­Trường Đại học Bách Khoa Hà Nội

Viện Công nghệ Thông Tin và Truyền Thông

Đồ án Tốt nghiệp Đại học

Ứng dụng

Gọi xe Giao hàng

Hoàng Khắc Hiếu

Hà Nội, 06/2018

Trường Đại học Bách Khoa Hà Nội

Viện Công nghệ Thông Tin và Truyền Thông

Đồ án Tốt nghiệp Đại học

Ứng dụng

Gọi xe Giao hàng

|  |  |
| --- | --- |
| Sinh viên thực hiện | Hoàng Khắc Hiếu |
| Người hướng dẫn | TS. Nguyễn Nhất Hải |

­­

Hà Nội, 06/2018

# Lời cam kết

Họ và tên sinh viên: Hoàng Khắc Hiếu

Điện thoại liên lạc: 01636458600 Email: hoangkhachieu96@gmail.com

Lớp: CN – CNTT 01 – K59 Hệ đào tạo: Đại học chính quy

Em – *Hoàng Khắc Hiếu* – cam kết Đồ án Tốt nghiệp (ĐATN) là công trình nghiên cứu của bản thân em dưới sự hướng dẫn của *TS. Nguyễn Nhất Hải*. Các kết quả nêu trong ĐATN là trung thực, là thành quả của riêng em, không sao chép theo bất kỳ công trình nào khác. Tất cả những tham khảo trong ĐATN – bao gồm hình ảnh, bảng biểu, số liệu, và các câu từ trích dẫn – đều được ghi rõ ràng và đầy đủ nguồn gốc trong danh mục tài liệu tham khảo. Em xin hoàn toàn chịu trách nhiệm với dù chỉ một sao chép vi phạm quy chế của nhà trường.

|  |  |
| --- | --- |
|  | *Hà Nội, ngày tháng năm 2018*  Tác giả ĐATN  *Hoàng Khắc Hiếu* |

# Lời cảm ơn

Để luận văn này đạt kết quả tốt đẹp, em đã nhận được sự hỗ trợ, giúp đỡ của nhiều cơ quan, tổ chức, cá nhân. Với tình cảm sâu sắc, chân thành, cho phép em được bày tỏ lòng biết ơn sâu sắc đến tất cả các cá nhân và cơ quan đã tạo điều kiện giúp đỡ trong quá trình học tập và nghiên cứu đề tài.

Trước hết em xin gửi tới các thầy cô khoa Công nghệ Thông tin Trường đại học Bách Khoa Hà Nội lời chào trân trọng, lời chúc sức khỏe và lời cảm ơn sâu sắc. Với sự quan tâm, dạy dỗ, chỉ bảo tận tình chu đáo của thầy cô, đến nay em đã có thể hoàn thành luận văn, đề tài với điều kiện tốt nhất.

Đặc biệt em xin gửi lời cảm ơn chân thành nhất tới thầy giáo – TS. Nguyễn Nhất Hải đã quan tâm giúp đỡ, hướng dẫn em hoàn thành tốt luận văn này trong thời gian qua.

Em xin bày tỏ lòng biết ơn đến lãnh đạo Trường Đại học Bách Khoa Hà Nội, các Khoa Phòng ban chức năng đã trực tiếp và gián tiếp giúp đỡ em trong suốt quá trình học tập và nghiên cứu đề tài.

Với điều kiện thời gian cũng như kinh nghiệm còn hạn chế của một học viên, luận văn này không thể tránh được những thiếu sót. Em rất mong nhận được sự chỉ bảo, đóng góp ý kiến của các thầy cô để em có điều kiện bổ sung, nâng cao ý thức của mình, phục vụ tốt hơn công tác thực tế sau này.

Em xin chân thành cảm ơn!

# Tóm tắt

Hiện nay tại Việt Nam, với sự phát triển không ngừng của nền kinh tế kèm theo đó là sự chuyển đổi từ mô hình thương mại truyền thống sang thương mại điện tử ngày càng mạnh mẽ, nhu cầu vận chuyển hàng hóa của người dân ngày càng lớn.Tuy nhiên, với thách thức về giao thông do vấn đề mật độ dân cư cao, hạ tầng giao thông hạn chế nên không phải ai cũng có thể đầu tư phương tiện hàng vận chuyển riêng cho mình. Nhiều dịch vụ giao hàng, chuyển hàng ra đời nhưng với mô hình thiếu khả năng tiếp cận thị trường nên gặp khó khăn trong việc phân phối, cung cấp dịch vụ của mình trong khi đó phía người dân cũng phải loay hoay tìm kiếm thông tin mỗi khi có nhu cầu hàng vận chuyển, gây ra nhiều bất cập.

Chính vì nhu cầu cấp thiết đó, đồng thời nhận thấy tiềm năng về sự phổ biến của Interner và điện thoại thông minh tại Việt Nam, ĐATN được ra đời với mục tiêu xây dựng một hệ thống thông tin kết nối giữa tài xế cung cấp dịch vụ với người dân có nhu cầu vận chuyển hàng hóa. Hệ thống tận dụng các thế mạnh về công nghệ của thiết bị di động thông minh dựa trên kết quả nghiên cứu nhu cầu thị trường và phân tích các ứng dụng tương tự đã có. Từ đó tập trung xây dựng một thiết kế tổng thể cho hệ thống, tích hợp các công nghệ mới, dịch vụ mới, hữu ích cho người sử dụng.

# Abstract

Mục này khuyến khích sinh viên viết lại mục “Tóm tắt” đồ án tốt nghiệp ở trang trước bằng tiếng Anh. Phần này phải có đầy đủ các nội dung như trong phần tóm tắt bằng tiếng Việt.

Sinh viên không nhất thiết phải trình bày mục này. Nhưng nếu lựa chọn trình bày, sinh viên cần đảm bảo câu từ và ngữ pháp chuẩn tắc, nếu không sẽ có tác dụng ngược, gây phản cảm.

# Mục lục

[Lời cam kết iii](#_Toc512713170)

[Lời cảm ơn iv](#_Toc512713171)

[Tóm tắt v](#_Toc512713172)

[Abstract vi](#_Toc512713173)

[Mục lục vii](#_Toc512713174)

[Danh mục hình vẽ xii](#_Toc512713175)

[Danh mục bảng xiii](#_Toc512713176)

[Danh mục công thức xiv](#_Toc512713177)

[Danh mục các từ viết tắt xv](#_Toc512713178)

[Danh mục thuật ngữ xvi](#_Toc512713179)

[Chương 1 Giới thiệu đề tài 1](#_Toc512713180)

[1.1 Đặt vấn đề 1](#_Toc512713181)

[1.2 Mục tiêu và phạm vi đề tài 1](#_Toc512713182)

[1.3 Định hướng giải pháp 2](#_Toc512713183)

[1.4 Bố cục đồ án 2](#_Toc512713184)

[Chương 2 Khảo sát và phân tích yêu cầu 5](#_Toc512713185)

[2.1 Khảo sát hiện trạng 5](#_Toc512713186)

[2.2 Tổng quan chức năng 6](#_Toc512713187)

[2.2.1 Biểu đồ use case tổng quan 6](#_Toc512713188)

[2.2.2 Biểu đồ use case phân rã XYZ 6](#_Toc512713189)

[2.2.3 Quy trình nghiệp vụ 6](#_Toc512713190)

[2.3 Đặc tả chức năng 7](#_Toc512713191)

[2.3.1 Đặc tả use case A 7](#_Toc512713192)

[2.3.2 Đặc tả use case B 7](#_Toc512713193)

[2.4 Yêu cầu phi chức năng 7](#_Toc512713194)

[Chương 3 Công nghệ sử dụng 8](#_Toc512713195)

[Chương 4 Phát triển và triển khai ứng dụng 9](#_Toc512713196)

[4.1 Thiết kế kiến trúc 9](#_Toc512713197)

[4.1.1 Lựa chọn kiến trúc phần mềm 9](#_Toc512713198)

[4.1.2 Thiết kế tổng quan 9](#_Toc512713199)

[4.1.3 Thiết kế chi tiết gói 10](#_Toc512713200)

[4.2 Thiết kế chi tiết 11](#_Toc512713201)

[4.2.1 Thiết kế giao diện 11](#_Toc512713202)

[4.2.2 Thiết kế lớp 12](#_Toc512713203)

[4.2.3 Thiết kế cơ sở dữ liệu 12](#_Toc512713204)

[4.3 Xây dựng ứng dụng 12](#_Toc512713205)

[4.3.1 Thư viện và công cụ sử dụng 12](#_Toc512713206)

[4.3.2 Kết quả đạt được 13](#_Toc512713207)

[4.3.3 Minh hoạ các chức năng chính 13](#_Toc512713208)

[4.4 Kiểm thử 13](#_Toc512713209)

[4.5 Triển khai 13](#_Toc512713210)

[Chương 5 Các giải pháp và đóng góp nổi bật 14](#_Toc512713211)

[Chương 6 Kết luận và hướng phát triển 16](#_Toc512713212)

[6.1 Kết luận 16](#_Toc512713213)

[6.2 Hướng phát triển 16](#_Toc512713214)

[Tài liệu tham khảo 17](#_Toc512713215)

[Phụ lục A-1](#_Toc512713216)

[A Hướng dẫn viết đồ án tốt nghiệp A-1](#_Toc512713217)

[A.1 Quy định chung A-1](#_Toc512713218)

[A.2 Tạo đề mục A-2](#_Toc512713219)

[A.3 Bảng biểu A-3](#_Toc512713220)

[A.4 Hình vẽ A-4](#_Toc512713221)

[A.5 Tài liệu tham khảo A-5](#_Toc512713222)

[A.6 Công thức toán học A-5](#_Toc512713223)

[A.7 Tham chiếu chéo A-6](#_Toc512713224)

[A.8 Cập nhật mục lục và tham chiếu chéo A-6](#_Toc512713225)

[A.9 In quyển đồ án tốt nghiệp A-7](#_Toc512713226)

[B Đặc tả use case B-7](#_Toc512713227)

[B.1 Đặc tả use case “Thống kê tình hình mượn sách” B-7](#_Toc512713228)

[B.2 Đặc tả use case “Đăng ký làm thẻ mượn” B-7](#_Toc512713229)

[C Công nghệ sử dụng C-8](#_Toc512713230)

[C.1 Công nghệ bảo mật dữ liệu C-8](#_Toc512713231)

[C.2 Công nghệ blockchain C-8](#_Toc512713232)

[D Thiết kế gói D-8](#_Toc512713233)

[D.1 Thiết kế gói cho kiến trúc tổng quan D-8](#_Toc512713234)

[D.2 Thiết kế gói cho chức năng “Trả sách” D-8](#_Toc512713235)

[E Thiết kế lớp E-8](#_Toc512713236)

# Danh mục hình vẽ

[**Hình 1** Ví dụ biểu đồ phụ thuộc gói 10](#_Toc512675503)

[**Hình 2** Ví dụ thiết kế gói 11](#_Toc512675504)

[**Hình 3** Ví dụ hình vẽ A-4](#_Toc512675505)

# Danh mục bảng

[**Bảng 1** Danh sách thư viện và công cụ sử dụng 12](#_Toc512675506)

[**Bảng 2** Ví dụ sử dụng bảng A-3](#_Toc512675507)

# Danh mục công thức

[**Công thức 1** Khai triển Newton A-6](#_Toc512675508)

Lưu ý: Nếu ĐATN có ít hơn ba công thức toán học, sinh viên có thể xóa bỏ mục này.

# Danh mục các từ viết tắt

|  |  |
| --- | --- |
| API | Application Programming Interface  Giao diện lập trình ứng dụng |
| **EUD** | End-User Development  Phát triển ứng dụng người dùng cuối |
| **GWT** | Google Web Toolkit  Công cụ lập trình Javascript bằng Java của Google |
| **HTML** | HyperText Markup Language  Ngôn ngữ đánh dấu siêu văn bản |
| **CNTT** | Công nghệ thông tin |
| **ĐATN** | Đồ án tốt nghiệp |
| **SV** | Sinh viên |

# Danh mục thuật ngữ

|  |  |
| --- | --- |
| Browser | Trình duyệt |
| **Cache memory** | Bộ nhớ đệm |
| **E-commerce** | Thương mại điện tử |
| **Bloatware** | Ứng dụng nhà sản xuất tích hợp vào thiết bị |
| **Interpreter** | Trình thông dịch |
| **Compiler** | Trình biên dịch |

# Giới thiệu đề tài

## Đặt vấn đề

Ngày nay, với sự phổ cập rộng rãi của minh, điện thoại di động đã trở thành một trong những công cụ liên lạc thiết yếu của con người. Ngoài chức năng liên lạc, cùng với sự phát triển của công nghệ, kết nối toàn cầu, điện thoại di động còn được trang bị nhiều ứng dụng giải trí, định vị, mua sắm, thanh toán trực tuyến, Các ứng dụng di động cũng được chú trọng phát triển, ứng dụng nhiều công nghệ mới, góp phần nâng cao hiệu của công việc, tiết kiệm thời gian, tiền bạc cho con người.

Hiện nay tại Việt Nam, với hơn một nửa dân số thường xuyên được tiếp cận với Internet, kèm theo đó là sự chuyển đổi từ mô hình thương mại truyền thống sang thương mại điện tử ngày càng mạnh mẽ, nhu cầu hàng vận chuyển hàng hóa của người dân ngày càng lớn. Tuy nhiên, với thách thức về giao thông do vấn đề mật độ dân cư cao, hạ tầng giao thông hạn chế nên không phải ai cũng có thể đầu tư phương tiện hàng vận chuyển riêng cho mình. Không có khả năng quảng cáo, tiếp cận thị trường khiến cho những tài xế cung cấp dịch vụ hàng vận chuyển gặp khó khăn trong việc cung cấp dịch vụ của mình trong khi đó phía người dân cũng phải loay hoay tìm kiếm thông tin mỗi khi có nhu cầu hàng vận chuyển, gây ra nhiều bất cập.

Chính vì nhu cầu cấp thiết đó, với mục tiêu xây dựng một hệ thống thông tin kết nối giữa tài xế với người dân có nhu cầu hàng tận dụng thế mạnh của thiết bị di động thông minh đồng thời kết hợp nghiên cứu phân tích các ứng dụng đã có trên thị trường. Đồ án tập trung xây dựng thiết kế tổng thể cho ứng dụng, tích hợp các công nghệ , dịch vụ mới, hữu ích cho người sử dụng.

## Mục tiêu và phạm vi đề tài

Sinh viên trước tiên cần trình bày tổng quan các kết quả của các nghiên cứu hiện nay cho bài toán giới thiệu ở phần 1.1 (đối với đề tài nghiên cứu), hoặc về các sản phẩm hiện tại/về nhu cầu của người dùng (đối với đề tài ứng dụng). Tiếp đến, sinh viên tiến hành so sánh và đánh giá tổng quan các sản phẩm/nghiên cứu này.

Dựa trên các phân tích và đánh giá ở trên, sinh viên khái quát lại các hạn chế hiện tại đang gặp phải. Trên cơ sở đó, sinh viên sẽ hướng tới giải quyết vấn đề cụ thể gì, khắc phục hạn chế gì, phát triển phần mềm gì, tạo nên đột phá gì, v.v.

Trong phần này, sinh viên lưu ý chỉ trình bày tổng quan, không đi vào chi tiết của vấn đề hoặc giải pháp. Nội dung chi tiết sẽ được trình bày trong các chương tiếp theo, đặc biệt là trong Chương 5.

## Định hướng giải pháp

Từ việc xác định rõ nhiệm vụ cần giải quyết ở phần 1.2, sinh viên đề xuất định hướng giải pháp của mình theo trình tự sau: (i) Sinh viên trước tiên trình bày sẽ giải quyết vấn đề theo định hướng, phương pháp, thuật toán, kỹ thuật, hay công nghệ nào; Tiếp theo, (ii) sinh viên mô tả ngắn gọn giải pháp của mình là gì (khi đi theo định hướng/phương pháp nêu trên); và sau cùng, (iii) sinh viên trình bày đóng góp chính của đồ án là gì, kết quả đạt được là gì.

Sinh viên lưu ý không giải thích hoặc phân tích chi tiết công nghệ/thuật toán trong phần này. Sinh viên chỉ cần nêu tên định hướng công nghệ/thuật toán, mô tả ngắn gọn trong một đến hai câu và giải thích nhanh lý do lựa chọn.

## Bố cục đồ án

Phần còn lại của báo cáo đồ án tốt nghiệp này được tổ chức như sau.

Chương 2: Phân tích yêu cầu bài toán.

Nội dung chương 2 trình bày kết quả khảo sát các ứng dụng tương tự trên thị trường và phân tích các công nghệ dùng trong hệ thống.

Chương 3: Phân tích thiết kế hệ thống.

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 4:

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..

# ­­Khảo sát và phân tích yêu cầu

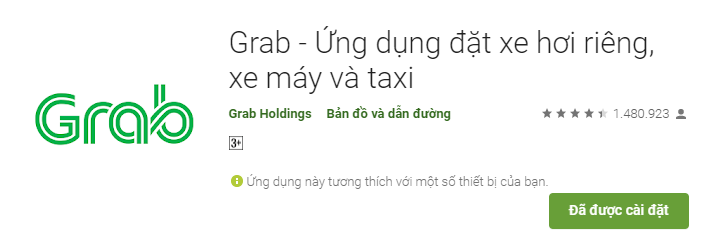
Tiếp nối chương I, ở chương này sẽ tập trung phân tích yêu cầu của bài toán dựa trên việc khảo sát các ứng dụng tương tự có trên thị trường, từ đó đưa ra được danh sách các chức năng cần có cho hệ thống. Phân tích các công nghệ cần thiết để xây dựng nên hệ thống.

## Khảo sát hiện trạng

Theo như bài toán đặt ra, chúng ta cần xây dựng một hệ thống có các chức năng cho phép tìm kiếm, kết nối và trao đổi thông tin giữa khách hàng có nhu cầu giao hàng với tài xế cung cấp các dịch vụ này.

Hiện nay trên thị trường có một số hệ thống, ứng dụng có tính năng tương tự với yêu cầu bài toán, có thể kể đến như sau:

* **Grab giao hàng**: Grab là ứng dụng gọi xe theo yêu cầu của công ty Grab. Ứng dụng cung cấp dịch vụ đa dạng dựa theo nhu cầu của khách hàng. Trong đó có dịch vụ giao hàng theo yêu cầu.

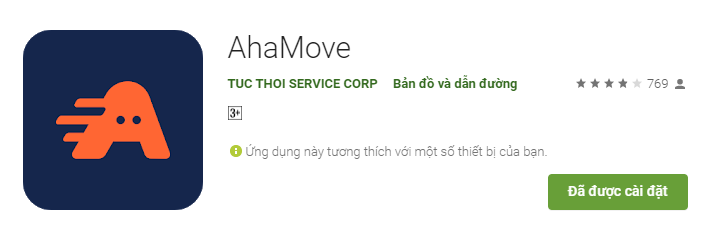


Các chức năng chính của ứng dụng:

1. Đăng nhập.
2. Quản lý thông tin cá nhân.
3. Gọi xe giao hàng theo yêu cầu.
4. Nhận yêu cầu giao hàng.
5. Quản lý lịch sử sử dụng.
6. Ví điện tử.
7. Theo dõi chuyến đi qua bản đồ.

Với lợi thế sử dụng chung nền tảng mạnh mẽ của Grab. Grab giao hàng có số lượng tài xế lớn, phủ rộng khắp với các nhiều chức năng đầy đủ, tiện ích. Tuy nhiên hiện nay Grab giao hàng vẫn còn một số điểm hạn chế như chỉ hỗ trợ giao hàng bằng xe máy, loại hình dịch vụ giao hàng còn hạn chế, giới hạn chỉ một điểm đến trên đơn hàng.

* **AhaMove:** AhaMove là ứng dụng đặt xe cho nhu cầu giao hàng. Kết nối khách hàng với đội ngũ tài xế xe giàu kinh nghiêm. Khi bạn cần chuyển các đơn hàng gấp hoặc cần lên lịch chuyển hàng chủ động, AhaMove sẽ giúp bạn đặt xe một cách hiệu quả.

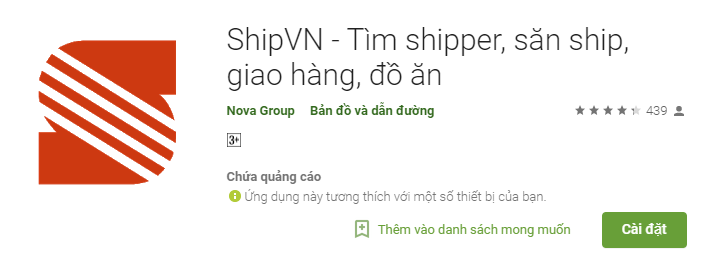


Các chức năng chính của ứng dụng:

1. Đăng nhập.
2. Quản lý thông tin cá nhân.
3. Gọi xe giao hàng theo yêu cầu.
4. Nhận yêu cầu giao hàng.
5. Quản lý lịch sử sử dụng.
6. Ví điện tử.
7. Theo dõi chuyến đi qua bản đồ.

Có thể nói AhaMove là một ứng dụng cung cấp đầy đủ các tính năng của một ứng dụng kết nối giao hàng, cung cấp đa dạng các loại dịch vụ giao hàng cho khách hàng lựa chọn với nhiều tính năng hay. Tuy nhiên ứng dụng vẫn chỉ dừng lại ở mức độ giao hàng bằng xe máy, chưa hỗ trợ liên kết với các tài xế xe tải.

* ShipVN: ShipVN là hệ thống dịch vụ phần mềm thông minh hỗ trợ tương tác giữa người bán hàng (shop) và người giao hàng (shipper) thông qua thiết bị di động (điện thoại) hoặc máy tính bảng.

****

Các chức năng chính của ứng dụng:

1. Đăng nhập.
2. Quản lý thông tin cá nhân.
3. Gọi xe giao hàng theo yêu cầu.
4. Nhận yêu cầu giao hàng.
5. Ví điện tử.

ShipVn là một ứng dụng cung cấp dịch vụ kết nối giữa các shipper với các shop có nhu cầu. Ứng dụng cung cấp đa dạng các loại hình giao hàng, chuyển hàng, liên kết đăng nhận tin yêu cầu giao hàng từ facebook. Tuy nhiên tương tự như hai ứng dụng khác ở trên, ShipVn vẫn còn một vài hạn chế như giao diện khó sử dụng, đối tượng hướng đến còn hạn chế, chưa có hệ thống tính giá đồng bộ.

Từ việc khảo sát các ứng dụng kể trên, với việc kết hợp các ưu điểm của các ứng dụng với nhau và hạn chế các điểm yếu, sinh viên đề xuất ra hệ thống các chức năng cần có cho một ứng dụng gọi xe giao hàng, đồng thời cũng đưa ra một số chức năng mới bổ sung, được chia thanh hai nhóm chức năng là nhóm chức năng cơ bản cần có và nâng cao, cụ thể như bảng:

|  |  |  |
| --- | --- | --- |
|  | Tên chức năng | Mô tả |
| Nhóm chức năng cơ bản | Đăng nhập | Cho phép tài xế, khách hàng đăng nhập vào hệ thống bằng số điện thoại, gmail, facebook,… |
| Quản lý thông tin cá nhân | Cho phép khách hàng và tài xế quản lý các thông tin cá nhân của mình. Đối với tài xế cho phép quản lý loại hình dịch vụ cung cấp và quản lý thông tin về phương tiện. |
| Gọi xe giao hàng theo yêu cầu | Cho phép khách hàng thực hiện yêu cầu gọi xe giao hàng theo nhu cầu. Hệ thống sẽ kết nối khách hàng với tài xế gần nhất. |
| Nhận yêu cầu giao hàng | Khi tài xế sử dụng chức năng này, hệ thống sẽ kết nối tài xế với khách hàng ở xung quanh khu vực hiện tại đang có nhu cầu giao hàng. |
| Theo dõi chuyến đi qua bản đồ | Cho phép khách hàng cũng như tài xế theo dõi trực tiếp quá trình vận chuyển hàng thông qua giao diện bản đồ. Các thông tin về tuyến đường, khoảng cách, thời gian, vị trí hiện tại đều được thể hiện lên bản đồ. |
| Quản lý lịch sử sử dụng | Cho phép tài xế, khách hàng xem lại lịch sử sử dụng dịch vụ của mình. |
| Nhóm chức năng nâng cao | Quản lý ví điện tử | Cho phép tài xế, khách hàng tạo ví điện tử GrapPay để thanh toán cho các dịch vụ mà mình sử dụng |
| Giao hàng tiết kiệm | Cho phép khách hàng đặt nhiều điểm dừng trên mỗi yêu cầu giao hàng nhằm |
| Phân loại xe theo nhu cầu | Cho phép khách hàng chọn loại xe phù hợp với loại hàng hóa cần chuyển. Dựa trên kích thước và khối lượng của hàng hóa, khách hàng có thể chọn xe máy hoặc ô tô để tối ưu hóa chi phí dịch vụ |
| Gọi ý hành trình | thiết lập tuyến đường di chuyển ngắn nhất giữa các đơn hàng hiện có của tài xế. |

Việc xây dựng nên hệ thống các chức năng ở trên giúp ĐATN tập trung xây dựng nên một ứng dụng đầy đủ chức năng đáp ứng được nhu cầu của người dùng.

## Tổng quan chức năng

Hệ thống xây dựng với mục đích kết nối khách hàng có nhu cầu đi lại vận chuyển với tài xế cung cấp dịch vụ , là kênh trung gian giúp người dùng và lái xe có thể tìm kiếm thông tin. Dựa vào đó, ta thấy hệ thống cần có 3 tác nhân chính gồm: khách hàng, tài xế và quản trị viên. Dựa trên mỗi tác nhân mà có từng loại chức năng khác nhau.

### Biểu đồ use case tổng quan

Vai trò của mỗi tác nhân được thể hiện cụ thể:

1. **Khách hàng**: là đối tượng có nhu cầu về dịch vụ hàng vận chuyển. Do đó khách hàng có thể thực hiện gọi xe theo yêu cầu, xem lại lịch sử gọi xe của mình. Sau mỗi chuyến đi khách hàng có thể đánh giá mức độ hài lòng. Nếu đánh giá cao một số tài xế, khách hàng có thể thêm tài xế vào danh sách ưa thích để dễ dàng liên hệ.Bên cạnh đó, khách hàng còn có thể quản lý các thông tin cá nhân, phương thức thanh toán của mình.
2. **Tài xế**: là đối tượng cung cấp dịch vụ hàng vận chuyển. Do đó tài xế có thể nhận các yêu cầu gọi xe. Thông tin về vị trí điểm đi, điểm đến, tuyến đường sẽ được hiện thị lên bản đồ cùng với thông tin khách hàng cũng như chi phí của chuyến đi. Ngoài ra, tài xế có thể quản lý dịch vụ mình cung cấp, quản lý thông tin cá nhân, thông tin phương tiện. Hệ thống cũng cho phép tài xe xem lại lịch sử hoạt động, thực hiện đánh giá chất lượng sau mỗi chuyến đi
3. **Quản trị viên**: Quản trị viên sẽ có thể quản trị thông tin của khách hàng và tài xế. Khi có một số vấn đề cần được thông báo cho người dùng, quản trị viên có thể gửi thông báo đến họ. Ngoài ra quản trị viên có thể quản lý việc sử dụng các loại mã khuyến mãi, giảm giá.

Từ vai trò của các tác nhân ở trên, ta có thể xây dựng nên biểu đồ usecase tổng quát của hệ thống như hình:



### Biểu đồ usecase phân rã

#### Biểu đồ usecase phân rã chức năng Quản lý danh sách tài xế ưa thích

Sau mỗi chuyến đi nếu khách hàng cảm thấy hài lòng với dịch vụ mà tài xế mang lại và có nhu cầu làm việc tiếp trong thời gian tới thì khách hàng có thể đánh dấu tài xế vào mục ưa thích. Với chức năng này, khách hàng có thể xem danh sách tài xế ưa thích, thêm, sửa thông tin, xóa, liên hệ với tài xế ưa thích.



#### Biểu đồ usecase phân rã cho chức năng quản lý lịch sử gọi xe

Sau mỗi chuyến đi hệ thống tự động ghi lại lịch sử. Người dùng có thể sử dụng chức năng này để xem lại thông tin về chuyến đi của mình, thêm lịch sử mới. Ngoài ra chức năng này cũng cung cấp khả năng sửa, xóa lịch sử các chuyến đi.

****

#### Biểu đồ usecase phân rã cho chức năng quản lý phương tiện

Với chức năng quản lý phương tiện, tài xế có thể thực hiện các thao tác xem, sửa, xóa phương tiện thông qua hệ thống.

****

#### Biểu đồ usecase cho chức năng quản trị người dùng

Hệ thống cung cấp cho quản trị viên bộ tính năng quản lý người dùng, cho phép quản trị viên có thể tìm kiếm người dùng, xem thông tin, chặn những người dùng xấu hoặc bỏ chặn.

****

#### Biểu đồ usecase cho chức năng quản lý dịch vụ

Chức năng quản lý dịch vụ cho phép quản trị viên quả lý các dịch vụ vận chuyển, đi lại và các chương trình khuyến mãi mà hệ thống cung cấp. Quản trị viên có thể thêm, sửa thông tin, đóng mở các loại dịch vụ.

****

### Quy trình nghiệp vụ

Với 3 đối tượng sử dụng hệ thông có 3 luồng quy trình nghiệp vụ chính là:

1. Quy trình sử dụng ứng dụng của tài xế.
2. Quy trình sử dụng dịch vụ của khách hàng.
3. Quy trình sử dụng dịch vụ của quản trị viên.

Cụ thể các quy trinh nghiệp vụ như sau:

* 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 tài khoản, đăng ký thông tin phương tiện và đăng nhập với vai trò tài xế vào hệ thống. Tài xế có thể thực hiện nhận các yêu cầu dịch vụ hàng vận chuyển, quản lý lịch sử, quản lý các thông tin cá nhân,…

* Quy trình nghiệp vụ của khách hàng



Sau khi đăng ký và đăng nhập với vai trò khách hàng vào hệ thống. Khách hàng có thể bắt đầu sử dụng các dịch vụ của hệ thống như gọi xe theo yêu cầu, quản lý các thông tin cá nhân, quản lý lịch sử, …

* 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.

## Đặc tả chức năng

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Mã usecase** | UC001 | **Tên usecase** | Đăng ký tài khoản |
| **Tác nhân** | Người dùng | | |
| **Mô tả** | Usecase mô tả quá trình người dùng đăng ký tài khoản trong hệ thống | | |
| **Tiền điều kiện** | Người dùng yêu cầu đăng ký tài khoản | | |
| **Luồng sự kiện chính** | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 1 | Người dùng | Chọn chức năng đăng ký tài khoản bằng số điện thoại | | 2 | Hệ thống | Hiển thị form nhập số điện thoại | | 3 | Người dùng | Nhập số điện thoại và gửi yêu cầu đăng ký | | 4 | Hệ thống | Kiểm tra số điện thoại | | 5 | Hệ thống | Gửi mã xác nhận đến đến số điện thoại và hiển thị form điền mã xác nhận | | 6 | Người dùng | Điền mã xác nhận và gửi yêu cầu | | 7 | Hệ thống | Kiểm tra mã xác nhận | | 8 | Hệ thống | Chuyển vào giao diện chính của ứng dụng | | | |
| **Luồng sự kiện thay thế** | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 1a | Người dùng | Chọn chức năng đăng ký tài khoản bằng gmail / facebook | | 2a.1 | Hệ thống | Hiển thị form chọn tài khoản facebook / gmail | | 2a.2 | Người dùng | Chọn tài khoản facebook/ email. | | 2a.3 | Hệ thống | Hiển thị form điền thông tin cá nhân và số điện thoại | | 4a | Hệ thống | Thông báo lỗi nếu số điện thoại nhập vào là sai | | 7a | Hệ thống | Thông báo lỗi nếu nhập sau mã xác nhận | | | |
| **Hậu điều kiện** | Đăng ký tài khoản thanh công hoặc thất bại | | |

### Đặc tả use case UC002 – Đăng nhập tài khoản

|  |  |  |  |
| --- | --- | --- | --- |
| **Mã usecase** | UC002 | **Tên usecase** | Đăng nhập |
| **Tác nhân** | Người dùng | | |
| **Mô tả** | Usecase mô tả quá trình người dùng đăng nhập tài khoản trong hệ thống | | |
| **Tiền điều kiện** | Người dùng yêu cầu đăng nhập hệ thống | | |
| **Luồng sự kiện chính** | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 1 | Người dùng | Chọn chức năng đăng nhập tài khoản bằng số điện thoại | | 2 | Hệ thống | Hiển thị form nhập số điện thoại | | 3 | Người dùng | Nhập số điện thoại và gửi yêu cầu đăng ký | | 4 | Hệ thống | Kiểm tra số điện thoại | | 5 | Hệ thống | Gửi mã xác nhận đến đến số điện thoại và hiển thị form điền mã xác nhận | | 6 | Người dùng | Điền mã xác nhận và gửi yêu cầu | | 7 | Hệ thống | Kiểm tra mã xác nhận | | 8 | Hệ thống | Chuyển vào giao diện chính của ứng dụng | | | |
| **Luồng sự kiện thay thế** | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 1a | Người dùng | Chọn chức năng đăng nhập tài khoản bằng gmail / facebook | | 2a.1 | Hệ thống | Hiển thị form chọn tài khoản facebook / gmail | | 2a.2 | Người dùng | Chọn tài khoản facebook/ email. | | 2a.3 | Hệ thống | Kiểm tra tài khoản và đăng nhập vào hệ thống | | 4a | Hệ thống | Thông báo lỗi nếu số điện thoại nhập vào là sai | | 7a | Hệ thống | Thông báo lỗi nếu nhập sau mã xác nhận | | | |
| **Hậu điều kiện** | Đăng nhập thành công hoặc thất bại | | |

### Đặc tả use case UC003 – Khách hàng thực hiện gọi xe

|  |  |  |  |
| --- | --- | --- | --- |
| **Mã usecase** | UC003 | **Tên usecase** | Khách hàng thực hiện gọi xe |
| **Tác nhân** | Khách hàng | | |
| **Mô tả** | Usecase mô tả quá khách hang sử dụng dịch vụ gọi xe theo yêu cầu | | |
| **Tiền điều kiện** | Khách hàng đã đăng nhập vào hệ thống | | |
| **Luồng sự kiện chính** | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 1 | Người dùng | Vào màn hình chính của ứng dụng | | 2 | Người dùng | Chọn loại hình vận chuyển và loại phương tiện vận chuyển | | 3 | Người dùng | Nhập vào điểm đi | | 4 | Người dùng | Nhập vào điểm đến và thông tin người nhận tại địa điểm này | | 5 | Người dùng | Gửi yêu cầu gọi xe | | 6 | Hệ thống | Hiển thị form xác nhận yêu cầu gọi xe | | 7 | Người dùng | Xác nhận yêu cầu | | 8 | Hệ thống | Tìm kiếm và kết nối khách hàng với tài xế gần nhất | | 9 | Hệ thống | Khi tài xế nhận yêu cầu gọi xe, hệ thống hiển thị thông tin về vị trí, thông tin cá nhân của tài xế, phương tiện, thời gian ước tính từ vị trí của tài xế đến vị trí của khách hàng. | | | |
| **Luồng sự kiện thay thế** | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 7a | Người dùng | Hủy yêu cầu | | 8a | Hệ thống | Trở về giao diện chính nếu người dùng hủy yêu cầu | | 9a | Hệ thống | Chuyển yêu cầu sang tài xế khác nếu tài xế trước đó không chấp nhận hoặc thông báo cho khách hàng nếu không tìm được tài xế | | | |
| **Hậu điều kiện** | Không | | |

### Đặc tả use case UC004 – Nhận yêu cầu gọi xe

|  |  |  |  |
| --- | --- | --- | --- |
| **Mã usecase** | UC004 | **Tên usecase** | Nhận yêu cầu gọi xe |
| **Tác nhân** | Tài xế | | |
| **Mô tả** | Usecase mô tả quá trình tài xế nhận yêu cầu gọi xe | | |
| **Tiền điều kiện** | Tài xế đã đăng nhập vào hệ thống và kích hoạt chế độ nhận yêu cầu gọi xe, có yêu cầu gọi xe từ khách hàng | | |
| **Luồng sự kiện chính** | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 1 | Hệ thống | Hiển thị thông tin về yêu cầu gọi xe và thời gian đếm ngược nhận yêu cầu gọi xe | | 2 | Tài xế | Xác nhận đồng ý nhận yêu cầu gọi xe | | 3 | Hệ thống | Hiển thị bản đồ thể hiện vị trí hiện tại của tài xế, khách hàng, tuyến đường từ vị trí cảu tài xế đến vị trí khách hàng | | 4 | Tài xế | Xác nhận đã nhận hàng từ khách hàng | | 5 | Hệ thống | Thêm đơn hàng vừa nhận vào danh sách chuyển hàng của tài xế | | | |
| **Luồng sự kiện thay thế** | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 2a | Tài xế | Không nhận yêu cầu gọi xe hoặc để thời gian nhận trôi qua | | 3a | Hệ thống | Tắt chế độ nhận yêu cầu gọi xe của tài xế trong 5 phút | | | |
| **Hậu điều kiện** | Không | | |

### Đặc tả use case UC005 – Kết thúc đơn hàng

|  |  |  |  |
| --- | --- | --- | --- |
| **Mã usecase** | UC005 | **Tên usecase** | Kết thúc đơn hàng |
| **Tác nhân** | Tài xế | | |
| **Mô tả** | Usecase mô tả quá trình tài xế xác nhận thực hiện thành công yêu cầu chuyển hàng | | |
| **Tiền điều kiện** | Tài xế đã đăng nhập vào hệ thống, nhận yêu cầu chuyển hàng từ khách hàng và đã chuyển hàng đến địa điểm cần giao hoặc hủy yêu cầu vì một lý do. | | |
| **Luồng sự kiện chính** | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 1 | Tài xế | Gửi yêu cầu kết thúc đơn hàng cần chuyển | | 2 | Hệ thống | Hiển thị form xác nhận loại kết thúc | | 3 | Tài xế | Xác nhận đã chuyển hàng đến nơi yêu cầu | | 4 | Hệ thống | Hiển thị chuyển thành công đơn hàng và thông báo cho khách hàng | | | |
| **Luồng sự kiện thay thế** | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 4a | Tài xế | Hiển thị hủy đơn hàng thành công và thông báo cho khách hàng | | | |
| **Hậu điều kiện** | Không | | |

### Đặc tả use case UC006 – Đánh giá quá trình sử dụng dịch vụ

|  |  |  |  |
| --- | --- | --- | --- |
| **Mã usecase** | UC006 | **Tên usecase** | Đánh giá quá trình sử dụng dịch vụ |
| **Tác nhân** | Khách hàng | | |
| **Mô tả** | Usecase mô tả quá trình khách hàng đánh giá về chất lượng của dịch vụ vửa sử dụng | | |
| **Tiền điều kiện** | Khách hàng đã đăng nhập vào hệ thống, thực hiện yêu cầu gọi xe chuyển hàng, tài xế đã chấp nhập và chuyển hàng đến địa chỉ hoặc đơn hàng bị hủy. | | |
| **Luồng sự kiện chính** | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 1 | Hệ thống | Thông báo tình trạng của đơn hàng ( thành công hoặc đã bị hủy) | | 2 | Khách hàng | Xác nhận tình trạng của đơn hàng | | 3 | Hệ thống | Hiển thị form đánh giá về đơn hàng | | 4 | Khách hàng | Đánh giá chất lượng của đơn hàng và gửi xác nhận | | 5 | Hệ thống | Hiển thị thông báo xác nhận thành công | | | |
| **Luồng sự kiện thay thế** | Không | | |
| **Hậu điều kiện** | Không | | |

### Đặc tả use case UC007 – Xem thông tin lịch sử sử dụng

|  |  |  |  |
| --- | --- | --- | --- |
| **Mã usecase** | UC007 | **Tên usecase** | Xem thông tin lịch sử sử dụng |
| **Tác nhân** | Khách hàng/tài xế | | |
| **Mô tả** | Usecase mô tả quá trình khách hàng / tài xế xem lại lịch sử sử dụng dịch vụ của mình | | |
| **Tiền điều kiện** | Khách hàng / tài xế đã đăng nhập vào hệ thống | | |
| **Luồng sự kiện chính** | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 1 | Khách hàng / tài xế | Chọn chức năng xem lại lịch sử | | 2 | Hệ thống | Hiển thị giao diện lịch sử các dịch vụ đã sử dụng | | 3 | Khách hàng / tài xế | Tìm kiếm lịch sử sử dụng | | 4 | Hệ thống | Hiển thị danh sách lịch sử sử dụng | | 5 | Khách hàng / tài xế | Chọn 1 lịch sử | | 6 | Hệ thống | Hiển thị thông tin của lịch sử của dịch vụ đó. | | | |
| **Luồng sự kiện thay thế** | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 4a | Hệ thống | Thông báo nếu lịch sử sử dụng trống | | | |
| **Hậu điều kiện** | Không | | |

### Đặc tả use case UC008 – Khóa người dùng

|  |  |  |  |
| --- | --- | --- | --- |
| **Mã use case** | UC008 | **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 | | |

Như vậy chương 2 đã đưa được kết quả khảo sát và phân tích yêu cầu. Xây dựng được hệ thống các chức năng cần có và phân tích các chức năng này. Từ đó tạo nền tảng để xây phân tích dựng ứng dụng ở các chương tiếp theo. Cụ thể là lựa chọn các công nghệ sử dụng ở chương 3.

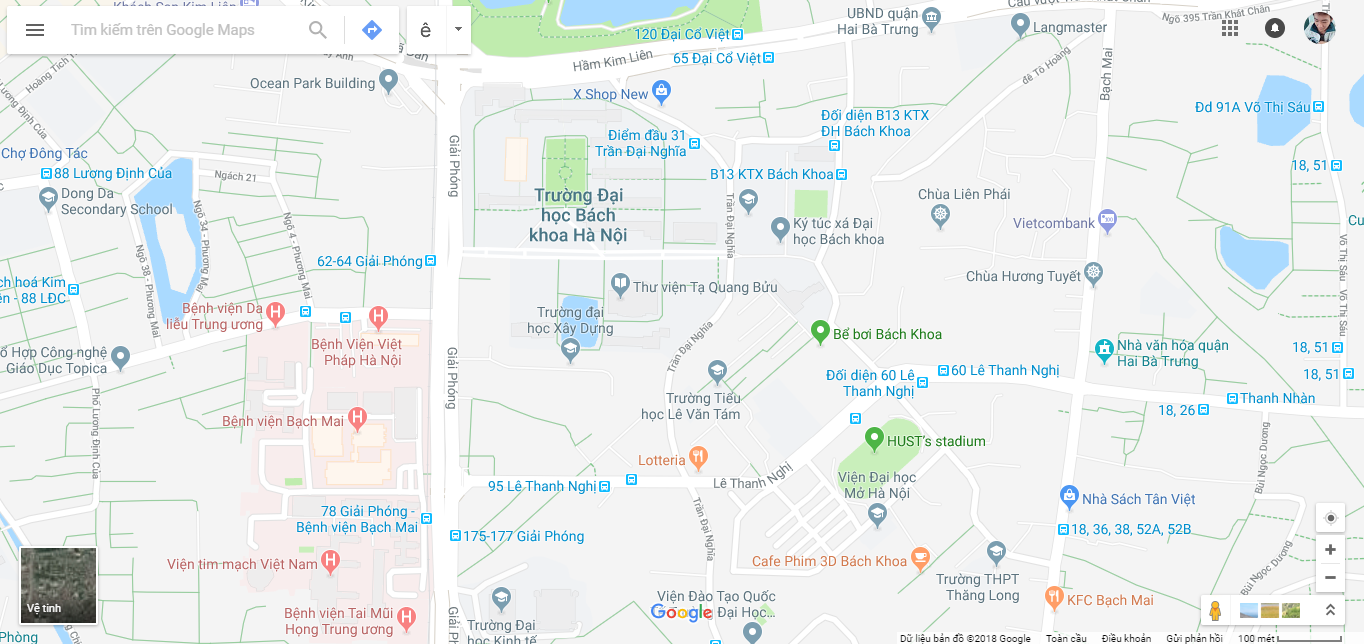
# Công nghệ sử dụng

Như đã trình bày tóm tắt trong phần phân tích một số định hướng giải pháp ở chương 1, và dựa trên việc phân tích yêu cầu ở chương 2, phần này sẽ tập trung vào phân tích và giải thích rõ các giải pháp tích hợp để xây dựng ứng dụng. 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 xử lý dữ liệu theo thời gian thực với Google Firebase Realtime Database
3. Giải pháp xác thực danh tính với Google Firebase Authencation
4. Giải pháp lưu trữ và quản lý tập tin với Google Firebase Cloud Storage).
5. Giải pháp thiết kế giao diện

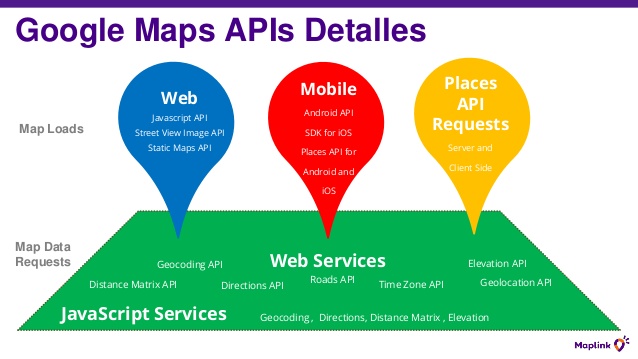
## Giải pháp bản đồ với Google Maps API

Google Map là một dịch vụ ứng dụng và công nghệ bản đồ trực tuyến miễn phí được cung cấp bởi Google. Nó cho phép thấy bản đồ đường sá, đường đi cho xe đạp, cho người đi bộ và xe hơi, và những địa điểm kinh doanh trong khu vực cũng như khắp nơi trên thế giới. Với nhiều tính năng vượt trội, Google Map là dịch vụ bản đồ số được ưa chuộng và sử dụng nhiều nhất trên thế giới.



Tháng 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.

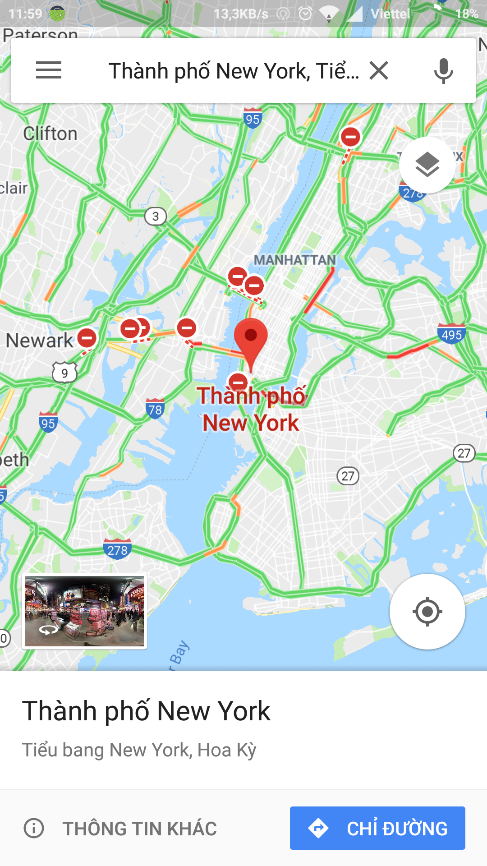
Google Maps API cung cấp rất nhiều các thư viện hỗ trợ mạnh mẽ cho nhà phát triển bao gồm các bộ SDK trên ứng dụng (như Place API, Android API, iOS SDK, Javascript API...) cùng với các Web Service API (như Directions API, Roads API...).



Đến thời điểm hiện tại, Google Map API đã được nâng cấp lên phiên bản v3, không chỉ hỗ trợ cho các máy để bàn truyền thống mà cho cả các thiết bị di động; các ứng dụng nhanh hơn và nhiều hơn .

### Tích hợp Google Map bằng Android API

Với mục tiêu ban đầu là xây dựng hệ thống phía client trên nền tảng Android nên ứng dụng được tích hợp Android API để xử lý hiển thị bản đồ. Việc tích hợp và sử dụng Android API khá dễ dàng. Google đã cung cấp các tài liệu hướng dẫn chi tiết để lập trình viên có thể tiếp cận và phát triển ứng dụng bản đồ của mình.



*Với Google Maps Android API, bạn có thể thêm bản đồ dựa trên dữ liệu Google Maps vào ứng dụng của bạn. API tự động xử lý quyền truy cập vào các máy chủ của Google Maps, tải dữ liệu, hiển thị bản đồ và phản ứng lại các thao tác trên bản đồ. Bạn cũng có thể sử dụng các hàm gọi API để thêm các điểm đánh dấu, đa giác và lớp phủ vào một bản đồ cơ bản và để thay đổi chế độ xem của người dùng về một khu vực bản đồ cụ thể.*

### Xác định vị trí với Google Play services location API

Google Play services location API là dịch vụ cho phép chúng ta lấy được vị trí hiện tại của thiết bị. Hiện nay có 3 nguồn cung cấp để lấy vị trí bao gồm:

1. GPS: nguồn này xác định vị trí dựa trên vệ tinh. Tùy thuộc vào điều kiện trong nhà hay ngoài trời mà nguồn này cần một khoảng thời gian để có thể trả về vị trí hiện tại với độ chính xác phụ thuộc.
2. NETWORK PROVIDER : nguồn này xác định vị trí dựa trên cột thu phát sóng của mạng di động và các điểm truy cập WIFI. Kết quả vị trí thu được nhờ tìm kiếm trong mạng.
3. PASSIVE PROVIDER: nguồn này trả về vị trí sinh ra bởi các nguồn khác. Ví dụ bạn nhận được vị trí cập nhật một cách thụ động khi mà các ứng dụng khác hoặc dịch vụ khác yêu cầu mà ko phải bạn thực sự yêu cầu điều này.

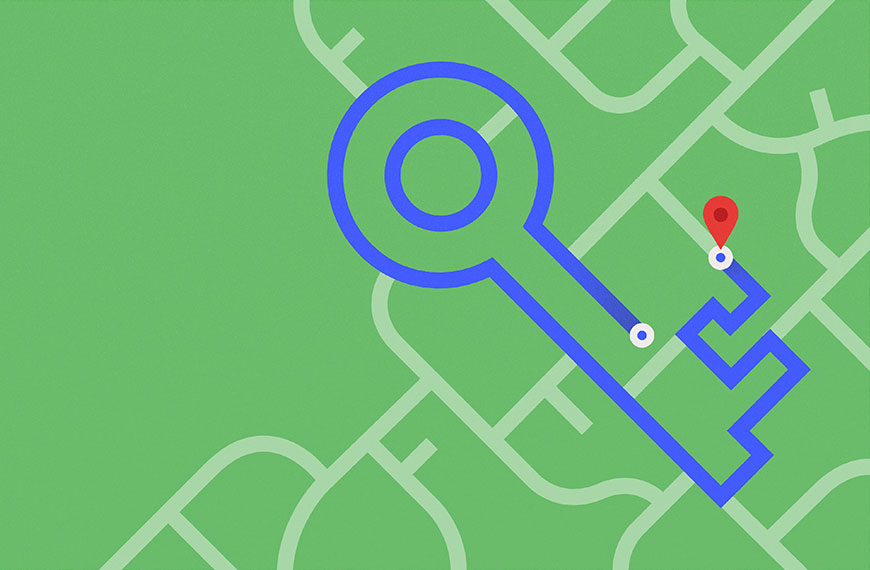
Google Play services location API là cách mới mà Google giới thiệu để có thể lấy được vị trí hiện tại của thiết bị. API này dùng "Fused Location Provider" - tự động chọn nguồn cung cấp vị trí để sử dụng dựa trên độ chính xác và mức tiêu thụ pin. API này cũng được Google khuyến khích sử dụng thay thế cho Android Location API cũ.

Việc tích hợp Google Play services location API vào ứng dụng khá đơn giản. Lập trình viên chỉ cần tích hợp gói Google Play services vào ứng dụng và thực hiện gọi các hàm API để lấy dữ liệu về vị trí.

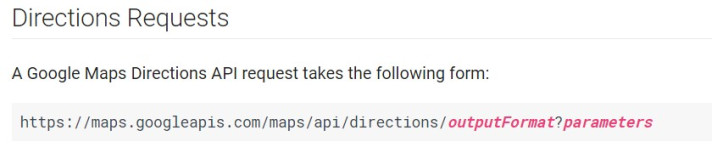
### Xác định đường đi với Google Directions API

Việc xác định đường đi từ đó xác định được khoảng cách, thời gian di chuyển trung bình là một trong những tính năng quan trọng của ứng dụng. Chi phí phát sinh cơ bản của một hợp đồng hàng vận chuyển trên ứng dụng là kết quả tính toán từ chức năng này.

Để hiển thị đường đi từ một điểm đến một điểm khác, hệ thống lựa chọn sử dụng dịch vụ Directions API. Dịch vụ này giúp 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.



Việc tích hợp vụ **Directions API** được thông qua bằng giao diện HTTP với cấu trúc **Request – Response**. Trong đó **Request** với các yêu cầu được xây dựng dưới dạng chuỗi URL, sử dụng chuỗi văn bản hoặc vĩ độ / kinh độ để xác định vị trí, cùng với khóa API của bạn.



Ở đây chúng ta chú ý đến outputFormat sẽ là kiểu dữ liệu mà ta muốn nhận về từ respone. Hiện tại Google hỗ trợ 2 định dạng là JSON và XML.

Ở phần parameters ta cần phải xác định 3 yếu tố bắt buộc sau đây:

1. Origin: Là địa điểm mà ta chọn là địa điểm xuất phát, bao gồm các định dạng được chấp nhận như sau: Địa chỉ, giá trị vĩ độ / kinh độ văn bản, hoặc ID nơi bạn muốn tính chỉ đường.
2. Destination: mục này là địa điểm mà ta chọn làm địa điểm đích đến, và cấu trúc cũng tương tự như Origin.
3. Key: Khóa API cho ứng dụng . Việc sử dụng khóa này giúp Google xác định được mức hạn ngạch Request mỗi ngày của ứng dụng. Chúng ta có thể tăng mức hạn ngạch này bằng cách sử dụng các gói cước của Google.

Bên cạnh đó, ta có thể bổ sung thêm 1 số parameters tùy chọn khác như sau:

1. Mode: chọn loại phương tiện mà ta muốn di chuyểnAlternatives: có thể trả về nhiều hơn 1 tuyến đường khả dụng.
2. Avoid: tuyến đường trả về cần phải tránh 1 số vấn đề như tránh đường cao tốc, tránh đi qua các trạm thu phí cầu đường.
3. … (tham khảo thêm trên trang document của Directions API).

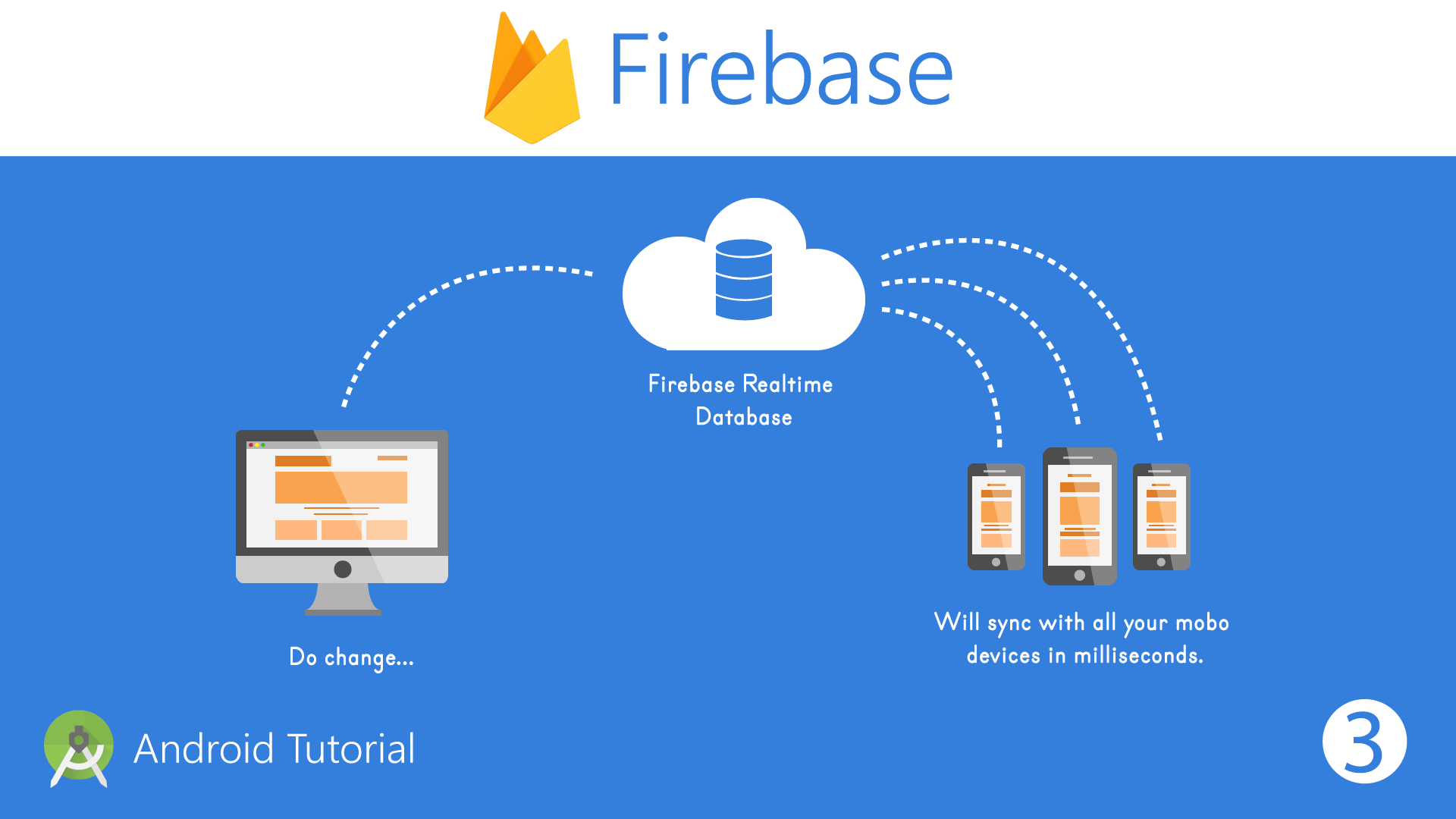
Sau khi gửi 1 Request lên, một chuỗi Response về chỉ đường được trả lại dưới định dạng được yêu cầu trong đường dẫn của URL Request (dạng JSON hoặc XML). Cụ thể chuỗi Response này được chia thành các step nhỏ, mỗi step là 1 đoạn trong tuyến đường di chuyển giữa 2 điểm.

|  |
| --- |
| "status": "OK",  "geocoded\_waypoints" : [  { "geocoder\_status" : "OK",  "place\_id" : "ChIJ7cv00DwsDogRAMDACa2m4K8",  "types" : [ "locality", "political" ]},  { "geocoder\_status" : "OK",  "place\_id" : "ChIJ69Pk6jdlyIcRDqM1KDY3Fpg",  "types" : [ "locality", "political" ]},  { "geocoder\_status" : "OK",  "place\_id" : "ChIJgdL4flSKrYcRnTpP0XQSojM",  "types" : [ "locality", "political" ]},  { "geocoder\_status" : "OK",  "place\_id" : "ChIJE9on3F3HwoAR9AhGJW\_fL-I",  "types" : [ "locality", "political" ]}],  ... |

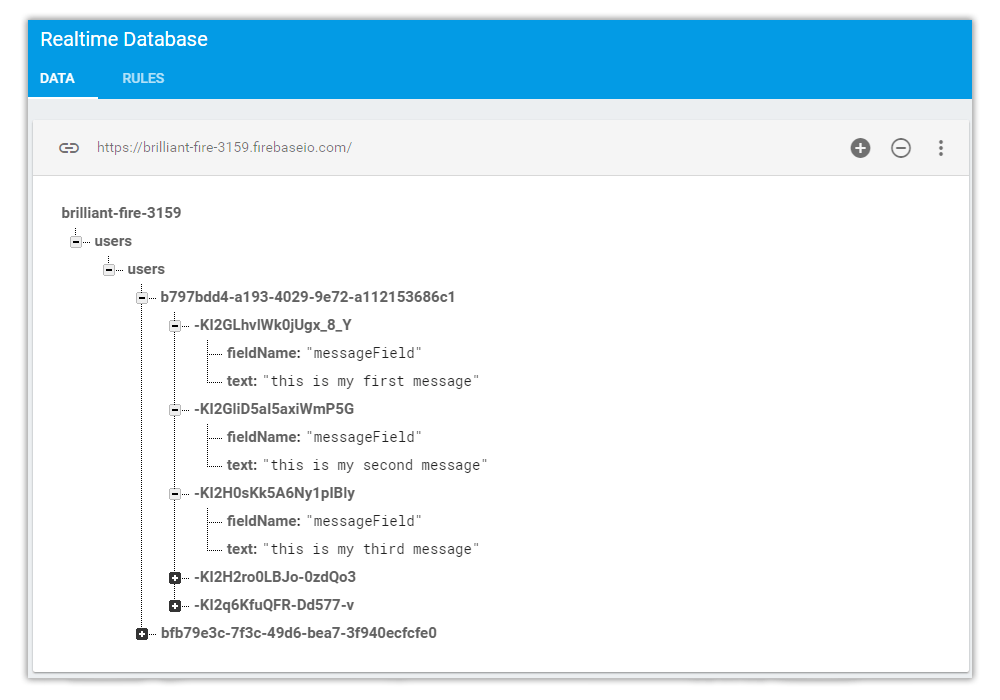
Sau khi nhận được chuỗi Respone, lập trinh viên chỉ cần xử lý chuỗi này dựa trên yêu cầu của bài toán.

## Xử lý dữ liệu theo thời gian thực với Firebase Realtime Database

Firebase Realtime Database là một Cloud hosted database hỗ trợ đa nền tảng: Android, IOS, Web, C++, Unity. Tất cả dữ liệu được lưu trữ ở định dạng JSON. Khi có một sự thay đổi dữ liệu nào thì có sự phản hồi ngay lập tức, hiển thị đồng bồ trên các tất cả các nền tảng và các thiết bị liên kết với nó.



Khác với các kiểu dữ liệu thường thấy như SQL database được tổ chức theo dạng bảng, Firebase Realtime được tổ chức theo dạng cây (trees), với mỗi nhánh giống như một container, chỉ chứa hoặc là dữ liệu ứng với nhánh đó (tức là value tương ứng với key), hoặc một tập hợp các nhánh con cũng được tổ chức theo một cách tương tự.



Mỗi nhánh được đại diện bởi một key và value tương ứng. Các key luôn có dạng là String. Còn các value thì có thể là một đối tượng kiểu String, một đối tượng kiểu Int, hay là một Object, được thể hiện dưới dạng một nhánh con. Và nhánh con đó được coi như là một value ứng với key đó. Do được tổ chức theo hình thức này, dữ liệu được gửi về các thiết bị có định dạng là JSON.

Các key cùng tầng phải đảm bảo được tính duy nhất. Các key khác tầng có thể được phép giống nhau. Để đảm bảo các key cùng tầng là duy nhất, Firebase cung cấp cho chúng ta các hàm để yêu cầu Firebase server tạo ngẫu nhiên các giá trị key duy nhất, ví dụ như; push() trong Android và Web, hay childByAutoId() trên iOS.

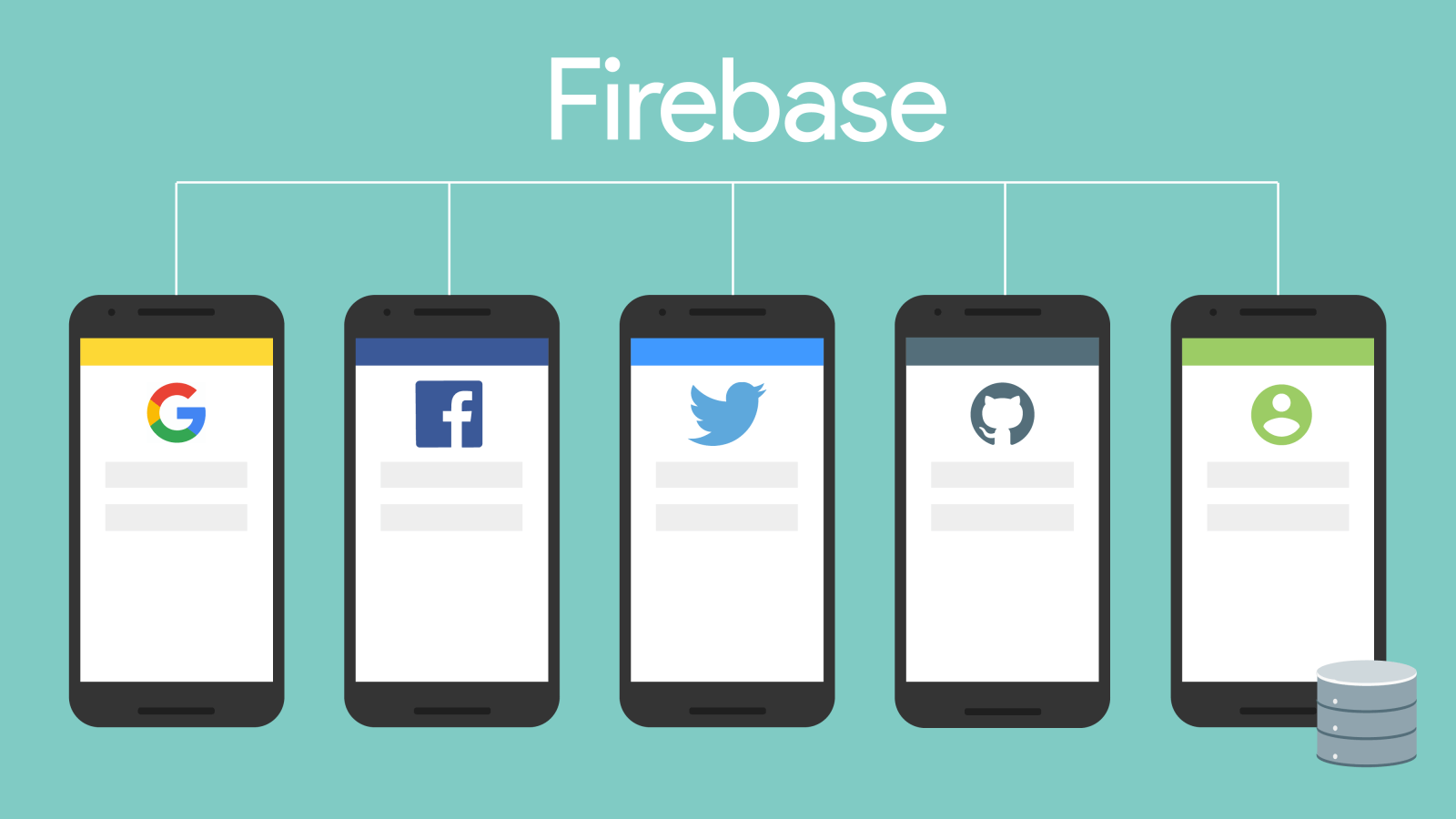
Khi bạn thêm một bộ dữ liệu mới, tức bao gồm 1 bộ key – value mới, thì sẽ xảy ra hai trường hợp sau:

1. Key chưa tồn tại: Firebase Database sẽ tạo một Object mới tại vị trí mà bạn muốn lưu, với key – value là các giá trị bạn đã định nghĩa. Event này được gọi là “add”, tức là thêm.
2. Key đã tồn tại: Firebase Database sẽ sửa value tại vị trí của key đó theo value mới. Event này được gọi là “chànge”.

Việc tích hợp Firebase Realtime Database vào ứng dụng khá đơn giản. Lập trình viên chỉ cần tạo Project trên trang chủ Firebase và lựa chọn liên kết với Firebase Realtime Database trên IDE Android studio bằng tài khoản và Project đã tạo.

## Xác thực người dùng với Firebase Authencations

Firebase Authencations là giải pháp giúp xây dựng các hệ thống xác thực an toàn một cách dễ dàng, đồng thời nâng cao trải nghiệm đăng nhập và cập nhật cho người dùng. Nó cung cấp khá nhiều giải pháp đầu cuối như: hỗ trợ tài khoản email, mật khẩu, xác thực số điện thoại, đăng nhập qua tài khoản Google, Twitter, Facebook và GitHub.



Firebase Authentication hoạt động như thế nào?

1. Để xác thực người dùng đăng nhập vào ứng dụng của bạn, trước tiên bạn nhận được thông tin xác thực từ người dùng. Những thông tin đăng nhập này có thể là địa chỉ email và mật khẩu của người dùng, hoặc mã OAuth từ một nhà cung cấp nhận dạng liên đới (Facebook, Twitter,…)
2. Sau đó, bạn chuyển những chứng chỉ này tới SDK Authentication Firebase. Các dịch vụ phụ trợ của Firebase Authentication sẽ xác minh các thông tin và trả lời cho khách hàng.
3. Sau khi đăng nhập thành công, ứng dụng của bạn có thể truy cập vào thông tin tiểu sử cơ bản của người dùng và bạn có thể kiểm soát quyền truy cập của người dùng vào dữ liệu được lưu trữ trong ứng dụng của mình.

Lập trình viên có thể tích hợp Firebase Aunthencation vào ứng dụng bằng cách sử dụng FirebaseUI – phương thức xác thực qua bộ thư viện kéo thả bằng trợ lý Assistant trên Android Studio hoặc sử dụng Firebase Authentication SDK theo cách thủ công.

## Lưu trữ dữ liệu với Firebase Storage

Firebase Storage là dịch vụ được xây dựng cho mục đích lưu trữ và quản lý các nội dung mà người dùng ứng dụng tạo ra như ảnh, videos hay dữ liệu dạng file.

Firebase Storage cung cấp các API cho việc uploads và download các file từ ứng dụng của bạn một cách bảo mật và bạn không cần quan tâm đến chất lượng đường truyền mạng.

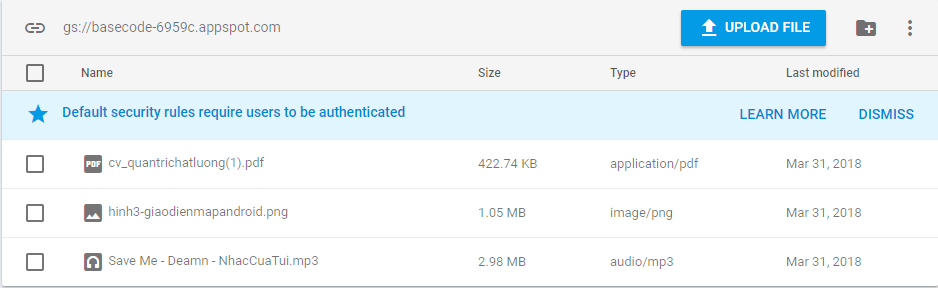
Với việc được xây dựng trên nền tảng Google Cloud Platform nên Firebase Storage có nhiều lợi thế, cụ thể:

1. Robust operations (Hoạt động mạnh mẽ): Với Firebase Storage bạn có thể thực hiện tải dữ liệu lên và xuống bất kể chất lượng của đường truyền mạng. Hỗ trợ tiếp tục tải lên / tải xuống tại nơi người dùng dừng lại, giúp tiết kiệm thời gian và băng thông.
2. Strong security (Tính bảo mật mạnh): Được tích hợp Firebase Authentication cho việc bảo mật nên bạn có thể dễ dàng quản lý quyền truy cập vào các files cho từng đối tượng người dùng.
3. High scalability (Khả năng mở rộng): Firebase Storage được xây dựng trên nền tảng Google Cloud Platform nên khả năng mở rộng có thể lên đến hàng Petabyte dữ liệu.

*Firebase Storage hoạt động như thế nào?*

1. Lập trình viên sử dụng Firebase SDK cho Cloud để tải lên và tải các tệp trực tiếp từ thiết bị của người dùng. Nếu chất lượng đường truyền không ổn định, người dùng có thể thử lại tại nơi có đường truyền ổn định hơn.
2. Cloud Storage lưu trữ tệp mà người dùng tải lên vào Google Cloud Storage, giúp các tệp này có thể truy cập qua cả Firebase và Google Cloud. Điều này cho phép bạn linh hoạt trong việc tải lên hoặc xuống tệp tin từ người dùng đồng thời thực hiện xử lý các công việc phía sever như lọc hình ảnh hoặc chuyển mã video thông qua Google Cloud Platform.
3. Firebase Storage sử dụng các Reference để chỉ đến một bucket hoặc một file, bạn có thể làm bất cứ thứ gì với bucket, files thông qua Reference đó như upload, download hay thay đổi metadata hoặc xóa file. Reference mang tính chất lightweight vì vậy bạn có thể sử dụng nó bao nhiêu tùy thích hoặc có thể sử dụng lại một Reference cho nhiều hành động khác nhau.

Việc tích hợp Firebase Storage vào ứng dụng khá đơn giản thông qua Firebase Assistant trên Android Studio IDE.



## Giải pháp thiết kế giao diện - Material Design

Material Design là một ngôn ngữ thiết kế được phát triển vào năm 2014 bởi Google. Mở rộng dựa trên mô típ "thẻ" có mặt trên Google Now, Material Design đem đến phong cách tự do hơn với các cách bố trí dạng lưới, các phản hồi hoạt họa chuyển động, kéo giãn, và các hiệu ứng chiều sâu như ánh sáng và đổ bóng.

Google giới thiệu Material Design vào ngày 25 tháng 6 năm 2014, tại hội nghị Google I/O 2014.



Đặc điểm của Material Design

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

# Phát triển và triển khai ứng dụng

## Thiết kế kiến trúc

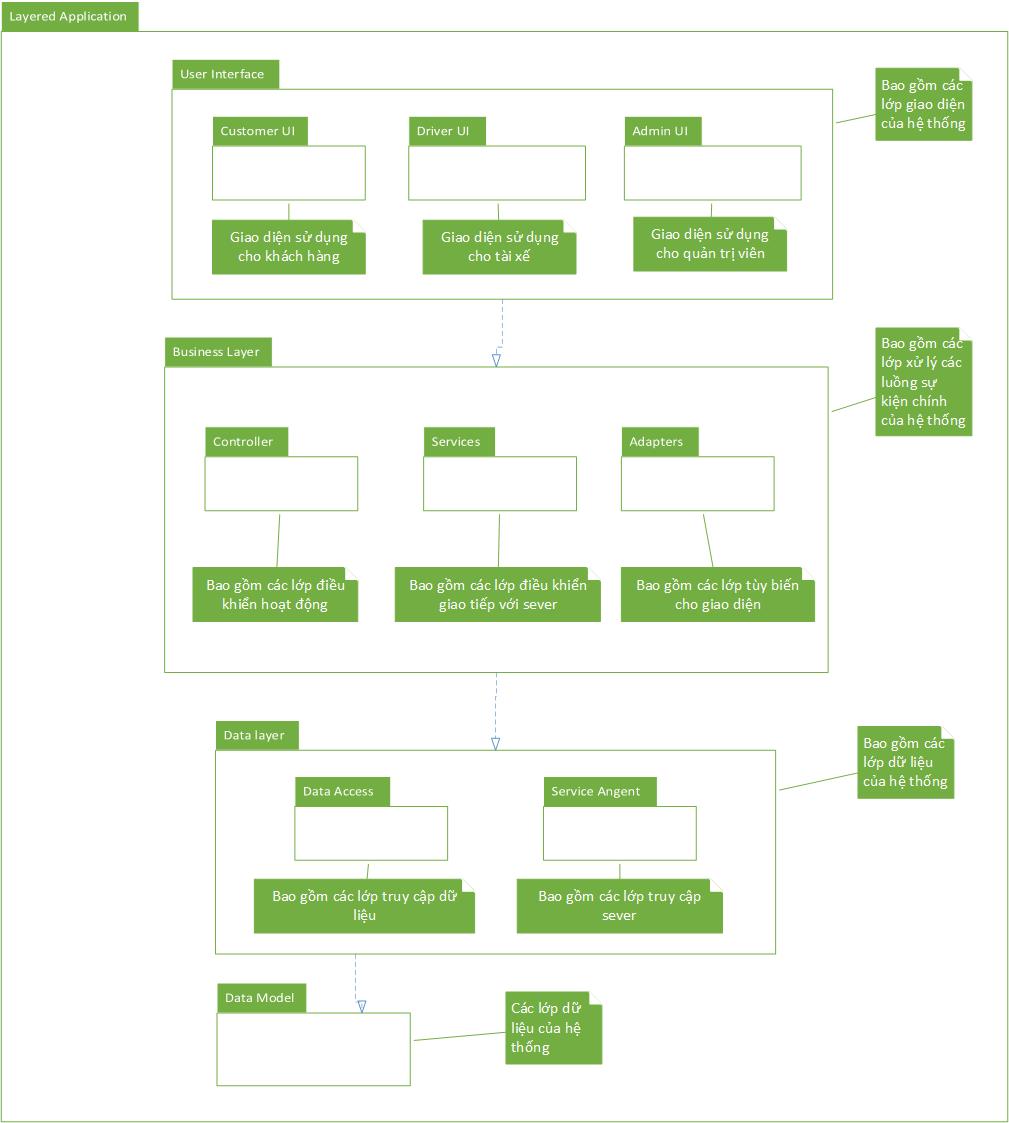
### Lựa chọn kiến trúc phần mềm

Mục này có độ dài từ một đến ba trang. Sinh viên cần lựa chọn kiến trúc phần mềm cho ứng dụng của mình như: kiến trúc ba lớp MVC, MVP, SOA, Microservice, v.v. rồi giải thích sơ bộ về kiến trúc đó (không giải thích chi tiết/dài dòng).

Sử dụng kiến trúc phần mềm đã chọn ở trên, sinh viên mô tả kiến trúc cụ thể cho ứng dụng của mình. Gợi ý: sinh viên áp dụng lý thuyết chung vào hệ thống/sản phẩm của mình như thế nào, có thay đổi, bổ sung hoặc cải tiến gì không. Ví dụ, thành phần M trong kiến trúc lý thuyết MVC sẽ là *những thành phần cụ thể nào* (ví dụ: là interface I + class C1 + class C2, v.v.) trong kiến trúc phần mềm của sinh viên.

### Thiết kế tổng quan

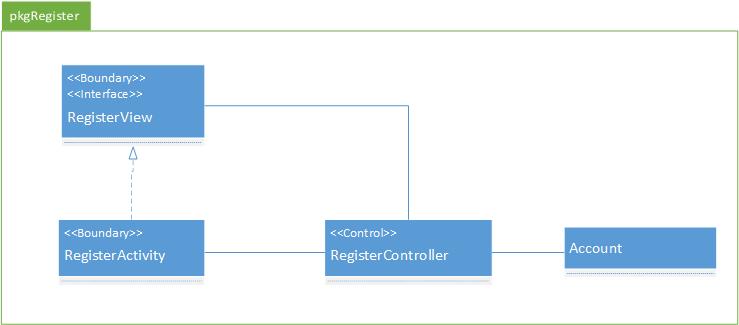
Dựa trên kiến trúc phần mềm, hệ thống được xây dựng thành 3 tầng xử lý chính, mỗi tâng có chức năng cụ thể như sau:



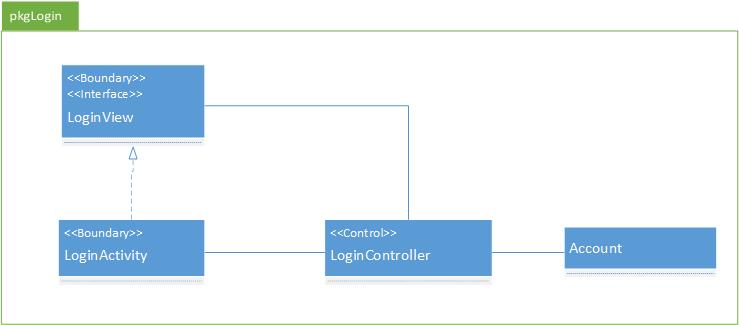
### Thiết kế chi tiết gói

Sau đây em xin trình bày biểu đồ lớp phân tích gộp 1 số use case liên quan tới nhau của một vài chức năng của ứng dụng:

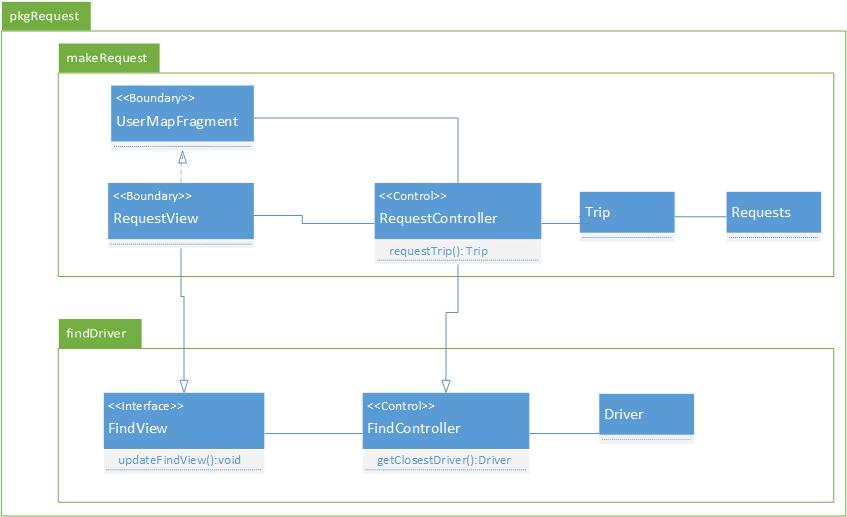
* Thiết kế chi tiết gói chức năng Đăng ký



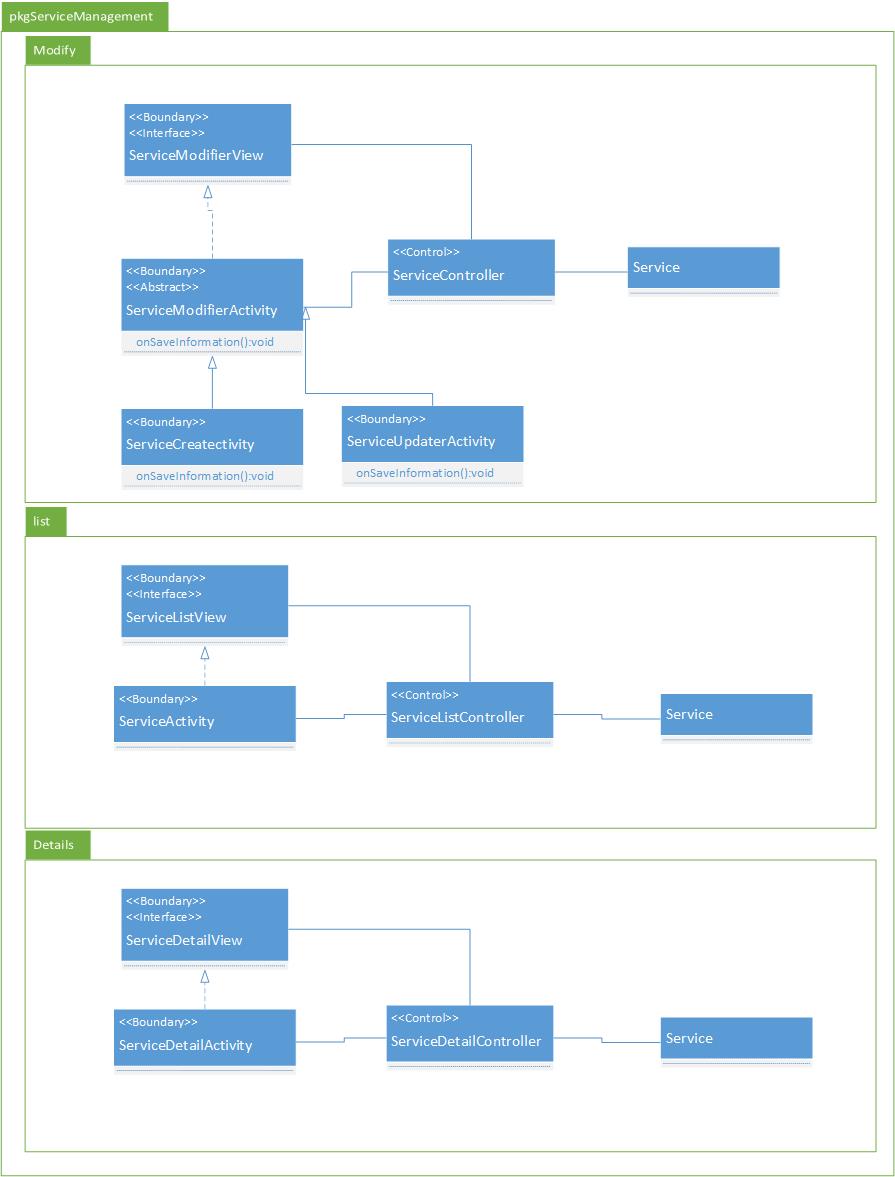
* Thiết kế chi tiết gói chức năng Đăng nhập



* Thiết kế chi tiết gói chức năng Gọi xe



* Thiết kế chi tiết gói chức năng Quản lý dịch vụ

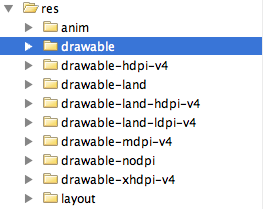


## Thiết kế chi tiết

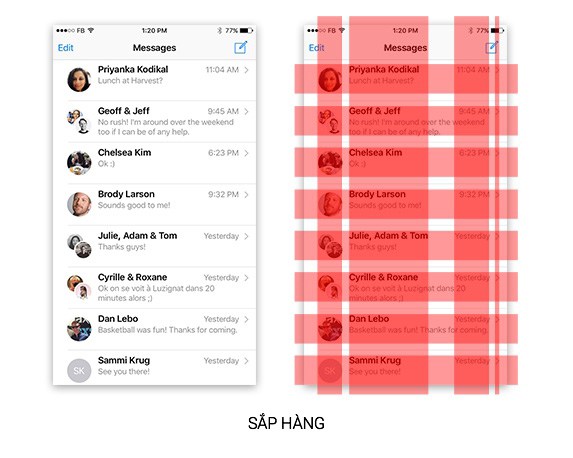
### Thiết kế giao diện

Việc thiết kế giao diện cho ứng dụng cần đảm bảo tính đơn giản, dễ sử dụng cho người dùng. Do đó cần thống nhất các quy chuẩn thiết kế cho ứng dụng. Để đảm bảo yêu cầu này sinh viên đưa ra các quy tắc thiết kế giao diện sau :

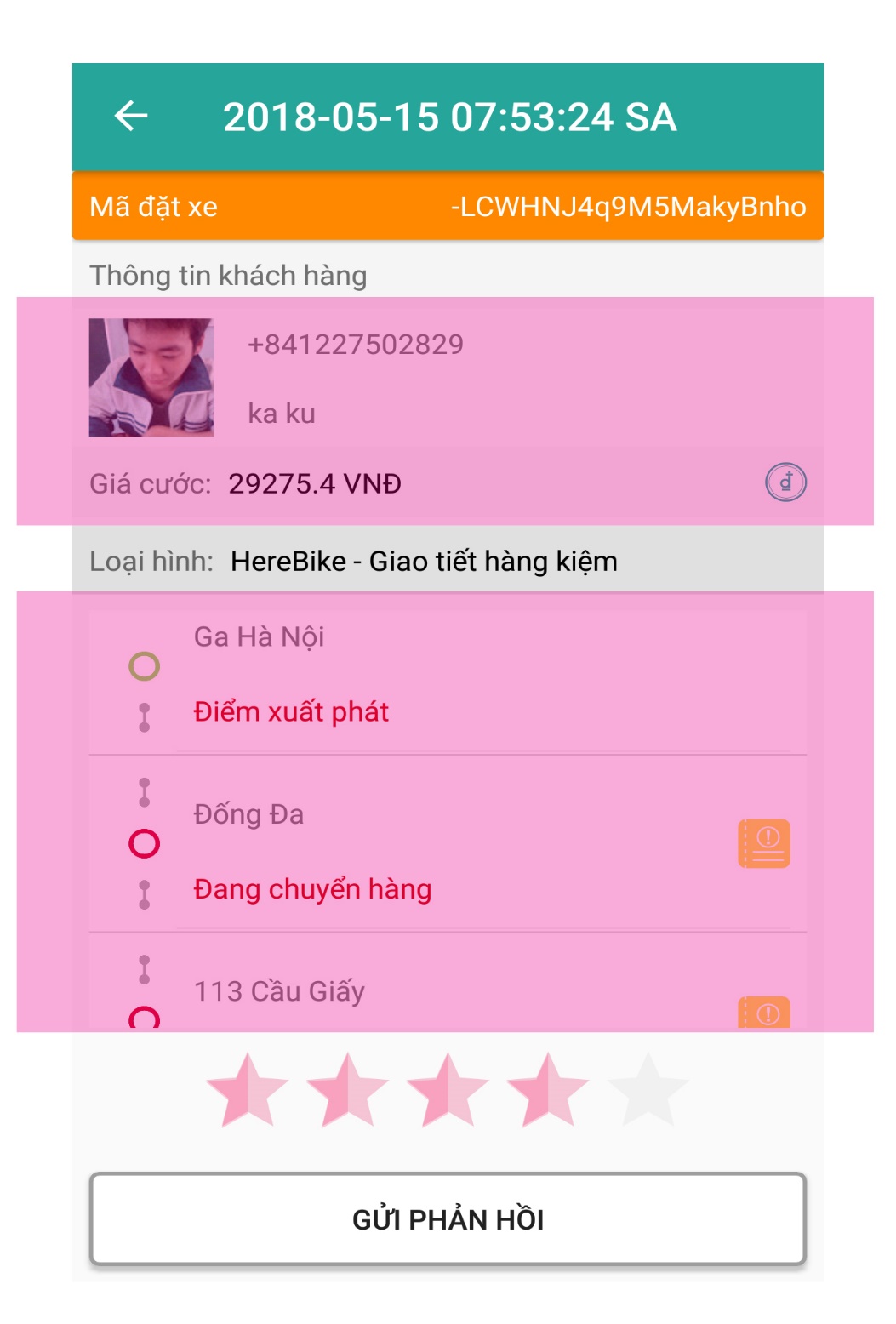
1. Quy định về màn hình: Hệ thống xây dựng trên hệ điều hành Android. Hiện nay trên thị trường các hàng sản xuất điện thoại Android rất nhiều, với nhiều loại điện thoại khác nhau, kích cỡ màn hình khác nhau. Do đó ứng dụng cần thiết kế phù hợp với các loại thiết bị. Không để bị mất thông tin, gây khó khăn cho người dùng. Các file hình ảnh, thiết kế cần được tùy chỉnh theo từng loại màn hình theo quy chuẩn thiết kế của Android



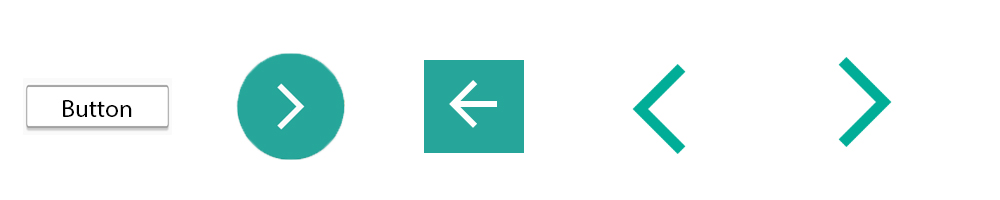
1. Quy định về sắp xếp thông tin: Bất kỳ thông tin được sắp xếp phải thẳng hàng với đối tượng đầu xuyên suốt trong các nguyên tắc thiết kế.Dựa trên nguyên tắc trái sang phải dưới giống như thói quen đọc sách.



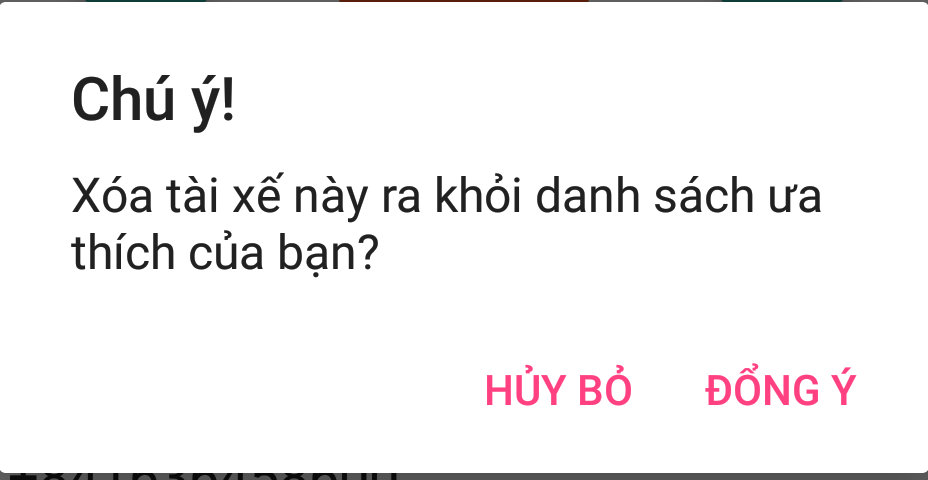
1. Tổ chức thông tin theo nguyên tắc quan hệ: Tổ chức thông tin theo các nguyên tắc quan hệ, gần gũi với các thông tin khác có liên quan, nhóm chúng lại với nhau. Nếu nhiều mục tồn tại gần nhau có quan hệ chặt chẽ nó sẽ trở thành một nhóm trực quan, chứ không phải là các yếu tố bị cô lập, đơn lẻ. Điều này sẽ giúp làm giảm sự nhầm lẫn và cung cấp cho người đọc một cấu trúc rõ ràng, dễ tiếp cận.



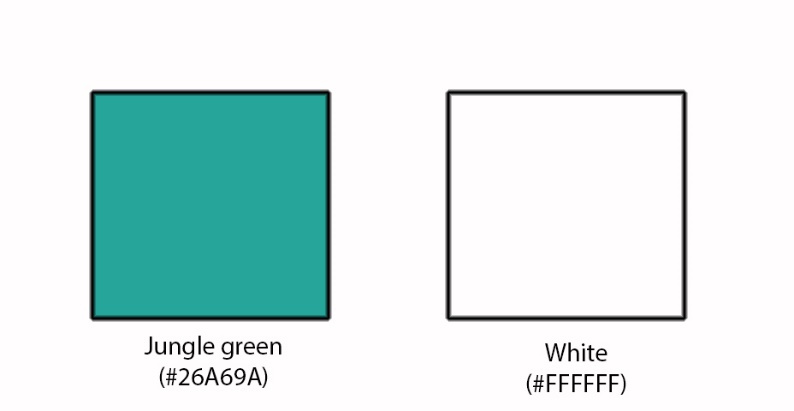
1. Thiết kế nút, điều khiển: Các nút xác nhận thiết kế đơn giản hình chữ nhật, viền xám, nền trắng, các phím điều khiển chức năng hình tròn, nền xanh với icon trắng, các phim điều hướng tiếp tục, quay lại được thiết kế như hình:



1. Hiển thị các thông điệp: Các thông điệp của ứng dụng hiển thị chính giữa màn hình. Với tiêu đề, nội dung và các phim xác nhận (nếu có). Nội dung của thông điệp ngắn gọn, dễ hiểu, như hình:



1. Nguyên tắc phối màu: ứng dụng sử dụng 2 màu chính là Jungle Green (#26A69A) và màu trắng. Thiết kế cần phối hợp hai màu này một cách hợp lý để làm nổi bật các thông tin nhưng vẫn đảm bảo tính đơn giản cho giao diện hệ thống.



### Thiết kế cơ sở dữ liệu

#### Yêu cầu của bài toán

Một người dùng khi sử dụng hệ thống có thể đăng ký làm một trong hai loại người dùng : Lái xe, khách hàng. Các thông tin của người dùng bao gồm họ tên, số điện thoại, địa chỉ email, hình ảnh đại diện.

Lái xe sở hữu phương tiện giao thông với các thông tin: biển số, và loại phương tiện (xe máy, ô tô), hãng sản xuất. Tài xế cũng cần cung cấp cho hệ thống các thông tin về giấy phép lái xe, giấy tờ xe.

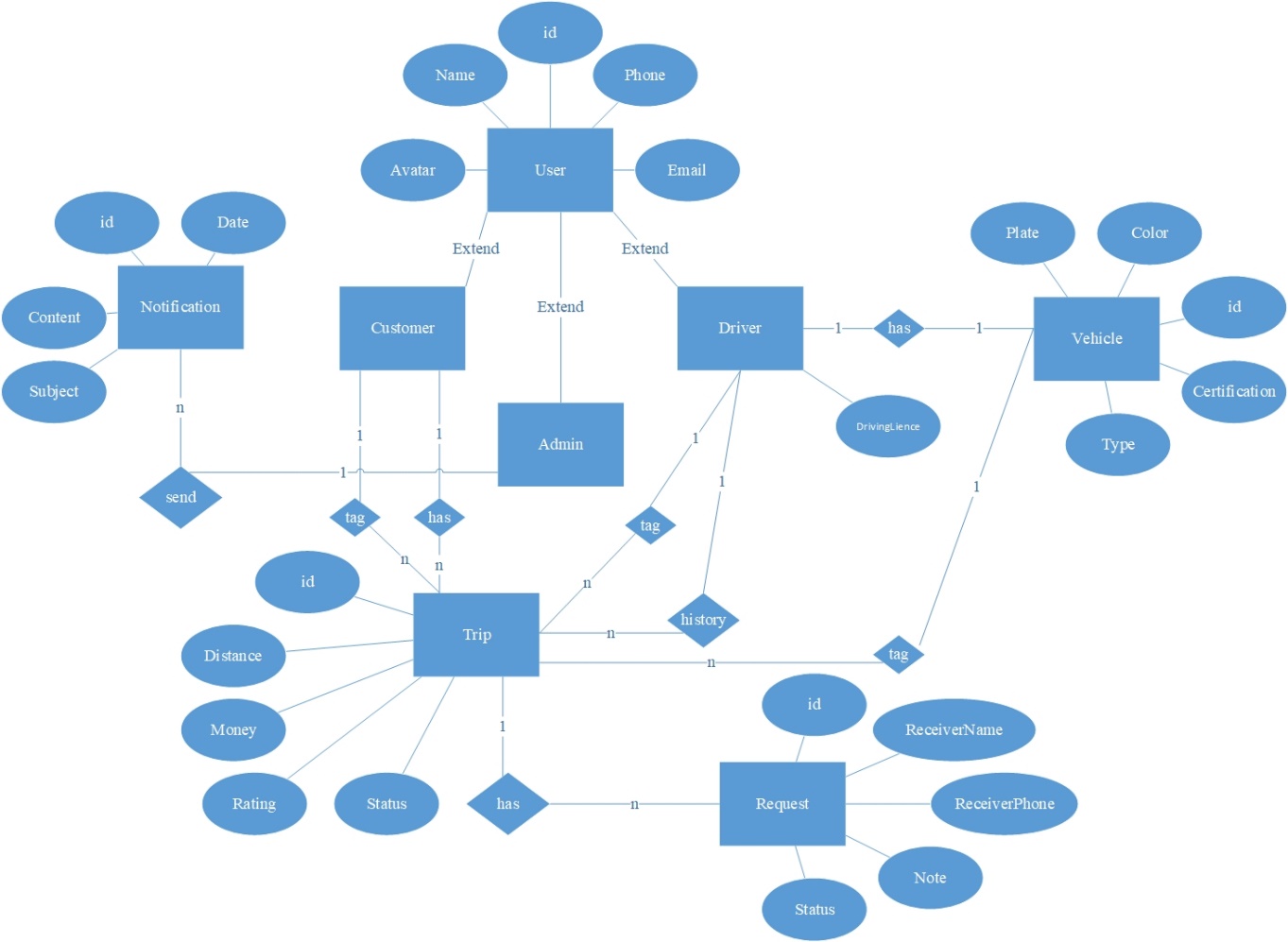
Khi khách hàng thực hiện yêu cầu gọi xe hệ thống sẽ tự động lưu dưới dạng hành trình với thông tin của khách hàng, thông tin tài xế, phương tiện sử dụng, các thông tin về dịch vụ sử dụng, quãng đường phát sinh, chi phí sử dụng, trạng thái của hành trình… Cả khách hàng và tài xế đều có thể xem lại thông tin hành trình mà mình tham gia

Mỗi 1 hành trình có thể có 1 hoặc nhiều điểm đến. Mỗi điểm đến chứa thông tin của người nhận, loại hàng hóa, ghi chú cho tài xế, và trạng thái xử lý.

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

Từ những yêu cầu bài toán ở trên có thể xây dựng thành biểu đồ thực thể liên kết sau:



Trong đó User là thực thể đại diện cho người dùng với các thông tin về id, tên người dùng, số điện thoại, email, và đường dẫn đến hình đại diện.

Customer,Admin, Driver là 3 thực thể con của User ứng với 3 loại người dùng của hệ thống.

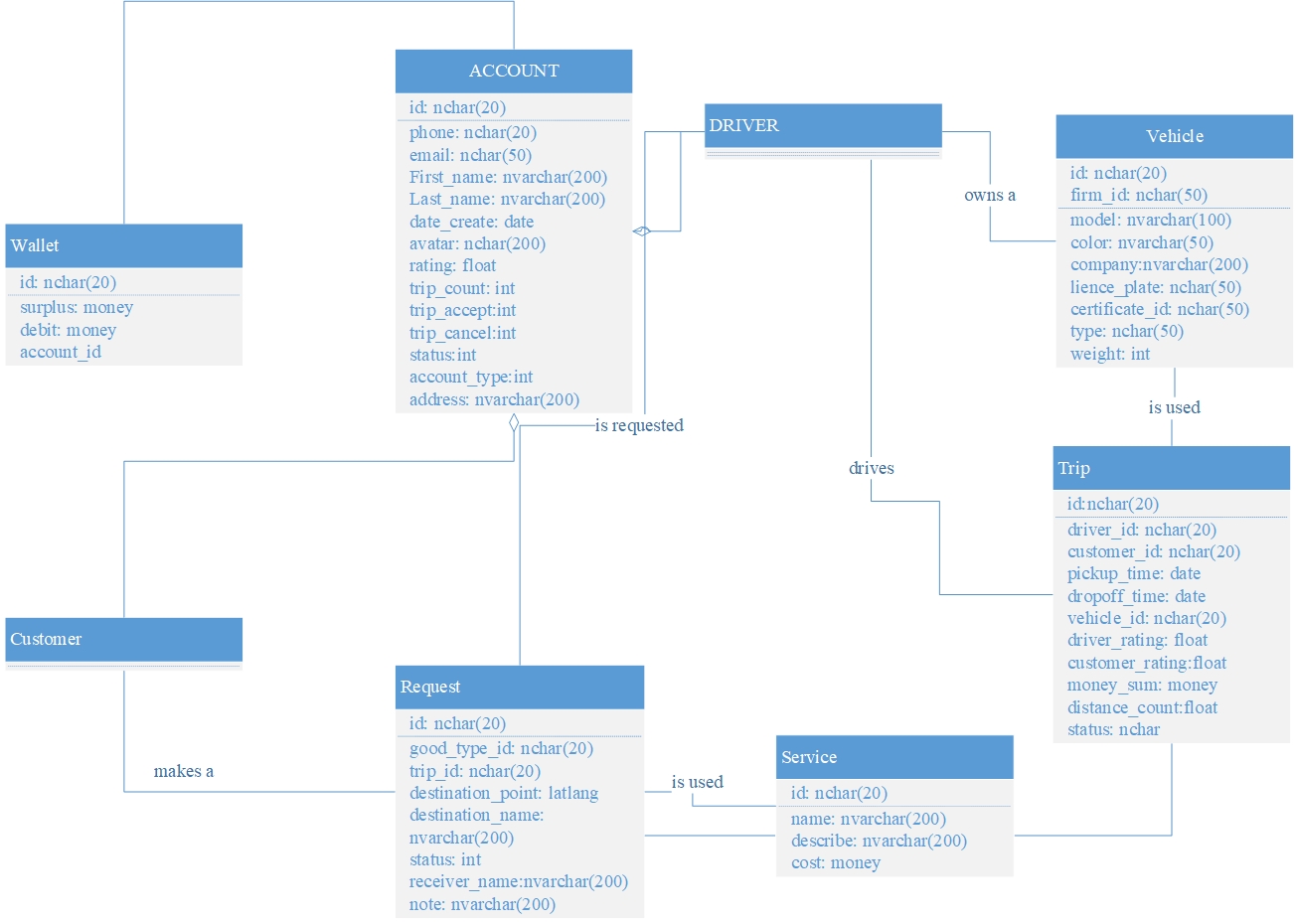
Trip là thông tin hành trình được hệ thống tạo ra khi người dùng yêu cầu giao hàng và tài xế chấp nhận yêu cầu đó. Bao gồm các thông tin về id, quãng đường, số tiền thanh toán, đánh giá, trạng thái hành trình. Mỗi khách hàng và tài xế có thể có nhiều hành trình và có thể lưu hành trình đó vào lịch sử sử dụng.

Một Trip có thể có nhiều điểm dừng gọi là Request, bao gồm các thông tin về id, tên người nhận, số điện thoại người nhận, loại hàng hóa vận chuyển, ghi chú,..

Mỗi tài xế sở hữu một loại phương tiện biểu diễn dưới dạng thực thể Device với các thông tin về id, biển số, màu sắc, hãng, loại (ô tô, xe máy). Tài xế dùng phương tiện trong mỗi hành trình xử lý yêu cầu của khách hàng.

Admin có thể gửi thông báo đến người dùng dưới dạng thực thể liên kết Notification với các thông tin về id, tiêu đề, ngày đăng, nội dung thông báo.

#### Biểu đồ lớp



#### Mô tả chi tiết lớp

* Lớp Account: lớp này mô tả các thông tin của tài khoản người dùng hệ thống.

|  |  |  |
| --- | --- | --- |
| Tên | Kiểu | Mô tả |
| id | int | Mã người dùng |
| phone | nchar | Số điện thoại |
| email | nchar | Địa chỉ thư điện tử |
| first\_name | nchar | Tên người dùng |
| last\_name | nchar | Họ người dùng |
| date\_created | date | Ngày kích hoạt |
| avartar | nchart | Đường dẫn ảnh đại diện |
| rating | int | Độ đánh giá |
| trip\_count | int | Số lượng dịch vụ đã sử dụng |
| trip\_accept | int | Số lượt dịch vụ thành công |
| trip\_cancel | int | Số lượt dịch vụ thất bại |
| status | boolean | Trạng thái (chặn/ không bị chặn) |
| account\_type | int | loại tài khoản |

* Lớp Vehicle: lớp này mô tả các thông tin phương tiện của tài xế

|  |  |  |
| --- | --- | --- |
| Tên | Kiểu | Mô tả |
| id | nchar | Mã phương tiện |
| firm\_id | nchar | Biển số xe |
| model | nchar | Loại xe |
| color | nchar | Màu xe |
| company | nvarchar | Hãng xe |
| lience\_palate | nchar | Số đăng ký xe |
| type | int | Loại xe (ô tô hoặc xe máy) |
| weight | int | Tải trọng của phương tiện |

* Lớp Request: lớp này mô tả một yêu cầu điểm dừng trên hành trình

|  |  |  |  |
| --- | --- | --- | --- |
| Tên | Kiểu | Mô tả | |
| id | int | Mã yêu cầu |
| good\_type | nchar | Mã loại hàng hóa |
| destination\_point | latlang | Điểm đến |
| Destination\_name | Nvarchar | Tên điểm đến |
| trip\_id | nchar | Mã đơn hàng |
| Receiver\_name | Nchar | Tên người nhận |
| note | nchar | Ghi chú |
| status | int | Trạng thái |
| Receiver\_phone | Nchar | Số điện thoại người nhận |

* Lớp Trip: lớp này mô tả một yêu cầu điểm dừng trên hành trình

|  |  |  |  |
| --- | --- | --- | --- |
| Tên | Kiểu | Mô tả | |
| id | int | Mã đơn hàng |
| driver\_id | nchar | Mã tài xế |
| customer\_id | nchar | Mã khách hàng |
| pickup\_time | date | Thời gian nhận |
| vehicle\_id | ncahr | Mã phương tiện |
| driver\_rating | int | Đánh giá tài xế |
| customer\_rating | int | Đánh giá khách hàng |
| money\_sum | money | Tổng chi phí phát sinh |
| distance\_count | float | Tổng quãng đường |
| status | nchar | Trạng thái của hành trình |

* Lớp Wallet: lớp này mô tả thông tin ví của khách hàng

|  |  |  |  |
| --- | --- | --- | --- |
| Tên | Kiểu | Mô tả | |
| id | int | Mã ví |
| surplus | money | số dư |
| debit | money | số nợ |
| account\_id | nchar | Mã người dùng |

* Lớp Service: Bảng này mô tả các loại dịch vụ riêng mà tài xế cung cấp. Vd: bốc xếp, dỡ hàng lên phương tiện, chi phí đỗ xe, chi phí cầu đường

|  |  |  |  |
| --- | --- | --- | --- |
| Tên | Kiểu | Mô tả | |
| id | int | Mã dịch vụ |
| name | nvarchar | Tên dịch vụ |
| describe | nvarchar | Mô tả về dịch vụ |
| cost | money | Giá dịch vụ |

## Xây dựng ứng dụng

### Thư viện và công cụ sử dụng

Sinh viên liệt kê các công cụ, ngôn ngữ lập trình, API, thư viện, IDE, công cụ kiểm thử, v.v. mà mình sử dụng để phát triển ứng dụng. Mỗi công cụ phải được chỉ rõ phiên bản sử dụng. SV nên kẻ bảng mô tả tương tự như Bảng 1. Nếu có nhiều nội dung trình bày, sinh viên cần xoay ngang bảng.

**Bảng 1** Danh sách thư viện và công cụ sử dụng

|  |  |  |
| --- | --- | --- |
| **Mục đích** | **Công cụ** | **Địa chỉ URL** |
| IDE lập trình | Eclipse Oxygen 64 bit | http://www.eclipse.org/ |
| v.v. | v.v. | v.v. |

### Kết quả đạt được

Sinh viên trước tiên mô tả kết quả đạt được của mình là gì, ví dụ như các sản phẩm được đóng gói là gì, bao gồm những thành phần nào, ý nghĩa, vai trò?

Sinh viên cần thống kê các thông tin về ứng dụng của mình như: số dòng code, số lớp, số gói, dung lượng toàn bộ mã nguồn, dung lượng của từng sản phẩm đóng gói, v.v. Tương tự như phần liệt kê về công cụ sử dụng, sinh viên cũng nên dùng bảng để mô tả phần thống kê này.

### Minh hoạ các chức năng chính

Sinh viên lựa chọn và đưa ra màn hình cho các chức năng chính, quan trọng, và thú vị nhất. Mỗi giao diện cần phải có lời giải thích ngắn gọn. Khi giải thích, sinh viên có thể kết hợp với các chú thích ở trong hình ảnh giao diện.

## Kiểm thử

Phần này có độ dài từ hai đến ba trang. Sinh viên thiết kế các trường hợp kiểm thử cho hai đến ba chức năng quan trọng nhất. Sinh viên cần chỉ rõ các kỹ thuật kiểm thử đã sử dụng. Chi tiết các trường hợp kiểm thử khác, nếu muốn trình bày, sinh viên đưa vào phần phụ lục.

Sinh viên sau cùng tổng kết về số lượng các trường hợp kiểm thử và kết quả kiểm thử. Sinh viên cần phân tích lý do nếu kết quả kiểm thử không đạt.

## Triển khai

Sinh viên trình bày mô hình và/hoặc cách thức triển khai thử nghiệm/thực tế. Ứng dụng của sinh viên được triển khai trên server/thiết bị gì, cấu hình như thế nào. Kết quả triển khai thử nghiệm nếu có (số lượng người dùng, số lượng truy cập, thời gian phản hồi, phản hồi người dùng, khả năng chịu tải, các thống kê, v.v.)

# Các giải pháp và đóng góp nổi bật

Chương này có độ dài tối thiểu 5 trang, tối đa không giới hạn. Sinh viên cần trình bày tất cả những nội dung đóng góp mà mình thấy tâm đắc nhất trong suốt quá trình làm ĐATN. Đó có thể là một loạt các vấn đề khó khăn mà sinh viên đã từng bước giải quyết được, là giải thuật cho một bài toán cụ thể, là giải pháp tổng quát cho một lớp bài toán, hoặc là mô hình/kiến trúc hữu hiệu nào đó được sinh viên thiết kế.

Chương này là **cơ sở quan trọng** để các thầy cô đánh giá sinh viên. Vì vậy, sinh viên cần phát huy tính sáng tạo, khả năng phân tích, phản biện, lập luận, tổng quát hóa vấn đề và tập trung viết cho thật tốt.

Mỗi giải pháp hoặc đóng góp của sinh viên cần được trình bày trong một mục độc lập bao gồm ba mục con: (i) dẫn dắt/giới thiệu về bài toán/vấn đề, (ii) giải pháp, và (iii) kết quả đạt được (nếu có).

Sinh viên lưu ý **không trình bày lặp lại nội dung**. Những nội dung đã trình bày chi tiết trong các chương trước không được trình bày lại trong chương này. Vì vậy, với nội dung hay, mang tính đóng góp/giải pháp, sinh viên chỉ nên tóm lược/mô tả sơ bộ trong các chương trước, đồng thời tạo tham chiếu chéo tới đề mục tương ứng trong Chương 5 này. Chi tiết thông tin về đóng góp/giải pháp được trình bày trong mục đó.

Ví dụ, trong Chương 4, sinh viên có thiết kế được kiến trúc đặc sắc gì đó, là sự kết hợp của các kiến trúc MVC, MVP, SOA, v.v. Khi đó, sinh viên sẽ chỉ mô tả ngắn gọn kiến trúc đó ở Chương 4, rồi thêm các câu có dạng: “Chi tiết về kiến trúc này sẽ được trình bày trong Mục 5.1”.

# 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

Qua quá trình thực hiện ĐATN em đã thu được các kết quả sau:

* Xây dựng thành công ứng dụng Gọi xe chuyển hàng Online với đầy đủ các chức năng cơ bản đã đặt ra, tích hợp vào một số tính năng mới giúp giải quyết các vấn đề còn tồn đọng so với các ứng dụng khác trên thị trường như chức năng phân loại xe, chuyển hàng tiết kiệm, tối ưu hóa đường đi, …
* 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:

1. Tìm hiểu và nắm được các kiến thức cơ bản về các dịch vụ mà Google cung cấp: Google Maps API, Firebase.
2. Nắm được những kiến thức cơ bản về lập trình trên nền tảng Android
3. Cũng cố kiến thức về phân tích, thiết kế và phát triển phần mềm, hệ thống..

## Hướng phát triển

Trong phần này, sinh viên trình bày định hướng công việc trong tương lai để hoàn thiện sản phẩm hoặc nghiên cứu của mình.

Trước tiên, sinh viên trình bày các công việc cần thiết để hoàn thiện các chức năng/nhiệm vụ đã làm. Sau đó sinh viên phân tích các hướng đi mới cho phép cải thiện và nâng cấp các chức năng/nhiệm vụ đã làm.

# Tài liệu tham khảo

Lưu ý: Sinh viên không được đưa bài giảng/slide, các trang Wikipedia, hoặc các trang web thông thường làm tài liệu tham khảo.

Một trang web được phép dùng làm tài liệu tham khảo **chỉ khi** nó là công bố chính thống của cá nhân hoặc tổ chức nào đó. Ví dụ, trang web đặc tả ngôn ngữ XML của tổ chức W3C https://www.w3.org/TR/2008/REC-xml-20081126/ là TLTK hợp lệ.

Có năm loại tài liệu tham khảo mà sinh viên phải tuân thủ đúng quy định về cách thức liệt kê thông tin như sau:

\* **Bài báo đăng trên tạp chí khoa học**: Tên tác giả, tên bài báo, tên tạp chí, volume, từ trang đến trang (nếu có), nhà xuất bản, năm xuất bản.

1. Hovy E. H., Automated Discourse Generation Using Discourse Structure Relations, Artificial Intelligence, Elsevier Science Publishers, 63: 341-385, 1993.

\* **Sách**: Tên tác giả, tên sách, volume (nếu có), lần tái bản (nếu có), nhà xuất bản, năm xuất bản.

1. Peterson L. L. and Davie B. S. , Computer Networks: A Systems Approach, 2nd ed., Mogran-Kaufmann, 1999.
2. Nguyễn Thúc Hải, Mạng máy tính và các hệ thống mở, Nhà xuất bản giáo dục, 1999.

\* **Tập san Báo cáo Hội nghị Khoa học**: Tên tác giả, tên báo cáo, tên hội nghị, ngày (nếu có), địa điểm hội nghị, năm xuất bản.

1. Poesio M. and Di Eugenio B., Discourse Structure and Anaphoric Accessibility, In Proc. of the ESSLLI Workshop on Information Structure, Discourse Structure and Discourse Semantics, Helsinki, 2001.

\* **Đồ án tốt nghiệp, Luận văn Thạc sĩ, Tiến sĩ**: Tên tác giả, tên đồ án/luận văn, loại đồ án/luận văn, tên trường, địa điểm, năm xuất bản.

1. Knott D., A Data-Driven Methodology for Motivating a Set of Coherence Relations, Ph.D. Thesis, University of Edinburgh, UK, 1996.

\* **Tài liệu tham khảo từ Internet** : Tên tác giả (nếu có), tựa đề, cơ quan (nếu có), địa chỉ trang web, thời gian lần cuối truy cập trang web.

1. Berners-Lee T., Hypertext Transfer Protocol (HTTP), CERN, ftp:/info.cern.ch/pub/www/doc/http-spec.txt.Z, last visited May 2010.
2. Princeton University, WordNet, http://www.cogsci.princeton.edu/~wn/index.shtml, last visited May 2010.

# Phụ lục

Phần phụ lục là không bắt buộc. Nếu sinh viên không có nhu cầu trình bày thêm, có thể xóa bỏ phần này. Lưu ý là phần phụ lục chỉ được đánh chỉ mục đến cấp 2, sinh viên không được phép chia nhỏ hơn nữa.

Hướng dẫn viết đồ án tốt nghiệp

Quy định chung

Dưới đây là một số quy định và hướng dẫn viết đồ án tốt nghiệp mà bắt buộc sinh viên phải **đọc kỹ và tuân thủ nghiêm ngặt**.

Sinh viên cần đảm bảo tính thống nhất toàn báo cáo (font chữ, căn dòng hai bên, hình ảnh, bảng, margin trang, đánh số trang, v.v.). Để làm được như vậy, sinh viên chỉ cần sử dụng các định dạng theo đúng template ĐATN này. Khi paste nội dung văn bản từ tài liệu khác của mình, sinh viên cần chọn kiểu Copy là “Text Only” để định dạng văn bản của template không bị phá vỡ/vi phạm.

Tuyệt đối cấm sinh viên đạo văn. Sinh viên cần ghi rõ nguồn cho tất cả những gì không tự mình viết/vẽ lên, bao gồm các câu trích dẫn, các hình ảnh, bảng biểu, v.v. Khi bị phát hiện, sinh viên sẽ **không được phép bảo vệ ĐATN**.

Tất cả các hình vẽ, bảng biểu, công thức, và tài liệu tham khảo trong ĐATN nhất thiết phải được SV giải thích và tham chiếu tới ít nhất một lần. Không chấp nhận các trường hợp sinh viên đưa ra hình ảnh, bảng biểu tùy hứng và không có lời mô tả/giải thích nào.

Sinh viên tuyệt đối không trình bày ĐATN theo kiểu viết ý hoặc gạch đầu dòng. ĐATN không phải là một slide thuyết trình; khi người đọc không hiểu sẽ không có ai giải thích hộ. Sinh viên cần viết thành các đoạn văn và phân tích, diễn giải đầy đủ, rõ ràng. Câu văn cần đúng ngữ pháp, đầy đủ chủ ngữ, vị ngữ và các thành phần câu.

Khi thực sự cần liệt kê, sinh viên nên liệt kê theo phong cách khoa học với các ký tự La Mã. Ví dụ, nhiều sinh viên luôn cảm thấy hối hận vì (i) chưa cố gắng hết mình, (ii) chưa sắp xếp thời gian học/chơi một cách hợp lý, (iii) chưa tìm được người yêu để chia sẻ quãng đời sinh viên vất vả, và (iv) viết ĐATN một cách cẩu thả.

Trong một số trường hợp nhất thiết phải dùng các bullet để liệt kê, sinh viên cần thống nhất Style cho toàn bộ các bullet các cấp mà mình sử dụng đến trong báo cáo. Nếu dùng bullet cấp 1 là hình tròn đen, toàn bộ báo cáo cần thống nhất cách dùng như vậy; ví dụ như sau:

* Đây là mục 1 – Thực sự không còn cách nào khác em mới dùng đến bullet trong báo cáo.
* Đây là mục 2 – Nghĩ lại thì em có thể không cần dùng bullet cũng được. Nên em sẽ xóa bullet và tổ chức lại hai mục này trong báo cáo của mình cho khoa học hơn. Em muốn thầy cô và người đọc cảm nhận được tâm huyết của em trong từng trang báo cáo ĐATN.

Tạo đề mục

Đề mục giúp tạo bố cục cho tài liệu. Để các tính năng tự động – ví dụ tính năng cập nhật mục lục, hoặc tính năng tham chiếu chéo – của Word hoạt động được, sinh viên cần tuân thủ theo các style đã tạo trong tài liệu này. Để hiển thị các style này, sinh viên vào tab Home trong thanh Ribbon của Word.

Để tạo đề mục cấp 1, 2, 3, 4, 5, sinh viên gõ tiêu đề cho đề mục của mình rồi chọn các style là Heading 1, 2, 3, 4, 5 tương ứng. Sinh viên hạn chế dùng tới đề mục cấp 4, và phải trong trường hợp thực sự cần thiết mới dùng đến đề mục cấp 5.

Phần phụ lục chỉ cho phép có hai cấp tiêu đề. Hai style tương ứng với hai cấp này là “Heading 7, Phụ lục cấp 1” và “Heading 8, Phụ lục cấp 2”.

Bảng biểu

Sinh viên lưu ý không để bảng tràn ra lề (margin) trên, dưới, trái hoặc phải của trang. Do không gian nhỏ hẹp, bảng nên có font là 12pt, nhỏ hơn một chút so với font thông thường (13pt) của ĐATN. Độ dãn dòng của bảng nên là 1 line. Căn lề bảng là căn giữa, nhưng nội dung văn bản trong bảng nên được căn lề trái.

Sinh viên có thể viết tắt các từ trong bảng để tiết kiệm không gian nhưng phải giải thích các từ viết tắt này ở phần Chú thích bảng. Ví dụ áp dụng được minh họa trong Bảng 2.

**Bảng 2** Ví dụ sử dụng bảng

Chú thích: Y: Year; RS: Risk Set; G: Graduated; AB: Academically Excluded;C: Censored; HRG: Hazard Ratio – Graduated

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Y** | **RS** | **G** | **AE** | **C** | **HRG**  **(%)** |
| **1** | 11.959 | 0 | 725 | 1619 | 0,0 |
| **2** | 10.457 | 0 | 474 | 1513 | 0,0 |
| **3** | 7.365 | 1213 | 335 | 966 | 16,7 |
| **4** | 900 | 599 | 145 | 405 | 55,3 |

Sinh viên được tùy ý lựa chọn định dạng (template) cho các bảng trong ĐATN của mình, nhưng phải đảm bảo tính nhất quán trên toàn ĐATN. Template bảng phải đảm bảo phần heading cột trong bảng có font in đậm và nổi bật (highlighted) hơn so với các nội dung khác trong bảng.

Vì bảng có thể kéo dài nhiều trang, tiêu đề của bảng nên để ở phần đầu của bảng. Sinh viên không thêm tiêu đề bảng bằng tay. Để thêm tiêu đề bảng tự động, sinh viên nhấn chuột phải vào bảng, chọn “Insert Caption”, chọn “Label” là “Bảng”, rồi nhấn nút “OK”. Sau đó, sinh viên nhập vào nội dung tiêu đề và căn chỉnh “Center” cho tiêu đề này. Lưu ý, sinh viên cần bôi đậm bằng tay cụm từ **Bảng n**. Kết quả thu được có dạng như sau “**Bảng 2** Ví dụ sử dụng bảng”. Sinh viên nên xoay ngang trang giấy với bảng có nhiều cột hoặc nhiều nội dung văn bản.

Hình vẽ

Tương tự như bảng, sinh viên không được để hình vẽ tràn lề trang. Căn lề cho hình vẽ là căn giữa (Center). Cách thêm tiêu đề hình vẽ tương tự như cách thêm tiêu đề bảng, nhưng sinh viên chọn “Label” là “Hình” thay vì “Bảng”.



**Hình 3** Ví dụ hình vẽ

Tiêu đề hình vẽ phải đặt ở dưới hình vẽ. Nếu hình vẽ được copy từ trên mạng, sinh viên bắt buộc phải ghi rõ nguồn. Sinh viên nên thống nhất công cụ sử dụng và style cho hình vẽ trong toàn ĐATN. Các chi tiết trong hình vẽ phải được bố trí gọn gàng; chữ trong hình phải đảm bảo nhìn được rõ nét khi in báo cáo trên giấy A4. Khi resize ảnh, cần giữ nguyên tỷ lệ dài rộng, tránh làm méo hoặc vỡ hình. Ví dụ hình vẽ được minh họa trong Hình 3.

Tài liệu tham khảo

Sinh viên cần hạn chế tối đa dùng trang Web làm tài liệu tham khảo. Chỉ chấp nhận trang Web làm TLTK khi trang đó là nơi công bố chính thức của tổ chức hoặc cá nhân nào đó. Ví dụ, trang web đặc tả ngôn ngữ XML của tổ chức W3C https://www.w3.org/TR/2008/REC-xml-20081126/ là TLTK hợp lệ.

Khi giới thiệu về công cụ, API, thư viện, hoặc nền tảng nào đó, sinh viên có thể đưa ra địa chỉ URL của các tiện ích này. Sinh viên lưu ý địa chỉ URL đó không phải là tài liệu tham khảo. Trong các trường hợp tương tự như vậy, sinh viên nên tạo “Footnote”. Sinh viên tạo “Footnote” bằng cách vào mục “References”, chọn “Insert Footnote”. Ví dụ tạo Footnote như sau: TensorFlow[[1]](#footnote-1) là nền tảng học máy mã nguồn mở đang được sử dụng rộng rãi hiện nay. Lưu ý: số Footnote phải đặt sát với từ được mô tả. Như trong ví dụ trên, số 1 được đặt ngay cạnh chữ TensorFlow (không có dấu cách).

Công thức toán học

Giống như bảng, hình vẽ, và tài liệu tham khảo, công thức toán học cần được đánh số, giải thích, và tham chiếu đầy đủ.

Để thêm tiêu đề (caption) cho công thức, sinh viên đặt con trỏ văn bản vào dòng văn bản dưới công thức. Sau đó vào menu “Insert” (lưu ý là menu Insert, không phải là tab Insert trong thanh Ribbon) và chọn mục “Caption”. Từ Popup Menu hiện ra, sinh viên chọn “Label” là “Công thức”, rồi nhấn nút “OK”. Sinh viên bôi đậm chữ **Công thức n**, và căn tiêu đề ra giữa. Công thức 1 là ví dụ mẫu cho sinh viên tham khảo.

**Công thức 1** Khai triển Newton

Tham chiếu chéo

Tham chiếu chéo (Cross-reference) là tiện ích hữu hiệu cho người viết báo cáo. Nó giúp tạo các liên kết tham chiếu (hyperlink) tới các hình ảnh, bảng biểu, tài liệu tham khảo, và các đề mục một cách tự động. Ví dụ, ngay trong câu này, một tham chiếu đã được tạo ra tới mục 2.1. Người đọc dễ dàng nhấp chuột vào liên kết 2.1 để ngay lập tức chuyển đến mục đó.

Để tạo tham chiếu chéo tới các đề mục (heading), sinh viên vào tab “References”, rồi tìm và nhấn chọn “Cross-refrence”. SV chọn “Refrence type” là “Heading” và chọn “Insert Reference to” là “Heading number (no context)”. Sau đó, SV chọn phần đề mục muốn tham chiếu rồi bấm “Insert”.

Để tạo tham chiếu chéo tới các hình vẽ, bảng biểu và công thức, sinh viên cũng vào tab “References” và chọn “Cross-refrence”. SV chọn “Refrence type” là “Hình”, “Bảng”, hoặc “Công thức’. SV chọn “Insert Reference to” là “Only label and number”, trỏ đến phần muốn tham chiếu rồi bấm “Insert”. Nếu font chữ trong liên kết tham chiếu tạo ra được in đậm (bold), SV chuyển về dạng thường cho chuẩn tắc.

Thực hiện tương tự các bước mô tả ở trên, sinh viên có thể tạo tham chiếu chéo tới các tài liệu tham khảo. Sinh viên chọn “Refrence type” là “Numbered item”, chọn “Insert Reference to” là “Paragraph number”, trỏ đến phần tài liệu tham khảo muốn tham chiếu rồi bấm “Insert”. Ví dụ, tham chiếu chéo tới tài liệu tham khảo [2], [3], [4] đã được tạo.

Cập nhật mục lục và tham chiếu chéo

Trong suốt quá trình viết ĐATN, sinh viên sẽ tạo ra nhiều xáo trộn như thay đổi vị trí hình và bảng, thay đổi thứ tự tài liệu tham khảo, thay đổi tên đề mục, v.v. Vì vậy, để hoàn thành ĐATN, sinh viên cần cập nhật lại các thành phần mục lục, danh sách hình ảnh/bảng/công thức và tham chiếu chéo được tạo tự động trong ĐATN của mình.

Để cập nhật các thành phần này, sinh viên bấm Ctrl+A để chọn toàn báo cáo, nhấn chuột phải và chọn “Update Field”, rồi liên tục chọn mục “Entire table” khi được hỏi trong hộp thoại Popup Menu. Sau đó sinh viên search chữ “Error” trên toàn báo cáo để kiểm tra xem có lỗi đánh chỉ mục hoặc lỗi tham chiếu nào không. Đồng thời, sinh viên nên tự soát lại bằng tay toàn bộ nội dung quyển ĐATN để tránh mọi sai sót.

In quyển đồ án tốt nghiệp

Do hiện nay có nhiều phiên bản Word cho nhiều nền tảng máy tính, sinh viên nhất thiết phải xuất ĐATN ra định dạng PDF rồi mang tới cửa hàng in ấn để tránh sai sót.

Sinh viên chỉ cần in bìa giấy cứng mà không cần đóng quyển để tiết kiệm chi phí và vận chuyển ĐATN dễ dàng hơn.

Quyển ĐATN nên được in trên các trang giấy trắng đủ dày. Mỗi tờ giấy A4 chỉ được in một mặt.

Đặc tả use case

Nếu trong nội dung chính không đủ không gian cho các use case khác (ngoài các use case nghiệp vụ chính) thì đặc tả thêm cho các use case đó ở đây.

Đặc tả use case “Thống kê tình hình mượn sách”

Cách trình bày tương tự như phần hướng dẫn ở mục 2.3.

Đặc tả use case “Đăng ký làm thẻ mượn”

Cách trình bày tương tự như phần hướng dẫn ở mục 2.3.

Công nghệ sử dụng

Công nghệ bảo mật dữ liệu

Công nghệ blockchain

Thiết kế gói

Thiết kế gói cho kiến trúc tổng quan

Thiết kế gói cho chức năng “Trả sách”

Thiết kế lớp

1. https://www.tensorflow.org/, lần truy cập cuối: 28/06/2018 [↑](#footnote-ref-1)