

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

KHOA CÔNG NGHỆ THÔNG TIN 1



ĐỒ ÁN

TỐT NGHIỆP ĐẠI HỌC

ĐỀ TÀI:

Phát triển hệ thống đặt xe trực tuyến tích hợp thanh toán điện tử và theo dõi GPS thời gian thực trên ứng dụng di động

Giảng viên hướng dẫn : TS. NGUYỄN TẤT THẮNG

Sinh viên thực hiện : ĐINH MẠNH HOÀNG

Lớp : D20HTTT02

Mã sinh viên : B20DCCN271

Hệ : ĐẠI HỌC CHÍNH QUY

Hà Nội, tháng ... năm 202...

LỜI CẢM ƠN

(GỢI Ý Ở DƯỚI, CÓ THỂ CHỈNH SỬA)

Lời đầu tiên, em xin gửi lời cảm ơn chân thành và sâu sắc tới thầy TS. Nguyễn Tất Thắng, người đã trực tiếp hướng dẫn tận tình, chu đáo, chia sẻ những ý kiến và kinh nghiệm quý báu trong suốt quá trình em thực tập và thực hiện đồ án tốt.

Sau đó, em xin gửi lời cảm ơn các thầy, cô trong Học Viện nói chung và khoa CNTT1 nói riêng đã luôn nhiệt huyết, tận tình trong từng bài giảng và tạo điều kiện thuận lợi nhất cho em trong thời gian học tập và nghiên cứu tại trường Học Viện Công Nghệ Bưu Chính Viễn Thông.

Con xin được gửi lời cảm ơn tới bố mẹ và những người thân yêu đã luôn lo lắng, động viên, ủng hộ và tạo điều kiện cho con được học tập tốt. Là chỗ dựa tinh thần và những người tiếp sức cho con có được thành công trong cuộc sống.

Cuối cùng, tôi xin gửi lời chúc tốt đẹp nhất đến những người bạn của tôi và các thầy cô tham gia đợt bảo vệ tốt nghiệp trong khóa này. Chúc cho mọi người luôn vui vẻ và thành công trong cuộc sống.

Em xin chân thành cảm ơn!

Hà Nội, tháng ... năm ...

Sinh viên thực hiện

...

NHẬN XÉT, ĐÁNH GIÁ, CHO ĐIỂM

(Của người hướng dẫn)

[illegible]

Điểm:..... (bằng chữ:)

Đồng ý/Không đồng ý cho sinh viên bảo vệ trước hội đồng chấm đồ án tốt nghiệp?

....., ngày..... tháng..... năm 202...

CÁN BỘ GIẢNG VIÊN HƯỚNG DẪN

NHẬN XÉT, ĐÁNH GIÁ, CHO ĐIỂM

(Của giáo viên phản biện)

[illegible]

Điểm:..... (bằng chữ:)

Đồng ý/Không đồng ý cho sinh viên bảo vệ trước hội đồng chấm đồ án tốt nghiệp?

....., ngày..... tháng..... năm 202...

CÁN BỘ GIẢNG VIÊN PHẢN BIỆN

MỤC LỤC

LỜI CẢM ƠN	i
NHẬN XÉT, ĐÁNH GIÁ, CHO ĐIỂM.....	ii
NHẬN XÉT, ĐÁNH GIÁ, CHO ĐIỂM.....	iv
MỤC LỤC	vi
BẢNG VIẾT TẮT VÀ THUẬT NGỮ	viii
DANH SÁCH HÌNH VẼ.....	ix
DANH SÁCH BẢNG.....	x
MỞ ĐẦU	1
CHƯƠNG I. (CHÈN TÊN CHƯƠNG VIẾT HOA)	3
1.1 (Chèn tên mục viết thường).....	Error! Bookmark not defined.
1.1.1 (Chèn tên mục viết thường)	Error! Bookmark not defined.
1.1.2	Error! Bookmark not defined.
1.2	Error! Bookmark not defined.
1.2.1	Error! Bookmark not defined.
1.2.2	Error! Bookmark not defined.
1.3	Error! Bookmark not defined.
1.3.1	Error! Bookmark not defined.
1.3.2	Error! Bookmark not defined.
1.3 Kết luận Chương I	5
CHƯƠNG II. (CHÈN TÊN CHƯƠNG VIẾT HOA).....	6
2.1	6
2.1.1	6
2.1.2	6
2.2	7
2.2.1	7

2.2.2	7
2.3 Kết luận Chương II.....	17
CHƯƠNG III. (CHÈN TÊN CHƯƠNG VIẾT HOA).....	18
3.1	38
3.2	38
3.3 Kết luận Chương III	39
CHƯƠNG	40
.....	40
.....	40
3.3 Kết luận Chương	40
CHƯƠNG ... KẾT LUẬN	41
...1 Kết quả đạt được	41
...2 Hạn chế của hệ thống	41
...3 Định hướng phát triển hệ thống.....	41
DANH MỤC TÀI LIỆU THAM KHẢO.....	42

(Phần này chỉ cần chọn toàn bộ các dòng và click chuột trái, chọn Update Field, text tự động cập nhật, nếu sai format font thì format lại font: Times New Roman size 13. Để làm tự động như thế thì ở tiêu đề các đề mục, phải chọn đúng Heading. Tên chương là Heading 1, đề mục 2 số là Heading 2, 3 số là Heading 3 v.v.)

BẢNG VIẾT TẮT VÀ THUẬT NGỮ

TỪ VIẾT TẮT VÀ THUẬT NGỮ	Ý NGHĨA
API	Application Programming Interface
MQTT	Message Queuing Telemetry Transport
IoT	Internet of Things
Client	Người dùng
Server	Máy chủ

DANH SÁCH HÌNH VẼ

Hình 1.1 (Chèn tên hình vào đây).....	Error! Bookmark not defined.
Hình 1.(...) (Chèn tên hình vào đây)	Error! Bookmark not defined.
Hình 2.1 (Chèn tên hình vào đây).....	7
Hình 2.2	Error! Bookmark not defined.
Hình 2.3 (Chèn tên hình vào đây).....	Error! Bookmark not defined.
Hình 2.4 (Chèn tên hình vào đây).....	Error! Bookmark not defined.
Hình 3.1 (Chèn tên hình vào đây).....	38
Hình ... (Chèn tên hình vào đây).....	40

(Phần này chỉ cần chọn toàn bộ các dòng và click chuột trái, chọn Update Field, text tự động cập nhật, nếu sai format font thì format lại font: Times New Roman size 13. Để làm tự động như thế thì ở các hình, tên hình phải chọn là Heading 7)

DANH SÁCH BẢNG

Bảng 1.1 (Chèn tên bảng vào đây nếu có bảng) **Error! Bookmark not defined.**

(Phần này chỉ cần chọn toàn bộ các dòng và click chuột trái, chọn Update Field, text tự động cập nhật, nếu sai format font thì format lại font: Times New Roman size 13. Để làm tự động như thế thì ở các bảng, tên bảng phải chọn là Heading 8)

MỞ ĐẦU

Trong bối cảnh xã hội hiện đại, nhu cầu di chuyển ngày càng tăng cao và đòi hỏi sự tiện lợi, an toàn và hiệu quả. Các hệ thống đặt xe trực tuyến đã trở thành một phần không thể thiếu trong cuộc sống hàng ngày của người dân, đặc biệt là ở các đô thị lớn. Tuy nhiên, nhiều hệ thống đặt xe hiện nay vẫn gặp phải những hạn chế về tính năng và khả năng đáp ứng nhu cầu người dùng. Việc tích hợp thanh toán tự động và định vị GPS thời gian thực vào hệ thống đặt xe trực tuyến không chỉ nâng cao trải nghiệm người dùng mà còn tăng cường hiệu quả hoạt động cho các nhà cung cấp dịch vụ vận tải. Ngoài ra, việc tính toán giá thành hợp lý một cách minh bạch cũng là một vấn đề cần được giải quyết.

Với mong muốn đóng góp, học tập và cải thiện một số điểm trên, đề tài “Hệ thống đặt xe trực tuyến tích hợp thanh toán tự động và định vị GPS thời gian thực cho ứng dụng di động” đã được lựa chọn để nghiên cứu và phát triển. Đề tài này không chỉ mang tính ứng dụng cao mà còn mở ra nhiều cơ hội phát triển và mở rộng trong tương lai.

Nội dung của đồ án “Phát triển hệ thống đặt xe trực tuyến tích hợp thanh toán điện tử và theo dõi GPS thời gian thực trên ứng dụng di động” bao gồm các phần sau:

Chương I: GIỚI THIỆU

(Sơ lược Chương 1)

...

Chương II: CƠ SỞ LÝ THUYẾT

(Sơ lược Chương 2)

...

Chương III: THIẾT KẾ HỆ THỐNG

(Sơ lược Chương 3)

...

Chương IV: CÔNG NGHỆ VÀ CÔNG CỤ SỬ DỤNG

Chương V: TRIỂN KHAI HỆ THỐNG

Chương VI: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Kết luận

(Sơ lược Kết luận)

...

CHƯƠNG I. GIỚI THIỆU

Chương này sẽ trình bày lý do chọn đề tài, mục tiêu nghiên cứu, phạm vi nghiên cứu và cấu trúc tổng thể của báo cáo. Các phần tiếp theo của báo cáo sẽ đi sâu vào chi tiết từng khía cạnh của hệ thống, từ phân tích yêu cầu, thiết kế, triển khai đến kiểm thử và đánh giá.

1.1 Mục tiêu nghiên cứu:

Phát triển hệ thống đặt xe trực tuyến: Mục tiêu chính của đề án là xây dựng một ứng dụng di động cho phép khách hàng dễ dàng đặt xe, tài xế dễ dàng nhận chuyến xe. Hệ thống phải đảm bảo tính ổn định, nhanh chóng, và bảo mật trong quá trình xử lý các yêu cầu đặt xe.

Tích hợp tính năng thanh toán điện tử: Hệ thống sẽ cung cấp các phương thức thanh toán điện tử linh hoạt, giúp người dùng có thể thanh toán nhanh chóng và an toàn mà không cần dùng tiền mặt. Điều này sẽ giúp tăng tính tiện lợi và giảm các rủi ro liên quan đến giao dịch trực tiếp.

Theo dõi hành trình xe theo thời gian thực: Ứng dụng sẽ tích hợp tính năng GPS để người dùng có thể theo dõi vị trí của xe theo thời gian thực, giúp họ biết được xe đang ở đâu và có thể xem được thời gian xe tới đón. Điều này không chỉ cải thiện trải nghiệm người dùng mà còn hỗ trợ các nhà quản lý dịch vụ điều phối và giám sát phương tiện hiệu quả hơn.

Hệ thống quản lý: Hệ thống quản lý ra đời với mục đích giúp người quản lý có thể dễ dàng quản lý không chỉ thông tin người dùng và tài xế, mà thêm vào đó là thông tin vị trí và trạng thái của các xe thông qua hệ thống định vị GPS.

1.2 Phạm vi nghiên cứu:

Phát triển ứng dụng di động cho hệ thống đặt xe trực tuyến: Đề tài tập trung vào việc xây dựng một ứng dụng di động cho nền tảng iOS và Android, nơi người dùng có thể thực hiện các thao tác đặt xe, theo dõi vị trí xe và thanh toán điện tử. Các tính năng chính sẽ bao gồm đặt xe, theo dõi vị trí của xe, thanh toán, quản lý thông tin cá nhân, đánh giá...

Tích hợp hệ thống thanh toán điện tử: Ứng dụng sẽ hỗ trợ các phương thức thanh toán trực tuyến phổ biến như ví điện tử và các dịch vụ chuyển khoản trực tuyến.

Đề tài sẽ không đi sâu vào phát triển các cổng thanh toán mà sẽ tích hợp các giải pháp thanh toán hiện có.

Theo dõi GPS thời gian thực: Hệ thống sẽ tích hợp công nghệ định vị GPS để cung cấp cho người dùng khả năng theo dõi vị trí của xe trong suốt hành trình. Điều này yêu cầu sự liên kết giữa ứng dụng di động và máy chủ.

Giới hạn của đề tài:

- Đề tài chỉ tập trung phát triển hệ thống trong khu vực, không bao gồm giải pháp Quốc Tế.
- Hạn chế về bản đồ: Sử dụng API bản đồ của Google Map, nhiều thứ sẽ bị hạn chế như: số lượt tìm kiếm, các địa điểm, template của bản đồ sẽ không được làm một cách đầy đủ.
- Hạn chế về Cloud: Sử dụng cloud của Firebase sẽ có hạn chế về chi phí và tốc độ xử lý.
- Hạn chế về phần cứng: Chưa phát triển được thiết bị kiểm soát GPS cho phương tiện của taxi, vì vậy việc kiểm soát GPS hiện tại chỉ kiểm soát thông qua thiết bị di động

Các chức năng nâng cao như phân tích mật độ khách hàng, thông tin khách hàng, điều phối xe thông minh sẽ nằm ngoài phạm vi dự án

1.3 Tính cấp thiết của đề tài:

Thời đại công nghệ phát triển, mọi dịch vụ đều chuyển hướng sang trực tuyến, và dịch vụ di chuyển cũng không ngoại lệ. Hàng loạt dịch vụ đặt xe trực tuyến được ra đời nhằm mang đến sự tiện lợi và nhanh chóng cho người dùng, đặc biệt là ở các đô thị lớn.

Các dịch vụ đặt xe truyền thống thường sẽ sử dụng tiền mặt để thanh toán, hoặc chuyển khoản qua tài khoản ngân hàng của taxi để thanh toán, gây bất cập nếu khách hàng không mang tiền mặt hay taxi không có tiền trả lại. Vì vậy, hệ thống ra đời tích hợp thanh toán trực tuyến giúp thuận tiện hơn trong việc thanh toán.

Các dịch vụ đặt xe trực tuyến thường sẽ nhận cuộc gọi từ khách hàng, sau đó người quản lý sẽ khoanh vùng các taxi hoạt động trong khu vực của khách hàng và

báo đảm tới xe của các tài xế đó, sau đó tài xế sẽ báo lại qua đảm để nhận chuyển xe. Điều này vẫn làm một cách thủ công và không khoa học, cùng như khách hàng sẽ không thể theo dõi vị trí của xe. Vì vậy, hệ thống đưa ra công nghệ định vị GPS thời gian thực để giải quyết những vấn đề trên, vừa giúp người quản lý dễ dàng quản lý, vừa giúp khách hàng dễ dàng theo dõi chuyển xe của mình.

Hệ thống cung cấp thêm hệ thống quản lý cho phép người quản lý có thể theo dõi vị trí và trạng thái của các tài xế theo thời gian thực, giúp tăng sự chặt chẽ trong việc vận hành của hệ thống, giúp nâng cao chất lượng dịch vụ cũng như xử lý kịp thời nếu có vấn đề xảy ra.

1.4 Kết luận Chương I

Trong chương 1, đồ án đã nêu ra Mục tiêu, phạm vi nghiên cứu cũng như tính cấp thiết của đề tài “Phát triển hệ thống đặt xe trực tuyến tích hợp thanh toán điện tử và theo dõi GPS thời gian thực cho ứng dụng di động”. Với phạm vi phát triển ứng dụng di động cho nền tảng iOS và Android, tích hợp hệ thống thanh toán điện tử, và công nghệ định vị GPS thời gian thực, hệ thống đặt xe trực tuyến sẽ đáp ứng được hầu hết các nhu cầu cơ bản của một hệ thống đặt xe trực tuyến. Tuy có những giới hạn nhưng đây vẫn là đề tài mang tính cấp thiết và ứng dụng cao cho đời sống. Những phần tiếp theo của tài liệu sẽ đi sâu vào chi tiết các yêu cầu kỹ thuật và chức năng cụ thể của hệ thống, nhằm cung cấp một cái nhìn toàn diện và rõ ràng hơn cho các bên liên quan và các nhà phát triển.

CHƯƠNG II. CƠ SỞ LÝ THUYẾT

Chương II sẽ đưa ra các cơ sở lý thuyết về các vấn đề liên quan đến đề tài, thêm vào đó là công nghệ sử dụng để làm rõ hơn nhưng vấn đề đó.

2.1 Tổng quan về dịch vụ đặt xe trực tuyến

2.1.1 Lịch sử phát triển của dịch vụ đặt xe nói chung

Từ xa xưa, nhu cầu di chuyển đã là một trong những nhu cầu thiết yếu trong cuộc sống. Từ những thế kỷ 10-11, những dịch vụ di chuyển sơ khai như xe kéo, xe thò đã phát triển và là một phần không thể thiếu được trong cuộc sống người dân lúc bấy giờ. Dần dần, công nghệ kỹ thuật phát triển, phương tiện di chuyển cũng phát triển dẫn tới dịch vụ vận chuyển cũng phát triển theo.

Dịch vụ taxi truyền thống xuất hiện từ cuối thế kỷ 19 và đầu thế kỷ 20, đáp ứng nhu cầu di chuyển cá nhân của người dân trong các thành phố lớn. Những chiếc xe đầu tiên được sử dụng làm taxi là xe ngựa và sau đó là ô tô. Việc gọi taxi truyền thống thường diễn ra qua điện thoại hoặc trực tiếp tại các bãi đậu xe, nơi các xe taxi chờ đợi khách hàng. Tuy nhiên, dịch vụ này có nhiều hạn chế về thời gian chờ đợi, tính tiện lợi, và an toàn.

Vào đầu những năm 2000, với sự phát triển nhanh chóng của công nghệ di động và internet, các ứng dụng đặt xe trực tuyến bắt đầu xuất hiện. Các ứng dụng này tận dụng công nghệ GPS, bản đồ số và các phương thức thanh toán đa dạng để cung cấp dịch vụ đặt xe nhanh chóng, tiện lợi và an toàn hơn so với dịch vụ taxi truyền thống.

2.1.2 Những hệ thống đặt xe trực tuyến phổ biến tại Việt Nam

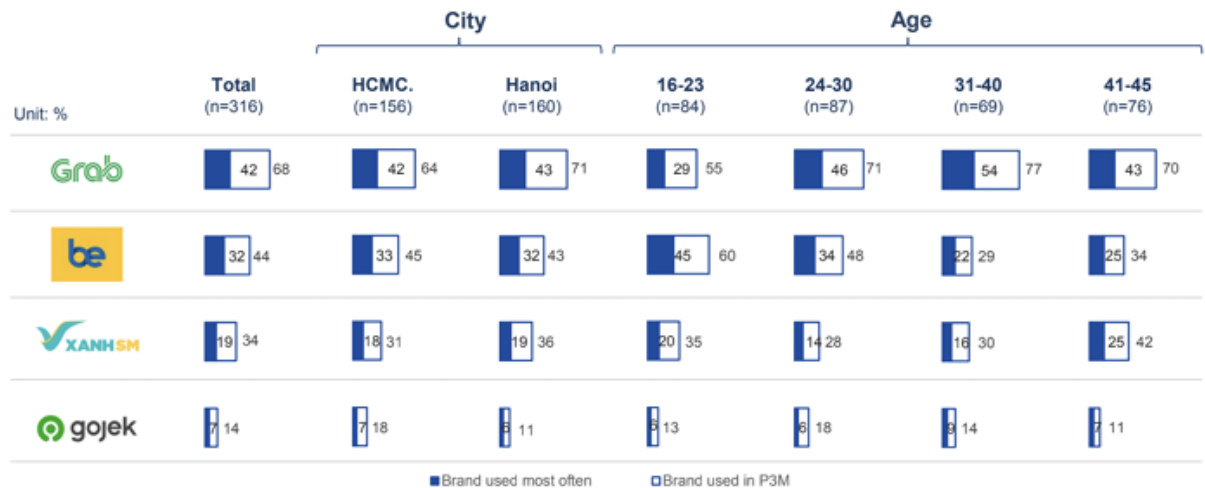
Grab: Là một sản phẩm của công ty Grab Holdings Inc, ra đời từ năm 2012, và hiện đang là hệ thống đặt xe trực tuyến có thị phần cao nhất tại Việt Nam.

Gojex là sản phẩm của PT Aplikasi Karya Anak Bangsa (Indonesia), được ra mắt vào năm 2010, tuy dẫn đầu vào thị trường Việt Nam sau nhưng được đánh giá là một trong những đối thủ nặng ký của Grab tại thị trường Việt Nam. Tuy nhiên, vào ngày 16/09/2024, Gojex công bố rút lui khỏi Việt Nam.

Be của công ty Cổ phần BE GROUP, là một ứng dụng đặt xe được phát triển bởi công ty Việt Nam. Ứng dụng này cung cấp các dịch vụ như beBike (xe ôm), beCar (xe ô tô), beDelivery (giao hàng), và beFinancial (dịch vụ tài chính). Be nhanh chóng

trở thành một đối thủ cạnh tranh mạnh mẽ trên thị trường nhờ vào sự hiểu biết sâu sắc về nhu cầu và thói quen của người tiêu dùng Việt Nam.

Xanh SM, chú ngựa ô trong thị trường đặt xe trực tuyến ở Việt Nam, tuy nhiên với lợi thế “chủ nhà”, Xanh SM cũng nhanh chóng nhập cuộc và cũng trở thành thương hiệu khá quen thuộc với người dân Việt Nam.



Hình 2.1 Mức độ phổ biến các thương hiệu gọi xe tại Việt Nam

Tham khảo Q&M

2.2 Hệ thống thanh toán điện tử

2.2.1 Hệ thống thanh toán điện tử

Hệ thống thanh toán điện tử (Electronic Payment System - EPS) là một phương thức cho phép người tiêu dùng thực hiện các giao dịch tài chính qua mạng Internet mà không cần sử dụng tiền mặt. Các giao dịch này có thể bao gồm thanh toán hóa đơn, mua sắm trực tuyến, chuyển tiền, và nhiều dịch vụ tài chính khác....

2.2.2 Các thành phần của hệ thống thanh toán điện tử

Người dùng: Là cá nhân hoặc tổ chức thực hiện giao dịch thanh toán

Ngân hàng hoặc tổ chức tài chính: Cung cấp dịch vụ tài khoản và hỗ trợ các giao dịch thanh toán.

Nhà cung cấp dịch vụ thanh toán: Là các công ty trung gian, như PayPal, Stripe, hoặc các công ty thanh toán điện tử khác, giúp xử lý và thực hiện giao dịch giữa người mua và người bán.

Cổng thanh toán: Là phần mềm cho phép chuyển đổi thông tin thanh toán giữa người dùng và ngân hàng. Cổng thanh toán giúp bảo mật thông tin và mã hóa dữ liệu trong quá trình giao dịch.

Hệ thống bảo mật: Đảm bảo tính an toàn cho các giao dịch thông qua các phương thức mã hóa, xác thực người dùng, và giám sát gian lận.

2.2.3 Các phương thức thanh toán điện tử:

Thẻ tín dụng và thẻ ghi nợ: Là phương thức phổ biến cho thanh toán trực tuyến. Người dùng có thể nhập thông tin thẻ để thực hiện giao dịch.

Ví điện tử (E-wallet): Là ứng dụng cho phép người dùng lưu trữ tiền điện tử và thực hiện thanh toán nhanh chóng. Các ví điện tử phổ biến như MoMo, ZaloPay, và GrabPay, hoặc có thể là bất cứ ví nào trong các ứng dụng di động.

Chuyển khoản ngân hàng trực tuyến: Người dùng có thể chuyển tiền từ tài khoản ngân hàng của mình đến tài khoản ngân hàng của người nhận thông qua Internet. Đây hiện tại vẫn đang là phương pháp thanh toán phổ biến nhất.

Thanh toán qua mã QR: Người dùng quét mã QR để thực hiện thanh toán, thường được sử dụng trong các giao dịch bán lẻ.

Thanh toán di động: Sử dụng các ứng dụng trên điện thoại di động để thực hiện giao dịch, chẳng hạn như Apple Pay hoặc Google Pay. Phương pháp này thường được sử dụng cho những ứng dụng trả tiền trên AppStore hay Google Play.

2.3 GPS (Global Positioning System):

GPS (Global Positioning System) là một hệ thống định vị toàn cầu sử dụng vệ tinh để xác định vị trí chính xác của một thiết bị bất kỳ trên mặt đất. Hệ thống này được phát triển bởi Bộ Quốc phòng Hoa Kỳ và hiện nay đã trở thành một công nghệ phổ biến được sử dụng rộng rãi trong nhiều lĩnh vực, bao gồm vận tải, logistics, du lịch và các ứng dụng di động.

2.3.1 Nguyên lý hoạt động của GPS

Vệ tinh GPS: Hệ thống bao gồm khoảng 24 vệ tinh hoạt động trong quỹ đạo quanh trái đất. Các vệ tinh này phát sóng tín hiệu định vị đến các thiết bị nhận GPS.

Thiết bị nhận GPS: Thiết bị này (chẳng hạn như điện thoại di động, máy định vị hoặc ô tô) nhận tín hiệu từ ít nhất 4 vệ tinh để xác định vị trí chính xác của nó trên bề mặt trái đất thông qua phép tính khoảng cách.

Tính toán vị trí: Khi thiết bị nhận được tín hiệu từ vệ tinh, nó tính toán khoảng cách đến từng vệ tinh dựa trên thời gian tín hiệu được phát ra và thời gian tín hiệu được nhận. Từ đó, thiết bị có thể xác định vị trí 2D (tọa độ) hoặc 3D (tọa độ + độ cao).

2.3.2 Sự chính xác của GPS

Các máy thu GPS ngày nay cực kì chính xác, nhờ vào thiết kế nhiều kênh hoạt động song song của chúng. Các máy thu 12 kênh song song (của Garmin) nhanh chóng khóa vào các quả vệ tinh khi mới bật lên và chúng duy trì kết nối bền vững, thậm chí trong tán lá rậm rạp hoặc thành phố với các toà nhà cao tầng. Trạng thái của khí quyển và các nguồn gây sai số khác có thể ảnh hưởng tới độ chính xác của máy thu GPS. Các máy thu GPS có độ chính xác trung bình trong vòng 15 mét.

Các máy thu mới hơn với khả năng WAAS (Wide Area Augmentation System) có thể tăng độ chính xác trung bình tới dưới 3 mét. Không cần thêm thiết bị hay mất phí để có được lợi điểm của WAAS. Người dùng cũng có thể có độ chính xác tốt hơn với GPS vi sai (Differential GPS, DGPS) sửa lỗi các tín hiệu GPS để có độ chính xác trong khoảng 3 đến 5 mét. Cục Phòng vệ Bờ biển Mỹ vận hành dịch vụ sửa lỗi này. Hệ thống bao gồm một mạng các đài thu tín hiệu GPS và phát tín hiệu đã sửa lỗi bằng các máy phát hiệu. Để thu được tín hiệu đã sửa lỗi, người dùng phải có máy thu tín hiệu vi sai bao gồm cả ăng-ten để dùng với máy thu GPS của họ.

2.4 Công nghệ sử dụng:

2.4.1 Flutter:

Flutter là một framework phát triển ứng dụng di động mã nguồn mở được phát triển bởi Google. Thư viện này cho phép các nhà phát triển xây dựng các ứng dụng di động cho cả hệ điều hành IOS và Android với duy nhất một mã nguồn.

Flutter sử dụng ngôn ngữ lập trình Dart – một ngôn ngữ cũng được phát triển bởi Google. Flutter cung cấp một loạt các công cụ, thư viện và hỗ trợ phong phú, giúp nhà phát triển xây dựng các ứng dụng phức tạp và tùy chỉnh một cách dễ dàng. Đối với ứng dụng di động, Flutter cung cấp Hot Reload, cho phép nhà phát triển ngay lập tức thấy được sự thay đổi trong ứng dụng mà không cần phải khởi động lại toàn bộ ứng dụng. Điều này giúp tăng tốc quá trình phát triển và thử nghiệm.

Nhờ sự tiện lợi và dễ sử dụng, Flutter trở nên rất phổ biến, có hơn 30.000 sao trên trang Github của mình. Khi Flutter được giới thiệu, đã có 1 lượng lớn các lập trình

viên cũng như những nhà phát triển Flutter nói riêng đưa ra sự quan tâm về nó. Vì vậy, cộng đồng phát triển của Flutter vô cùng lớn và tiềm năng.

Tóm lại, Flutter là một framework khá toàn diện, linh hoạt và mang tính hiệu quả cao trong việc phát triển ứng dụng di động đa nền tảng. Với sự hỗ trợ từ Google nói riêng và cộng đồng ủng hộ Flutter nói chung, các lập trình viên có thể tự tin xây dựng những ứng dụng chất lượng, đáp ứng nhu cầu ngày càng cao của người dùng.

2.4.2 Nodejs:

Node.js là một môi trường thực thi mã nguồn mở mạnh mẽ, được xây dựng trên JavaScript runtime của Chrome, V8. Với khả năng chạy mã JavaScript ở phía máy chủ, Node.js mang đến sự linh hoạt vượt trội trong việc phát triển các ứng dụng mạng và API đa dạng. Một trong những lý do quan trọng khiến tôi chọn Node.js để phát triển Back-end cho ứng dụng di động là khả năng xử lý nhanh chóng và hiệu quả, nhờ vào mô hình làm việc không đồng bộ và sự kiện.

Node.js hoạt động trên mô hình single-threaded, nhưng với non-blocking I/O, nó có thể xử lý nhiều yêu cầu I/O một cách hiệu quả mà không làm chậm quá trình thực thi. Điều này đặc biệt quan trọng đối với các ứng dụng thời gian thực và các dự án đòi hỏi hiệu suất cao. Khả năng xử lý nhiều kết nối đồng thời mà không gây ra tình trạng tắc nghẽn là một điểm mạnh nổi bật của Node.js, khiến nó trở thành lựa chọn lý tưởng cho việc phát triển backend của các ứng dụng di động, nơi mà hiệu suất và khả năng đáp ứng nhanh chóng là yếu tố then chốt.

Ngoài ra, Node.js còn có một cộng đồng mạnh mẽ và tích cực. Cộng đồng này không chỉ cung cấp nền tảng để phát triển ứng dụng nhanh chóng mà còn hỗ trợ tích hợp dễ dàng với nhiều thư viện và module mở rộng. Điều này giúp tiết kiệm thời gian và công sức của nhà phát triển, cho phép tập trung vào việc xây dựng các tính năng quan trọng và cải thiện trải nghiệm người dùng.

Node.js thường được sử dụng để xây dựng các ứng dụng mạng, API, và dịch vụ Back-end, nhờ vào khả năng mở rộng và hiệu suất cao. Sự linh hoạt của Node.js còn nằm ở việc dễ dàng tích hợp với các công nghệ khác, cho phép tạo ra các giải pháp phức tạp và mạnh mẽ. Việc sử dụng Node.js cho backend của ứng dụng điện thoại không chỉ giúp đảm bảo hiệu suất cao mà còn mang lại khả năng mở rộng dễ dàng, đáp ứng tốt nhu cầu phát triển trong tương lai.

Với tất cả những lợi ích này, Node.js là một lựa chọn hoàn hảo cho việc phát triển backend của ứng dụng điện thoại, giúp tối ưu hóa hiệu suất và đảm bảo trải nghiệm người dùng mượt mà. Sự hỗ trợ mạnh mẽ từ cộng đồng và khả năng tích hợp linh hoạt cũng là những yếu tố quan trọng giúp tôi tin tưởng vào lựa chọn này.

2.4.3 PostgreSQL:

PostgreSQL là một hệ thống quản trị cơ sở dữ liệu quan hệ-đối tượng (object-relational database management system) có mục đích chung, hệ thống cơ sở dữ liệu mã nguồn mở tiên tiến nhất hiện nay.

PostgreSQL sở hữu một hệ tính năng đa dạng giúp hỗ trợ các nhà phát triển xây dựng app, các nhà quản trị bảo vệ toàn vẹn dữ liệu, và tạo ra một môi trường chịu lỗi fault-tolerant giúp quản lý dữ liệu bất kể tập dữ liệu lớn hay nhỏ. Bên cạnh hệ thống nguồn mở và miễn phí, PostgreSQL cũng có khả năng mở rộng tuyệt vời. Chúng ta có thể định nghĩa các kiểu dữ liệu riêng, xây dựng các hàm tùy chỉnh, hay viết mã từ các ngôn ngữ lập trình khác nhau mà không cần biên dịch lại cơ sở dữ liệu.

Một số điểm nổi bật của PostgreSQL:

- Bảo mật: Xác thực: GSSAPI, SSPI, LDAP, SCRAM-SHA-256, Certificate và các hình thức khác
- Hiệu suất cao: Có thể xử lý được tập dữ liệu lớn và đa dạng, đồng thời cung cấp các tính năng như phân vùng bảng, lập danh mục nâng cao và truy vấn đọc song song giúp tăng hiệu suất truy vấn.
- Tính linh hoạt: Cho phép định nghĩa các kiểu dữ liệu tùy chỉnh, xây dựng các hàm tùy chỉnh và viết mã từ các ngôn ngữ lập trình khác nhau. Điều này giúp cho các nhà phát triển có thể tùy chỉnh cơ sở dữ liệu cho phù hợp với nhu cầu của ứng dụng.
- Hỗ trợ trên nhiều hệ điều hành
- Cập nhật thường xuyên: Được phát triển và cập nhật thường xuyên, với các phiên bản mới được phát hành để cải thiện tính năng, tăng hiệu suất và sửa các lỗi bảo mật.

2.4.4 MQTT Protocol (Message Queuing Telemetry Transport):

- MQTT là một phương thức truyền tin tiêu chuẩn hay một bộ các tiêu chuẩn sử dụng cho việc giao tiếp giữa các máy với nhau. Các thiết bị thông minh, các thiết bị IoT (Internet of Things) thường cần phải gửi và nhận thông tin

với một lượng thông tin và băng thông nhỏ, vì vậy cần phải dùng MQTT để đảm bảo sự dễ dàng trong việc triển khai cũng như sự hiệu quả trong việc truyền tin. Đây cũng chính là một phương thức truyền tin thời gian thực đang được sử dụng rộng rãi hỗ trợ truyền tin từ các thiết bị tới đám mây (Cloud) và từ đám mây tới các thiết bị.

- Nguyên tắc hoạt động của MQTT: Giao thức MQTT hoạt động dựa trên nguyên tắc mô hình publish/subscribe (xuất bản/đăng ký). Trong giao tiếp mạng truyền thống, client và server giao tiếp trực tiếp với nhau. Client (người dùng) yêu cầu tài nguyên hoặc dữ liệu từ server (máy chủ), sau đó server xử lý và gửi lại phản hồi. Tuy nhiên, MQTT sử dụng mô hình publish/subscribe để tách rời bên gửi thông điệp (nơi thông điệp được gửi) với bên nhận thông điệp (nơi thông điệp được nhận). Thay vào đó, một thành phần thứ ba gọi là message broker (bên trung gian trung chuyển thông điệp) xử lý việc giao tiếp giữa bên gửi và bên nhận thông điệp. Nhiệm vụ của broker là lọc tất cả các thông điệp đến từ bên gửi và chuyển chúng đúng cách đến bên nhận thông điệp. Broker tách rời bên gửi và bên nhận thông điệp.
- Các thành phần của MQTT:
 - MQTT Client: là bất kỳ thiết bị nào chạy thư viện MQTT, có thể gửi hoặc nhận thông điệp
 - MQTT Broker: là hệ thống xử lý việc giao tiếp giữa các client, nhận, lọc và chuyển tiếp thông điệp đến các client đã đăng ký nhận thông điệp
 - MQTT Connection: là kết nối của client và broker qua giao thức TCP/IP.
- Hoạt động của MQTT
 - MQTT Topic: đề cập đến các từ khoá được sử dụng để lọc thông điệp cho các MQTT Client. Các topic được tổ chức theo thứ bậc, tương tự như đường dẫn đến tệp hoặc thư mục.
 - Quá trình xuất bản bản tin qua MQTT: các MQTT client sẽ xuất bản thông điệp qua các topic dưới dạng byte. Dữ liệu này có thể được định dạng là văn bản, nhị phân, tệp XML hoặc JSON
 - Quá trình đăng ký MQTT: các MQTT Client sẽ đăng ký các topic để có thể nhận được các thông điệp được xuất bản từ các Client qua thông điệp đó.

Khi một Client gửi thông điệp thì tất cả Client (kể cả chính nó) đăng ký nhận thông điệp của topic tương ứng sẽ nhận được thông điệp mà Client đó đã gửi đi.

- Như vậy, MQTT Protocol là một giao thức truyền tin tin cậy và hiệu quả trong việc truyền thông điệp cho các thiết bị, đặc biệt là thiết bị IoT. Vì vậy, việc sử dụng MQTT cho đồ án là hoàn toàn hợp lý và có cơ hội phát triển trong tương lai.

2.4.5 Phương pháp phát triển: Mô hình Waterfall

Mô hình Waterfall (Mô hình thác nước) là một phương pháp phát triển phần mềm quản lý dựa trên quy trình thiết kế tuần tự và liên tiếp. Trong mô hình waterfall, các giai đoạn nối tiếp nhau, mỗi giai đoạn được bắt đầu khi giai đoạn trước của nó kết thúc.

Mô hình Waterfall gồm 6 giai đoạn:

1. *Giai đoạn yêu cầu (Requirement Analysis):*

Giai đoạn yêu cầu là giai đoạn nhóm thực hiện sẽ thực hiện việc tìm kiếm yêu cầu của các bên liên quan phù hợp với dự án để hiểu rõ ràng về những thứ cần xây dựng cũng như những chức năng của hệ thống.

Giai đoạn này nhóm sẽ đi tìm hiểu các vấn đề như nhu cầu kinh doanh cần giải quyết, yêu cầu của người dùng đối với hệ thống hay các ràng buộc và rủi ro đi kèm... nhằm mục đích cuối cùng là hiểu rõ mục tiêu và yêu cầu của dự án.

2. *Giai đoạn thiết kế (Design):*

Đây là giai đoạn sau khi đã xác định yêu cầu một cách tường minh, đội ngũ phát triển bắt đầu thiết kế kiến trúc hệ thống. Bản thiết kế này gồm giao diện người dùng, các module sử dụng phần mềm, kiến trúc cơ sở dữ liệu hay môi trường làm việc. Chính nó sẽ mô tả chi tiết logic của hệ thống sẽ được thực thi thế nào.

3. *Giai đoạn phát triển (Development):*

Giai đoạn phát triển là giai đoạn các thiết kế ở giai đoạn trên được làm một cách chi tiết và thực tế hơn, các thuật toán, cấu trúc dữ liệu cũng như giao diện sẽ được xác định rõ ràng và kết nối liên kết một cách liên mạch với nhau đảm bảo những bước hoạt động chính của hệ thống.

4. *Giai đoạn kiểm thử (Testing):*

Sau khi giai đoạn phát triển thực hiện xong, đội ngũ phát triển sẽ phải tiếp tục tiến hành hoạt động phát hiện và sửa các lỗi trong phần mềm. Các loại kiểm thử như Kiểm thử đơn vị (Unit testing), Kiểm thử tích hợp (Integration testing) và kiểm thử hệ thống (System testing) sẽ được thực hiện để đảm bảo chất lượng sản phẩm.

5. *Giai đoạn triển khai (Deployment):*

Sau khi đã kiểm thử và đảm bảo tính ổn định của hệ thống, đội ngũ tiến hành triển khai hệ thống vào môi trường thực - môi trường người dùng để bắt đầu sử dụng.

6. *Giai đoạn bảo trì (Maintenance):*

Giai đoạn bảo trì bao gồm việc sửa chữa các lỗi phát sinh sau khi phần mềm đã được phát hành, cũng như cập nhật và nâng cấp phần mềm khi cần thiết. Giai đoạn này trong các hệ thống phần mềm là một giai đoạn kéo dài, có thể là tới lúc hệ thống không còn được sử dụng và hoạt động nữa, nhằm đảm bảo hệ thống hoạt động trơn tru và hành động kịp thời với bất cứ tình huống nào xảy ra với hệ thống.



Hình 2.2 Mô hình Waterfall

Ưu điểm của mô hình Waterfall:

- Đơn giản và dễ sử dụng: Là mô hình có logic dễ hiểu, phù hợp với đội ngũ vừa và nhỏ, dễ dàng quản lý và theo dõi tiến độ của dự án
- Tính rõ ràng: Đây là 1 trong những yếu tố nổi bật của mô hình này, công việc diễn ra theo tuần tự từng giai đoạn nên các thao tác sẽ không bị chồng chéo, không bị quá tải cho bộ máy hệ thống

- Quản lý dễ dàng: Mỗi giai đoạn có một kết quả cụ thể và được hoàn thành trước khi chuyển sang giai đoạn tiếp theo. Vì vậy, việc quản lý sẽ trở nên dễ dàng hơn ở tất cả các giai đoạn

Nhược điểm của mô hình Waterfall:

- Thiếu linh hoạt: Một khi dự án đã chuyển sang giai đoạn sau, rất khó để quay lại và thay đổi yêu cầu hoặc thiết kế ban đầu.
- Khó khăn trong việc phản ứng với thay đổi: Nếu có sự thay đổi về yêu cầu hoặc điều kiện thị trường, mô hình này không thể đáp ứng kịp thời.
- Rủi ro cao: Do toàn bộ sản phẩm chỉ được kiểm thử sau khi hoàn thành, nếu phát hiện lỗi ở giai đoạn sau, việc sửa chữa sẽ rất tốn kém và mất thời gian.

2.5 Chi tiết hệ thống:

2.5.1 Kiến trúc hệ thống tổng quan: Hệ thống gồm các thành phần chính:

- Frontend: Xây dựng bằng Flutter cho IOS và Androi dành cho tài xế và khách hàng, xây dựng ReactJS cho Web dành cho người quản trị
- Backend: Sử dụng Nodejs xử lý logic hệ thống và giao tiếp với cơ sở dữ liệu
- Database: Sử dụng PostgreSQL lưu trữ toàn bộ dữ liệu hệ thống
- Real-time Communication: Dùng MQTT để hỗ trợ truyền tin thời gian thực cho việc cập nhật vị trí GPS và thông báo

2.5.2 Sơ đồ kiến trúc hệ thống:

2.5.3 Các công nghệ hỗ trợ và tích hợp:

2.5.3.1 Google Maps API:

- Cách thức sử dụng: Tạo một tài khoản Google Cloud, sau đó tạo 1 Dự án. Tiếp đến sẽ tạo API Key bằng cách vào phần "Credentials" để tạo theo hướng dẫn. Sau đó sẽ sử dụng API Key này để gọi tới các API tương ứng cho các dịch vụ của Google.
- Các dịch vụ sử dụng:
- **Maps API:** Đây là dịch vụ của Google sử dụng để hiển thị bản đồ cho các thiết bị, và trong hệ thống sử dụng **Maps SDK for IOS** để hiển thị bản đồ cho thiết bị IOS và **Maps SDK for Android** để hiển thị bản đồ cho thiết bị Android cũng như **Maps Javascript API** để hiển thị bản đồ cho web
- **Routes API:** Đây là dịch vụ chỉ đường của Google, sử dụng địa chỉ dưới dạng toạ độ của 2 địa điểm để tìm ra đường đi tối ưu nhất giữa 2 địa điểm đó. Đồng thời, dịch vụ này cũng cung cấp thời gian ước tính, khoảng cách cũng như chỉ dẫn đi đường cho tuyến đường được chọn.
- Các thành phần của dịch vụ
 - Vị trí điểm xuất phát
 - Vị trí điểm đích
 - Phương thức di chuyển (DRIVE, BICYCLE, WALK, TRANSIT)
 - Ưu tiên lộ trình (nếu có):
 - TRAFFIC_AWARE: Lộ trình tối ưu dựa trên tình trạng giao thông thời gian thực

TRAFFIC_AWARE_OPTIMAL: Lộ trình tối ưu nhất dựa trên giao thông và thời gian

UNSPECIFIED: không phụ thuộc vào giao thông

Các yếu tố khác:

avoidTolls: Tránh các tuyến đường có thu phí

avoidHighways: Tránh các tuyến đường cao tốc

avoidFerries: Tránh các tuyến đường có phà

- Các tiêu chí chỉ đường của Routes API để đưa ra tuyến đường hiệu quả nhất:

1. Thời gian di chuyển (yếu tố chính)
2. Khoảng cách
3. Số vòng di chuyển

- **Places API:** Dịch vụ này cho phép truy cập vào hơn 100 triệu địa điểm trên thế giới, và trong hệ thống nó còn được sử dụng để tìm kiếm địa điểm.

2.5.3.2 Dịch vụ định vị GPS:

- Sử dụng dịch vụ Geolocator của Flutter để định vị được vị trí của Thiết bị đang sử dụng. Dịch vụ này được sử dụng khi:
 - Tài xế bật trạng thái hoạt động, dữ liệu về vị trí sẽ được gửi lên Hệ thống
 - Trong chuyến xe, dữ liệu vị trí cũng như tuyến đường dự kiến của tài xế sẽ được gửi cho khách hàng
- Phương thức truyền thông tin dịch vụ GPS là thông qua giao thức MQTT đảm bảo việc truyền tin thời gian thực.

2.5.3.3 Dịch vụ đánh giá thời tiết (OpenWeather API): Đây là API sử dụng để đánh giá mức độ thời tiết của một vị trí trong thời gian thực, trong hệ thống sử dụng để tính giá của chuyến xe.

2.5.4 Các nội dung khác:

2.5.4.1 Bảng tính giá chuyến xe:

• **Bảng giá cước cơ bản**

Dịch vụ	Giá cước 2km đầu tiên (tối thiểu)	Giá cước mỗi km tiếp theo	Giá cước tính theo thời gian di chuyển (sau 2km đầu tiên)
Ô tô 4 chỗ	29.000đ	10.000đ	450đ/ phút
Ô tô 7 chỗ	34.000đ	13.000đ	550đ/ phút
Xe máy	13.500đ	4.300đ	350đ/ phút

• **Các loại phụ phí khác**

Loại phí	Giải thích		Giá
Phí nền tảng	Phí sử dụng nền tảng ứng dụng		4.000đ cho ô tô 2.000đ cho xe máy
Phụ phí thời tiết xấu	Phụ phí này sinh ra khi thời tiết không tốt	Mưa (to)	10%/Tổng tiền
		Thời tiết nóng (>38 độ C)	10%/Tổng tiền
Phí cầu đường, vé xe	Chi phí khi đi đường gặp phải		Thoả thuận giữa tài xế và người đặt xe

Công thức tính cước: *Giá cước tối thiểu + Giá cước mỗi km tiếp theo (sau 2km đầu tiên) + Giá cước tính theo thời gian di chuyển (sau 2km đầu tiên) + Phụ phí*

- ***Ví dụ: Chuyển ô tô 4 chỗ đi 10km, thời gian di chuyển sau 2km đầu tiên là 30 phút, thời tiết mưa to, không có phí cầu đường***
=> Giá cước: $29.000 + 8 * 10000 + 30 * 450 + 50000 + 4000 = 135.500đ$

2.3 Kết luận Chương II

Như vậy, chương II đã đưa ra rất chi tiết về những Cơ sở lý thuyết sẽ được sử dụng trong hệ thống. Ngoài ra, chương này không chỉ cung cấp nền tảng lý thuyết mà còn định hình rõ ràng các thành phần công nghệ và cách chúng phối hợp để tạo nên một hệ thống đặt xe trực tuyến hiện đại và hiệu quả. Đây sẽ là cơ sở vững chắc để triển khai các nội dung trong các chương tiếp theo, tập trung vào thiết kế chi tiết và triển khai thực tế hệ thống.

CHƯƠNG III. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

Chương 3 sẽ đưa ra chi tiết phân tích và thiết kế của hệ thống, bao gồm các biểu đồ, sơ đồ, các thiết kế chi tiết cho từng phần, từng Module của hệ thống.

(Chưa hoàn thiện)

3.1 Tổng quan hệ thống

3.1.1 Môi trường hệ thống

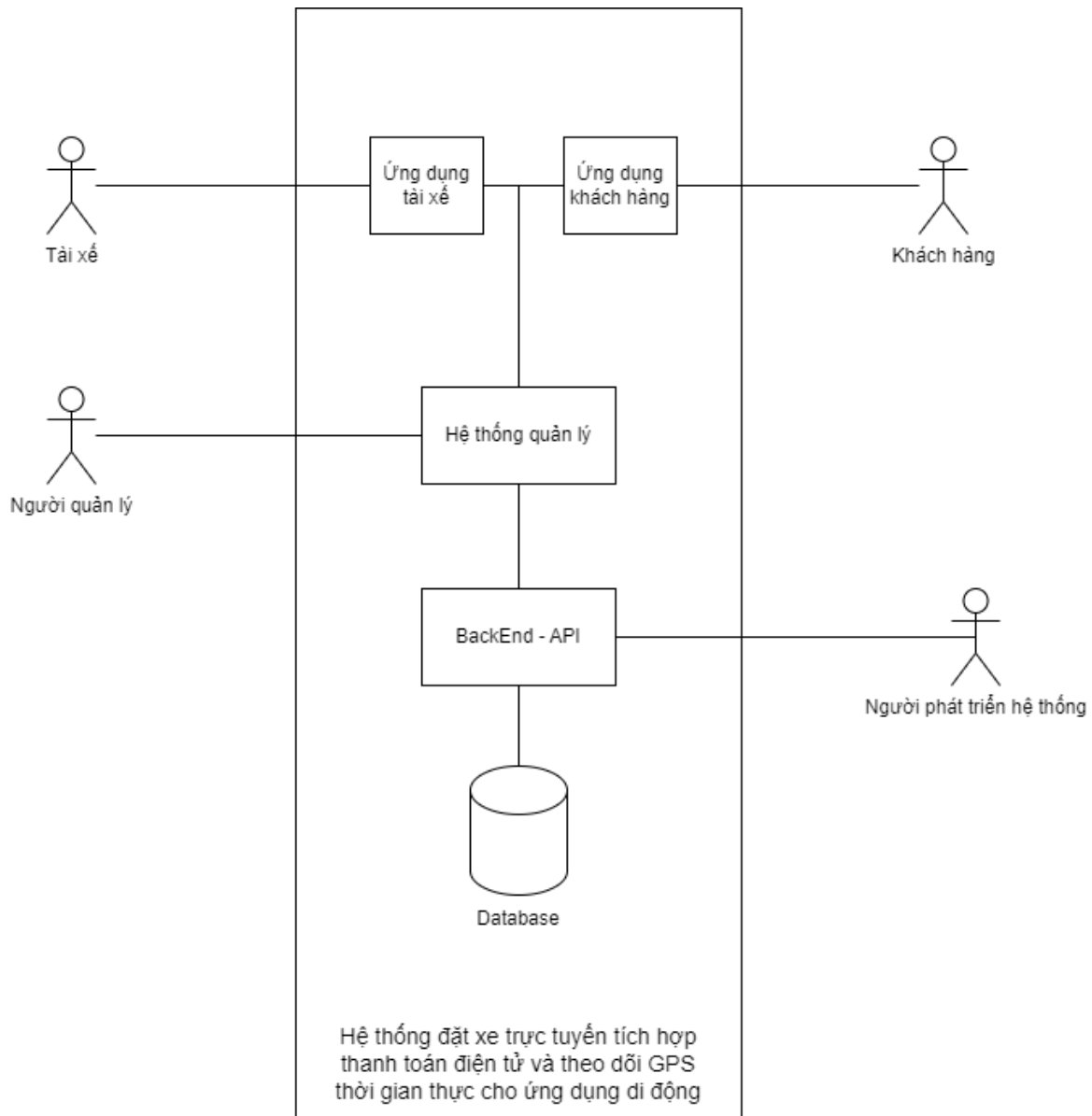


Figure 1 - System Environment

Hệ thống đặt xe trực tuyến tích hợp thanh toán điện tử và theo dõi GPS thời gian thực cho ứng dụng di động bao gồm 4 bên liên quan chính là: Khách hàng, tài xế, người quản lý, Người phát triển hệ thống

2.2 Sơ đồ Use case hệ thống

Các Use case:

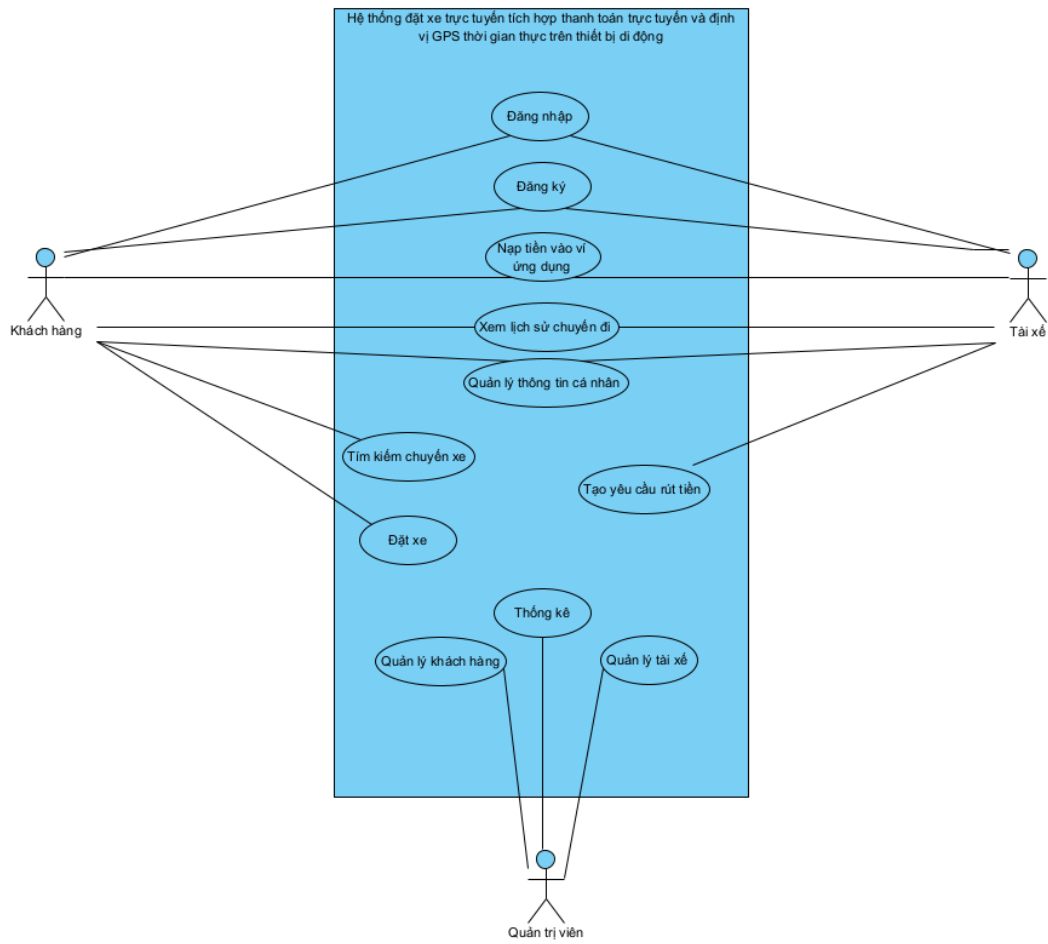
- Cho khách hàng:

- ☐ Đăng ký/Đăng nhập
- ☐ Đặt xe
- ☐ Tìm kiếm chuyến xe (chọn xe là extend cho tìm kiếm chuyến xe)
- ☐ Xem lịch sử chuyến đi
- ☐ Quản lý thông tin cá nhân
- ☐ Nạp tiền vào tài khoản ví
- ☐ Liên kết ví với ứng dụng
- ☐ Đánh giá chuyến đi
- ☐ Thanh toán (áp dụng mã giảm giá sẽ là extend của thanh toán)
- ☐ Huỷ chuyến xe

- Cho tài xế:

- ☐ Đăng ký/Đăng nhập
- ☐ Xác thực thông tin
- ☐ Nhận chuyến xe
- ☐ Xem lịch sử chuyến đi
- ☐ Xem lịch sử giao dịch
- ☐ Xem thống kê doanh thu
- ☐ Nạp tiền vào tài khoản
- ☐ Tạo yêu cầu rút tiền

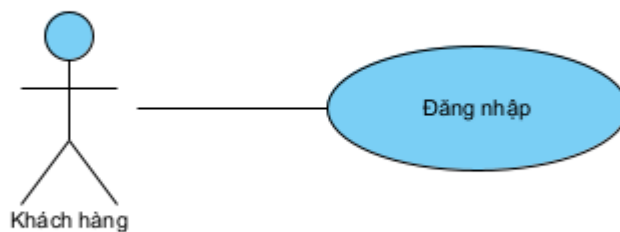
Use case **Tổng quát:**



2.2.1 Use Case dành cho Khách hàng

Use case: Đăng nhập

Diagram:

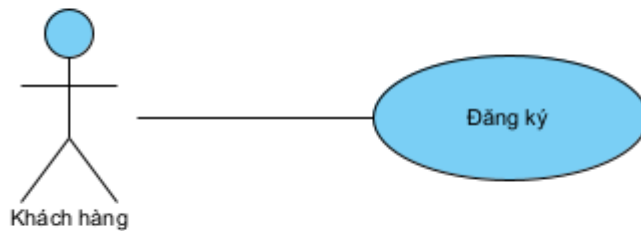


Mô tả ngắn gọn

Người dùng đăng nhập vào hệ thống thông qua số điện thoại

Use case: Đăng ký

Diagram:

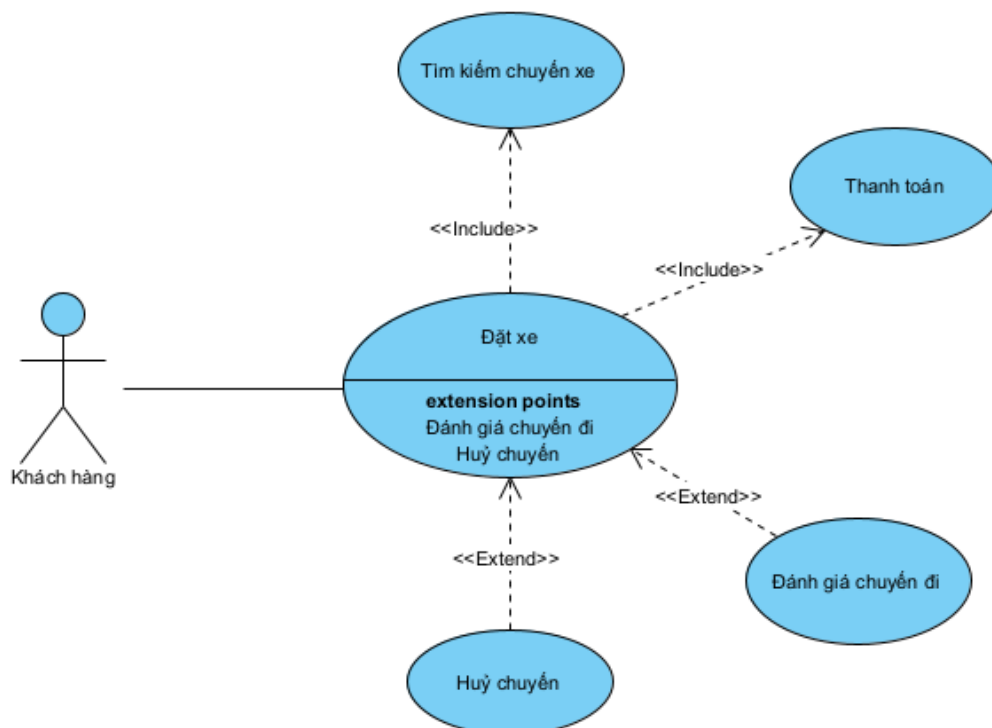


Mô tả ngắn gọn

Người dùng đăng ký vào hệ thống thông qua số điện thoại, sau đó nhập thông tin và mã 6 số

Use case: Đặt xe

Diagram:



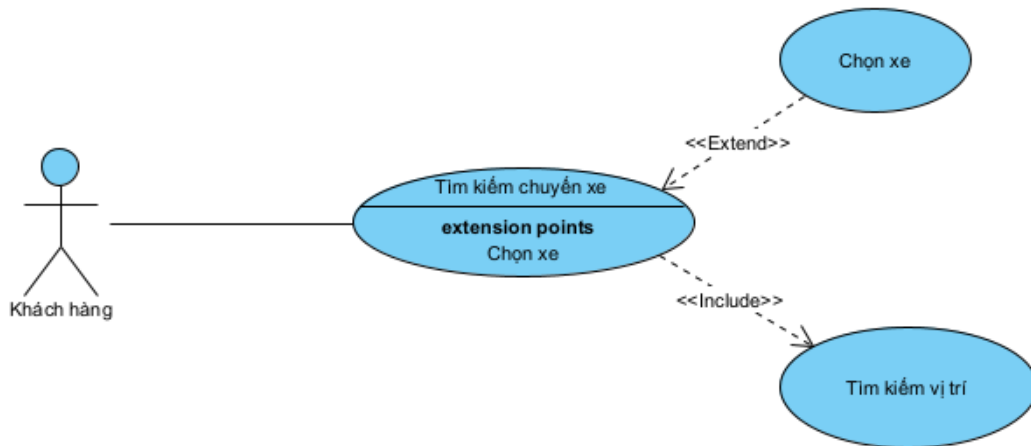
Mô tả ngắn gọn

Người dùng tìm kiếm vị trí để có thể tìm kiếm chuyến xe (có thể chọn xe, chọn phương thức thanh toán). Sau đó khi xác nhận đặt xe, hệ thống sẽ giúp tìm tài xế, khi tài xế nhận thì tài xế

sẽ thực hiện chuyển đi. Sau khi thực hiện chuyển đi, khách hàng tiến hành thanh toán, đánh giá chuyến đi. Khi tài xế chưa nhận chuyển xe, khách hàng có thể hủy chuyển xe.

Use case: Tìm kiếm chuyển xe

Diagram:

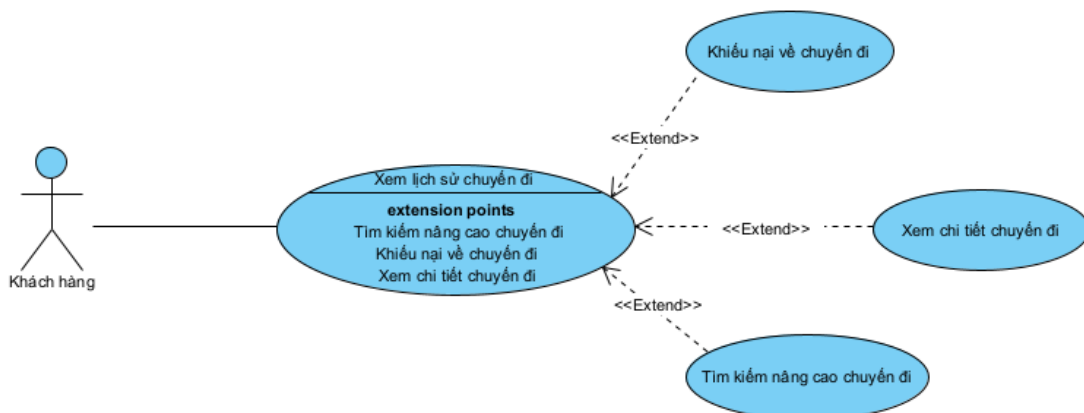


Mô tả ngắn gọn

Người dùng tìm kiếm vị trí điểm đón và điểm đến, sau đó có thể chọn xe, sau khi chọn xong hệ thống sẽ tính toán và đưa ra lộ trình, tính toán thời gian, chi phí

Use case: Xem lịch sử chuyển xe

Diagram:

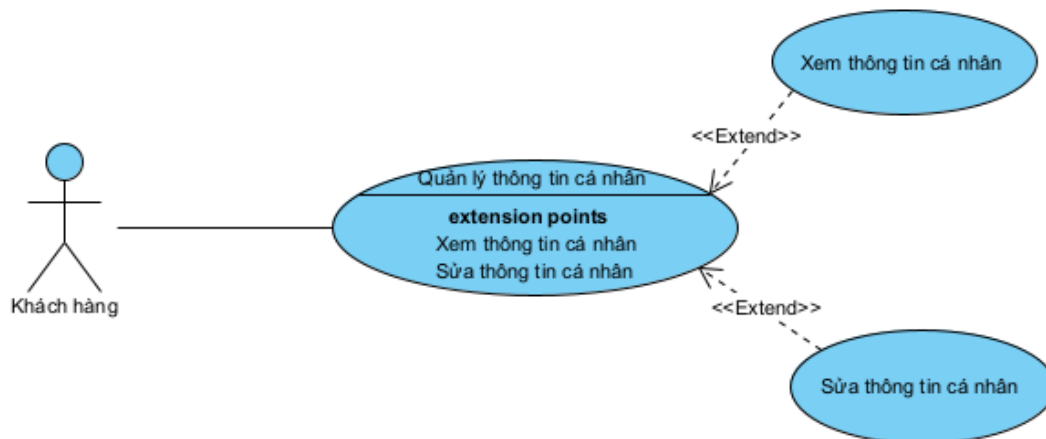


Mô tả ngắn gọn

Khách hàng xem lịch sử chuyến đi, có thể sử dụng các tính năng như sắp xếp, tìm kiếm... Người dùng có thể khiếu nại về chuyến đi và xem thông tin chi tiết chuyến đi

Use case: Quản lý thông tin cá nhân

Diagram:

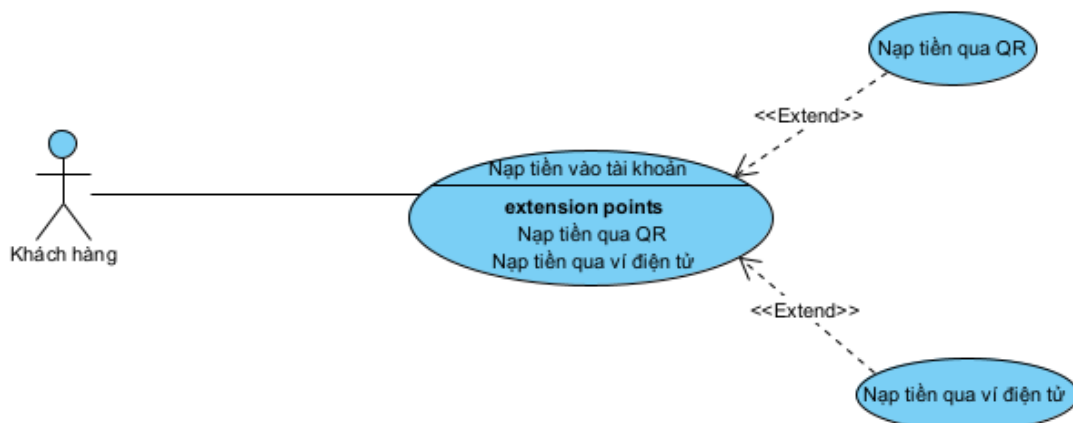


Mô tả ngắn gọn

Khách hàng có thể xem và sửa thông tin cá nhân như tên, số điện thoại,...

Use case: **Nạp tiền vào tài khoản ví**

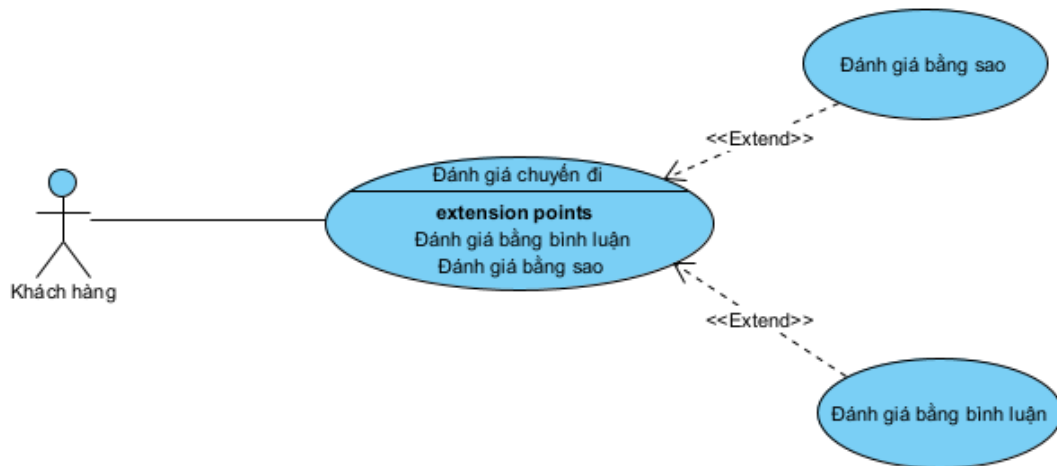
Diagram:



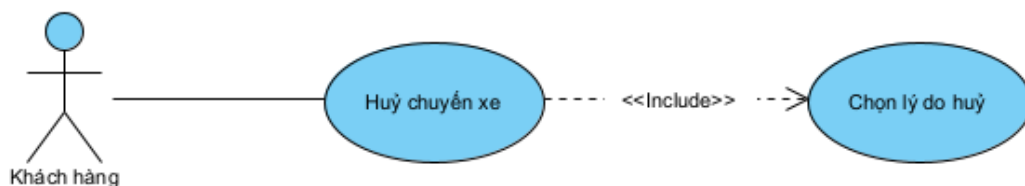
Mô tả ngắn gọn: Khách hàng có thể nạp tiền vào tài khoản ví ứng dụng thông qua việc sử dụng QR có sẵn của ứng dụng hoặc nạp tiền thông qua ví điện tử

Use case: **Liên kết ví với ứng dụng**

Diagram:

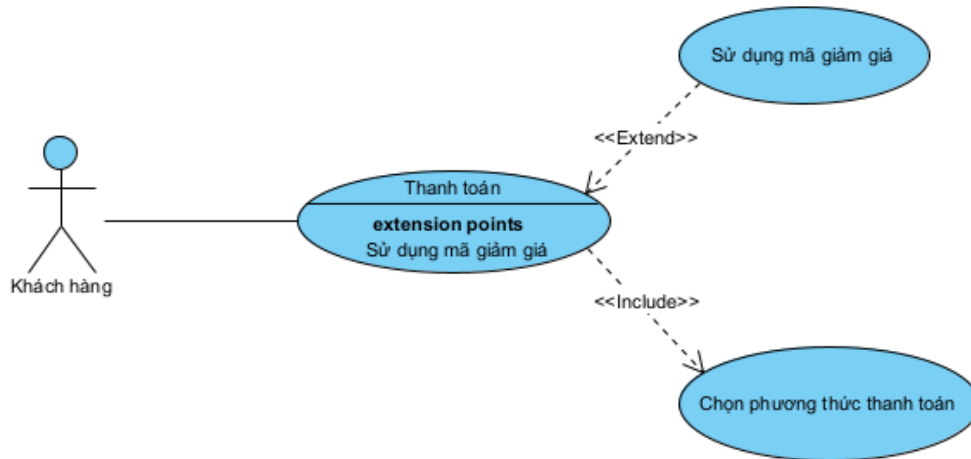
Mô tả ngắn gọn**Use case:** Đánh giá chuyến đi**Diagram:****Mô tả ngắn gọn**

Sau khi hoàn thành chuyến xe, khách hàng có thể đánh giá bằng cách cho sao hoặc đánh giá bằng bình luận

Use case: Huỷ chuyến xe**Diagram:****Mô tả ngắn gọn**

Khách hàng có thể huỷ chuyến xe trước khi tài xế nhận chuyến xe, khách hàng phải chọn lý do huỷ chuyến xe

Use case: Thanh toán**Diagram:**



Mô tả ngắn gọn

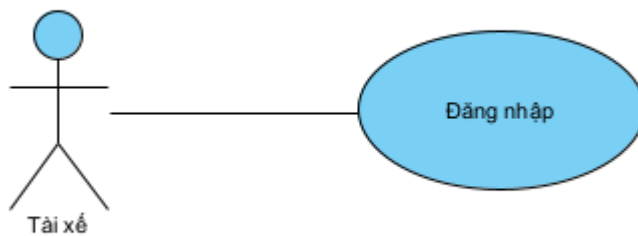
Người dùng thanh toán cho chuyến xe của mình cần chọn phương thức thanh toán và thanh toán, có thể chọn sử dụng mã giảm giá nếu có

2.2.2 Tài xế

Dưới đây là các Use case dành cho tài xế

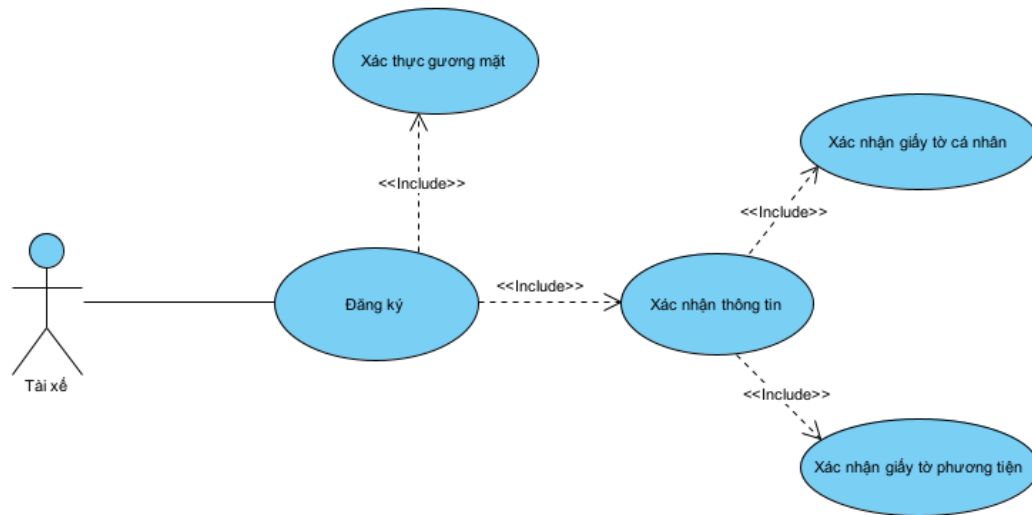
Use case: Đăng nhập

Diagram:

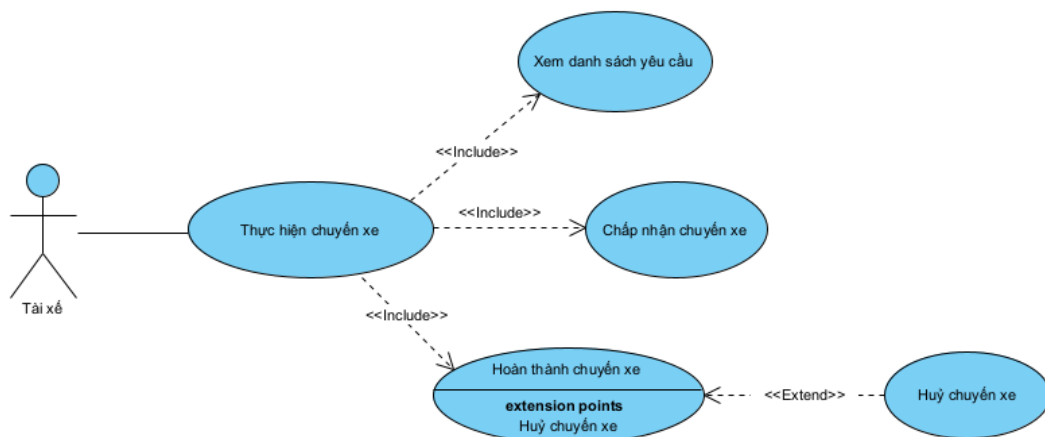


Mô tả ngắn gọn

Tài xế đăng nhập vào hệ thống bằng số điện thoại

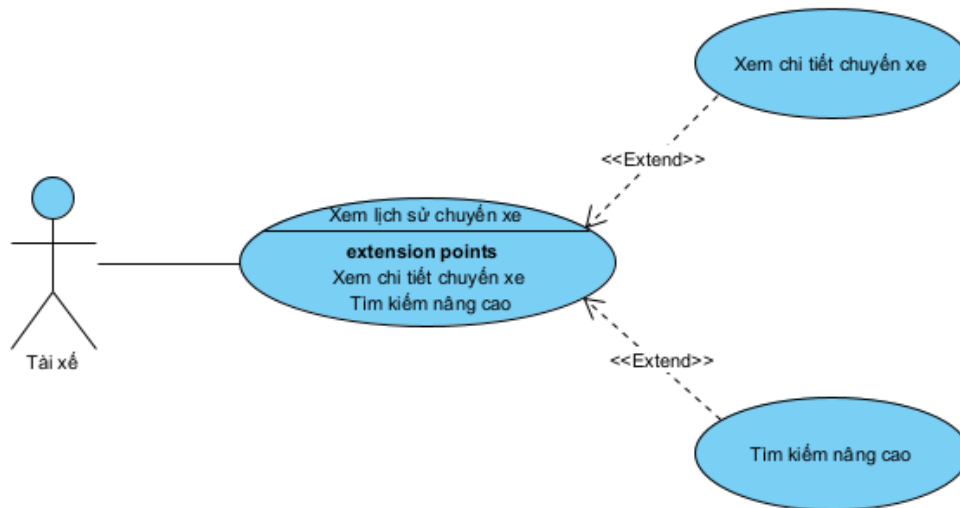
Use case: Đăng ký**Diagram:****Mô tả ngắn gọn**

Tài xế đăng ký vào hệ thống thông qua số điện thoại, sau đó tài xế phải xác thực khuôn mặt và xác thực các thông tin về cá nhân và phương tiện

Use case: Thực hiện chuyến xe**Diagram:****Mô tả ngắn gọn**

Sau khi bật trạng thái hoạt động, tài xế có thể xem danh sách yêu cầu chuyến xe phù hợp với tài xế, sau đó tài xế có thể chấp nhận chuyến xe. Sau khi chấp nhận, tài xế thực hiện quy trình chuyến xe. Trong quá trình thực hiện, tài xế có thể hủy chuyến xe khi chưa đón khách hàng. Sau cùng, tài xế cần hoàn thành chuyến xe.

Use case: Xem lịch sử chuyến xe**Diagram:**

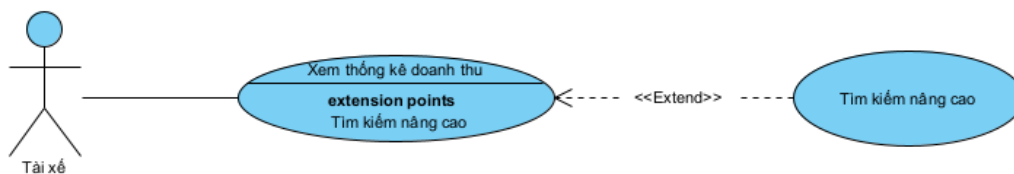


Mô tả ngắn gọn

Tài xế có thể xem lịch sử các chuyến xe, tìm kiếm nâng cao cho lịch sử chuyến xe, xem chi tiết chuyến xe.

Use case: Xem thống kê doanh thu

Diagram:



Mô tả ngắn gọn

Tài xế có thể xem thống kê doanh thu

Use case: **Nạp tiền vào tài khoản ví**

Diagram:

Mô tả ngắn gọn

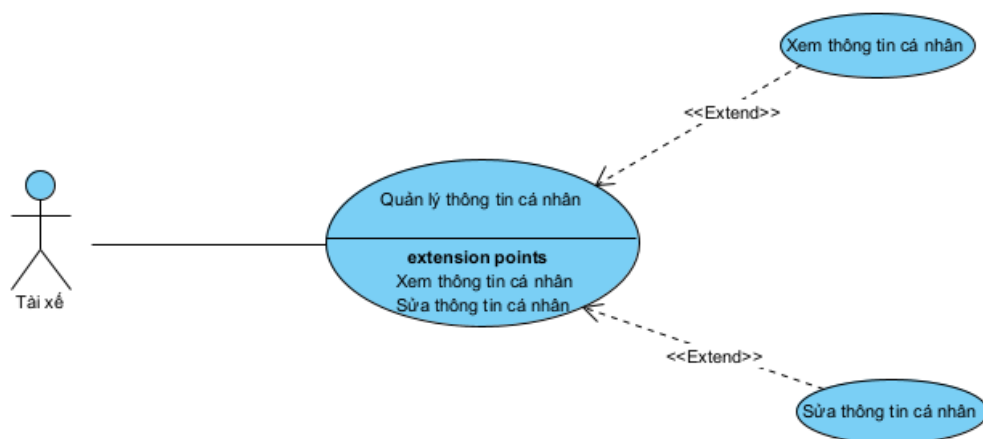
Use case: Tạo yêu cầu rút tiền

Diagram:

Mô tả ngắn gọn

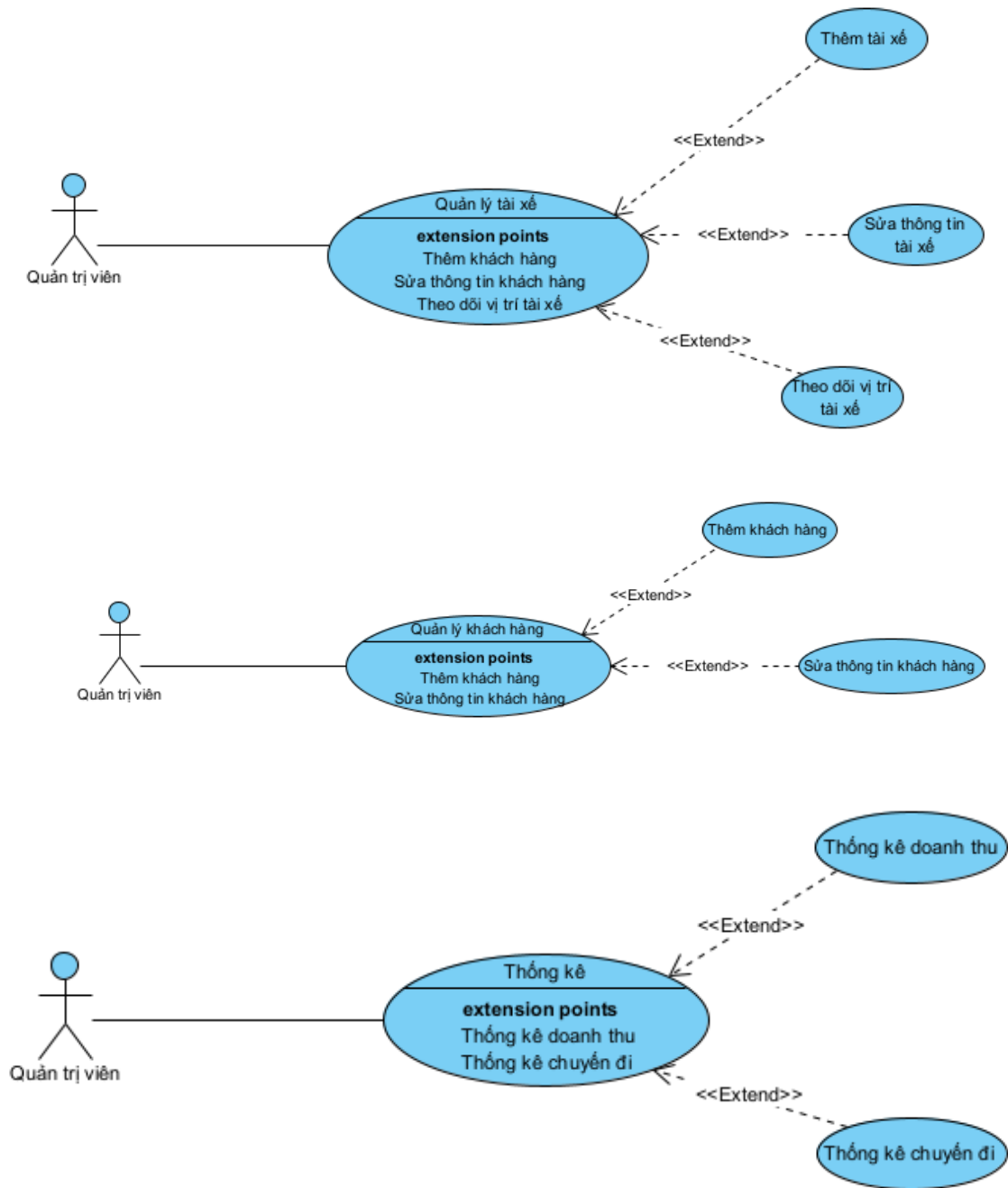
Use case: Quản lý thông tin cá nhân

Diagram:



Mô tả ngắn gọn

Tài xế có thể xem và sửa thông tin cá nhân như tên, số điện thoại,...



2.3 Đặc điểm người dùng

Khách hàng sẽ có kỹ năng sử dụng ứng dụng điện thoại và khả năng sử dụng công cụ tìm kiếm vị trí. Màn hình chính của hệ thống sẽ có chức năng tìm kiếm chuyến xe và khách hàng sẽ sử dụng từ màn hình này.

Tài xế sẽ có kỹ năng lái xe, kỹ năng sử dụng Internet và ứng dụng di động để thao tác với các yêu cầu nhận chuyến xe từ hệ thống.

Người quản lý dự kiến sẽ có kỹ năng sử dụng Internet và sử dụng các kỹ năng quản lý cơ bản, các thao tác cơ bản của hệ thống quản lý.

2.4 Yêu cầu phi chức năng

3.0. Requirements Specification

3.1 External Interface Requirements

3.2 Functional Requirement

3.2.1 Đăng nhập

Tên Use case	Đăng nhập
Actor	Khách hàng, Tài xế
Điều kiện tiên quyết	Khách hàng và tài xế đã tải về ứng dụng và đã có tài khoản
Sự kiện chính	<ol style="list-style-type: none"> 1. Khách hàng (Tài xế) mở ứng dụng 2. Màn hình nhập số điện thoại của ứng dụng hiện lên 3. Khách hàng (Tài xế) nhập số điện thoại và bấm nút Xác nhận 4. Màn hình xác nhận mã OTP hiện ra 5. Khách hàng (Tài xế) nhập mã OTP được gửi về điện thoại và bấm nút xác nhận 6. Hệ thống thông báo Đăng nhập thành công
Alternative Paths	
Postcondition	
Ngoại lệ	<ol style="list-style-type: none"> 3. Khách hàng (Tài xế) chưa có tài khoản của hệ thống, hệ thống sẽ thông báo khách hàng chưa có tài khoản và yêu cầu khách hàng đăng ký. 5. Khách hàng (Tài xế) chưa nhận được mã OTP qua số điện thoại, khách hàng bấm vào “Gửi lại” để hệ thống gửi lại mã OTP vào số điện thoại 6. Khách hàng (Tài xế) nhập sai mã OTP, hệ thống yêu cầu khách hàng nhập lại.
Ngoài ra	

3.2.2 Đăng ký

Tên Use case	Đăng ký
Actor	Khách hàng
Điều kiện tiên quyết	Khách hàng đã tải về ứng dụng và chưa có tài khoản
Sự kiện chính	<ol style="list-style-type: none"> 1. Khách hàng mở ứng dụng 2. Màn hình nhập số điện thoại của ứng dụng hiện lên 3. Khách hàng nhập số điện thoại và bấm nút Xác nhận 4. Màn hình xác nhận mã OTP hiện ra 5. Khách hàng nhập mã OTP được gửi về điện thoại và bấm nút xác nhận 6. Hệ thống thông báo Xác thực số điện thoại thành công và màn hình điền thông tin cá nhân hiện ra 7. Khách hàng điền các thông tin Họ và tên, email, giới tính và bấm vào nút Đăng ký 8. Hệ thống thông báo khách hàng Đăng ký thành công, màn hình xác lập mã 6 số hiện lên

	9. Khách hàng nhập mã 6 số (nhập thêm lần 2 để xác nhận) và bấm xác nhận 10. Hệ thống thông báo Xác lập mã 6 số thành công, màn hình chính hiện lên
Alternative Paths	
Postcondition	Thông tin tài khoản khách hàng sẽ được lưu vào hệ thống
Ngoại lệ	3. Khách hàng đã có tài khoản của hệ thống, hệ thống sẽ thông báo khách hàng đã có tài khoản và thông báo Đăng nhập thành công 5. Khách hàng chưa nhận được mã OTP qua số điện thoại, khách hàng bấm vào “Gửi lại” để hệ thống gửi lại mã OTP vào số điện thoại 6. Khách hàng nhập sai mã OTP, hệ thống yêu cầu khách hàng nhập lại. 9. Khách hàng nhập lại sai mã 6 số, hệ thống yêu cầu khách hàng nhập lại
Ngoài ra	

3.2.3 Đặt xe

Tên Use case	Đặt xe
Actor	Khách hàng
Tham chiếu	Tìm kiếm chuyến xe ở phần 3.2.4
Điều kiện tiên quyết	Khách hàng đã đăng nhập vào ứng dụng
Sự kiện chính	1. Khách hàng tiến hành Tìm kiếm chuyến xe 2. Khách hàng chọn phương thức thanh toán và chọn mã giảm giá (nếu có) 3. Khách hàng bấm vào nút Đặt xe 4. Hệ thống thông báo đặt xe thành công, khách hàng cần đợi cho tới khi có tài xế nhận chuyến xe 5. Sau khi có tài xế nhận chuyến xe, khách hàng sẽ đợi cho tới khi tài xế tới đón khách hàng tại điểm đón. 6. Khi tài xế xác nhận đã đón khách hàng, hệ thống sẽ thông báo khách hàng đang trong chuyến xe 7. Khi tài xế đã đưa khách hàng tới Điểm đến, khách hàng tiến hành thanh toán 8. Sau khi đã thanh toán, tài xế xác nhận kết thúc chuyến đi, hệ thống sẽ thông báo chuyến đi hoàn thành. 9. Màn hình hoàn thành chuyến đi hiện ra, khách hàng tiến hành đánh giá bằng sao (hoặc bằng bình luận) cho tài xế. 10. Khách hàng bấm vào Xác nhận, hệ thống thông báo Hoàn thành chuyến xe
Luồng thay thế	- Ở bước 5, nếu trong lúc tài xế đến mà tài xế hủy chuyến xe, khách hàng sẽ quay lại ở bước 2. - Ở bước 7, nếu khách hàng đã chọn thanh toán trực tuyến, hệ thống sẽ chuyển ngay tới bước 9.
Postcondition	
Ngoại lệ	

Ngoài ra	
-----------------	--

3.2.4 Tìm kiếm chuyến xe

Tên Use case	Tìm kiếm chuyến xe
Actor	Khách hàng
Điều kiện tiên quyết	Khách hàng đã đăng nhập vào ứng dụng
Sự kiện chính	<ol style="list-style-type: none"> 1. Khách hàng bấm vào ô “Bạn muốn đi đâu?” 2. Màn hình nhập điểm đón và điểm đến hiện ra 3. Khách hàng nhập vào điểm đón và điểm đến 4. Màn hình chọn loại xe hiện lên 5. Khách hàng chọn loại xe và bấm Tiếp 6. Màn hình bản đồ với các thông tin điểm đón, điểm đến, đường đi, thời gian, giá tiền ước tính hiện ra 7. Khách hàng kiểm tra thông tin bằng cách bấm vào Xem chi tiết chuyến xe 8. Màn hình chi tiết chuyến xe hiện ra
Luồng thay thế	<p>- Ở bước 2, nếu khách hàng không muốn nhập điểm đón điểm đến hoặc hệ thống không có địa điểm đó, khách hàng tiến hành chọn bằng bản đồ</p> <ol style="list-style-type: none"> 2. Khách hàng bấm vào nút Chọn trên bản đồ 3. Bản đồ với con trỏ địa điểm hiện lên 4. Khách hàng tiến hành chọn điểm đón và điểm đến trên bản đồ 5. Khách hàng xác nhận <p>Các bước tiếp theo sẽ bắt đầu từ bước 3 ở trên</p>
Postcondition	<ol style="list-style-type: none"> 1. Thông tin chuyến xe sẽ được lưu vào hệ thống 2. Khách hàng tới đúng địa chỉ Điểm đến
Ngoại lệ	<ol style="list-style-type: none"> 3. Hệ thống không xác nhận được địa điểm mà khách hàng nhập
Ngoài ra	

3.2.4 Xem lịch sử chuyến xe

Tên Use case	Xem lịch sử chuyến xe
Actor	Khách hàng
Điều kiện tiên quyết	Khách hàng đã đăng nhập vào ứng dụng
Sự kiện chính	<ol style="list-style-type: none"> 1. Khách hàng bấm vào Xem lịch sử chuyến xe 2. Màn hình lịch sử chuyến xe hiện ra 3. Khách hàng lựa chọn cách hiển thị lịch sử chuyến xe thông qua bộ lọc 4. Lịch sử chuyến xe hiện ra 5. Khách hàng bấm vào để xem chi tiết chuyến xe
Luồng thay thế	
Postcondition	
Ngoại lệ	<ol style="list-style-type: none"> 4. Khách hàng không có chuyến xe nào, hệ thống sẽ hiển thị là chưa có chuyến xe

Ngoài ra	
-----------------	--

3.2.5 Quản lý thông tin cá nhân

Tên Use case	Xem lịch sử chuyến xe
Actor	Khách hàng, tài xế
Điều kiện tiên quyết	Khách hàng (tài xế) đã đăng nhập vào ứng dụng
Sự kiện chính	<ol style="list-style-type: none"> 1. Khách hàng (tài xế) bấm vào phần Tài khoản 2. Màn hình tài khoản khách hàng hiện lên 3. Khách hàng (tài xế) bấm vào thông tin cá nhân 4. Màn hình thông tin cá nhân của khách hàng hiện lên 5. Khách hàng (tài xế) chọn sửa thông tin cá nhân 6. Màn hình sửa thông tin cá nhân của khách hàng hiện lên 7. Khách hàng (tài xế) chọn sửa các thông tin Tên, email, giới tính, số điện thoại và bấm nút Xác nhận 8. Hệ thống thông báo khách hàng sửa thông tin cá nhân thành công
Luồng thay thế	<p>Ở bước 7, nếu Khách hàng (tài xế) chọn sửa số điện thoại thì sẽ theo luồng sau</p> <ol style="list-style-type: none"> 8. Màn hình nhập mã OTP hiện ra 9. Khách hàng (tài xế) nhập vào mã OTP được gửi về số điện thoại mới thay đổi và bấm xác nhận 10. Hệ thống thông báo khách hàng sửa thông tin cá nhân thành công
Postcondition	Thông tin cá nhân khách hàng được cập nhật vào hệ thống
Ngoại lệ	7. Khách hàng (tài xế) nhập số điện thoại đã có sẵn trong hệ thống, hệ thống sẽ thông báo Số điện thoại đã tồn tại và yêu cầu khách hàng nhập lại
Ngoài ra	

3.2.6 Đánh giá chuyến xe

Tên Use case	Đánh giá chuyến xe
Actor	Khách hàng
Điều kiện tiên quyết	Khách hàng đã hoàn thành chuyến xe
Sự kiện chính	<ol style="list-style-type: none"> 1. Sau khi hoàn thành chuyến xe, khách hàng tiến hành đánh giá chuyến xe. Khách hàng chọn số sao để đánh giá cho tài xế hoặc chọn những câu đánh giá cho tài xế rồi bấm Xác nhận 2. Hệ thống xác nhận khách hàng đánh giá tài xế thành công và chuyển tới màn hình chính
Luồng thay thế	
Postcondition	Thông tin đánh giá sẽ được lưu vào hệ thống
Ngoại lệ	
Ngoài ra	

3.2.7 Huỷ chuyến xe

Tên Use case	Huỷ chuyến xe
Actor	Khách hàng
Điều kiện tiên quyết	Khách hàng đã đặt xe thành công
Sự kiện chính	<ol style="list-style-type: none"> 1. Khách hàng bấm vào nút huỷ chuyến xe 2. Màn hình huỷ chuyến xe hiện ra 3. Khách hàng chọn lý do huỷ chuyến xe và bấm xác nhận 4. Hệ thống hiển thị thông báo Chắc chắn huỷ 5. Khách hàng bấm chắc chắn 6. Hệ thống xác nhận huỷ chuyến xe
Luồng thay thế	Ở bước 4, nếu khách hàng bấm không xác nhận Huỷ thì hệ thống sẽ chuyển về màn hình đặt xe
Postcondition	Chuyến xe yêu cầu của khách hàng sẽ bị huỷ
Ngoại lệ	
Ngoài ra	

3.2.8 Thanh toán

Tên Use case	Thanh toán
Actor	Khách hàng
Điều kiện tiên quyết	Khách hàng đã hoàn thành chuyến xe
Sự kiện chính	<ol style="list-style-type: none"> 1. Sau khi đã hoàn thành chuyến xe, màn hình thanh toán hiện lên 2. Khách hàng tiến hành thanh toán dựa theo phương thức thanh toán đã chọn từ trước khi đặt xe <ol style="list-style-type: none"> 1. Khách hàng chọn thanh toán tiền mặt thì sẽ đưa tiền mặt cho tài xế 2. Khách hàng chọn thanh toán qua chuyển tiền qua tài khoản ngân hàng thì khách hàng tiến hành thanh toán qua mã QR được cung cấp từ ứng dụng thông qua ứng dụng ngân hàng của khách hàng 3. Khách hàng chọn thanh toán qua ví ứng dụng thì sau khi kết thúc chuyến đi, tiền trong ví ứng dụng sẽ tự động trừ 4. Khách hàng chọn thanh toán qua ví điện tử thì giao diện thanh toán qua ví điện tử của thiết bị di động hiện lên, khách hàng tiến hành thanh toán. 3. Hệ thống hiển thị thông báo thanh toán thành công, hoá đơn sẽ được hiển thị nếu khách hàng thanh toán trực tuyến
Luồng thay thế	
Postcondition	Thông tin thanh toán khách hàng sẽ được lưu lại vào hệ thống, và sẽ cập nhật lại ví của khách hàng cũng như tài xế
Ngoại lệ	
Ngoài ra	

3.2.9 Nạp tiền vào tài khoản ví

Tên Use case	Nạp tiền vào tài khoản ví
Actor	Khách hàng
Điều kiện tiên quyết	Khách hàng đã đăng nhập vào ứng dụng
Sự kiện chính	
Luồng thay thế	
Postcondition	
Ngoại lệ	
Ngoài ra	

3.2.8 Liên kết ví với ứng dụng

Tên Use case	Liên kết ví với ứng dụng
Actor	Khách hàng
Điều kiện tiên quyết	Khách hàng đã đăng nhập vào ứng dụng
Sự kiện chính	
Luồng thay thế	
Postcondition	
Ngoại lệ	
Ngoài ra	

3.2.8 Đăng ký

Tên Use case	Đăng ký
Actor	Tài xế
Điều kiện tiên quyết	Tài xế chưa có tài khoản của hệ thống và đã tải về ứng dụng
Sự kiện chính	<ol style="list-style-type: none"> 1. Tài xế mở ứng dụng lên 2. Màn hình nhập số điện thoại hiện ra 3. Tài xế nhập số điện thoại và bấm Xác nhận 4. Màn hình nhập mã OTP hiện lên 5. Tài xế nhập mã OTP được gửi về số điện thoại và bấm xác nhận 6. Hệ thống thông báo Xác nhận số điện thoại thành công, màn hình nhập thông tin cá nhân hiện lên 7. Tài xế nhập Họ và tên, email, giới tính và bấm Đăng ký 8. Màn hình nhập thông tin căn cước công dân và bằng lái xe hiện lên

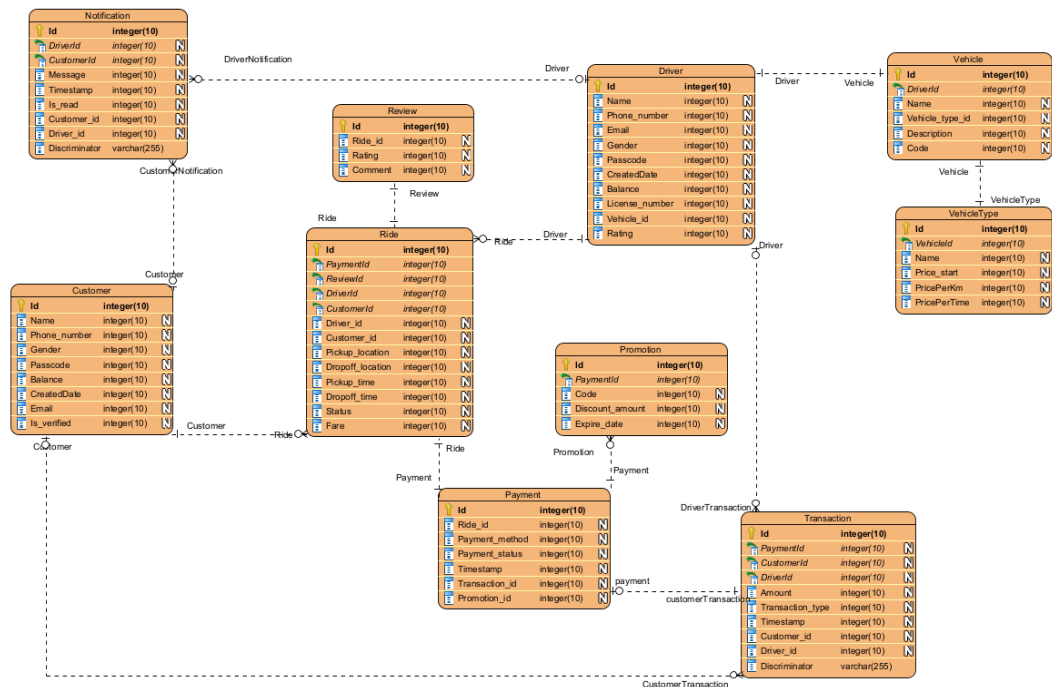
	9. Tài xế chụp ảnh theo hướng dẫn và đưa lên ứng dụng mặt trước, mặt sau của giấy tờ và bấm xác nhận 10. Hệ thống xác nhận thông tin của tài xế và thông báo Xác nhận thông tin thành công, màn hình Xác lập mật mã 6 số hiện lên 11. Tài xế nhập vào mã 6 số cùng với xác nhận lại mã 6 số, sau đó bấm Xác nhận 12. Hệ thống thông báo tài xế Đăng ký thành công, màn hình chính hiện lên
Luồng thay thế	Ở bước 6 nếu số điện thoại đã tồn tại thì hệ thống sẽ hiển thị ngay màn hình chính
Postcondition	Thông tin cá nhân khách hàng được cập nhật vào hệ thống
Ngoại lệ	5. Tài xế không nhận được mã OTP gửi về số điện thoại, tài xế bấm Gửi lại để hệ thống gửi lại mã OTP về số điện thoại 6. Tài xế nhập sai mã OTP, hệ thống thông báo tài xế nhập sai mã OTP và yêu cầu tài xế nhập lại 9. Tài xế chụp ảnh chưa đúng định dạng, hệ thống yêu cầu tài xế chụp ảnh lại đúng định dạng
Ngoài ra	

3.2.8 Thực hiện chuyển xe

Tên Use case	Thực hiện chuyển xe
Actor	Tài xế
Điều kiện tiên quyết	Tài xế đã đăng nhập vào hệ thống và mở trạng thái hoạt động
Sự kiện chính	1. Tài xế chọn xem Yêu cầu chuyển xe 2. Màn hình toàn bộ yêu cầu chuyển xe phù hợp với tài xế hiện ra 3. Tài xế bấm vào chuyển xe muốn nhận 4. Màn hình chi tiết chuyển xe hiện ra bao gồm Chi tiết thông tin khách hàng, khoảng cách, giá tiền, đường đi 5. Tài xế xem chi tiết chuyển xe, sau đó bấm Nhận chuyển xe 6. Giao diện trong chuyển xe hiện ra, hệ thống thông báo trạng thái chuyển đi Đang đón 7. Tài xế bắt đầu thực hiện chuyển xe 8. Tài xế đến địa điểm đón khách hàng, tài xế bấm Đã đón 9. Hệ thống thông báo chuyển trạng thái chuyển đi sang Đang di chuyển 10. Tài xế đưa khách hàng tới điểm đến, yêu cầu khách hàng thực hiện thanh toán 11. Sau khi khách hàng thanh toán, tài xế bấm Hoàn thành chuyển xe 12. Màn hình kết thúc chuyển xe hiện ra
Luồng thay thế	Ở bước 8, nếu tài xế quyết định Huỷ chuyển xe 9. Màn hình huỷ chuyển xe hiện lên 10. Tài xế nhập lý do huỷ chuyển và bấm xác nhận 11. Màn hình xác nhận huỷ hiện lên

	12. Tài xế bấm xác nhận hủy 13. Màn hình kết thúc chuyến đi hiện lên
Postcondition	Thông tin chuyến xe được lưu lại trong hệ thống, nếu tài xế hủy chuyến thì điểm của tài xế sẽ bị giảm
Ngoại lệ	
Ngoài ra	

Biểu đồ Thực thể liên hệ (Chưa hoàn thiện):



3.1 ...

....

....

(Chèn Hình vào đây nếu có)

Hình 3.1 (Chèn tên hình vào đây, copy dòng này cho các hình khác nếu bổ sung thêm hình, format Heading 7)

....

....

3.2 ...

....

....

3.3 Kết luận Chương III

....

....

CHƯƠNG ...

(Chèn giới thiệu mở chương ở đây).

...

....

....

(Chèn Hình vào đây nếu có)

Hình ... (Chèn tên hình vào đây, copy dòng này cho các hình khác nếu bổ sung thêm hình, format Heading 7)

....

....

...

....

....

3.3 Kết luận Chương ...

....

....

CHƯƠNG ... KẾT LUẬN

...1 Kết quả đạt được

....

....

...2 Hạn chế của hệ thống

....

....

...3 Định hướng phát triển hệ thống

....

....

DANH MỤC TÀI LIỆU THAM KHẢO

Tài liệu, giáo trình:

[1] Ví dụ: PGS.TS Trần Đình Quế, Phân tích và thiết kế hệ thống thông tin, Học viện Công nghệ Bưu chính Viễn thông, 2014.

[...] ...

[...] ...

Trang web:

[1] https://pma.edu.vn/blogs/mo-hinh-waterfall-la-gi-khi-nao-su-dung-waterfall/?gad_source=1&gclid=Cj0KCQjw1Yy5BhD-ARIsAI0RbXbJLmtv6pxGrrLcmydG9QhOoZys9_2Gpfnd9HYlt2G3UsOFsXAh9S8aAhrMEALw_wcB

[2] <https://aws.amazon.com/what-is/mqtt/>

[3]

<https://developers.google.com/maps/documentation/directions/overview?hl=vi>