Trường Đại học Bách Khoa Hà Nội

Viện Công nghệ Thông Tin và Truyền Thông

Đồ án Tốt nghiệp Đại học

Ứng dụng di động gọi xe với giá cả thỏa thuận giữa người dùng và tài xế

|  |  |
| --- | --- |
| Sinh viên thực hiện | Nguyễn Đình Đoàn |
| Người hướng dẫn | ThS. Vũ Đức Vượng |

­­

Hà Nội, 05/2019

# Lời cam kết

Họ và tên sinh viên : Nguyễn Đình Đoàn . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

Điện thoại liên lạc : 0395616595. . . . . . . . . . . . Email : nguyendinhdoanjdk@gmail.com. . .

Lớp : CNTT2-1. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Hệ đào tạo : Kĩ sư. . . . . . . . . . . . . . .

Tôi – *Nguyễn Đình Đoàn* – cam kết Đồ án Tốt nghiệp (ĐATN) là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của *ThS. Vũ Đức Vượng*. Các kết quả nêu trong ĐATN là trung thực, là thành quả của riêng tôi, 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. Tôi xin hoàn toàn chịu trách nhiệm với dù chỉ một sao chép vi phạm quy chế của nhà trường.

|  |  |
| --- | --- |
|  | *Hà Nội, ngày tháng năm*  Tác giả ĐATN  *Nguyễn Đình Đoàn* |

# Lời cảm ơn

Lời đầu tiên, em xin chân thành cảm ơn các thầy, cô ở trường Đại học Bách Khoa Hà Nội nói chung cũng như Viện Công nghệ Thông tin và Truyền Thông nói riêng, đã giảng dạy, và truyền đạt cho những thế hệ sinh viên chúng em nhiều kiến thúc bổ ích, lý thú những cách giải quyết vấn đề thực tế rất hay và sâu sắc. Đó chính là những hành trang vững chắc giúp chúng em bước vào cuộc sống thực tế sau này.

Đặc biệt, em xin gửi lời cảm ơn sâu sắc đến thầy Vũ Đức Vượng - giảng viên Bộ môn Công nghệ Phần mềm, Viện Công nghệ Thông tin và Truyền thông đã tận tình hướng dẫn, chỉ bảo em trong quá trình làm đồ án tốt nghiệp. Thầy là người hiền hậu, nhiệt tình luôn định hướng động viện viên em vượt qua những giai đoạn khó khăn của quá trình làm dự án cuối cùng của cuộc đời sinh viên này.

Cuối cùng, con cũng xin cảm ơn bố mẹ đã luôn bên cạnh chăm sóc, yêu thương con suốt những năm vừa qua. Bố mẹ và em gái ở bên động viên, ủng hộ con trong những giai đoạn khó khăn nhất của cuộc đời đã tạo cho con động lực để vượt qua giúp con mạnh mẽ và trưởng thành hơn.

|  |  |
| --- | --- |
|  | *Người làm đồ án, sinh viên*  *Nguyễn Đình Đoàn* |

# Tóm tắt

Công nghệ ngày càng phát triển người ta cũng sẽ tìm cho mình một cách di chuyển nhanh hơn và hiệu quả và tiết kiệm chi phí hơn. Taxi hay xe ôm công nghệ đã và đang trở nên phổ biến và ưa chuộng. Số dùng người dùng và tài xế tăng trưởng một cách chóng mặt giúp Uber hay Grab trở thành những công ty công nghệ khởi nghiệp hàng đầu. Cách sử dụng đơn giản, cước phí di chuyển rẻ hơn cho khách hàng và mức thu nhập hấp dẫn cho chủ xe…Rất nhiều lợi ích to lớn khiến cách di chuyển này trở thành xu hướng trong tương lai gần khi rất nhiều ứng dụng như vậy đã được ra đời như: Go Việt, Be… Nhưng tài xế lại không nhận được toàn bộ thu nhập của mình mà phải chia sẻ cho nhà phát hành ứng dụng, khách hàng không được chọn xe mình muốn đi mà nhận gợi ý từ hệ thống. Từ ý tưởng giải quyết hai vấn đề này em quyết định hiện thực hóa trong đồ án tốt nghiệp xây dựng ứng dụng đặt xe cho khách hàng và tài xế.

Nội dung đồ án sẽ trình bày chi tiết từ quá trình thiết kế, xây dựng ý tưởng, triển khai lập trình và kiểm thử. Cùng với đó là các giải pháp và đóng góp nổi bật cùng hướng phát triển ứng dụng trong tương lai. Tất cả được trình bày củ thể và chi tiết qua sáu chương sau: (i) Giới thiệu về đề tài, (ii) Khảo sát và phân tích yêu cầu, (iii) Công nghệ sử dụng, (iv) Phát triển và triển khai ứng dụng, (v) Các giải pháp và đóng góp nổi bật, (vi) Kết luận và hương phát triển trong tương lai.

# Mục lục

[Lời cam kết ii](#_Toc9091698)

[Lời cảm ơn iii](#_Toc9091699)

[Tóm tắt iv](#_Toc9091700)

[Mục lục v](#_Toc9091701)

[Danh mục hình vẽ viii](#_Toc9091702)

[Danh mục bảng xii](#_Toc9091703)

[Danh mục các từ viết tắt xiii](#_Toc9091704)

[Danh mục thuật ngữ xiv](#_Toc9091705)

[Chương 1 Giới thiệu đề tài 1](#_Toc9091706)

[1.1 Đặt vấn đề 1](#_Toc9091707)

[1.2 Mục tiêu và phạm vi đề tài 2](#_Toc9091708)

[1.3 Định hướng giải pháp 3](#_Toc9091709)

[1.4 Bố cục đồ án 4](#_Toc9091710)

[Chương 2 Khảo sát và phân tích yêu cầu 5](#_Toc9091711)

[2.1 Khảo sát hiện trạng 5](#_Toc9091712)

[2.2 Tổng quan chức năng 7](#_Toc9091713)

[2.2.1 Biểu đồ use case tổng quan 7](#_Toc9091714)

[2.2.2 Biểu đồ use case phân rã cho chức năng đặt cuốc xe 9](#_Toc9091715)

[2.2.3 Biểu đồ use case phân rã cho chức năng nhận cuốc xe 10](#_Toc9091716)

[2.2.4 Quy trình nghiệp vụ 11](#_Toc9091717)

[2.3 Đặc tả chức năng 12](#_Toc9091718)

[2.3.1 Đặc tả use case Đăng nhập 12](#_Toc9091719)

[2.3.2 Đặc tả use case Xác nhận số điện thoại 14](#_Toc9091720)

[2.3.3 Đặc tả use case Đặt cuốc xe 15](#_Toc9091721)

[2.3.4 Đặc tả use case Chấp nhận cuốc xe 19](#_Toc9091722)

[2.3.5 Đặc tả use case Xem thông tin thanh toán 20](#_Toc9091723)

[2.3.6 Đặc tả use case Đánh giá tài xế 22](#_Toc9091724)

[2.3.7 Đặc tả use case Xem lịch sử chuyến đi 22](#_Toc9091725)

[2.3.8 Đặc tả use case Cập nhật thông tin cá nhân 23](#_Toc9091726)

[2.4 Yêu cầu phi chức năng 25](#_Toc9091727)

[Chương 3 Công nghệ sử dụng 26](#_Toc9091728)

[3.1 Giới thiệu cơ bản về nền tảng Android 26](#_Toc9091729)

[3.1.1 Nền tảng Android 26](#_Toc9091730)

[3.1.2 Ưu điểm khi phát triển trên nền tảng Android 27](#_Toc9091731)

[3.2 Giới thiệu về Firebase 28](#_Toc9091732)

[3.2.1 Firebase Realtime Database 29](#_Toc9091733)

[3.2.2 Firebase Authentication 32](#_Toc9091734)

[3.2.3 Firebase Storage 34](#_Toc9091735)

[3.3 Giới thiệu về GeoFire 36](#_Toc9091736)

[3.4 Giới thiệu về Google API 38](#_Toc9091737)

[3.4.1 Google Map API cho Android 38](#_Toc9091738)

[3.4.2 Google Place API cho Android 43](#_Toc9091739)

[3.4.3 Google Direction API 45](#_Toc9091740)

[Chương 4 Phát triển và triển khai ứng dụng 48](#_Toc9091741)

[4.1 Thiết kế kiến trúc 48](#_Toc9091742)

[4.1.1 Mô hình client-server 48](#_Toc9091743)

[4.1.2 Áp dụng trong đồ án tốt nghiệp 49](#_Toc9091744)

[4.1.3 Thiết kế tổng quan của ứng dụng 49](#_Toc9091745)

[4.2 Thiết kế giao diện 51](#_Toc9091746)

[4.3 Thiết kế lớp 54](#_Toc9091747)

[4.4 Thiết kế cơ sở dữ liệu 58](#_Toc9091748)

[4.5 Xây dựng ứng dụng 61](#_Toc9091749)

[4.5.1 Thư viện và công cụ sử dụng 61](#_Toc9091750)

[4.5.2 Kết quả đạt được 63](#_Toc9091751)

[4.6 Kiểm thử 71](#_Toc9091752)

[Chương 5 Các giải pháp và đóng góp nổi bật 76](#_Toc9091753)

[5.1 Kết quả và những điểm mới trong việc phát triển ứng dụng 76](#_Toc9091754)

[5.2 Một số giải pháp về mặt công nghệ 77](#_Toc9091755)

[5.2.1 Áp dụng công nghệ Firebase 77](#_Toc9091756)

[5.2.2 Áp dụng API Map của Google 77](#_Toc9091757)

[Chương 6 Kết luận và hướng phát triển 79](#_Toc9091758)

[6.1 Kết luận 79](#_Toc9091759)

[6.2 Hướng phát triển 80](#_Toc9091760)

[Tài liệu tham khảo 82](#_Toc9091761)

# Danh mục hình vẽ

**Hình 1** Số liệu thống kê về tốc độ tăng trưởng của chuyến xe của Grab [1] 14

**Hình 2** Ứng dụng gọi xe Grab với nhiều hình thức sử dụng [2] 18

**Hình 3** Biểu đồ use case tổng quan cho ứng dụng 20

**Hình 4** Biểu đồ phân rã chức năng đặt cuốc xe 22

**Hình 5** Biểu đồ phân rã chức năng chấp nhận cuốc xe 23

**Hình 6** Biểu đồ quy trình nghiệp vụ của một cuốc xe 24

**Hình 7** Số liệu thống kê về thị phần hệ điều hành toàn cầu năm 2016 [3] 39

**Hình 8** Hình minh họa về sự phù hợp của Firebase khi phát triển ứng dụng di động [4] 41

**Hình 9** Thư viện Firebase Realtime Datababase 42

**Hình 10** Hình ảnh minh họa việc cấu hình Realtime Database trong ứng dụng 42

**Hình 11** Hình ảnh minh họa việc đọc dữ liệu từ Firebase Realtime Database 43

**Hình 12** Hình ảnh minh họa việc ghi dữ liệu từ Firebase Realtime Database 43

**Hình 13** Thư viện Firebase Authentication mới nhất từ Google 44

**Hình 14** Kích hoạt phương thức Phone number authentication của Firebase 45

**Hình 15** Cách hệ thống gửi một mã xác thực tới điện thoại người dùng. 45

**Hình 16** Các số điện thoại test của ứng dụng gọi xe 46

**Hình 17** Hình ảnh minh họa những người dùng đã đăng nhập hệ thống gọi xe với thỏa thuận từ tài xế và khách hàng 46

**Hình 18** Thư viện Firebase Storage 47

**Hình 19** Lấy một thể hiện của Firebase Storage 47

**Hình 20** Hình ảnh dữ liệu ảnh được lưu trong Firebase Storage 48

**Hình 21** Thư viện GeoFire 49

**Hình 22** Lấy một thể hiện của GeoFire 49

**Hình 23** Thiết lập dữ liệu vị trí 49

**Hình 24** Cách lấy dữ liệu vị trí với GeoFire 50

**Hình 25** Truy vấn dữ liệu vị trí quanh một bán kình nhất định 50

**Hình 26** Thư viện Map SDK cho Android 51

**Hình 27** Fragment để hiện thị Google Map trong ứng dụng Android 51

**Hình 28** Minh họa về hiện thị marker trong ứng dụng di động 52

**Hình 29** Minh họa về xử lí hiện thị marker trong các đoạn mã code 53

**Hình 30** Minh họa về hiện thị Polyline trên bản đồ 54

**Hình 31** Minh họa về cách xử lý hiện thị Polyline trên bản đồ 55

**Hình 32** Place Autocomplete trong Android 55

**Hình 33** Thư viện Place Autocomplete cho Android 56

**Hình 34** Đoạn mã dùng khởi tạo Place API 56

**Hình 35** Đoạn mã dùng để tạo hiện thị Place Autocomplete 57

**Hình 36** Minh họa đoạn mã để trả về địa chỉ từ Place Autocomplete 57

**Hình 37** Đường dẫn URL mà Direction API trả về 58

**Hình 38** Kết quả trả về dạng JSON của Direction API 58

**Hình 39** Mô hình client-server 60

**Hình 40** Mô hình client-server với Firebase 61

**Hình 41** Biểu đồ tổng quát 62

**Hình 42** Thiết kế màn hình đăng nhập 63

**Hình 43** Thiết kế màn hình xác thực số điện thoại 64

**Hình 44** Thiết kế màn hình khi khách hàng đặt cuốc xe 64

**Hình 45** Thiết kế màn hình khách hàng theo dõi thông tin cuốc xe 65

**Hình 46** Thiết kế màn hình chi phí cuốc xe và đánh giá tài xế 66

**Hình 47** Lớp UserActivity 67

**Hình 48** Lớp CallActivity 68

**Hình 49** Lớp TrackingActivity 69

**Hình 50** Cơ sở dữ liệu drivers 70

**Hình 51** Cơ sở dữ liệu users 71

**Hình 52** Cơ sở dữ liệu driver\_location 71

**Hình 53** Cơ sở dữ liệu pickup\_request 71

**Hình 54** Cơ sở dữ liệu rate\_driver 72

**Hình 55** Cơ sở dữ liệu tokens 72

***Hình 56*** *Cơ sở dữ liệu history\_user* 72

**Hình 57** Cơ sở dữ liệu history\_driver 72

**Hình 58** Màn hình chính của khách hàng 75

**Hình 59** Màn hình chính của tài xế 76

**Hình 60** Màn hình khi khách hàng tìm tài khế cho quốc xe 77

**Hình 61** Màn hình gọi cho tài xế 78

**Hình 62** Màn hình tài xế nhận được cuộc gọi 79

**Hình 63** Màn hình theo dõi tài xế của khách hàng 80

**Hình 64** Màn hình xử lí cuốc xe của tài xế 81

**Hình 65** Màn hình kết thúc cuốc xe của khách hàng 82

**Hình 66** Màn hình kết thúc cuốc xe của tài xế 83

# Danh mục bảng

[**Bảng 1** Đặc tả use case Đăng nhập 25](#_Toc9090975)

[**Bảng 2** Mô tả dữ liệu use case Đăng nhập 26](#_Toc9090976)

[**Bảng 3** Đặc tả use case Xác nhận số điện thoại 26](#_Toc9090977)

[**Bảng 4** Mô tả dữ liệu use case Xác thực số điện thoại 27](#_Toc9090978)

[**Bảng 5** Đặc tả use case Đặt cuốc xe 27](#_Toc9090979)

[**Bảng 6** Mô tả dữ liệu use case Đặt cuốc xe 30](#_Toc9090980)

[**Bảng 7** Mô tả dữ liệu đầu ra use case Đặt cuốc xe 30](#_Toc9090981)

[**Bảng 8** Đặc tả use case Chấp nhận cuốc xe 31](#_Toc9090982)

[**Bảng 9** Đặc tả use case Xem thông tin thanh toán (tác nhân khách hàng) 32](#_Toc9090983)

[**Bảng 10** Dữ liệu đầu ra use case Xem lịch sử chuyến đi 35](#_Toc9090984)

[**Bảng 6** Mô tả dữ liệu use case Cập nhật thông tin cá nhân 36](#_Toc9090985)

[**Bảng 12** Mô tả những phương thức chính của UserActivity 67](#_Toc9090986)

[**Bảng 13** Mô tả những phương thức chính của CallActivity 68](#_Toc9090987)

[**Bảng 14** Mô tả những phương thức chính của TrackingActivity 69](#_Toc9090988)

[**Bảng 15** Danh sách thư viện và công cụ sử dụng 73](#_Toc9090989)

[**Bảng 16** Thiết kế kiểm thử 83](#_Toc9090990)

# Danh mục các từ viết tắt

|  |  |  |
| --- | --- | --- |
| TMĐT | Thương mại điện tử. | |
| **UI/UX** | User Interface/User Experience.  Giao diện người dùng/Trải nghiệm người dùng. | |
| **REST** | Representational State Transfer.  Truyền đại diện trạng thái. | |
| **ER** | Entity Relationship  Thực thể liên kết. | |
| **API** | Application Programming Interface.  Giao diện lập trình ứng dụng. | |
| **BK** | Bách Khoa. | |
| **CNTT** | Công nghệ thông tin. |
| **ĐATN** | Đồ án tốt nghiệp. |
| **SV** | Sinh viên. |
|  |  | |

# Danh mục thuật ngữ

|  |  |  |  |
| --- | --- | --- | --- |
| TMĐT | | Thương mại điện tử. | |
| **Browser** | | Trình duyệt | |
| **Cache memory** | | Bộ nhớ đệm | |
| **E-commerce** | | Thương mại điện tử | |
| **Git** | | Hệ thống quản lý phiên bản mã nguồn. | | |
| **Branch** | | Nhánh quản lý nội dung, lịch sử mã nguồn độc lập (theo Git). | |
| **Issue** | | Vấn đề trên một Repository (theo Git). | |
| **Merge** | Ghép một nhánh vào nhánh khác (theo Git). | |
| **Repository** | Git: nơi lưu trữ quản lý, mã nguồn của dự án.  Mô hình đa lớp: nơi lưu trữ dữ liệu. | |
| **Asynchronous** | | Bất đồng bộ. | |
|  |  | |
|  | |  | |

# Giới thiệu đề tài

## Đặt vấn đề

Thời đại công nghệ thông tin bùng nổ mọi thứ đi thật nhanh và thật mạnh. Trước đây khi chưa có điện thoại hay mạng xã hội chúng ta gặp nhau để nói chuyện, tâm sự nhưng nay chỉ với chiếc điện thoại thông minh có kết nối Internet chúng ta có thể làm tất cả, chụp ảnh, quay phim, nhắn tin với bạn bè hay trò chuyện trực tiếp với nhau. Công nghệ khiến mọi thứ đều thay đổi, chúng ta làm việc với năng suất cao hơn, nhanh hơn và chính xác hơn. Các nhà máy sản suất ô tô có những chú robot lắp ráp, những chiếc xe ô tô tự lái có phần mềm điều khiển hay ti vi có thể tự động bật chỉ bằng lời nói. Công nghệ giúp chúng ta thay đổi mọi thứ và di chuyển cũng vậy.

Thay vì phải đi ra đường tìm kiếm một taxi nào đó, hay là vất vả đi một đoạn đường để chỉ tìm thấy một anh, một chú xe ôm. Chẳng những thế nhiều lúc chúng ta còn bị “chặt, chém” hét giá một cách thái quá cho chỉ những đoạn đường không đáng đến mức tiền như vậy. Những bác tài xế taxi, xe ôm phải đi lòng vòng trong thành phố, qua những con đường dài và đầy nắng hay đứng ở những trường học bệnh viện để chờ mong có một vị khách nào đó cần di chuyển và mình sẽ là người may mắn được chọn. Những người có xe, có thời gian rảnh rỗi muốn kiếm thêm thu nhập, những tài xế đi lại trên đường còn chỗ ngồi trên xe cho thể cho người khác đi nhờ. Quá nhiều thứ thúc đẩy để chúng ta tìm ra một cách di chuyển, tìm xe tốt hơn, hiệu quả hơn và rẻ hơn nữa so với cách truyền thống.

Trong bối cảnh công nghệ 4.0 phát triển, những ứng dụng gọi xe có thể hỗ trợ tài xế và khách hàng mọi lúc, mọi nơi. Rất nhiều ứng dụng taxi, xe ôm công nghệ đã dần trở nên phổ biến như: Grab, Uber...chỉ với một vài thao tác đơn giản chọn địa điểm bạn đến và nhấn đồng ý bạn có thể tìm được một tài xế với giá cước hợp lý. Mọi thứ đều tiện lợi và dễ dàng cho cả tài xế và khách hàng đã khiến lượng người dùng các ứng dụng gọi xe công nghệ này tăng trưởng với cấp số nhân và nhanh chóng trở thành những công ty startup tỉ đô trên toàn cầu.



**Hình 1** Số liệu thống kê về tốc độ tăng trưởng của chuyến xe của Grab [1]

Các ứng dụng gọi xe đã dần trở nên quen thuộc với người sử dụng, mọi người sử dụng nó hàng ngày cho việc đi học, đi làm hay thậm chí là đi du lịch. Mọi thứ thật tiện lợi và văn minh nhưng liệu có cách nào để người dùng có thể chọn những bác tài mình thích hay những anh những chị lái xe cho thể nâng thu nhập của mình cũng chỉ với quãng đường trước đó mà không phụ thuộc vào nhà phát triển ứng dụng. Hiện nay những ứng dụng như Uber, Be… đều tự động tìm kiếm lái xe và đưa ra một giá thành cho người sử dụng. Tài xế thường chỉ được hưởng 80% cuốc xe còn 20% là của hệ thống. Khách hàng không được chọn những lái xe mình yêu thích hay với giá cả mình có thể chấp nhận. Chính vì vậy tai sao chúng ta không phát triển một ứng dụng đặt xe hoàn toàn miễn phí có sự thỏa thuận giá cả giữa người dùng và tài xế. Tài xế có thể tự đưa giá mức chi phí cho từng km đường đi và khách hàng cũng có thể chọn xe mình thích. Người dùng thích đi xe sang như Audi, Mercedes sẽ phải trả chi phí cao hơn hoàn toàn tự nguyện, còn với chi phí rẻ hơn, tiết kiệm hơn những xe như Kia Morning có thể là một sự lựa chọn vừa đủ. Một ứng dụng đặt lợi ích của khách hàng và người lái xe lên cao nhất. Và đó có thể là một ứng dụng tuyệt vời.

## Mục tiêu và phạm vi đề tài

Tài liệu này đưa ra mô tả chi tiết cho quá trình phát triển ứng dụng gọi xe có giá cả thỏa thuận giữa người dùng và tài xế. Báo cáo mục đích và các tính năng của hệ thống, giao diện và dàng buộc của hệ thống cần thực hiện để phản ứng với những kích thích từ bên ngoài.

Ứng dụng gọi xe giữa khách hàng và tài xế với giá cả thỏa thuận được xây dựng có tính thực tế đáp ứng những mục tiêu sau:

* Tìm hiểu về công nghệ liên quan đến lập trình Android.
* Tìm hiểu dịch vụ lưu trữ giữa liệu thời gian thực Firebase như: Firebase Authentication, Firebase Realtime Database, Firebase Storage.
* Tìm hiểu về các API của Google phục vụ phát triển những ứng dụng sử dụng Google Map: Map API, Place API, Direction API.
* Áp dụng những kiến thức để xây dựng một ứng dụng tương đối hoàn chỉnh phục vụ việc gọi xe và giao tiếp giữa người dùng và lái xe theo thời gian thực. Việc phát triển ứng dụng tập chung vào tính năng gọi xe của người dùng và điều hướng di chuyển của lái xe tới vị trí điểm cần đến.
* Tìm hiểu phương hướng phát triển ứng dụng trong tương lai.

## Định hướng giải pháp

Những giải pháp cho việc phát triển ứng dụng gọi xe với thời gian ngắn đáp ứng được yêu cầu của đồ án tốt nghiệp:

* Xây dựng 2 ứng dụng cho tài xế và người dùng: Driver App và User App trên nền tảng hệ điều hành Android với ngôn ngữ chính là Java và IDE là Android Studio
* Áp dụng những kiến thức về Firebase để có một dịch vụ backend lưu trữ dữ liệu thời gian thực với hiệu năng cao và độ ổn định lớn và Firebase Authentication để xác thực đăng nhập với số điện thoại chính xác từ người sử dụng ứng dụng.
* Những API của Google phục vụ phát triển ứng dụng sử dụng bản đồ, Map API để hiện thị bản đồ và vị trí của người dùng, Place API dùng để tự động tìm kiếm những địa điểm trên bản đồ còn với Direction API giúp nhà phát triển xây dựng và chỉ dẫn đường đi, thời gian, khoảng cách và địa chỉ từ vị trí đầu tới vị trí cuối
* Thư viện GeoFire kết hợp với Firebase Realtime Database cho phép việc truy vấn và lưu trữ những khóa chính dựa trên vị trí địa lý.
* Sử dụng công cụ công cụ quản lý mã nguồn GitHub để lưu lại những bản sao của dữ án.
* Google Console giúp quản lý việc sử dụng những số lần gọi API cũng như những lối xảy ra trong quá trình sử dụng được hiện thị rõ số liệu theo biểu đồ hàng ngày.
* Sử dụng công cụ thiết kế và phân tích biểu đồ ca sử dụng, biểu đồ hoạt động là Astah Professional.

## Bố cục đồ án

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

Chương 2 trình bày quá trình khảo sát và phân tích yêu cầu của ứng dụng, cả về chức năng và phi chức năng hiện thực hóa qua những biểu đồ trực quan ca sử dụng, những đặc tả chi tiết về luồng hoạt động của một số chức năng và cả luồng hoạt động của ứng dụng.

Chương 3 trình bày một cách cụ thể những công nghệ chính được sử dụng trong việc xây dựng nên ứng dụng, những điểm chính, cách thiết lập và sử dụng căn bản của ưng dụng.

Chương 4 chi tiết cách triển khai của ứng dụng từ thiết kế kiến trúc, thiết kế giao diện đến thiết kế lớp và cơ sở dữ liệu. Việc xây dựng ứng dụng với chi tiết những thư viện và công cụ cũng được trình bày rõ ràng và cụ thể hơn về mục đích cũng như địa chỉ của thư viện hay công cụ ấy. Kết quả được đạt được mô tả cụ thể qua những hình ảnh về chức năng của hệ thống cả ứng dụng về phía khách hàng và phí tài xế. Việc kiểm thử là một việc vô cùng quan trọng của bất cứ ứng dùng nào cũng được trình bày cụ thể và chi tiết trong phần này của đồ án quá những testcase.

Chương 5 trình bày các giải pháp và đóng góp nổi bật của đồ án tốt nghiệp, mô tả những điểm mới và kết quả trong quá trình phát triển hệ thống đặt xe, những điểm được và chưa được của hệ thống và không thể thiếu những giải pháp quan trong để hình thành nên ứng dụng qua những công nghệ nổi bật.

Chương 6 sẽ trình bày lại những kết luận trong quá trình thực hiện đồ án, hướng phát triển trong tương lai, mong muốn của tác giả đồ án và những cải tiến của đồ án trong tương lai gần.

# ­­Khảo sát và phân tích yêu cầu

## Khảo sát hiện trạng

Những ứng dụng gọi xe hiện nay đề có những chức năng cơ bản mà chúng ta cần quan tâm đến, khảo sát giúp nhà phát triển ứng dụng có cách nhìn rõ hơn về những yêu cầu sử dụng của khách hàng, những tính năng cơ bản, những thứ được và chưa được để áp dụng vào bài toán thực tế. Một ứng dụng taxi công nghệ thường có những tính năng nổi bật giúp việc thao tác trở lên đơn giản cho cả người dùng và tài xế. Người dùng tìm kiếm địa điểm mình cần đến chọn lái xe mình thích và hệ thống sẽ gọi lái xe giúp bạn. Việc theo dõi thời gian và vị trí của tài xế giúp bạn nắm rõ và tiết kiệm được thời gian quý báu của mình. Tài xế cũng nhanh chóng nhận được thông tin cuộc gọi từ khách hàng, với những thao tác cơ bản, người dùng được định hướng tới vị trí của khách và bắt đầu di chuyển tới một vị trí cố định của khách. Việc áp dụng hình thức taxi công nghệ hiện nay không chỉ là giúp mọi người di chuyển từ vị trí này đến vị trí kia mà nó còn len lỏi trong nhiều công việc khác. Sau đây là một số hình thức triển khai phổ biến của những ứng dụng gọi xe công nghệ đặc biệt này tiểu biểu là của ứng dụng Grab.

* Grab Taxi là một hệ thống giúp kết nối những tài xế lái xe với khách hàng với việc tính giá tiền hiện thị trên đồng hồ tính phí của xe.
* Grab Car một dịch vụ gọi xe dành cho ô tô 4 đến 7 chỗ với nhiều tính năng tốt và tiện dụng, giá tiền được hiện thị ngay trên ứng dụng và khách hàng chỉ phải trả đúng số tiền này.
* Grab Bike ứng dụng gọi xe công nghệ hiện nay đã xuất hiện thường xuyên hơn trên phương tiện là xe máy. Di chuyển nhanh, cước phí rẻ là những ưu điểm của hình thức di chuyển này.
* Grab Express việc giao hàng và nhận hàng trở lên dễ dàng hơn bao giờ hết. Khách hàng có thể tự do gửi và nhận hàng mà không phải chờ đợi quá lâu.
* Grab Share bạn có thể đi chung xe với người khác chứ, nếu đồng ý bạn có thể giảm 30% đấy. Đó chính là ưu điểm nổi bật của ứng dụng này, trò chuyện với người đi chung sẽ vui vẻ hơn và thú vị hơn.
* Grab Food khi bạn thèm ăn một món gì đó chắc bạn lại phải xách xe lên và đi đến một quán ăn và thưởng thức nhưng nay khi có mặt của ứng dụng tiện ích này. Bạn chỉ phải ở nhà, văn phòng hay đâu đó chạm vào món ăn mình yêu thích và nó sẽ xuất hiện nhanh ngay bên bạn.



**Hình 2** Ứng dụng gọi xe Grab với nhiều hình thức sử dụng [2]

Và còn nhiều hình thức khác nữa chứ. Gọi xe công nghệ đang len lỏi vào nhiều lĩnh vực của đời sống với nhiều tính năng hướng tới khách hàng. Các chức năng cần phải dễ dàng sử dụng, giao diện trực quan, đẹp mắt và thân thiện tới người dùng. Các ứng dụng gọi xe đều tập chung phát triển sao cho khách hàng và tài xế có thể thương tác một cách chuẩn xác nhất, mọi thứ đề trong thời gian thực. Chỉ trong một thơi gian ngắn thôi để khách hàng và tài xế đều không phải chờ đợi lâu. Việc tối ưu hóa thời gian gọi xe giúp những ứng dụng sử dụng tốt hơn và người dùng yêu thích hơn. Ứng dụng gọi xe theo giá cả thỏa thuận giữa người dùng và tài xế cũng sẽ không khác ý tưởng đó.

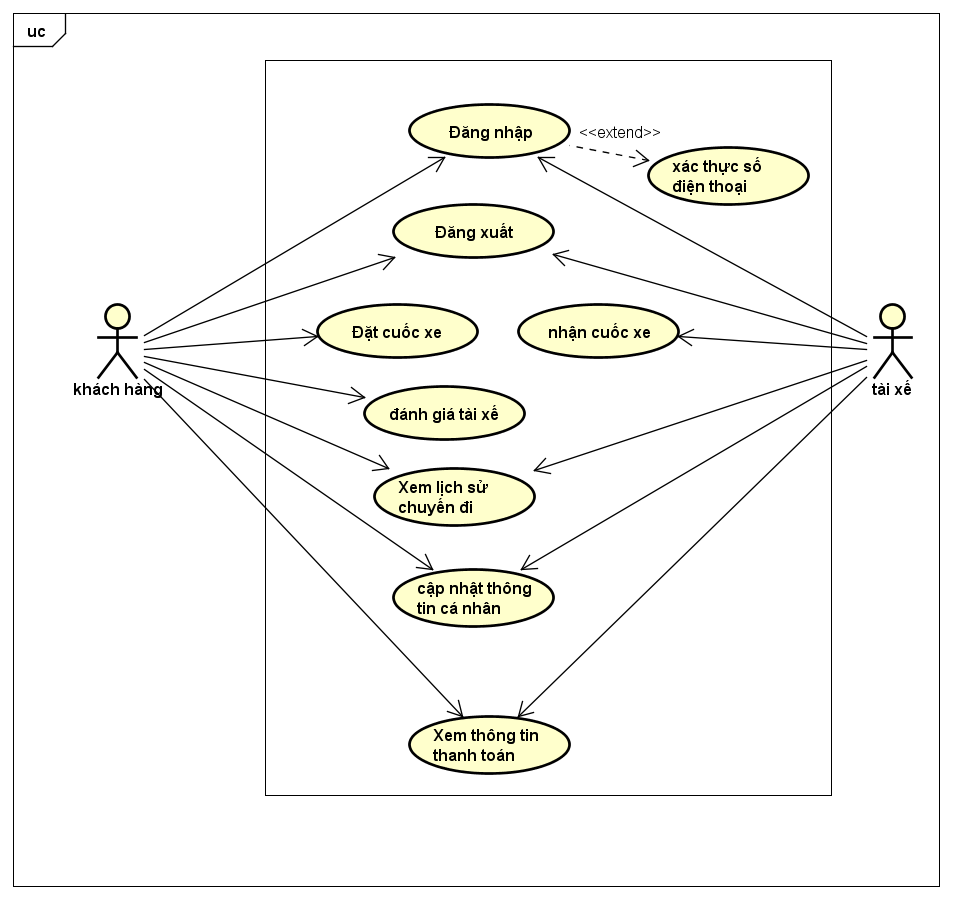
## Tổng quan chức năng

Hệ thống xây dựng hai ứng dụng dành riêng cho tài xế và khách hàng. Khách hàng của người muốn gọi xe để chi chuyển còn tài xế là chủ phương tiện muốn tìm kiếm người dùng của mình. Hai tác nhân chính này sẽ được thể hiện và mô tả các chức năng qua những biểu đồ ca sử dụng sau đây.

### Biểu đồ use case tổng quan

Biểu đồ use case tổng quan mô tả ngăn gọn những chức năng chính của người sử dụng và tài xế như sau:

* “Đăng nhập”: Cả khách hàng và tài xế đều cần đăng nhập vào hệ thống với một số điện thoại và email hợp lệ, về phía tài xế những thông tin thêm về chủ xe được yêu cầu như: tên loại xe, biển số xe và chi phí tính theo km mà tài xế đưa ra.
* “Đăng xuất”: Cả hai ứng dụng Driver App và User App đều cho phép lưu lại tài khoản của khách hàng sau lần đăng nhập, nếu muốn thoát khỏi ứng dụng người sử dụng app hãy đăng xuất và bắt đầu lại công đoạn đăng nhập.
* “Đặt cuốc xe”: Chức năng quan trọng nhất về phía khách hàng, tìm kiếm địa điểm, chọn chủ xe và bắt đầu chuyến đi.
* “Nhận cuốc xe”: Chức năng dành riêng cho phía tài xế, chấp nhận cuộc gọi từ phía người dùng User App, di chuyển đến vị trí của người dùng và bắt đầu chuyến đi một cách nhanh chóng.
* “Xem thông tin thanh toán”: Tài xế chọn chuyến đi kết thúc một số tiền được tính theo chi phí của lái xe sẽ được hiện nên cả phía ứng dụng của tài xế và khách hàng. Khách hàng chỉ cần trả số tiền đúng với ứng dụng đưa ra.
* “Đánh giá tài xế”: Chuyến đi kết thúc, khách hàng có thể đánh giá người lái xe của mình từ 1 – 5 theo thang điểm với 1 là rất tệ còn 5 thì tuyệt vời. Người dùng chỉ cần nhấn vào số sao tương ứng trên màn hình để đánh giá tài xế.
* “Xem lịch sử chuyến đi”: Lịch sử của cuốc xe khi tài xế nhấn bắt đầu đến khi kết thúc có các thông tin như: ngày, giờ của cuốc xe, thông tin địa điểm bắt đầu, địa điểm kết thúc cũng như chi phí của cuốc xe đó.
* “Cập nhật thông tin cá nhân”: Mỗi người dùng khi đăng nhập ứng dụng đều phải có các thông tin bắt buộc như: tên, email và số điện thoại. Khi cần thay đổi thông tin người sử dụng ứng dụng hãy chọn cập nhật ở đây người dùng có thể thay đổi cả ảnh đại điện của bản thân mình.

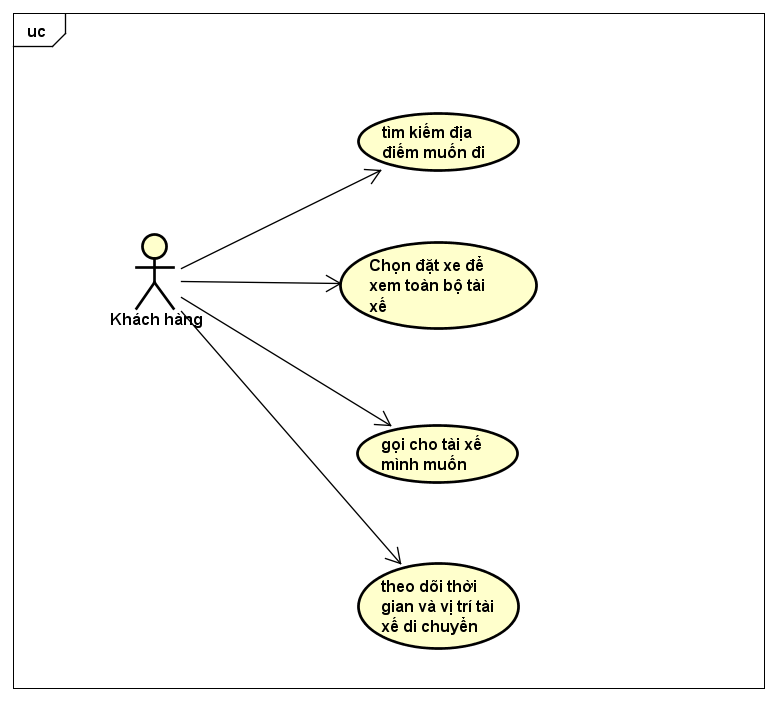


**Hình 3** Biểu đồ use case tổng quan cho ứng dụng

### Biểu đồ use case phân rã cho chức năng đặt cuốc xe

Ca sử dụng này mô tả các chức năng khi đặt cuốc xe của khách hàng:

* Khách hàng trước tiên phải tìm kiếm một địa điểm mình muốn đến, hệ thống sẽ tự động gợi ý cho khách chỉ với những kí tự đầu tiên
* Chọn đặt cuộc xe khách hàng sẽ được xem toàn bộ những tài xế của mình trong phạm vi 3 km quang vị trí của khách hàng mà có thể đưa khách hàng tới vị trí mong muốn, thông tin về tài xế cũng được hiện thị như: ảnh đại diện, tên, loại xe, chi phí và cả đánh giá của những người dùng trước đó về lái xe.
* Để xác nhận chọn những tài xế mình muốn và phù hợp với chi phí chuyến đi, khách hàng nhấn vào lái xe và hệ thống sẽ gọi cho tài xế mà khách hàng muốn cùng di chuyển.
* Nếu tài xế đồng ý người dùng có thể chuyển sang một giao diện mới theo dõi đây đầy đủ thời gian và vị trí tài xế di chuyển từ lúc tới vị trí của khách hàng cho đến khi cuốc xe kết thúc.

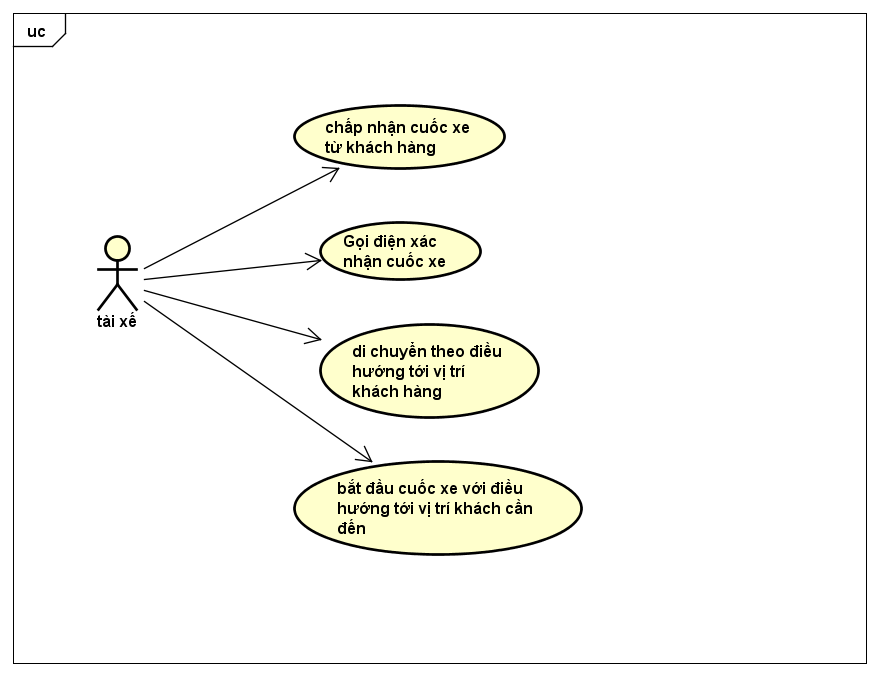


**Hình 4** Biểu đồ phân rã chức năng đặt cuốc xe

### Biểu đồ use case phân rã cho chức năng nhận cuốc xe

Ca sử dụng này mô tả chức năng chấp nhận cuốc xe của tài xế:

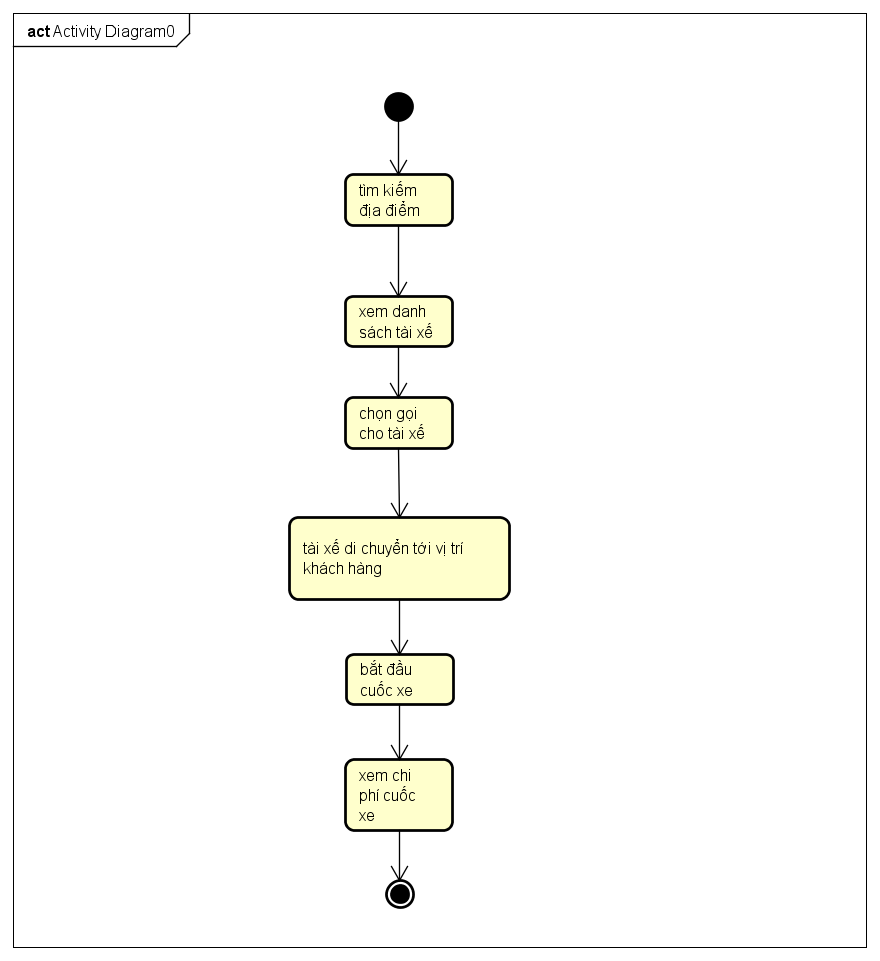
* Khi nhận được cuộc gọi từ khách hàng, tài xế có 30 (s) để chấp nhận cuộc gọi nếu đồng ý tài xế có thể di chuyển đến vị trí của khách hàng.
* Để đảm bảo chắc chắn rằng cuộc xe được đặt thành công, tài xế sẽ gọi cho khách hàng để xác nhận.
* Có sự đồng ý từ khách hàng, tài xế sẽ di chuyển tới bên khách hàng và bắt đầu cuốc xe
* Chọn điều hướng để biết chính xác đường đi đến vị trí khách hàng muốn đến, tài xế đưa khách đến vị trí cuối và kết thúc một hành trình cuốc xe.



**Hình 5** Biểu đồ phân rã chức năng chấp nhận cuốc xe

### Quy trình nghiệp vụ

Quy trình nghiệp vụ của hệ thống mô tả quá trình từ lúc bắt đầu tìm kiếm của khách hàng tới lúc tài xế bấm kết thúc chuyến đi. Quy trình mô tả một cách đầy đủ những giai đoạn chính của hệ thống.



**Hình 6** Biểu đồ quy trình nghiệp vụ của một cuốc xe

## Đặc tả chức năng

### Đặc tả use case Đăng nhập

Bảng dưới đặc tả chi tiết các yếu tố của use case “Đăng nhập”:

**Bảng 1** Đặc tả use case Đăng nhập

|  |  |  |  |
| --- | --- | --- | --- |
| Mã Use case | UC001 | Tên Use case | Đăng nhập |
| Tác nhân | Khách hàng | | |
| Tiền điều kiện | Không | | |
| Luồng sự kiện chính  (Thành công) | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | |  | Khách hàng | Thực hiện mở ứng dụng User App | |  | Hệ thống | Hiển thị giao diện đăng nhập | |  | Khách hàng | Nhập tên, email và số điện thoại (mô tả phía dưới \*) | |  | Khách hàng | Yêu cầu đăng nhập | |  | Hệ thống | Kiểm tra xem khách đã nhập các trường bắt buộc nhập hay chưa | |  | Hệ thống | Kiểm tra tên và email và số điện thoại có hợp lệ do khách nhập trong hệ thống hay không | |  | Hệ thống | Gọi use case “xác thực số điện thoại” kèm thông tin đăng nhập | | | |
| Luồng sự kiện thay thế | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 5a. | Hệ thống | Thông báo lỗi: Cần nhập các trường bắt buộc nhập nếu khách nhập thiếu | | 6a. | Hệ thống | Thông báo lỗi: email và số điện thoại không hợp lệ khách hàng cần nhập email và số điện thoại đúng | | | |
| Hậu điều kiện | Không | | |

\* Dữ liệu đầu vào của thông tin cá nhân gồm các trường dữ liệu sau:

**Bảng 2** Mô tả dữ liệu use case Đăng nhập

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Trường dữ liệu** | **Mô tả** | **Bắt buộc?** | **Điều kiện hợp lệ** | **Ví dụ** |
|  | Tên |  | Có |  | doannd |
|  | Email |  | Có |  | nguyendinhdoanjdk@gmail.com |
|  | Số điện thoại |  | Có |  | 0395616595 |

### Đặc tả use case Xác nhận số điện thoại

**Bảng 3** Đặc tả use case Xác nhận số điện thoại

|  |  |  |  |
| --- | --- | --- | --- |
| Mã Use case | UC002 | Tên Use case | Xác nhận số điện thoại |
| Tác nhân | Khách hàng | | |
| Tiền điều kiện | Khách hàng đăng nhập thành công | | |
| Luồng sự kiện chính  (Thành công) | | STT | Thực hiện bởi | Hành động | | --- | --- | --- | |  | Hệ thống | Gửi cho khách hàng một mã số có 6 chữ số | |  | Khách hàng | Nhập mã số từ tin nhắn vào ô mã xác thực (mô tả phí dư) | |  | Khách | Chọn xác thực số điện thoại | |  | Hệ thống | Kiểm tra mã xác thực có đúng hay không | |  | Hệ thống | Kiểm tra xem mã xác thực có trong thời gian timeout hay không. | |  | Hệ thống | Lưu thông tin tài khoản và chuyển người dùng sang màn chính | | | |
| Luồng sự kiện thay thế | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 4a. | Hệ thống | Thông báo lỗi nếu mã xác thực không hợp lệ | | 6a.1 | Hệ thống | Thông báo lỗi nếu như mã xác thực đã hết thời gian timeout | | 6a.2 | Hệ thống | Thông báo lỗi nếu có bất cứ điều gì không hợp lệ khi thực hiện xác thực. | | | |
| Hậu điều kiện | Không | | |

\* Dữ liệu đầu vào của xác thực số điện thoại gồm các trường dữ liệu sau:

**Bảng 4** Mô tả dữ liệu use case Xác thực số điện thoại

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Trường dữ liệu** | **Mô tả** | **Bắt buộc?** | **Điều kiện hợp lệ** | **Ví dụ** |
|  | Mã xác thực | Có 6 chữ số | Có | Số điện thoại hợp lệ | 147384 |

### Đặc tả use case Đặt cuốc xe

**Bảng 5** Đặc tả use case Đặt cuốc xe

|  |  |  |  |
| --- | --- | --- | --- |
| Mã Use case | UC003 | Tên Use case | Đặt cuốc xe |
| Tác nhân | Khách hàng | | |
| Tiền điều kiện | Không | | |
| Luồng sự kiện chính  (Thành công) | | STT | Thực hiện bởi | Hành động | | --- | --- | --- | |  | Khách hàng | Chọn tìm điếm địa điểm | |  | Hệ thống | Hiển thị giao diện tìm kiếm | |  | Khách | Nhập thông tin địa điểm cần tìm kiếm (mô tả phía dưới \*) | |  | Hệ thống | Gợi ý cho khách hàng địa điểm từ những kí tự đầu tiên | |  | Khách hàng | Chọn địa điểm mình cần di chuyển tới | |  | Hệ thống | Hiện thị một đường đi từ vị trí khách hàng tới điểm đến trên bản đồ | |  | Khách hàng | Chọn đặt cuốc xe | |  | Hệ thống | Tìm kiếm trong phạm vi 3 km hiện thị toàn bộ những tài xế có khả năng nhận cuốc xe của khách hàng với thông tin đầy đủ (mô tả phía dưới \*\*) | |  | Khách hàng | Chọn tài xế mình thích với giá cả phù hợp | |  | Hệ thống | Gửi một thông báo tới tài xế mà khách hàng chọn | |  | Hệ thống | Chuyển sang màn hình giao diện gọi cho tài xế | |  | Khách hàng | Chờ đợi sự đồng ý từ tài xế về chuyến đi (tối đa không quá 30s) | |  | Hệ thống | Nếu tài xế đồng ý hệ thống hiện thị giao diện theo dõi vị trí và thời gian mà tài xế tới vị trí của khách hàng và thông báo một notification cho khách hàng | |  | Hệ thống | Hiện thị quá trình di chuyển của tài xế một cách tương đối trên bàn đồ cùng thời gian mà tài xế gần tới nơi. | |  | Hệ thống | Khi tài xế tới vị trí của khách hàng trong bán kính 50m hệ thống hiện thị thông báo trên giao diện và gửi một notification thông báo tài xế đã đến cho khách hàng. | |  | Khách hàng | Lên xe của tài xế và bắt đầu cuốc xe khi tài xế nhấn bắt đầu chuyến đi | |  | Hệ thống | Hiện thị quá trình di chuyển của tài xế tới vị trí khách hàng muốn đến trên bản đồ cùng thời gian dự kiến | |  | Hệ thống | Khi tài xế chọn dừng cuốc xe, hệ thống gửi một notification thông báo tới người dùng | |  | Hệ thống | Chuyển sang giao diện xem chi phí cuốc xe và đánh giá người tài xế | | | |
| Luồng sự kiện thay thế | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 8a. | Hệ thống | Thông báo không tìm thấy tài xế hợp lệ quanh vị trí của khách hàng | | 12a. | Khách hàng | Hủy bỏ cuộc gọi tới khách hàng | | 14a | Khách hàng | Hủy bỏ cuốc xe | | | |
| Hậu điều kiện | Không | | |

\* Dữ liệu đầu vào khi tìm kiếm:

**Bảng 6** Mô tả dữ liệu use case Đặt cuốc xe

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Trường dữ liệu** | **Mô tả** | **Bắt buộc?** | **Điều kiện hợp lệ** | **Ví dụ** |
|  | Từ khóa nhập vào | Là địa điểm người dùng cần di chuyển đến | Có |  | Chỉ gần gõ một từ đầu tiên, hoặc gần đúng với địa điểm. |

\*\* Dữ liệu đầu ra khi hiển thị danh sách tài xế, dưới đây là các thông tin của một tài xế.

**Bảng 7** Mô tả dữ liệu đầu ra use case Đặt cuốc xe

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Trường dữ liệu** | **Mô tả** | **Định dạng hiển thị** | **Ví dụ** |
|  | Tên |  | Dạng chuỗi | Nguyễn Đình Đăng |
|  | Ảnh đại diện | Một hình ảnh được bo tròn với kích thức 50 x 50 | Ảnh định dạng PNG | Hình ảnh được hiện thị trên giao diện ứng dụng, nếu người dùng chưa thay đổi ảnh, một hình ảnh mặc định sẽ được hiện thị |
|  | Số sao | Một con số trung bình đánh giá từ phía khách hàng về cuốc xe kèm theo một icon hình ngôi sao nhỏ màu vàng | Số + hình ảnh | Chữ số 5 cộng với hình ngôi sao (5 là điểm trung bình lớn nhất mà tài xế có thể nhận được sau nhiều chuyến đi) |

### Đặc tả use case Chấp nhận cuốc xe

**Bảng 8** Đặc tả use case Chấp nhận cuốc xe

|  |  |  |  |
| --- | --- | --- | --- |
| Mã Use case | UC004 | Tên Use case | Chấp nhận cuốc xe |
| Tác nhân | Tài xế | | |
| Tiền điều kiện | Không | | |
| Luồng sự kiện chính  (Thành công) | | STT | Thực hiện bởi | Hành động | | --- | --- | --- | |  | Hệ thống | Gửi một notification với ứng dụng của tài xế | |  | Hệ thống | Hiện thị giao diện với thông tin địa điểm khách hàng muốn đến cũng như khoảng cách và thời gian dự diện (tài xế có tối đa 30s) để chấp nhận lời đề nghị của khách hàng. | |  | Tài xế | Chọn “Accept” để đồng ý cuốc xe | |  | Hệ thống | Hiện thị giao diện chính mô tả đường đi từ vị trí của tài xế đến vị trí của khách hàng, vị trí của khách hàng được mô tả quanh một hình tròn có bán kính 50m | |  | Tài xế | Gọi điện cho khách hàng để chắc chắn cuốc xe được đặt thành công | |  | Tài xế | Chọn điều hướng để biết chính xác đường đi tới vị trí của khách hàng | |  | Hệ thống | Hiện thị giao diện ứng dụng Google Map với đường đi chi tiết tới vị trí của khách hàng | |  | Tài xế | Di chuyển tới vị trí của khách hàng | |  | Hệ thống | Khi tới bán kính 50m của khách hàng, hệ thống chuyển sang giao diện điều hướng có thể bắt đầu cuốc xe | |  | Tài xế | Gọi cho khách hàng để khách hàng và bản thân biết rõ vị trí của nhau | |  | Tài xế | Nếu khách hàng tìm được vị trí của bạn, tài xế chọn bắt đầu chuyến đi | |  | Tài xế | Chọn điều hướng tới vị trí mà khách hàng muốn đến | |  | Hệ thống | Hiện thị Google Map dẫn đường chi tiết tới địa điểm của khách hàng muốn đến | |  | Tài xế | Khi đến vị trí cuối cuốc xe tài xế chọn kết thúc cuốc xe | | | |
| Luồng sự kiện thay thế | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 5a. | Tài xế | Hủy bỏ cuốc xe | | 6a | Hệ thống | Hủy bỏ cuốc xe | | | |
| Hậu điều kiện | Không | | |

### Đặc tả use case Xem thông tin thanh toán

**Bảng 9** Đặc tả use case Xem thông tin thanh toán (tác nhân khách hàng)

|  |  |  |  |
| --- | --- | --- | --- |
| Mã Use case | UC005 | Tên Use case | Xem thông tin thanh toán |
| Tác nhân | Khách hàng | | |
| Tiền điều kiện | Không | | |
| Luồng sự kiện chính  (Thành công) | | STT | Thực hiện bởi | Hành động | | --- | --- | --- | |  | Hệ thống | Hiện thị giao diện xem thông tin thành toán | |  | Hệ thống | Tính toán cước của chuyến đi từ vị trí bắt đầu đến vị trí kết thúc | |  | Hệ thống | Hiện thị chi tiết thông tin về chi phí và thời gian hiện tài (mô tả bên dưới \*) | |  | Khách hàng | Xem thông tin số tiền và thanh toán với tài xế | |  | Khách hàng | Chọn thoát để quay về giao diện chính của hệ thống | | | |
| Luồng sự kiện thay thế | Không | | |
| Hậu điều kiện | Không | | |

\* Dữ liệu đầu ra khi hiện thị thông tin thanh toán

**Bảng 10** Mô tả dữ liệu đầu ra use case Xem thông tin thanh toán

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Trường dữ liệu** | **Mô tả** | **Định dạng hiển thị** | **Ví dụ** |
|  | Thời gian | Được tính theo thời gian hiện tại | Hh:mm, dd/MM/yyyy | 10:24, 15/5/2019 |
|  | Chí phí cuốc xe | Số tiền khách hàng phải thanh toán cho cuốc xe | VND + số tiền + K | VND 100K |

### Đặc tả use case Đánh giá tài xế

**Bảng 11** Đặc tả use case Đánh giá tài xế

|  |  |  |  |
| --- | --- | --- | --- |
| Mã Use case | UC006 | Tên Use case | Đánh giá tài xế |
| Tác nhân | Khách hàng | | |
| Tiền điều kiện | Sau kết thúc cuốc xe | | |
| Luồng sự kiện chính  (Thành công) | | STT | Thực hiện bởi | Hành động | | --- | --- | --- | |  | Hệ thống | Hiện thị giao diện đánh giá tài xế | |  | Khách hàng | Đánh giá tài xế theo bằng cách chọn từ 1 – 5 sao tương ứng từ rất tệ đến rất tốt | |  | Khách hàng | Chọn thoát khỏi giao diện đánh giá tài xế | |  | Hệ thống | Hiện thị giao diện chính của ứng dụng | | | |
| Luồng sự kiện thay thế | Không | | |
| Hậu điều kiện | Không | | |

### Đặc tả use case Xem lịch sử chuyến đi

**Bảng 12** Đặc tả use case Xem lịch sử chuyến đi

|  |  |  |  |
| --- | --- | --- | --- |
| Mã Use case | UC007 | Tên Use case | Xem lịch sử chuyến đi |
| Tác nhân | Khách hàng | | |
| Tiền điều kiện | Không | | |
| Luồng sự kiện chính  (Thành công) | | STT | Thực hiện bởi | Hành động | | --- | --- | --- | |  | Khách hàng | Chọn chức năng xem lịch sử chuyến đi | |  | Hệ thống | Hiện thị giao diện xem lịch sử chuyến đi của bản (Mô tả dữ liệu đầu ra \*) | |  | Khách hàng | Lựa chọn thoát khỏi giao diện | |  | Hệ thống | Hiện thị giao diện chính của ứng dụng | | | |
| Luồng sự kiện thay thế | Không | | |
| Hậu điều kiện | Không | | |

\* Dữ liệu đầu ra khi xem lịch sử chuyến đi:

**Bảng 10** Dữ liệu đầu ra use case Xem lịch sử chuyến đi

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Trường dữ liệu** | **Mô tả** | **Định dạng hiển thị** | **Ví dụ** |
| 1 | Thời gian | Được tính theo thời gian hiện tại | Hh:mm, dd/MM/yyyy | 10:24, 15/5/2019 |
| 2. | Chí phí cuốc xe | Số tiền khách hàng phải thanh toán cho cuốc xe | VND + số tiền + K | VND 100K |
| 3. | Địa điểm bắt đầu | Vị trí bắt đầu cuốc xe từ lúc tài xế nhấn bắt đầu | Text | Hồ Hoàn Kiếm, Hà Nội |
| 4. | Địa điểm kết thúc | Vị trí kết thúc cuốc xe khi tài xế nhấn kết thúc | Text | Đại học Bách Khoa Hà Nội, 1 Đại Cồ Việt |

### Đặc tả use case Cập nhật thông tin cá nhân

|  |  |  |  |
| --- | --- | --- | --- |
| Mã Use case | UC008 | Tên Use case | Cập nhật thông tin cá nhân |
| Tác nhân | Khách hàng | | |
| Tiền điều kiện | Không | | |
| Luồng sự kiện chính  (Thành công) | | STT | Thực hiện bởi | Hành động | | --- | --- | --- | |  | Khách hàng | Chọn cập nhật thông tin cá nhân | |  | Hệ thống | Hiện thị giao diện cập nhật thông tin cá nhân cho khách hàng | |  | Khách hàng | Chọn và nhập thông tin cần thiết để cập nhật (Mô tả dữ liệu đầu vào \*) | |  | Khách hàng | Chọn cập nhật thông tin cá nhân | |  | Hệ thống | Kiểm tra thông tin có hợp lệ hay không | |  | Hệ thống | Nếu thông tin cập nhật hợp lệ thông báo thành công với khách hàng | | | |
| Luồng sự kiện thay thế | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 6a. | Hệ thống | Thông báo cập nhật thông tin cá nhân không thành công | | | |
| Hậu điều kiện | Không | | |

**Bảng 6** Mô tả dữ liệu use case Cập nhật thông tin cá nhân

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Trường dữ liệu** | **Mô tả** | **Bắt buộc?** | **Điều kiện hợp lệ** | **Ví dụ** |
|  | Ảnh đại diện | Ảnh đại diện cá nhân của người dùng | Không |  | Chọn một bức ảnh từ điện thoại và hệ thống sẽ tài lên giao diện ứng dụng |
|  | Tên | Tên của người dùng | Có |  | Nguyễn Đình Đoàn |
|  | Email | Email cá nhân của người dùng | Có |  | nguyendinhdoanjdk@gmail.com |
|  | Số điện thoại | Số điện thoại cá nhân của người dùng | Có |  | 0395616595 |

## Yêu cầu phi chức năng

Ứng dung phải đáp ứng một số yêu cầu phi chức năng giúp hệ thống hoạt động ổn đinh, mượt mà nhất có thể:

* Hệ thống hoạt động ổn định 24/24 giờ, người dùng và tài xế có thể tương tác bất cứ lúc nào
* Cả hai ứng dụng Driver App và User App đều phải hoạt động trên thời gian thực đảm bảo tương tác ổn định và độ trễ thấp
* Giao diện được thiết kế thân thiện với người dùng và dễ thao tác trong quá trình di chuyển
* Các ứng dụng được cài đặt trên hệ điều hành Android 4.0.3 trở nên
* Độ bảo tương đối tốt đảm bảo người dùng phải đăng nhập mới có thể sử dụng

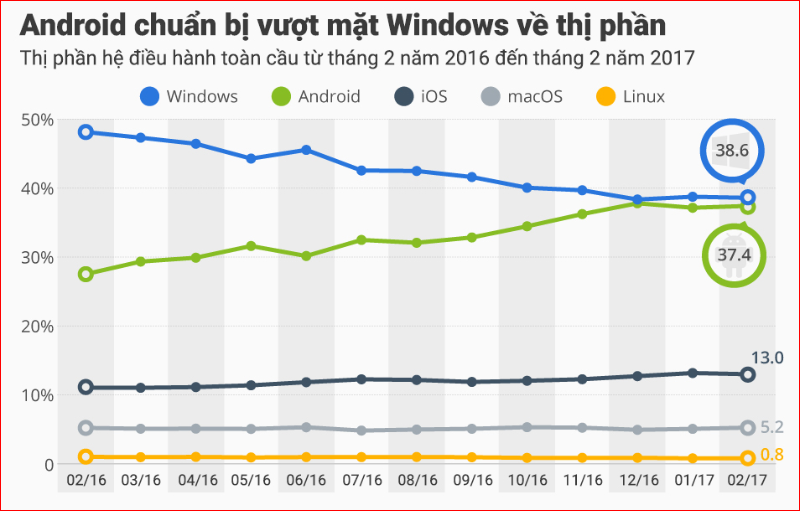
# Công nghệ sử dụng

## Giới thiệu cơ bản về nền tảng Android

Cả hai ứng dụng của tài xế và khách hàng đều được phát triển dựa trên hệ điều hành Android, chúng ta sẽ cùng tìm hiểu những đặc điểm chính của hệ điều hành này cũng như ưu điểm khi phát triển ứng dụng trên nền tảng di động phổ biến nhất hiện nay.

### Nền tảng Android

Android đã và đang là một trong những hệ điều hành di động phổ biến nhất trên thế giới. Thị phần ngày càng tăng của Android đã phản ánh lượng người dùng đồ sộ được phát triển dựa trên nền tảng Linux này. Chỉ mới ra mắt vào năm 2007 nhưng Android đã nhanh chóng trở vượt mặt iOS, Linux, macOS để trở thành hệ điều hành phổ biến thứ hai trên thế giới và được dự báo chỉ một thời gian ngắn nữa thôi Android sẽ vượt Window để trở thành hệ điều hành thống trị toàn cầu.



**Hình 7** Số liệu thống kê về thị phần hệ điều hành toàn cầu năm 2016 [3]

Android có mã nguồn mở và được Google phát hành theo giấy phép của Apache. Chính mã nguồn mở cùng với một giấy phép không có nhiều ràng buộc đã cho phép những nhà phát triển thiết bị, mạng di động và các lập trình viên nhiệt huyết được điều chỉnh và phân phối hệ điều hành này một cách tự do. Ngoài ra Android còn có một cộng động lập trình viên đông đảo chuyên viết các ứng dụng để mở rộng chức năng của thiết bị bằng ngôn ngữ Java và Kotlin. Chính những yếu tố này đã giúp Android trở thành nền tảng điện thoại thông minh phổ biến nhất trên thế giới. Chỉ vào tháng 10 năm 2012 sau 5 năm hình thành Android đã có khoảng 700.000 ứng dụng và số lượt tải trên chợ ứng dụng khi đó đã lên đến 25 tỷ lượt.[4] Qua tất cả chúng ta thấy rằng việc phát triển một ứng dụng trên nền tảng này là hoàn toàn phù hợp.

### Ưu điểm khi phát triển trên nền tảng Android

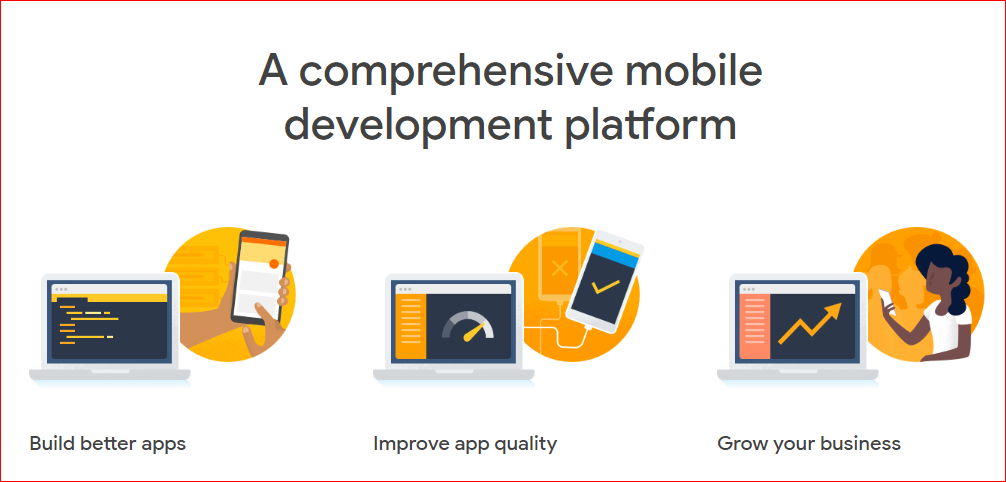
Hiện này có nhiều hệ điều hành hỗ trợ việc phát triển ứng dụng di động như macOS, Window Phone, Blackberry. Có một câu hỏi mà nhiều lập trình viên thương thắc mắc khi phát triển ứng dụng di động nên phát triển trên nền tảng nào. Có lẽ câu hỏi trên sẽ được trả lời một cách dễ dàng đối với những công ty lớn, đồ sộ và có nguồn nhân lực rồi dào. Nhưng với những công ty nhỏ hơn thì sao nhỉ, công ty startup hay chỉ là những sinh viên làm đồ án tốt nghiệp. Nền tảng cho phép phát triển mọi thứ được nhanh, tốn ít chi phí và tin cậy cao. Đó chính là Android, sau đây là những lý do chính khiến Android luông được chọn đầu tiên khi phát triển một ứng dụng:

* Thị phần lớn, chắc có lẽ đây là nguyên nhân đầu tiên khi chúng ta phát triển ứng dụng trên hệ điều hành này, năm 2015 Android đã chiếm đến 75% thị phần của những ứng dụng di động. Thị phần nhiều chứng tỏ nhiều người sử dụng, khi đó việc phát hành ứng dụng cũng có thể nhận được nhiều người sử dụng hơn và nhiều lượt tải hơn và tất nhiên cơ hội thành công cũng sẽ cao hơn so với nền tảng khác.
* Khả năng sinh lời cao, nếu ứng dụng của chúng ta cho phép người dùng tải miễn phí, vậy lợi nhuận ở đâu ra, câu trả lời chính là từ quảng cáo. Càng nhiều người sử dụng và xem quảng cáo từ ứng dụng của bạn, bạn càng có thể kiểm nhiều tiền hơn. Chẳng những thế những chức năng Premium trong ứng dụng có trả phí cũng là một cách kiểm tiền tốt từ việc phát triển ứng dụng trên nền tảng này.
* Rào cản thấp, khi việc phát triển ứng dụng iOS phải được thực hiện trên máy tính Mac thì Android lại ngoại nệ, bạn có thể lập trình nó trên macOS, Window hay là Linux. Mọi thứ thật dễ dàng nhất là với sinh viên không đủ kinh phí để sắm một chiếc Macbook. Chẳng những thế việc phát hành ứng dụng với chi phí cũng rẻ hơn rất nhiều chỉ có 25$ và chỉ thanh toán một lần còn Apple App Store thì khác đó là 99$ một năm. Vì vậy việc phát triển ứng dụng trên hệ điều hành này luôn đó lựa chọn ưu tiền hàng đầu.
* Tiếp theo đó là Java, một ngôn ngữ lập trình phổ biến và được chứng minh là vô cùng mạnh mẽ và ổn định, thật may mắn đó cũng chính là một ngôn ngữ để phát triển ứng dụng Android hiện nay bên cạnh Kotlin. Java còn rất quen thuộc với giới học sinh viên và lập trình viên, những ứng dụng, website có độ bảo mật tốt thường được phát triển bởi ngôn ngữ này. Cộng đồng lập trình viên lớn, đông đảo và sẵn sàng trợ giúp bạn. Điều này giúp các nhà lập trình viên dễ dàng phát triển ứng dụng và gặp ít khó khăn hơn nhiều so với những nền tảng khác.

## Giới thiệu về Firebase

Firebase có thể đã rất quen thuộc với cả lập trình viên di động và website những tính năng và ứng dụng quá tuyệt vời của nền tảng backend này đã được giới lập trình viên ủng hộ một cách mạnh mẽ.

Firebase giúp việc xây dựng ứng dụng của bạn nhanh hơn, tập chung hơn về logic của ứng dụng mà không cần quan tâm nhiều đến backend. Những chắc năng của Firebase như Firebase Analytics, Firebase database, Firebase Authentication đã giúp đỡ những nhà phát triển rất nhiều trong việc phát hành một ứng dụng thành công. Được xây dựng bởi Google và có thể mở rộng ứng dụng của bạn một cách mạnh mẽ, Firebase đã và đang được nhiều ứng dụng hàng đầu lựa chọn như Trivago, Wattpad, Doulingo hay Alibaba.com. Và đặt biệt Firebase hỗ trợ một cách toàn diện khi bạn phát triển một ứng dụng di động. App được build một cách tốt hơn, cải thiện được chất lượng ứng dụng, phát triển những logic khó đã đưa nền tảng này lại gần với lập trình viên hơn.



**Hình 8** Hình minh họa về sự phù hợp của Firebase khi phát triển ứng dụng di động [4]

### Firebase Realtime Database

Firebase Realtime Database là một tính đặc điểm tuyệt vời của giúp chúng ta làm việc với cơ sở dữ liệu. Có thể hiểu nôm na nó là một cơ sở dữ liệu dạng NoSQL được lưu và đồng bộ dữ liệu trên đám mây. Những dữ liệu này được đồng trên tất cả clients theo thời gian thực và vẫn khả dụng khi ứng dụng offline.

Realtime Database rất tốt vì nó có những tính năng tuyệt vời mà những lập trình viên cần sử dụng. Đầu tiên đó là realtime, trước đây với những ứng dụng bình thường khi có bất cứ sự thay đổi dữ liệu nào từ server người dùng thường phải tải lại trang web, hay mở lại ứng dụng. Nhưng hiện nay realtime database đã làm mọi thứ dễ dàng hơn. Tất cả dữ liệu sẽ được đồng bộ trên đám mây, khi có bất sự thay đổi nào từ cơ sở dữ liệu, mọi thiết bị kết nối tới sẽ nhận được những sự thay đó chỉ vài mili giây thôi thật tuyêt vời đúng không nào. Và thêm nữa Firebase Realtime Database giúp chúng ta đồng bộ dữ liệu dữ liệu trên bộ nhớ cache của máy khi chúng ta offline. Khi trực tuyến mọi thứ được kết nối loại và hệ thống dữ liệu sẽ tự động tải dữ liệu cho phía client. Mọi thứ đều hoàn toàn tự động và Firebase hoạt động thật hoàn hảo. Và nó có một tính năng đặc biết đó là có thể truy cập được cả từ phía ứng dụng di động và trình duyệt web mà không cần sử dụng bất cứ một server nào cả. Mọi ứng dụng truy cập dữ liệu realtime đều cần phải vượt qua các rule đó chính là những luật giúp dữ liệu của bạn được an toàn khỏi những truy cập không được phép từ bên ngoài.

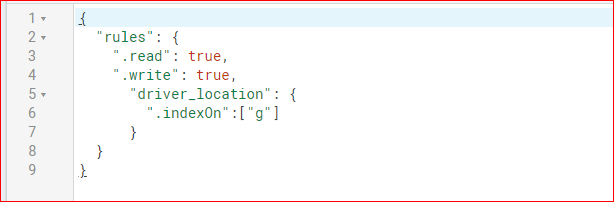
Chúng ta cũng sẽ cùng tìm hiểu cách cài đặt và sử dụng Firebase realtime database trong một ứng dụng Android.

Để sử dụng Firebase trong khi phát triển ứng dụng Android, chúng ta cần thêm thư viện mới nhất của nó trong build.gradle file:



**Hình 9** Thư viện Firebase Realtime Datababase

Tiếp theo một công việc quan trọng đó là cấu hình Realtime database để cho phép ứng dụng nào có thể đọc và ghi dữ liệu vào data base. Sau đây là hình ảnh mình hoạt cầu hình này:

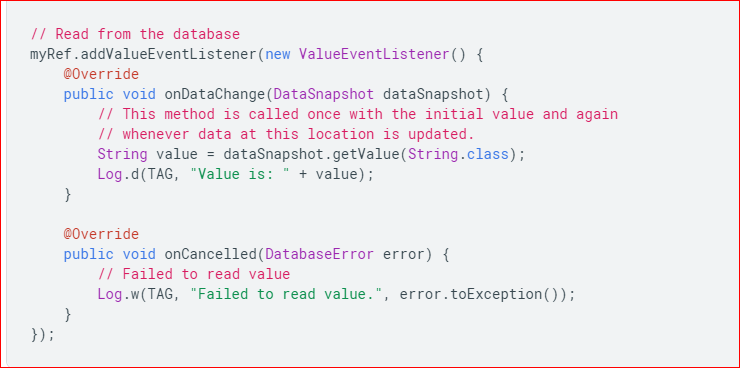


**Hình 10** Hình ảnh minh họa việc cấu hình Realtime Database trong ứng dụng

Hình ảnh phía trên mô tả quá trình cấu hình database cho ứng dụng gọi xe kết nối giữa tài xế và khách hàng. Nhìn qua chúng ta có thể hiểu rằng việc đọc ghi dữ liệu vào cơ sở dữ liệu này đã được cho phép khi nó được gán giá trị = true.

Vậy việc đọc và ghi dữ liệu trong thì thế nào nhỉ, sau đây là một vài minh họa về cách đọc và ghi dữ liệu từ nền tảng này.

Đầu tiên là dọc dữ liệu từ cơ sở dữ liệu, chúng ta phải thực hiện thao tác qua method addValueListener() để có thể tham chiếu tới cơ sở dữ liệu.



**Hình 11** Hình ảnh minh họa việc đọc dữ liệu từ Firebase Realtime Database

Viết dữ liệu chúng ta phải lấy thể hiện của Realtime Database qua method getInstance() và tham chiếu đến vị trí mà bạn muốn thay đổi dữ liệu.



**Hình 12** Hình ảnh minh họa việc ghi dữ liệu từ Firebase Realtime Database

Qua những tìm hiểu trên chúng ta đã hiểu rõ được cách sử dụng Firebase Realtim Database trong việc phát triển một ứng dụng Android giúp chúng ta phát triên những ứng dụng linh hoạt, có khả năng mở rộng cao và bảo mật tốt.

### Firebase Authentication

Authentication là một công việc quan trọng mà hầu hết những ứng dụng đều phải trải qua. Việc đáp ứng cho người dùng có thể đăng nhập, đăng ký hay đăng xuất trở trong ứng dung của mình. Nhưng những vấn đề về bảo mật hay bảo trì sẽ thường gây khó khăn trong quá trình phát triển, ngoài những hình thức Authentication bình thường, gần đây những ứng dụng thường cho phép người dùng đăng nhập qua những nền tảng mạng xã hội như Google, Twitter hay Facebook vậy có một nền tảng nào có hể hoạt động tốt tất cả những thứ đó, câu trả lời chính là Firebase Authentication.

Firebase có nhiều hình thức xác thực người dùng hiệu quả và được sử dụng rộng rãi như: Email and password based authentication hay Federated indentity provider intergation nhưng trong ứng dụng gọi xe của tài xế và khách hàng chúng ta cũng sẽ sử dụng hình thức authentication khác cũng khá phổ biến đó là Phone number authentication.

Chúng ta sẽ cùng tìm hiểu cách thiết lập và sử dụng việc authentication bằng số điện thoại qua những giai đoạn chính như sau:

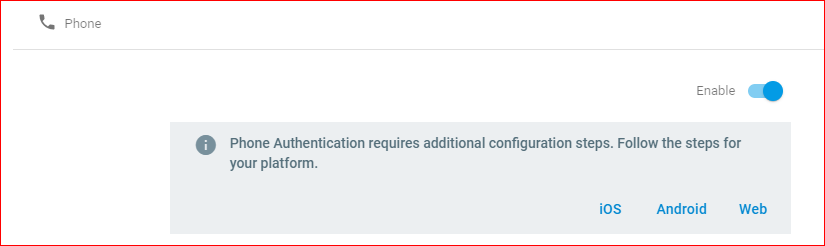
Để bắt đầu việc quan trong đầu tiên là một thư viện sẽ được thêm vào file build.gradle trong Android Studio



**Hình 13** Thư viện Firebase Authentication mới nhất từ Google

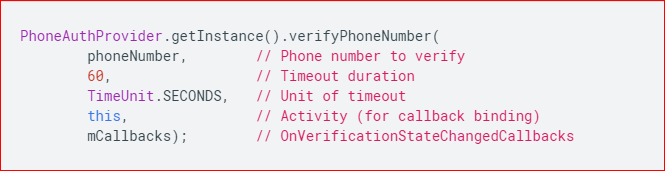
Khía cạnh bảo mật cũng trong authentication bằng số điện thoại dù không tốt bằng những hình thức khác vì bất cứ ai có số điện thoại hợp lý với mã xác thực SMS từ hệ thống cũng có thể đăng nhập vào ứng dụng. Nhưng do ứng dụng đặt xe không đòi hỏi độ bảo mật quá cao từ phía người dùng và tài xế do có sự tương tác trực tiếp và việc thanh toán được thực hiện bời tiền mặt. Firebase authentication với số điện thoại cũng khá một hình thức tốt và phù hợp với thực tế ứng dụng hiện nay.

Để sử dụng được Phone number authentication lập trình viên cần phải kích hoạt phương thức này trong Firebase project, vào mục Sign-In Method và kích hoạt phương thức đăng nhập bằng số điện thoại. Hình ảnh sau đây minh họa việc kích hoạt của dự án gọi xe:



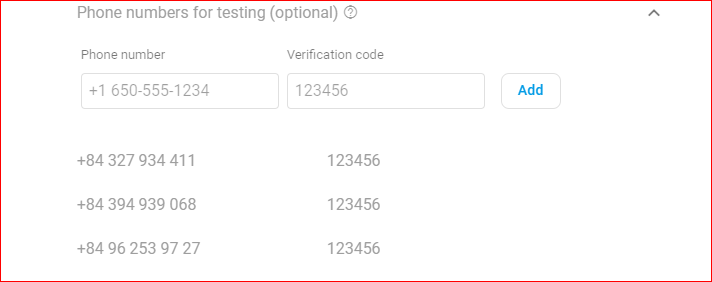
**Hình 14** Kích hoạt phương thức Phone number authentication của Firebase

Khi sử dụng việc xác nhận bằng số điện thoại, người sử dụng ứng dụng cần có một thiết bị thật để sử dụng chắc năng này, Firbase sẽ gửi một mã code gồm sau chữ số về phía máy của người dùng khi đó việc nhập đúng mã số giúp người sử dụng đăng nhập thành công.



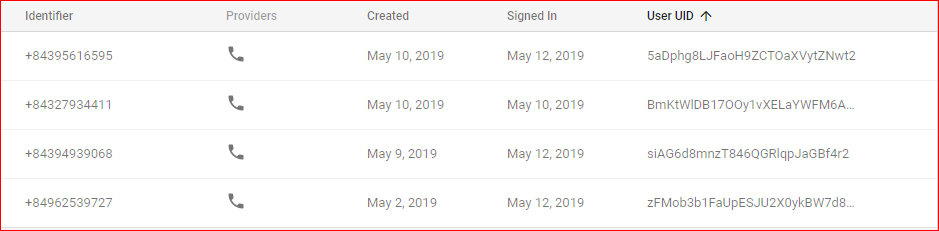
**Hình 15** Cách hệ thống gửi một mã xác thực tới điện thoại người dùng.

Chúng ta thấy rằng việc mã xác thực chỉ trong một thời gian timeout nhất định sẽ giúp hệ động đạt được độ bảo mật tốt hơn trong việc phát triển ứng dụng. Việc gửi quá nhiều lần mã xác thực đến một số điện thoại trong thời gian ngắn cũng sẽ không được phép. Việc này Firebase làm giúp các nhà phát triển ứng dụng tránh được những hacker đánh cặp tài khoản của người dùng. Vì vậy ứng dụng đặt xe có thỏa thuận giữa tài xế và khách hàng đã có những số điện thoại giúp ứng dụng kiểm thử nhiều trường hợp đăng nhập từ phía khách hàng.



**Hình 16** Các số điện thoại test của ứng dụng gọi xe

Sau khi đăng nhập thành công Firebase lưu lại các thông tin đăng nhập đầy đủ với số điện thoại ngày đăng nhập, ngày đăng xuất và cả mã người dùng. Tất cả những điều này giúp nhà phát triển ứng dụng quản lí được hệ thống của mình một cách hiểu quả và bảo mật tốt nhất.



**Hình 17** Hình ảnh minh họa những người dùng đã đăng nhập hệ thống gọi xe với thỏa thuận từ tài xế và khách hàng

### Firebase Storage

Chúng ta muốn lưu trữ dữ liệu dưới dảng ảnh, video hay file và Firebase có một tính năng quang trọng đó là Firebase Storage đây dịch vụ cho phép lý và lưu trữ nội dung có dữ liệu lớn từ phía nhà phát triển dụng.

Vậy nó có những ưu điểm gì để những nhà phát triển ứng dụng áp dụng nó vào trong dự án thực tế của mình. Sau đây là ba ưu điểm nổi bật của Firebase Storage:

* Firebase Storage giúp việc upload và dowload dữ liệu mà không phụ thuộc vào đường truyền mạng, dù đường truyền có gặp vấn đề hệ thống sẽ tạm dừng và lưu lại dữ liệu dã tải xong và bắt đầu lại khi có kết nối. Vì vậy Firebase Storage giúp tiết kiệm thời gian và băng thông.
* Firebase Storage được tính hợp với Firebase Authentication giúp việc bảo mật trở lên dễ dàng và việc quản lý truy cập dữ liệu dạng ảnh, file và video.
* Firebase Storage được xây dựng nên nền tảng Google Clound Platform nên khả năng mở rộng có thể lên tới Petabye dữ liệu.

Chúng ta sẽ cùng tìm hiểu cách thiết lập và sử dụng cũng như cách áp dụng Firebase Storage vào trong project ứng dụng gọi xe.

Đầu tiên như thường lệ chúng ta phải có thư viện giúp việc triển khai thành công:



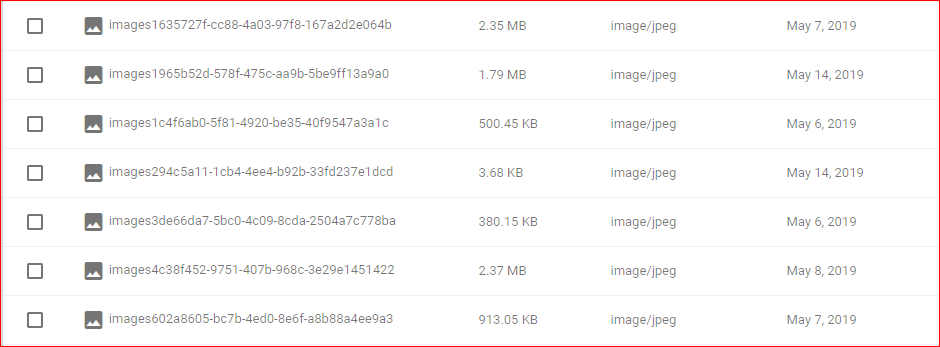
**Hình 18** Thư viện Firebase Storage

Việc thiết lập thật dễ dàng khi Firebase đã hỗ trợ rất nhiều vì điều này chỉ đơn giản là lấy một thể thiện của Firebase Storage bạn có thể áp dụng vào bất kì ứng dụng di động Android nào.



**Hình 19** Lấy một thể hiện của Firebase Storage

Vậy việc áp dụng Firebase Storage trong dự án của chúng ta là gì nhỉ? Đó chính là việc dùng để lưu và truy xuất một bức ảnh đại diện cá nhân cả ta. Sau đây là một ví dụ thực tế trong dự án khi lưu lại một file dạng ảnh trên Firebase Storage.



**Hình 20** Hình ảnh dữ liệu ảnh được lưu trong Firebase Storage

Vậy chúng ta đã tìm hiểu trong ba chức năng quan trọng của Firebase được áp dụng vào trong đồ án tốt nghiệp xây dựng ứng dụng đặt xe với giá cả thỏa thuận giữa người dùng và khách hàng, phần tiếp theo chúng ta sẽ cùng tìm hiểu một thư viện đặc biệt có vai trò quan trọng trong việc xử lí nghiệp vụ đó là GeoFire.

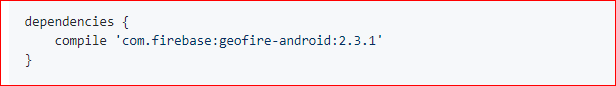
## Giới thiệu về GeoFire

Khi làm việc với dự án việc tìm những người dùng quanh một vị trí địa lý với khoảng cách nhất định đã có thể gặp nhiều khó khăn. Nhưng sau nhiều tuần tìm hiểu, có nên viết một thuật toán để giải quyết vấn đề này thì em đã tìm ra GeoFire một thư viện rất tốt làm việc được cả với Firebase Realtime Database chúng ta sẽ cùng tìm hiểu về GeoFire ngay thôi.

GeoFire là một thư viện mã nguồn mở cho Android/Java cho phép bạn lưu trữ và truy vấn sét một key dựa trên vị trí địa lý. Ở trung tâm, GeoFire lưu trữ những vị trí với một chuỗi key thường là id tại một vị trí địa lý. Đó là mục đích chính của GeoFire, ngoài ra nó còn có thể truy vấn keys trong một khu vực địa lý tại thời gian thực. GeoFire sử dụng Firebase Realtime Database để lưu trữ dữ liệu, cho phép truy vấn kết quả cập nhật theo thời gian thực nếu chúng có bất cứ sự thay đổi nào. GeoFire chỉ tải có chọn lọc những dữ liệu quanh vị trí nhất định. Luôn giữ cho ứng dụng của bạn nhẹ và phản ứng nhanh với những thay đổi, thậm chí là với dữ liệu cực kì lớn. [6]

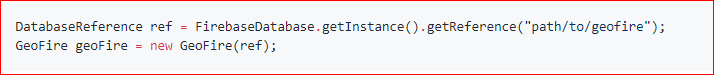
Như thường lệ chúng ta sẽ cùng tìm hiểu cách sử dụng và những thứ mà đã được áp dụng vào trong đồ án tốt nghiệp này.

Đầu tiên công việc quen thuộc một thư viện của GeoFire sẽ được thêm trong dự án.



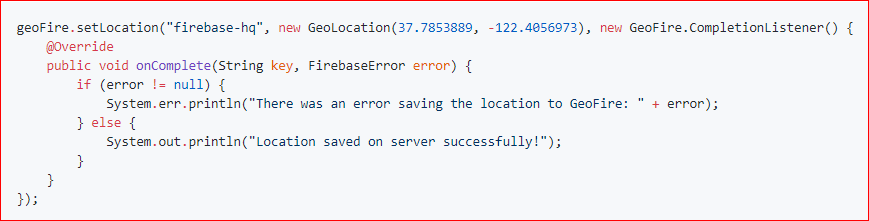
**Hình 21** Thư viện GeoFire

GeoFire giúp chúng ta đọc và ghi dữ liệu từ một vị trí địa lý và lưu vào trong Firebase Database. Đầu tiên việc tạo một instance của GeoFire và đính kèm nó với một bảng trong Firebase Realtime Database, ví dụ như sau:



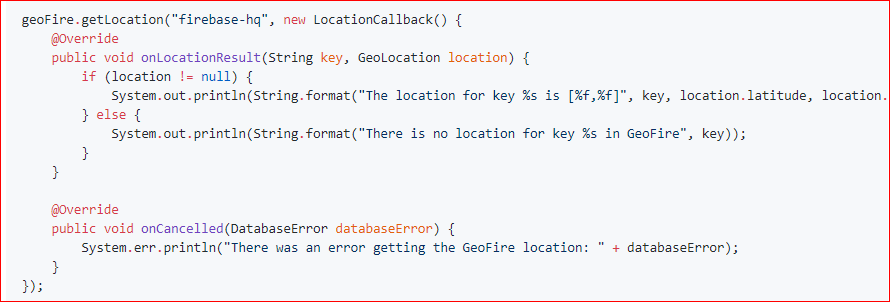
**Hình 22** Lấy một thể hiện của GeoFire

Việc lưu lại vị trí địa lý là cực kì quan trong đồ án tốt nghiệp này, mỗi sự thay đổi của vị trí GeoFire sẽ giúp chúng ta thay đổi ngay lập tức trong Firebase Database:



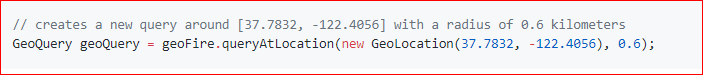
**Hình 23** Thiết lập dữ liệu vị trí

Tiếp theo là cách lấy vị trí dữ liệu với một key duy nhất được thực hiện như sau:



**Hình 24** Cách lấy dữ liệu vị trí với GeoFire

Một trong cách những truy vấn dữ liệu cực kì quan trọng và đóng vai trò lớn trong dữ án này đó là việc truy vấn dữ liệu quanh một vị trí nhất định, điều này được mô tả đơn giản như sau:



**Hình 25** Truy vấn dữ liệu vị trí quanh một bán kình nhất định

Qua hình trên chúng ta thấy rằng việc truy vấn vị trí có bán kình 0.6 km, mọi vị trí trong bán kình khu vực này có thể lấy được dữ liệu vị trí, giúp việc phát triển dự án nhanh và ít tốn chi phí hơn.

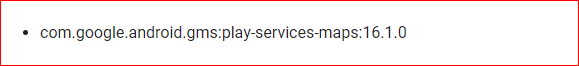
## Giới thiệu về Google API

### Google Map API cho Android

Với Map SDK cho Android, bạn có thể thêm một bản đồ dựa trên dữ liệu Google Map vào trong ứng dụng của bạn. API sẽ tự động xử lí truy cập dữ liệu Google Map từ server, tải dữ liệu về, hiện thị bản đồ cũng như phản ứng với những thay đổi. Bạn cũng thể sử dụng API gọi để thêm những marker, polygons và ghi đè lên một bản đồ đơn giản hơn. Bạn cũng có thể thay đổi được cách nhìn của người dùng từ một địa điểm cụ thể trên bản đồ. [7]

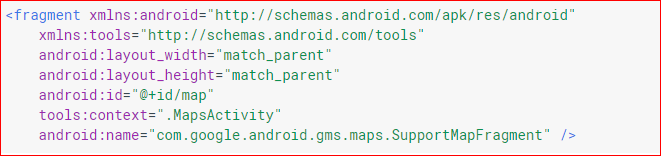
Google Map sẽ được chúng ta sử dụng nhiều trong quá trình phát triển đồ án này, đó là một thành phần quan trọng giúp chúng ta hiện thị được vị trí, đường đi những như những thay đổi về vị trí địa lý của khách hàng và lái xe khi di chuyển. Chúng ta sẽ cùng tìm hiểu một số cách thiết lập và sử dụng của Map SDK cho Android vào trong dự án này.

Một công việc quen thuộc thư viện Map SDK cần được thêm vào project



**Hình 26** Thư viện Map SDK cho Android

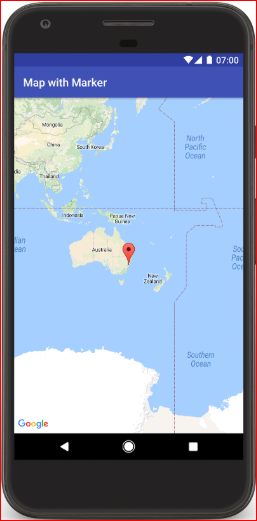
Để hiện thị được bản đồ trong một ứng dụng Android chúng ta cần một fragment và nó được minh họa qua những đoạn code xml sau:



**Hình 27** Fragment để hiện thị Google Map trong ứng dụng Android

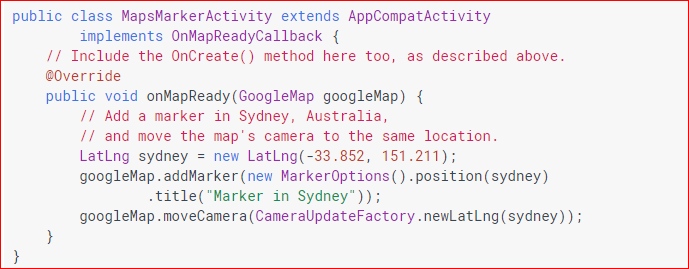
Việc làm tiếp theo chúng ta chỉ cần tham chiếu đến id của fragment để nó có thể hiện thị nhưng phần đó sẽ không được nói kĩ ở đây. Tiếp theo chúng ta sẽ cùng tìm hiểu hai phần quan trọng đó là hiện thị markers hay những icon trên bản đồ và vẽ một đường đi giữa hai vị trí mà sẽ được áp dụng rất nhiều trong đồ án này.

Đầu tiên là markers, marker sẽ được gọi và hiện thị trên một vị trí xác định của bản đồ, nó thường đại diện cho vị trí người dùng, tài xế hay là một địa điểm nhất định nào đó.



**Hình 28** Minh họa về hiện thị marker trong ứng dụng di động

Vậy sử dụng như nào để có thể hiện được nó. Chúng ta thường phải cung cấp cho marker một vị trí địa lý với kinh độ và vĩ độ để nó có thể hiện thị vị trí trên bản đồ. Cùng không quen di chuyển camera của bản đồ để đến đúng vị trí mà bạn mong muốn. Minh họa được thực hiện tại tọa độ thành phố Sydney như sau:



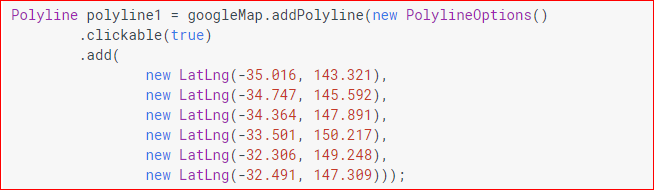
**Hình 29** Minh họa về xử lí hiện thị marker trong các đoạn mã code

Tiếp theo chúng ta cùng tìm hiểu về Polyline. Một đường đi có thể đường hiện thị giữa hai vị trí trên bản đồ, giúp người sử dụng hay tài xế có thể ước lượng được khoảng cách đi và đến giữa hai vị trí.



**Hình 30** Minh họa về hiện thị Polyline trên bản đồ

Tiếp theo một xử lí nhỏ về cách hiện thị Polyline trong các đoạn mã, chúng ta phải cung cấp ít nhất hai vị trí địa lý điểm đầu và cuối để polyline có thể được vẽ thành công, một vài những thay đổi về màu sắc hay độ rộng của Polyline cũng có thể được tùy chỉnh.

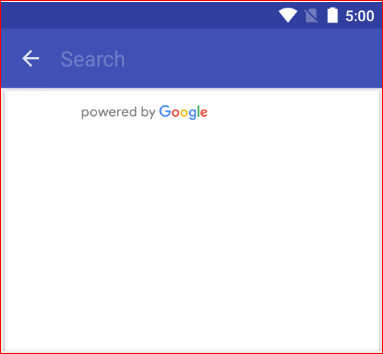


**Hình 31** Minh họa về cách xử lý hiện thị Polyline trên bản đồ

Qua đây chúng ta đã tìm hiểu được cách phát triển và sử dụng bản đồ trong xây dựng ứng dụng Android và cách áp dụng vào dự án thực tế. Phần tiếp theo chúng ta cùng tìm hiểu Google Place API giúp chúng ta tìm kiếm những địa điểm một cách nhanh chóng và đúng như ý muốn.

### Google Place API cho Android

Google Place API cho phép bạn xây dựng những ứng dụng xác định vị trí theo một ngữ cảnh địa phương. Với việc cung cấp thông tin về hơn 100 triệu địa điểm. Google Place API thực sự là một công vụ tuyệt với giúp các nhà phát triển ứng dụng xây dựng lên những hệ thống độc đáo.



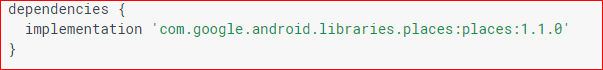
**Hình 32** Place Autocomplete trong Android

Place SKD cho Android có ba đặc điểm cũng như là công cụ nổi bật đó là:

* Place Picker cho phép người dùng có tìm kiếm các địa điểm xung quanh họ, có thể quán ăn, nhà hàng, khách san…
* Place Autocomplete giúp người dùng tìm kiếm những địa điểm một cách dễ dàng hơn, chỉ việc nhập địa chỉ, Place Autocomplete sẽ tự động hoàn thành truy vấn và lấy được địa điểm một cách chính xác.
* Place Detail cho phép người dùng lấy chi tiết về một địa điểm, họ có thể biết thêm những thông tin về địa điểm đó như hình ảnh, số điện thoại hay là trang web…

Trong việc phát triển ứng dụng gọi xe giữa khách hàng và tài xế. Ứng dụng của chúng ta sẽ sử dụng Place Autocomplete trợ giúp tìm kiếm địa điểm người dùng muốn đến một cách dễ dàng. Sau đây sẽ là cách thiết lập và sử dụng của API đặc biệt này.

Áp dụng vào dự án một API của Google thì không thể thiếu thư viện của nó. Chúng ta sẽ thêm thư viện Place API cho Android vào build.gradle file:



**Hình 33** Thư viện Place Autocomplete cho Android

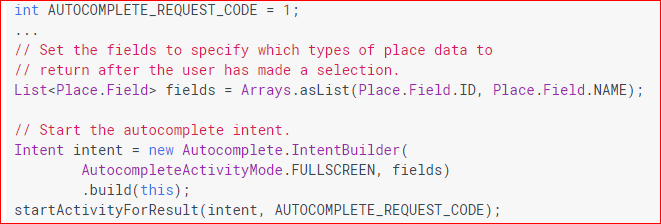
Tiếp theo việc khởi tạo Place API để sử dụng được thực hiện phải bao gồm một Google API Key và nó sẽ được thực hiện qua đoạn mã sau:



**Hình 34** Đoạn mã dùng khởi tạo Place API

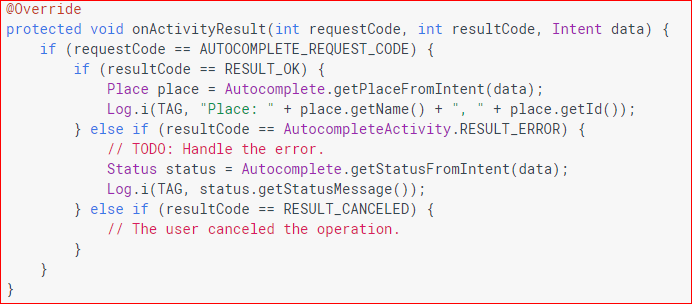
Tiếp theo chính là phần quan trọng khi chúng ta cùng tim hiểu về Place Autocomplete cách triển khai trong code để nó có thể hoạt động một cách chính xác.

Trong ứng dụng của chúng ta, việc xây dựng một giao diện riêng cho Place autocomplete là cần thiết vì vậy chúng ta sẽ sử dụng đối tượng Intent để hiện thị API của Google này.



**Hình 35** Đoạn mã dùng để tạo hiện thị Place Autocomplete

Sau dó chúng ta sẽ gọi phương thức onActivityResult() để lấy kết quả trả về, minh họa đoạn code như sau:



**Hình 36** Minh họa đoạn mã để trả về địa chỉ từ Place Autocomplete

### Google Direction API

Trong việc phát triển ứng dụng với Google Map chắc nhiều lập trình viên đã quen thuộc với Direction API, việc chỉ hướng đi, tính khoảng cách, thời gain dự kiến đã trở thành quen thuộc. Google đã cung cấp cho nhà phát triển một API tuyệt vời vậy cùng tìm hiểu nó xem nó hoạt động như thế nào để có thể áp dụng vào đồ án tốt nghiệp này.

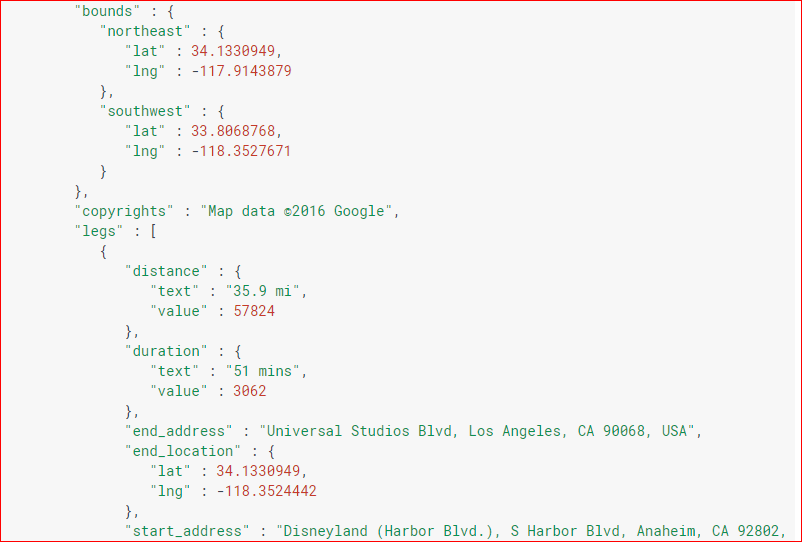
Với việc giao tiếp qua một giao thức HTTP Interface. Mỗi khi có một request đến API này sẽ trả về một URL bạn có thể sử dụng địa điểm, tọa độ địa lý kèm theo một API Key của Google mọi việc sẽ thành công. Đây là một đường dẫn URL như thế:





**Hình 37** Đường dẫn URL mà Direction API trả về

Bạn thử copy một đường dẫn URL như thế vào ô trình duyệt của Google bạn sẽ thầy một điều đặt biệt, kết quả đã được trả ra dưới dạng JSON với đầy đủ những thông tin chi tiết về đường đi tối ưu qua những điểm, thời gian, khoảng cách, địa chỉ đầu tiên và cuối cùng nữa. Vậy thật đơn giản những lập trình viên chúng ta chỉ phải đọc JSON và hiện thị lên giao diện cho người sử dụng. Dưới đây là minh họa về kết quả trả về như thế:



**Hình 38** Kết quả trả về dạng JSON của Direction API

Mặc dù phần kết quả còn thiếu nhiều dữ liệu nhưng chúng ta cũng đã có cách nhìn tổng quan về dữ liệu trả về của một Direction API này.

Qua tất cả những công nghệ đã được giới thiệu phía trên, đó chính là những công nghệ cốt lõi đóng vai trò quan trọng nhất trong việc xây dựng lên hai ứng dụng Driver App và User App của hệ thống đặt xe giữa người dùng và tài xế với giá cả thỏa thuận. Phần tiếp theo chúng ta sẽ cùng tìm hiểu việc phát triển ứng dùng một cách chi tiết nhất từ thiết kế kiến trúc, thiết kế chi tiết đến việc xây dựng ứng dụng và kiểm thử chương trình.

# Phát triển và triển khai ứng dụng

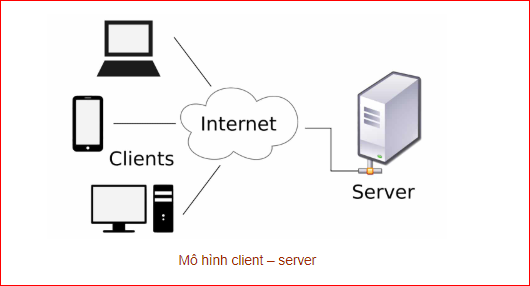
## Thiết kế kiến trúc

### Mô hình client-server

Mô hình client-server được hiểu nôm na đó chính là mô hình khách chủ trong đó phía client bao gồm có thể có các máy tính, điện thoại hay các thiết bị cầm tay nói chúng. Còn server chính là nơi lưu trữ dữ liệu và tài nguyên của hệ thống.

Trong mô hình này các máy khách tức là client sẽ gửi những request đến server và yêu cầu xử lí khi xử lí yêu cầu xong từ phía client, máy chủ trả về những response cho phía client.

Mô hình client-server khá là phổ biến khi chúng ta thực hiện xây dựng một ứng dụng di động, bởi vì chúng ta không thể biết phía người dùng sử dụng web, điện thoại iOS hay Android.

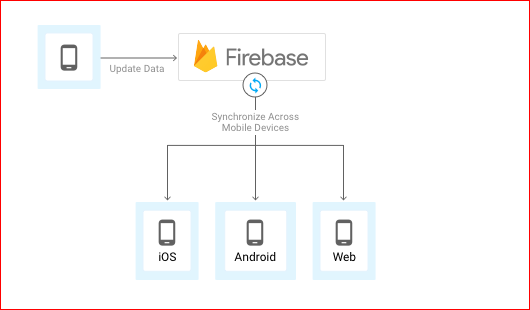


**Hình 39** Mô hình client-server

### Áp dụng trong đồ án tốt nghiệp

Ứng dung đặt xe giữa người dùng và tài xế với giá cả thỏa thuận cũng được xây dựng theo mô hình client-server. Trong đó Firebase sẽ đóng vai trò phía backend lưu trữ và xử lí dữ liệu, chúng ta chỉ cần gọi API và tập chung hơn về phía logic của ứng dụng.

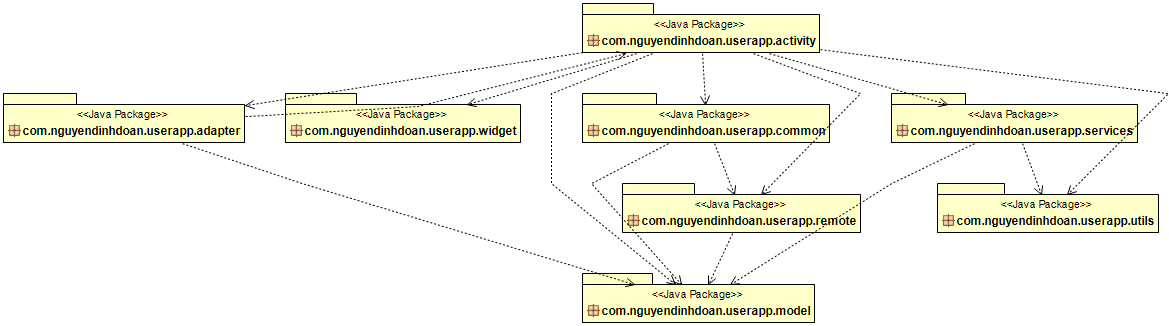
Firebase đã có sự kết hợp giữa nền tảng đám mây với máy chủ cực kì mạnh mẽ của Google chính vì vậy Firebase cung cấp cho chúng ta những công cụ hiệu quả để quản lí và lưu trữ dữ liệu đặt biệt là ở thời gian thực.



**Hình 40** Mô hình client-server với Firebase

### Thiết kế tổng quan của ứng dụng

Trong phần thiết kế tổng quan của ứng dụng, chúng ta sẽ cùng tìm hiểu cấu trúc gói của dự án, sự phụ thuộc diễn ra như thế nào và mô hình chung của dự án. Do hệ thống có hai ứng dụng dành riêng cho tài xế và người dùng, mà lại có cấu trúc tương tự nhau. Vì vậy em xin phép thầy cô chỉ bày trình bày thiết kế tổng quan của User App.



**Hình 41** Biểu đồ tổng quát

Biểu đồ thể hiện các gói riêng biệt với chức năng khác nhau. Sau đây chúng ta cùng tìm hiểu chi tiết chức năng và nhiệm vụ của từng gói trong dự án User App.

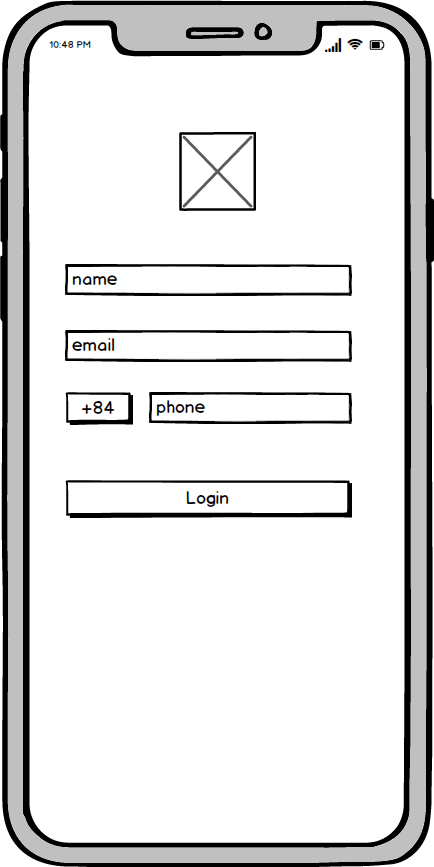
* “activity”: Các lớp activity vừa có nhiệm vụ lấy dữ liệu từ server và hiện thị lên giao diện cho người sử dụng.
* “adapter”:
* “widget”: Những tùy chỉnh về giao diện được xử lí ở đây
* “common”: Chứa những thuộc tính, phương thức hay hằng số dùng chung
* “services”: Xử lí những dịch vụ gửi notification và xử lí token cho ứng dụng
* “remote”: Kết nối tới server qua việc gọi API và lấy dữ liệu về
* “utils”: Chứa những phương thức dùng chung
* “model”: Có nhiệm vụ lưu trữ những thực thể tương ứng với cơ sở dữ liệu

## Thiết kế giao diện

Ứng dụng được xây dựng với mục tiêu thân thiện với người dùng, chính vì vậy giao diện đơn giản nhưng tinh tế luôn mà mục tiêu mà dự án hướng tới.

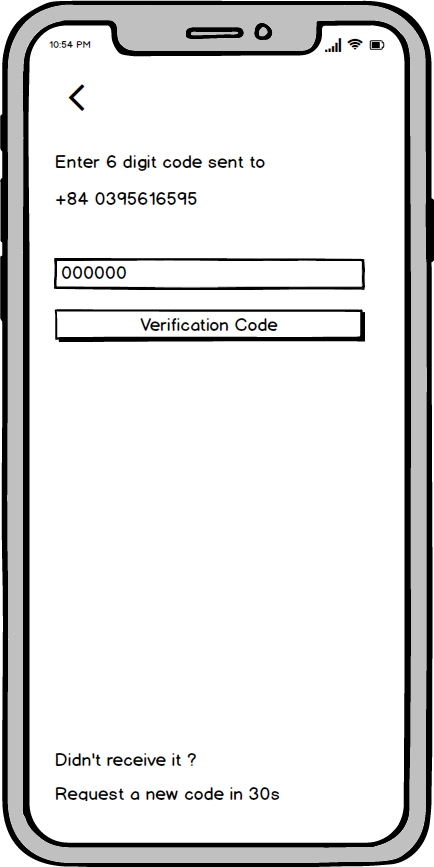
Hệ thống đặt xe được xây dựng trên hai ứng dụng dành riêng cho khách hàng và tài xế là User App và Driver App. Hai ứng dụng đều có những màn hình giao diện có đôi chút khác nhau nhưng tựu chung lại phong cách, kích thước chuẩn và cách thiết kế không có sự thay đổi. Chính vì điều đó trong phần thiết kế giao diện của đồ án tốt nghiệp này, em sẽ trình bày một cách chi tiết cách thiết kế của những màn hình chính của ứng dụng User App.

Đầu tiên đó là màn hình đăng nhập được thiết kế để người sử dụng ứng dụng điền những thông tin được yêu cầu khi sử dụng hệ thống.



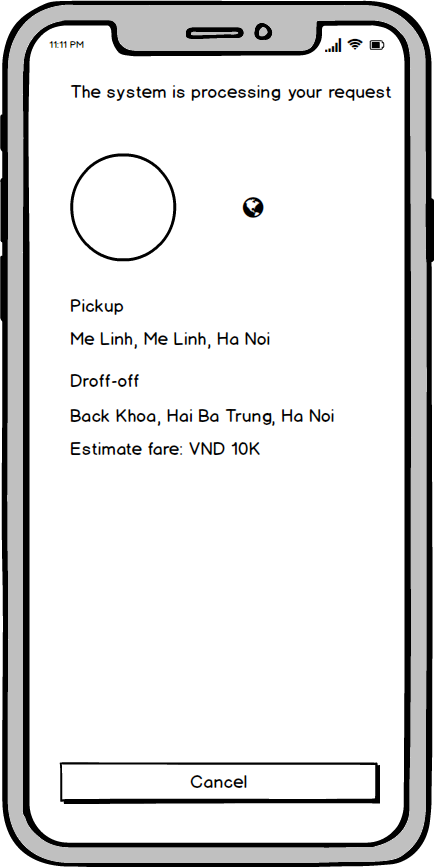
**Hình 42** Thiết kế màn hình đăng nhập

Tiếp theo là màn hình xác thực số điện thoại của khách hàng.



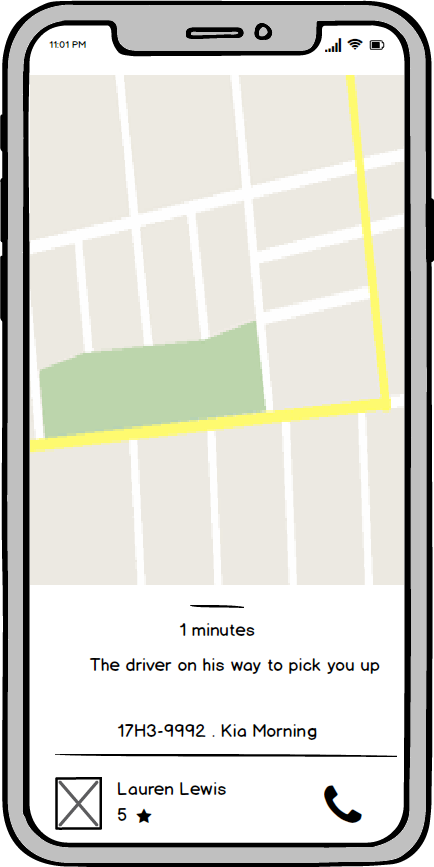
**Hình 43** Thiết kế màn hình xác thực số điện thoại

Màn hình khi khách hàng đặt cuốc xe.



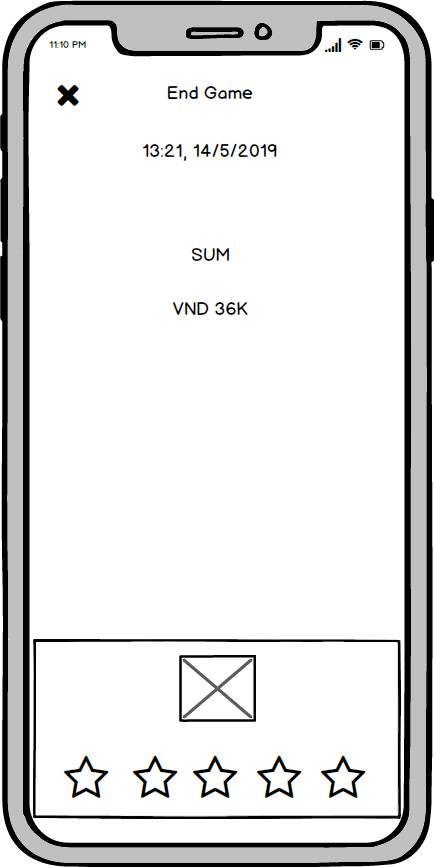
**Hình 44** Thiết kế màn hình khi khách hàng đặt cuốc xe

Màn hình thiết kế người dùng theo dõi thông tin chuyến đi.



**Hình 45** Thiết kế màn hình khách hàng theo dõi thông tin cuốc xe

Màn hình kết thúc với thông tin chi tiết chi phí cuốc xe và cho phép người dùng đánh giá tài xế.

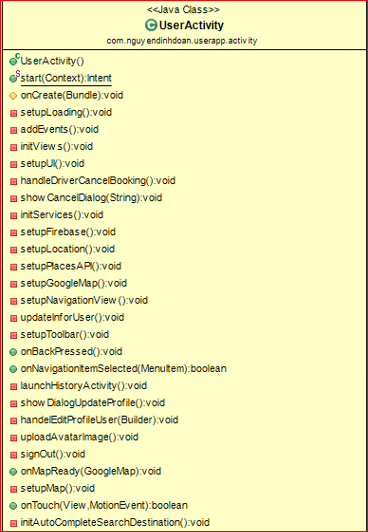
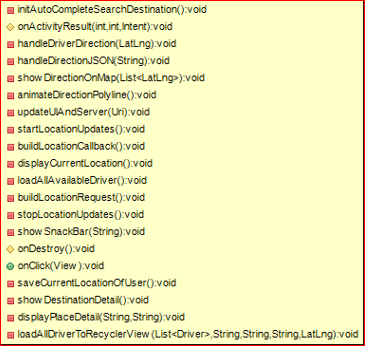


**Hình 46** Thiết kế màn hình chi phí cuốc xe và đánh giá tài xế

## Thiết kế lớp

Hai ứng dụng User App và Driver App được xây dựng và tạo thành bởi nhiều lớp nhưng vẫn có những lớp tạo lên chức năng chính của ứng dụng. Sau đây em xin được trình bày 3 đóng vai trò quan trọng trong ứng dụng dành cho khách hàng User App như sau.

Đầu tiên là UserActivity lớp đóng vai trò quan trọng khi hiện thị thông tin toàn bộ những tài xế có trong phạm vi 3km quanh vị trí của người dùng giúp người dùng nhìn thấy được chi phí và những đánh giá về tài xế đó. Do lớp quá dài chính vì vậy em xin phép chỉ hiện thị thông tin những phương thức trong lớp mà bỏ qua thuộc tính.

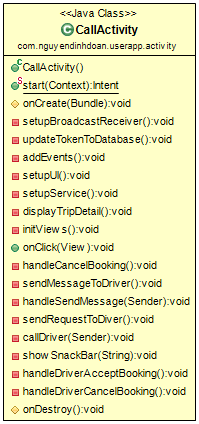
 

**Hình 47** Lớp UserActivity

**Bảng 12** Mô tả những phương thức chính của UserActivity

|  |  |  |
| --- | --- | --- |
| Phương thức | Tham số | Chức năng |
| loadAllAvailableDriver |  | Hiện thị những tài xế trong phạm vi 3 km quanh vị trí hiện tài của người dùng. Với điều kiện là các tài xế đang rảnh và có thể chấp nhận cuốc xe từ người dùng |
| loadAllDriverToRecyclerView | driverList, distance, locationAddress, destinationAddress, destinationLocation | Hiện thị thông tin chi tiết về những về tài xế, tên, đánh giá sao, chi phí cho cuốc phí và vị trí bắt đầu chuyến đi và vị trí kết thúc của chuyến đi. |
| handleDriverDirection | destinationLocation | Hiện thị một đường đi mô phỏng trên bản đồ từ vị trí của người dùng đến vị trí mà người dùng muốn đến trên bản đồ, bao gôm một animation chạy từ điểm đầu đến điểm cuối. |
| displayCurrentLocation |  | Hiện thị vị trí hiện tài của người dùng trên bạn đồ với một icon, vị trí này sẽ được cập nhật liên tục khi người dùn thay đổi vị trí dựa trên GPS của điện thoại. |

Tiếp theo là CallActivity lớp này được hiện thị khi người dùng chọn vào một tài xế bất kì và hệ thống bắt đầu gửi một thông báo tới tài xế.

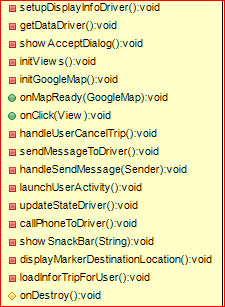
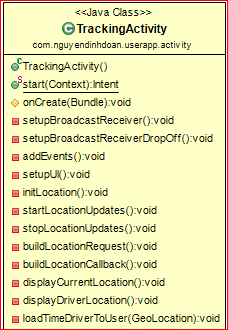


**Hình 48** Lớp CallActivity

**Bảng 13** Mô tả những phương thức chính của CallActivity

|  |  |  |
| --- | --- | --- |
| Phương thức | Tham số | Chức năng |
| updateTokenToDatabase |  | Cập nhật một token của người dùng tới Firebase Realtime Database điều này được sử dụng khi để nhận biết chính xác người sẽ gửi thông báo tới cho tài xế. |
| sendRequestToDiver |  | Gửi một thông báo tới tài xế, tài xế sẽ nhận được id của người dùng cùng với vị trí hiện tại của người dùng, vị trí họ muốn đến cùng với địa chỉ của điểm đến. |
| handleCancelBooking |  | Khi người sử dụng đột ngột không muốn đặt cuốc xe nữa, họ sẽ nhấn hủy khi đó phương thức này sẽ được gọi. Một thông báo sẽ được gửi đến cho tài xế về sự thay đổi của người dùng. |
| displayTripDetail |  | Hiện thị thông tin chi tiết cuốc xe mà người dùng đã đặt, gồm có địa điểm bắt đầu, địa điểm kết thúc và cước phí cho chuyến đi. |

Lớp tiếp theo đó là TrackingActivity



**Hình 49** Lớp TrackingActivity

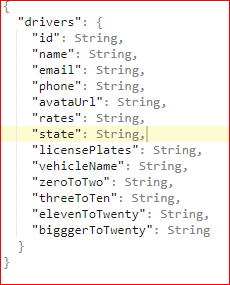
**Bảng 14** Mô tả những phương thức chính của TrackingActivity

|  |  |  |
| --- | --- | --- |
| Phương thức | Tham số | Chức năng |
| loadInforTripForUser |  | Hiện thị những thông tin về thời gian tài xế đến vị trí của người dùng khi mà tài xế đã chấp nhận cuốc xe. |
| displayDriverLocation |  | Hiện thị vị trí của tài xế theo thời gain thực, người dùng sẽ theo dõi được quãng đường và địa điểm hiện tại của tài xế. |
| handleUserCancelTrip |  | Xử lí khi chuyến đi chưa được tài xế nhấn bắt đầu, người dùng vẫn có thể hủy cuốc xe đó, khi đó dựa vào token id của tài xế, ta sẽ gửi một thông biết cho tài xế biết. |
| callPhoneToDriver |  | Phương thức này sẽ xử lí việc gọi điện cho tài xế, số điện thoại của tài xế sẽ lấy dược từ cơ sử dữ liệu thời gain thực khi ta đã có id. |

## Thiết kế cơ sở dữ liệu

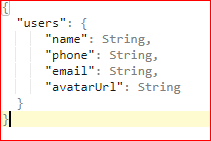
Cơ sở dữ liệu của ứng dụng User App và Driver App là dạng NoSQL được lưu trữ theo kiểu key- value qua các đối tượng dạng json. Sau nay là chi tiết về những đối tượng trong cơ sở dữ liệu.

Đầu tiên là drivers:



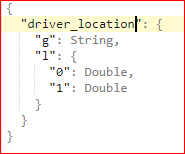
**Hình 50** Cơ sở dữ liệu drivers

Tiếp theo users:



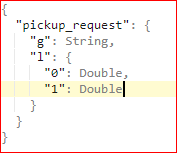
**Hình 51** Cơ sở dữ liệu users

Đối tượng tiếp theo là driver\_location:



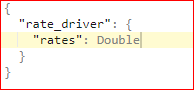
**Hình 52** Cơ sở dữ liệu driver\_location

Tiếp theo là pickup\_request:



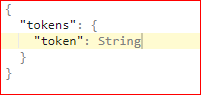
**Hình 53** Cơ sở dữ liệu pickup\_request

Tiếp theo là rate\_driver:



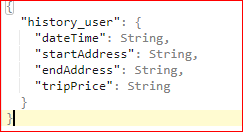
**Hình 54** Cơ sở dữ liệu rate\_driver

Tiếp theo là tokens:



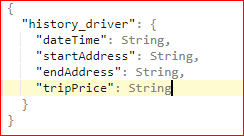
**Hình 55** Cơ sở dữ liệu tokens

Tiếp theo là history\_user:



***Hình 56*** *Cơ sở dữ liệu history\_user*

Tiếp theo là history\_driver:



**Hình 57** Cơ sở dữ liệu history\_driver

Việc sử dụng NoSQL có vai trò quan trọng trong việc xây dựng những cơ sở dữ liệu lớn, truy vấn tốc độ cao và ít có ràng buộc về tài nguyên hệ thống và khả năng chịu lỗi. Việc áp dụng cơ sở dữ liệu không quan hệ còn phù hợp khi chúng ta xây dựng một ứng dụng thời gian thực, phản hồi ngay lập tức với những thay đổi ở phía client.

## Xây dựng ứng dụng

### Thư viện và công cụ sử dụng

Trong hệ thống đặt xe ngoài có sử dụng nhiều công cụ và thư hiện để năng cao năng suất và tối ưu hóa thơi gian phát triển ứng dụng. Các thư viện bên thứ được chọn lựa khi có những đánh giá cao từ nhiều nhà phát triển cũng như được bảo trì thường xuyên.

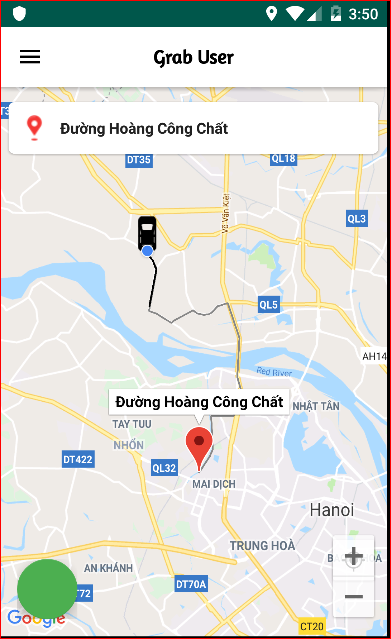
**Bảng 15** Danh sách thư viện và công cụ sử dụng

|  |  |  |
| --- | --- | --- |
| Mục đích | Công cụ / Thư viện | Địa chỉ URL |
| IDE lập trình Android | Android Studio IDE | https://developer.android.com/studio |
| Hộ trợ gọi API | Retrofit | https://square.github.io/retrofit/ |
| Dịch ngược Polyline Map | Map utils | https://github.com/googlemaps/android-maps-utils |
| Xin quyền từ hệ thống | Dexter | https://github.com/Karumi/Dexter |
| Bo tròn ảnh | Circle Image View | https://github.com/hdodenhof/CircleImageView |
| Hỗ tợ tải ảnh từ Internet | Glide | https://github.com/bumptech/glide |
| Hiện thị Loading | Spots Dialog | https://github.com/d-max/spots-dialog |
| Hỗ trợ cắt ảnh theo ý muốn | Image Cropper | https://github.com/ArthurHub/Android-Image-Cropper |
| Hiện thị Loading bo tròn với chuyển động | Loading Indicator View | https://github.com/81813780/AVLoadingIndicatorView |
| Lưu trữ và truy vấn vị trí | Geo Fire | https://github.com/firebase/geofire-java |
| Lưu trữ, truy vấn, xác thực cơ sở dữ liệu, đăng nhập, lưu file ảnh | Firebase | https://firebase.google.com/ |
| Xử lí hiện thị bản đồ | Google Map SDK Android | https://developers.google.com/maps/documentation/android-sdk/intro |
| Hỗ trợ tìm kiếm | Google Place SDK Android | https://developers.google.com/places/android-sdk/intro |
| Hỗ trợ hướng dẫn đường đi | Google Direction API | https://developers.google.com/maps/documentation/directions/start |
| Lưu trữ mã nguồn | Github | https://github.com |
| Thiết kế biểu đồ ca sử dụng, hoạt động | Astah | http://astah.net/ |
| Quản lí dự án với Google | Google Console | https://console.developers.google.com |

### Kết quả đạt được

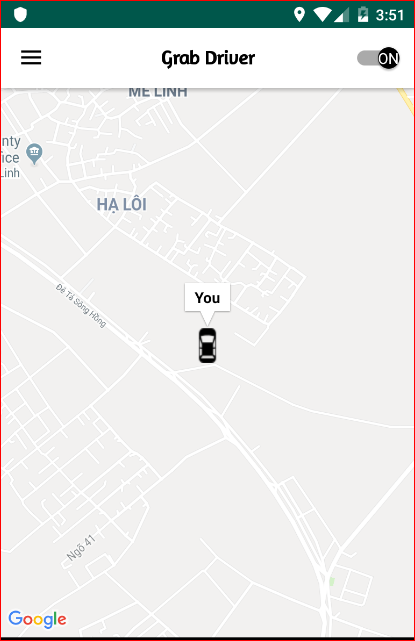
Phần này em xin trình bày và minh hoạ những màn hình chính của ứng dụng User App và Driver App. Sau mỗi bức ảnh là mô tả về màn hình cũng như chắc năng mà người dùng hệ thống có thực hiện.

Đầu tiên là màn hình chính của khách hàng khi tìm kiếm một địa điểm đến.



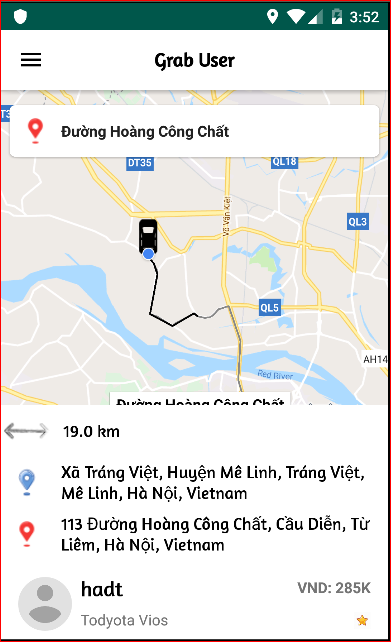
**Hình 58** Màn hình chính của khách hàng

Màn hình chính của tài xế như sau:



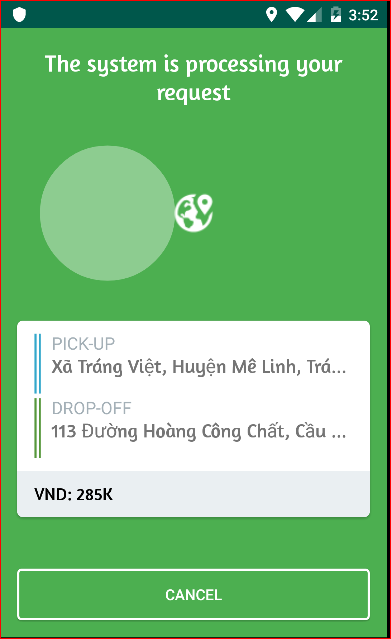
**Hình 59** Màn hình chính của tài xế

Màn hình khi khách hàng chọn tài xế cho cuốc xe:



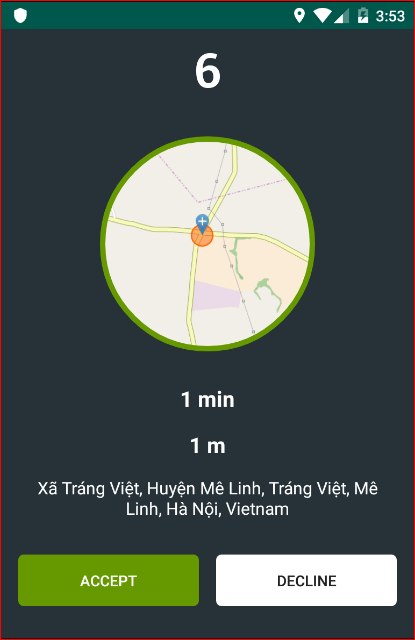
**Hình 60** Màn hình khi khách hàng tìm tài khế cho quốc xe

Màn hình hiện thị đang gọi tài xế



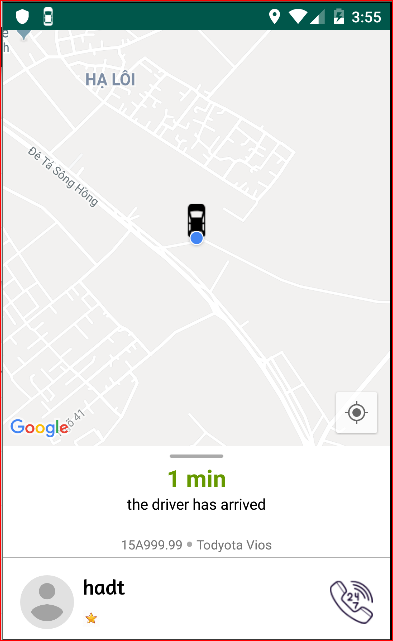
**Hình 61** Màn hình gọi cho tài xế

Màn hình tài xế nhận được cuộc gọi đến



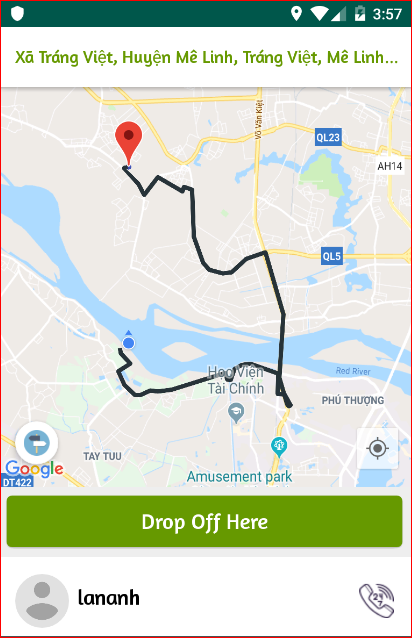
**Hình 62** Màn hình tài xế nhận được cuộc gọi

Màn hình hiện thị theo dõi tài xế của khách hàng trong chuyến đi



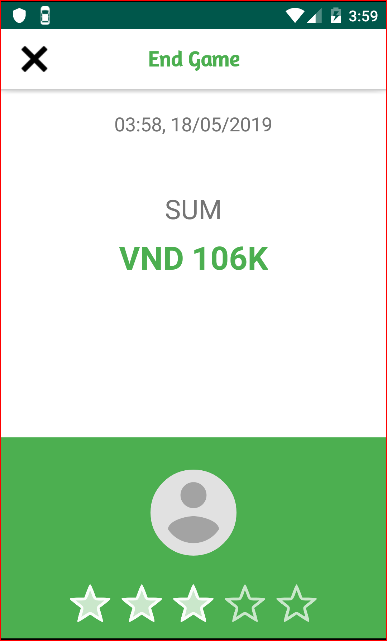
**Hình 63** Màn hình theo dõi tài xế của khách hàng

Màn hình xử lí của tài xế:



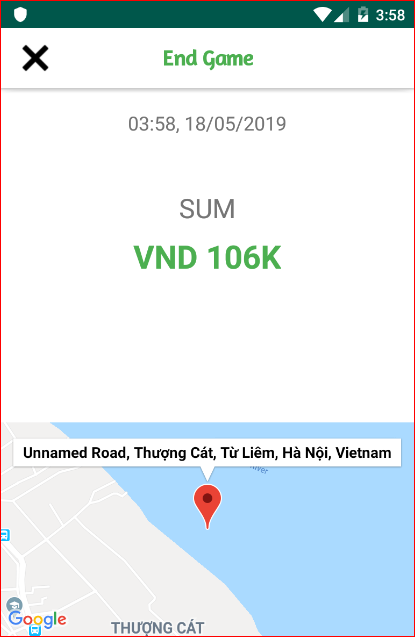
**Hình 64** Màn hình xử lí cuốc xe của tài xế

Màn hình kết thúc cuốc xe của khách hàng



**Hình 65** Màn hình kết thúc cuốc xe của khách hàng

Màn hình kết thúc cuốc xe của tài xế



**Hình 66** Màn hình kết thúc cuốc xe của tài xế

## Kiểm thử

Qua trình kiểm thử của dự án tập trung vào kiểm thử hộp đen, đó chính là những gì người dùng thao tác và sử dụng hệ thống. Kiểm thử giúp chương trình giảm tối đa lỗi có thể khi ứng dụng đến tay người sử dụng. Dưới đây là những testcase quản trọng của ứng dụng User App và Driver App. Testcase được thiết kế đơn giản nhưng mô tả đúng tên, trình tự hoạt dộng cũng như đầu ra mong đợi khi kiểm thử một chức năng của chương trình.

**Bảng 16** Thiết kế kiểm thử

|  |  |  |  |
| --- | --- | --- | --- |
| No. | TestCase Name | Action(Conditions if need) | Expected Output |
| 1. | Đăng nhập | - Nhập thông tin tên, email, số điện thoại  - Nhấn chọn “login” | - Hệ thống không báo lỗi, thông tin nhập hợp lệ.  - Màn hình xác thực số điện thoại được hiện thị, người dùng nhận mã số 6 số. |
| 2. | Xác thực số điện thoại | - Nhập mã số điện thoại nhận được  - Nhấn chọn “xác thực mã” | - Thực hiện xác thực thành công, hệ thống không báo lỗi. |
| 3. | Người dùng tìm kiếm địa điểm đến | - Nhập địa điểm cần di chuyển đến  - Nhấn chọn địa điểm hệ thống gợi ý người dùng | - Hệ thống gợi ý một danh sách địa điểm từ kí tự người dùng đánh.  - Hiện thị một đường đi từ địa điểm hiện tại của người dùng đến địa điểm đến. |
| 4. | Người dùng đặt cuốc xe | - Nhấn chọn “đặt cuốc xe”  - Chọn tài xế mình muốn | - Hiện thị danh sách tài xế trong phạm vi 3km có thể nhận cuốc xe.  - Hệ thống gửi thông báo cho tài xế được chọn, màn hình tài xế xuất hiện thông tin cuốc xe. |
| 5. | Người dùng theo dõi di chuyển của tài xế |  | - Tài xế chấp nhận cuốc xe thông tin về vị trí và thời gian tài xế tới hiện thị trên màn hình người dùng |
| 6. | Người dùng theo dõi cuốc xe |  | - Quãng đường di chuyển và thời gian ước lượng tới địa điểm đến được hiện và thực thi trên màn hình. |
| 7. | Người dùng đánh giá tài xế | - Chọn từ 1 đến 5 sao trên màn hình  - Chọn thoát khỏi màn hình “end game” | - Hệ thống lưu lại đánh giá của người dùng.  - Lưu đánh giá vào cơ sở dữ liệu và tính trung bình sao cho tài xế. |
| 8. | Tài xế nhận được cuốc xe |  | - Màn hình hiện thị địa điểm bắt đầu, kết thúc của cuốc xe cùng khoảng cách và thơi gian dự kiến của cuốc xe.  - Tài xế có 30s để đồng ý cuốc xe |
| 9. | Tài xế đồng ý cuốc xe | - chọn “accept” trên màn hình | - Hệ thống hiện thị vị trí của tài xế và người dùng cùng đường đi dự kiến trên màn hình  - Thông tin về người dùng được hiện thị: tên, ảnh đại diện và tài xế có thể gọi cho khách hàng |
| 10. | Tài xế di chuyển đến vị trí của khách hàng | - Nhấn điều hướng trên màn hình | - Màn hình điều hướng chi tiết đường đi cho tài xế tới vị trí của người dùng |
| 11. | Tài xế bắt đầu chuyến đi | - Nhấn chọn “start trip” trên màn hình  - Nhấn chọn “điều hướng” | - Hệ thống bắt đầu tính tiền từ vị trí hiện tại, một thông báo sẽ được gửi cho người dùng  - Màn hình điều hướng chi tiết đường đi tới vị trí của người dùng muốn đến. |
| 12. | Tài xế kết thúc cuốc xe | - Nhấn chọn “end trip” trên màn hình | - Màn hình hiện thị chi phí của cuốc xe và người dùng phải thanh toán. |
| 13 | Tài xế chọn chế độ không làm việc | - Gạt thanh switch sang bên trái nếu đang hoạt động | - Vị trí của tài xế không được hiện thị và người dùng không thấy được tài xế. |
| 14 | Tài xế đang làm việc |  | - Khi tài xế đang có cuốc xe, khách hàng không thể nhìn thấy tài xế và họ không đặt cuốc được với tài xế. |
| 15. | Tài xế tới gần vị trí của người dùng lúc người dùng đặt cuốc xe |  | - Hệ thống cập nhật thời gian thực hiện thị thông tin về tài xế cho khách hàng. |

# Các giải pháp và đóng góp nổi bật

## Kết quả và những điểm mới trong việc phát triển ứng dụng

Hệ thống được xây dựng với hai ứng dụng rành riêng cho tài xế và khách hàng. Mỗi ứng dụng được xử lí những chức năng yêu cầu phù hợp với thực tế sử dụng của người dùng. Ứng dụng được xây dựng đơn giản nhưng thân thiện với người dùng, cung cấp đầy đủ thông tin, chỉ dẫn đường đi, và theo dõi tới tài xế. Dù vậy do thời gian ngắn để phát triển và kình nghiệm chưa có ứng dụng còn rất nhiều điều thiết xót và chức năng chưa hoàn thiện đầy đủ. Nhưng cả hai ứng dụng cũng đã đáp ứng cơ bản về chức năng đặt và nhận cuốc xe từ khách hàng và tài xế và hứa hẹn ứng dụng sẽ được cập nhật thêm nhiều trong tương lai.

Ứng dụng đặt xe dành cho khách hàn và tài xế với giá cả thỏa thuận được xây dựng tương đối hoàn thiện về chức năng cũng như giải pháp việc tạo ra cho người dùng việc lựa chọn những tài xế mà mình thích theo ý muốn, phù hợp với chi phí và ý thích của nhà phát triển, tạo cho người dùng sự tự do quản lí được chuyến đi của mình. Việc nhìn thấy được đánh giá sao của tài xế cũng giúp khách hàng biết được người lái xe nào thì phù hợp với bản thân, những loại xe cao cấp hơn có thể có chi phí cuốc xe lớn hơn và ngược lại, việc tùy ý lựa chọn tài xế và không phụ thuộc vào sự lựa chọn của hệ thống như Grab hay Uber cũng sẽ tiết kiệm thời gian hơn cho việc tìm kiếm.

Về phía tài xế được tự mình đưa ra cước phí cho từng km đường đi thật sự là tuyệt vời. Cước phí dưới 2km, từ 2km đến 10km, từ 10km đến 20km và lớn hơn 20km. Nhìn qua thì thật giống với taxi những tài xế đã tối ưu được thu nhập của mình mà không phụ thuộc vào bất cứ hãng taxi nào, cũng như là những khoản tiền phải triết khấu trước khi cho những hãng. Không phải chịu ràng buộc về thời gian địa điểm hay nhiều thứ khác nữa. Tài xế khi có xe thời gian rảnh rỗi hay đang đi trên đường hoàn toàn có thể sử dụng ứng dụng để kiếm thêm thu nhập cho bản thân và gia đình. Việc sử dụng ứng dụng với giá cả thỏa thuận cũng khác biệt hơn so với Grab hay Uber bỏ qua thủ tục đăng ký rắc rối, những nguyên tắc phải theo như mặc quần áo hay giờ đi lại. Ứng dụng cho còn cho phép tài xế nhận được toàn bộ thù lao của mình trong mỗi chuyến đi, tự mình đặt giá cước tài xế có thể tham khảo từ nhiều nguồn khác nhau nhận thấy được chiếc xe của mình đáng với giá cao hơn, hay dịch vụ đi xe tuyệt vời .Tài xế sẽ có thể tự mình xây dựng chi phí và chiến lược của mỗi cuốc xe sao cho thu hút được khách hàng nhất có thể và tài xế cũng có thể đăng xuất để thay đổi chi phí từng km của mình.

## Một số giải pháp về mặt công nghệ

### Áp dụng công nghệ Firebase

Việc sử dụng Firebase trong phát triển ứng dụng đã giảm rất nhiều thời gian và chi phí phát triển ứng dụng. Việc không phải xây dựng một backend đã giúp lập trình viên tối ưu hóa thời gian và tập chung vào phát triển nghiệp vụ phía client. Trong quá trình phát triển việc áp dụng những công nghệ của Firebase như Firebase Authentication, Firebase Realtime Database, Firebase Storage đã giúp ích rất nhiều với một ứng dụng thời gian thực. Việc cập nhật bất cứ cự thay đổi nào ngay tại thời điểm đó rất phù hợp với ứng dụng gọi xe đòi hỏi thời gian chờ đời của khách hàng hay tài xế phải ở mức thấp nhất. Firebase còn giúp những nhà phát triển tận dụng được ưu thế vì tốc độ truy vấn nhanh, khả năng lưu trữ dữ liệu lớn NoSQL mà không phụ thuộc quá nhiều về tài nguyên của hệ thống. Một lợi ích to lớn khác không thể không nói đến là độ bảo mật tốt của nền tảng được phát triển bởi Google này. Sử dụng nhiều Rule và những ràng buộc chống lại những xâm nhập hay truy cập trái phép giúp những nhà phát triển an tâm hơn về độ an toàn của dữ liệu hệ thống. Firebase thực sự rất phù hợp với những công ty nhỏ, những startup hay học sinh, sinh viên nhưng Firebase cũng đã được tin dùng bởi những công ty và tập đoàn hàng đầu.

### Áp dụng API Map của Google

Chúng ta phát triển một ứng dụng trên hệ điều hành của Google đó là Android, sử dụng công cụ backend của Google đó là Firebase và lại phát triển một ứng dụng bản đồ của Google. Chính vì vậy việc sử dụng các API bản đồ của tập đoàn công nghệ hàng đầu này đó là điều chắc chắn. Google đã xây dựng những API tuyệt vời cho các nhà phát triển đặc biệt trên nền tảng di động hay Android. Những việc hiện thị, tìm kiếm hay chỉ đường di chuyển được thực hiện chính xác, tốc độ cao thực sự là một điều tuyệt vời khi phát triển một ứng dụng. Google Map API cho Android cho ta việc hiện thị bản đồ, giúp ích nhiều hơn khi vị trí của người dùng hiện tại được hiện thị với độ chính xác cáo. Việc tìm kiếm địa điểm, vị trí nhà hàng, khách san… đã có Google Place API hàng triệu địa chỉ có sẵn trợ giúp những lập trình viên nhiều, nhiều cho việc hiện thị địa điểm và giảm thiểu tối đa những thao tác mà người sử dụng phải làm. Chỉ đường bên bản đồ, việc chỉ đường cho ô tô, xe máy, xe bus hay đi bộ đã được Google Direction API xử lí một cách hoàn hảo. Các nhà phát triển có thể chọn đường đi theo ý muốn hoặc API này sẽ lựa chọn đường đi tối ưu nhất với thời gian ngắn nhất cho những lập trình viên.

# Kết luận và hướng phát triển

## Kết luận

Qua trình phát triển hệ thông đặt xe cũng đã gặp nhiều khó khăn thử thách nhưng với sự giúp đỡ của thầy hướng dẫn cùng nỗ lực của bản thân ứng dụng đã được hoàn thành một cách cơ bản và có thể áp dụng ý tưởng này vào trong thực tế. Ứng dụng được xây dựng và thiết kế từ bước đầu đến khi hoàn thành và thực hiện việc kiểm thử. Những chức năng của khác hàng hay tài xế đều được tham khảo và nghiên cứu cũng như được hướng dẫn từ thầy Vũ Đức Vượng đúng theo thực tế và mục tiêu mà đồ án hướng tới. Việc gọi xe của khách hàng hay nhận cuốc xe của tài xế đều được thực hiện và xử lí theo hướng thời gian thực. Phần tối ưu và hiệu năng của ứng dụng thật sự còn chưa được tốt vì bản thân chưa có nhiều kình nghiệm khi phát triển một ứng dụng di động. Nhưng tựu chung lại ứng dụng đặt xe có giá cả thỏa thuận giữa khách hàng và tài xế cũng đã đáp ứng cơ bản việc thiết kế, xây dựng và kiểm thử ứng dụng.

Bản thân đã cố gắng giải quyết những khó khăn của dự án, việc hiện thị những thông tin tài xế trong một khoảng cách nhất định đã làm tốn rất nhiều thời gian và công sức. Nhưng nó cũng chính là một niềm vui khi phát triển một ứng dụng thực tế. Việc ứng dụng đột nhiên cố lỗi hay sửa những lỗi xảy ra trong quá trình phát triển thực sự khiến bản thân cảm thấy căng thẳng nhiều. Nhưng có sự động viên từ thầy, gia đình và bạn bè bản thân đã cố gắng và vượt qua những khó khăn đó. Việc tìm hiểu tất cả những giai đoạn từ đầu đến khi kết thúc cơ bản việc phát triển một ứng dụng thực tế đã cho em nhiều kinh nghiệm hơn, nhiều ý tưởng và cách nhìn hơn khi phát triển bản sau này.

Qua đồ án này, em thấy rằng việc phát triển một ứng dụng gọi xe với thỏa thuận giữa tài xế và khách hàng là một ý tưởng tốt khi nó đặt lợi ích của tài xế lên và khách hàng lên hàng đầu. Có lẽ ứng dụng sẽ thu hút được lượng người dùng là lái xe lớn hơn. Nhưng việc lợi nhuận của nhà phát hành ứng dụng cũng cần được tính đến thêm và nhiều chức năng cần được cải thiện và tối ưu. Hướng phát triển ứng dụng trong tương lai sẽ được em trình bày một cách cụ thể hơn trong phần sau của đồ án này.

## Hướng phát triển

Trong tương lai ứng dụng ứng dụng gọi xe giữa người dùng với khách hàng sẽ được xây dựng thêm những chắc năng để cải thiện khả năng sử dụng cũng như tích hợp thêm những tính năng bổ ích và hướng tới người dùng

* Xây dựng chức năng giúp khách hàng có thể thanh toán bằng thẻ tín dụng, Visa, Master … bên cạnh việc thanh toán tiền mặt, việc tích hợp những hình thức như ví điện tử cũng là một cách khác để giảm bớt thao tác thanh toán từ khách hàng.
* Nghiên cứu thêm hướng phát triển với hình thức đi chung xe có thể giảm chi phí của khách hàng đi rất nhiều, có người đi chung cũng có thể tạo thêm không khí vui vẻ của chuyến đi.
* Tích cựu tham khảo những ứng dụng gọi xe như Grab, Uber để cải thiện thêm chức năng, theo dõi những đánh giá thấp 1 sao nên Google Play Store, Apple App Store để thu thập thêm thông tin đánh giá và phát triển ứng dụng.
* Cố gắng phát triển theo ứng tối ưu hóa và nâng cao hiệu năng của chương trình giảm dung lượng khi cài đặt ứng dụng cũng như chuyển đổi ngôn ngữ phát triển từ Java sang Kotlin và một mô hình phát triển cụ thể như MVVM để nâng cao chất lượng chương trình.
* Cố gắng sửa lỗi và kiểm thử chương trình nhiều hơn nữa, cả hộp đen và hộp trắng, kiểm thử mức độ phủ nhánh của những dòng code. Độ phủ càng lớn thì mức độ tin cậy càng cao
* Tích hợp Dependency Injection giảm độ phụ thuộc giữa các module với nhau và tăng khả năng viết unit-test.
* Đưa ứng dụng nên chợ Google Play Store để thu hút người sử dụng cũng như thêm nguồn thu nhập hỗ trợ việc phát triển
* Phát triển ứng dụng trên hệ điều hành di động của Apple là iOS nhằm hướng tới toàn bộ phân khúc người dùng phổ biến của di động thông minh.

Với bản thân trong tương lai việc học hỏi thêm nhiều kiến thức về lập trình và thiết kế đặc biệt liên quan đến phát triển những ứng dụng di động, đưa được những ứng dụng của bản thân nên chợ ứng dụng cũng là một ước mơ mà em đang hướng đến. Những ứng dụng di động đã và đang sẽ thay thế dần máy tính trở thành người bạn, người sẻ chia giúp đỡ đến những người sử dụng trong tương lai. Vì vậy mong muốn trở thành một nhà phát triển ứng dụng di động bản em sẽ cố gắng học hỏi và trau dồi thêm những kiến thức về cấu trúc dữ liệu và giải thuật, những công nghệ mới và cách triển khai và giải quyết vấn đề trong tương lai.

# Tài liệu tham khảo