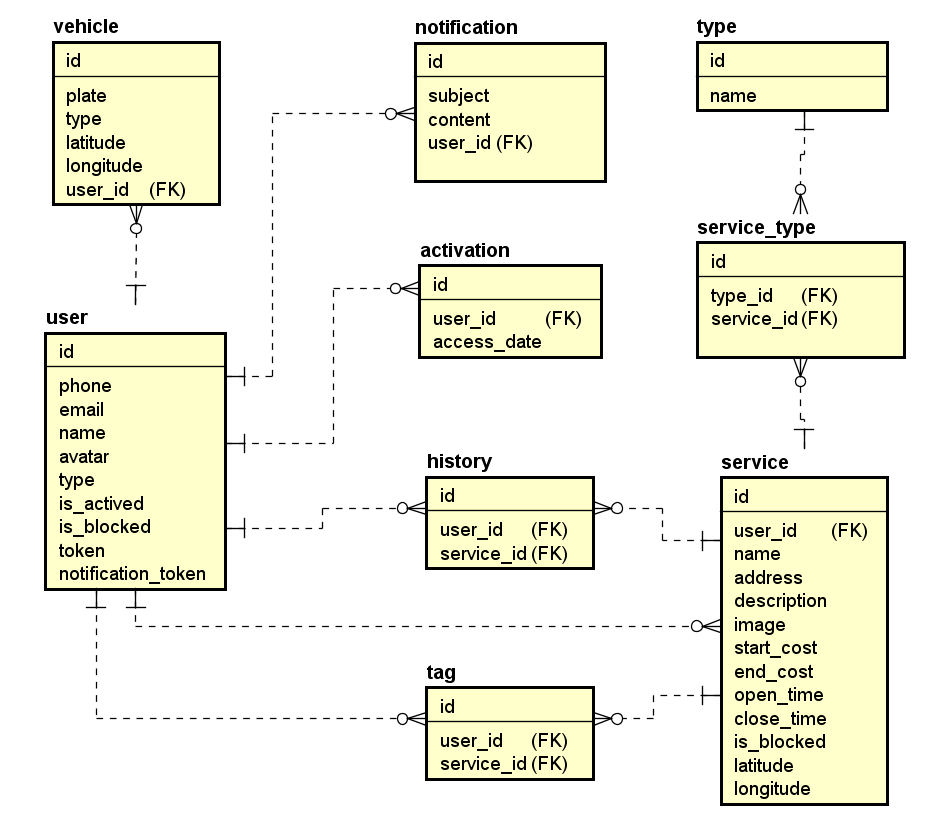
Hệ thống có quản trị viên có thể gửi thông báo đến người dùng, nội dung của thông báo bao gồm: chủ đề và nội dung, ngày gửi thông báo.

### Biểu đồ thực thể liên kết



Hình 55: Biểu đồ thực thể liên kết

### Thiết kế cơ sở dữ liệu



Hình : Biểu đồ cơ sở dữ liệu

### Thiết kế chi tiết bảng

* Bảng user

|  |  |  |  |
| --- | --- | --- | --- |
| Tên | Kiểu | Mô tả | Ràng buộc |
| id | int | Mã người dùng | Khóa chính |
| phone | varchar | Số điện thoại | Duy nhất |
| email | varchar | Địa chỉ thư điện tử |  |
| name | varchar | Tên người dùng |  |
| avatar | varchar | Đường dẫn ảnh đại diện |  |
| type | int | Loại người dùng |  |
| is\_actived | boolean | Trạng thái kích hoạt |  |
| is\_blocked | boolean | Trạng thái bị chặn |  |
| token | varchar | Khóa đăng nhập |  |
| notification\_token | varchar | Khóa thông báo |  |

Bảng : Bảng user

* Bảng service

|  |  |  |  |
| --- | --- | --- | --- |
| Tên | Kiểu | Mô tả | Ràng buộc |
| id | int | Mã dịch vụ | Khóa chính |
| user\_id | int | Mã người dùng nhà cung cấp | Khóa ngoài |
| name | varchar | Tên dịch vụ |  |
| address | varchar | Địa chỉ dịch vụ |  |
| description | varchar | Mô tả dịch vụ |  |
| image | varchar | Đường dẫn ảnh dịch vụ |  |
| start\_cost | int | Giá bắt đầu |  |
| end\_cost | int | Giá kết thúc |  |
| open\_time | datetime | Thời gian bắt đầu |  |
| end\_time | datetime | Thời gian kết thúc |  |
| is\_blocked | boolean | Trạng thái bị chặn |  |
| latitude | float | Vĩ độ |  |
| longitude | float | Kinh độ |  |

Bảng : Bảng dịch vụ

* Bảng type

|  |  |  |  |
| --- | --- | --- | --- |
| Tên | Kiểu | Mô tả | Ràng buộc |
| id | int | Mã loại dịch vụ | Khóa chính |
| name | varchar | Tên loại dịch vụ |  |

Bảng : Bảng type

* Bảng service\_type

|  |  |  |  |
| --- | --- | --- | --- |
| Tên | Kiểu | Mô tả | Ràng buộc |
| id | int | Mã bảng nối | Khóa chính |
| service\_id | int | Mã dịch vụ | Khóa ngoài |
| type\_id | int | Mã loại dịch vụ | Khóa ngoài |

Bảng : Bảng nối service\_type

* Bảng vehicle

|  |  |  |  |
| --- | --- | --- | --- |
| Tên | Kiểu | Mô tả | Ràng buộc |
| id | int | Mã phương tiện | Khóa chính |
| user\_id | int | Mã người dùng lái xe | Khóa ngoài |
| type | int | Loại phương tiện |  |
| plate | varchar | Biển số |  |
| latitude | float | Vĩ độ |  |
| longitude | float | Kinh độ |  |

Bảng : Bảng vehicle

* Bảng notification

|  |  |  |  |
| --- | --- | --- | --- |
| Tên | Kiểu | Mô tả | Ràng buộc |
| id | int | Mã thông báo | Khóa chính |
| user\_id | int | Mã người dùng admin | Khóa ngoài |
| content | varchar | Nội dung thông báo |  |
| subject | varchar | Chủ đề thông báo |  |
| date | datetime | Ngày thông báo |  |

Bảng : Bảng notification

* Bảng history

|  |  |  |  |
| --- | --- | --- | --- |
| Tên | Kiểu | Mô tả | Ràng buộc |
| id | int | Mã lịch sử | Khóa chính |
| user\_id | int | Mã người dùng | Khóa ngoài |
| service\_id | int | Mã dịch vụ | Khóa ngoài |

Bảng : Bảng history

* Bảng tag

|  |  |  |  |
| --- | --- | --- | --- |
| Tên | Kiểu | Mô tả | Ràng buộc |
| id | int | Mã đánh dấu | Khóa chính |
| user\_id | int | Mã người dùng | Khóa ngoài |
| service\_id | int | Mã dịch vụ | Khóa ngoài |

Bảng : Bảng tag

* Bảng activation

|  |  |  |  |
| --- | --- | --- | --- |
| Tên | Kiểu | Mô tả | Ràng buộc |
| id | int | Mã hoạt động | Khóa chính |
| user\_id | int | Mã người dùng | Khóa ngoài |
| accsess\_date | datetime | Ngày hoạt động |  |

Bảng : Bảng activation

## Đánh giá kết quả hệ thống

* Ưu điểm
* Hệ thống đã được phát triển hoàn chỉnh bao gồm cả phía client và server, có thể chạy thử nghiệm (demo).
* Qua cài đặt trong thực tế thì phần mêm hoạt động ổn định, cung cấp được nhiều dịch vụ cho người dùng.
* Phản hồi giữa người dùng và hệ thống thực hiện một cách nhanh chóng, chính xác.
* Hệ thống cho phép các nhà cung cấp dịch vụ bên ngoài có thể tham gia và đăng tải lên các dịch vụ mà họ mong muốn cung cấp
* Nâng cao khả năng an toàn, an ninh thông tin của hệ thống thông qua giải pháp xác thực người dùng.
* Nhược điểm
* Các thiết bị di động rất đa dạng với nhiều kích cỡ màn hình khác nên việc tích hợp giao diện là rất khó khăn.
* Các chức năng tải ảnh lên hệ thống chưa thực sự hoạt động tốt với các ảnh có kích thước dung lượng lớn.
* Chưa có chức năng cài đặt tuy biến giao diện cho người dùng.
* Đối với các nhà cung cấp dịch vụ, khi đăng ký dịch vụ lên hệ thống, cần phải có một đội ngũ nhân viên đi kiểm tra, xác thực dịch vụ. Chức năng này hiện đang nằm ngoài thiết kế của hệ thống, cần được bổ sung trong thời gian tới.

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Trong phần này em sẽ tổng kết lại những gì đồ án đã đạt được, chưa đạt được và hướng phát triển tương lai cho hệ thống.

## Kết luận

Đồ án tốt nghiệp đã thu được các kết quả sau:

1. Tìm hiểu và làm chủ được một số công nghệ quan trọng trong việc lập trình ứng dụng cho thiết bị di động hiện nay:

* Tìm hiểu và nắm được các kiến thức cơ bản về các dịch vụ mà Google cung cấp: Google Maps API, Firebase Cloud Mesaging (FCM).
* Tìm hiểu các dịch vụ xác thực qua số điện thoại Nexmo, dịch vụ lưu trữ file Amazon S3…
* Tìm hiểu và nắm được kiến trúc hệ thống client-server.
* Nắm được những kiến thức cơ bản về lập trình trên nền tảng Android (client side), cũng như với framework Laravel (server side).
* Cũng cố kiến thức về phân tích, thiết kế và phát triển phần mềm, hệ thống.

## Hướng phát triển

Qua đánh giá hệ thống trong *Mục [4.8]* em nhận thấy sản phẩm của mình vẫn còn nhiều hạn chế vì thế trong tương lại em sẽ cố gắng hoàn thiện ứng dụng, khắc phục những nhược điểm còn tồn tại của ứng dụng, những lỗi phát sinh trong quá trình sử dụng ứng dụng

Trong tương lai, để ứng dụng ngày một hoàn thiện và có thể triển khai thực tế một cách rộng rãi, đem đến một sự trải nghiệm tốt nhất cho người dùng thì phải bổ sung và cải thiện một số chức năng sau:

* Cải thiện lại giao diện cho ứng dụng một cách đẹp hơn, có tính thẩm mỹ cao, dễ sử dụng.
* Tối ưu lại về cách quản lý cơ sở dữ liệu.
* Nghiên cứu và phát triển thêm một số tính năng: cho phép thanh toán qua ứng dụng, đăng các tin tức, cho phép admin và lái xe có thể đề xuất thêm thông tin dịch vụ vào hệ thống, các chức đăng review và đánh giá dịch vụ…
* Tối ưu hóa, nâng cao hiệu năng hệ thống để phục vụ được lượng người dùng lớn.

TÀI LIỆU THAM KHẢO

[1]. “Google Maps API”. Google Developers. Website: <https://developers.google.com/maps>

[2]. “Firebase Cloud Messaging”. Google Developers. Retrieved 2016-07-20

[3]. “Nexmo API”. Nexmo Developers. Website: <https://developer.nexmo.com>

[4]. "PHP and MySQL: Working with Google Maps". Ian Rose. Syntaxxx. Retrieved October 13, 2014.

[5]. “Official Google Blog: The world is your JavaScript-enabled oyster”

[6]. “Material Degisn”. Google Developers. Website: <https://material.io>

[7]. “Google Announces Launch of Google Maps for mobile with my Location Technology”. News from Google. Google. November 28, 2007. Retrieved April 25, 2017

[8]. “Phân tích thiết kế hệ thống thông tin”. Nguyễn Văn Ba. NXB Giáo Dục. Retrieved 2004.

[9]. “Phát triển hệ thống hướng đối tượng với UML 2.0 và C++”. Nguyễn Văn Ba

[10]. “Các biểu mẫu Graduation Research”. HaNoi University of Science and Technology.

[11]. “Getting Started with Amazon S3”. AWS. Website: <https://aws.amazon.com/s3>

[12]. “The DCI Architecture: A New Vision of Object-Oriented Programming”. Trygve Reenskaug and James Coplien. Retrieved March 20, 2009.

[13]. “MVP: Model-View-Presenter. The Taligent Programming Model for C++ and Java”. Mike Potel