

**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 TẮT THẮNG**

**Sinh viên thực hiện : ĐINH MẠNH HOÀNG**

**Lớp : D20HTTT02**

**Mã sinh viên : B20DCCN271**

**Hệ : ĐẠI HỌC CHÍNH QUY**

*Hà Nội, tháng 12 năm 2024*

## LỜI CẢM ƠN

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 chỉ bảo và chia sẻ những ý kiến, kinh nghiệm quý báu trong suốt quá trình thực hiện đồ án tốt nghiệp của em.

Thêm vào đó, em cũng bày tỏ lòng biết ơn tới các thầy cô trong Học viện Công nghệ Bru chính Viễn thông, đặc biệt là các thầy cô trong khoa CNTT1, những người đã không ngừng tận tâm giảng dạy, truyền đạt kiến thức, và tạo mọi điều kiện thuận lợi nhất để em có thể học tập và nghiên cứu hiệu quả trong suốt thời gian qua.

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, đạt được nhiều thành công trong sự nghiệp và trong cuộc sống.

*Em xin chân thành cảm ơn!*

*Hà Nội, tháng 12 năm 2024*

Sinh viên thực hiện

**Hoàng**

**Đình Mạnh Hoàng**

**NHẬN XÉT, ĐÁNH GIÁ, CHO ĐIỂM**  
**(Của người hướng dẫn)**

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

**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)**

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

...

**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 .....	iii
MỤC LỤC .....	iv
BẢNG VIẾT TẮT VÀ THUẬT NGỮ.....	vii
DANH SÁCH HÌNH VẼ.....	viii
DANH SÁCH BẢNG.....	x
MỞ ĐẦU .....	1
CHƯƠNG I. GIỚI THIỆU .....	3
1.1 Tổng quan về dịch vụ đặt xe trực tuyến .....	3
1.1.1 Lịch sử phát triển của dịch vụ đặt xe nói chung.....	3
1.1.2 Những hệ thống đặt xe trực tuyến phổ biến tại Việt Nam.....	3
1.2 Hệ thống thanh toán điện tử .....	4
1.2.1 Hệ thống thanh toán điện tử .....	4
1.2.2 Các thành phần của hệ thống thanh toán điện tử.....	4
1.2.3 Các phương thức thanh toán điện tử .....	5
1.3 GPS (Global Positioning System) .....	5
1.3.1 Nguyên lý hoạt động của GPS .....	5
1.3.2 Sự chính xác của GPS .....	6
1.4 Mục tiêu nghiên cứu.....	6
1.5 Phạm vi nghiên cứu.....	7
1.6 Tính cấp thiết của đề tài .....	8
1.7 Kết luận Chương I.....	8
CHƯƠNG II. PHƯƠNG PHÁP VÀ CÔNG NGHỆ SỬ DỤNG.....	9
2.1 Phương pháp phát triển: <i>Mô hình Waterfall</i> .....	9
2.2 Công nghệ sử dụng.....	11
2.2.1 Flutter .....	11
2.2.2 Nodejs .....	12
2.2.3 PostgreSQL .....	13
2.2.4 MQTT Protocol (Message Queuing Telemetry Transport).....	13
2.3 Tổng quan hệ thống.....	15
2.3.1 Kiến trúc hệ thống tổng quan .....	15
2.3.2 Sơ đồ kiến trúc hệ thống.....	15

2.3.3 Các công nghệ hỗ trợ và tích hợp.....	15
2.3.4 Mô hình phân loại đánh giá.....	17
2.3.5 Các nội dung khác .....	19
2.4 Kết luận Chương II.....	20
<b>CHƯƠNG III. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG .....</b>	<b>21</b>
3.1 Tổng quan hệ thống.....	21
3.1.1 Môi trường hệ thống.....	21
3.2 Sơ đồ Usecase .....	22
3.2.1 Usecase Tổng quát .....	22
3.2.2 Usecase Đăng nhập .....	23
3.2.3 Usecase Quản lý thông tin cá nhân .....	23
3.2.4 Usecase Xem lịch sử chuyến đi.....	24
3.2.5 Usecase Nạp tiền vào .....	24
3.2.6 Usecase Đăng ký .....	25
3.2.7 Usecase Đặt xe .....	25
3.2.8 Usecase Thực hiện chuyển xe .....	26
3.2.9 Usecase Tìm kiếm chuyển xe.....	26
3.2.10 Usecase Chọn chuyển xe.....	27
3.2.11 Usecase Đánh giá chuyển xe.....	27
3.2.12 Usecase Thanh toán.....	28
3.2.13 Usecase Tạo yêu cầu rút tiền.....	28
3.2.14 Usecase Quản lý khách hàng.....	28
3.2.15 Usecase Quản lý tài xế .....	29
3.2.16 Usecase Thống kê.....	29
3.3 Biểu đồ hoạt động (Activity Diagram).....	30
3.3.1 Đăng nhập .....	30
3.3.2 Đăng ký .....	30
3.3.3 Khách hàng đặt xe .....	31
3.3.4 Tài xế nhận chuyển xe.....	31
3.3.5 Thực hiện chuyển xe .....	32
3.3.6 Nạp tiền vào tài khoản ví.....	32
3.3.7 Rút tiền khỏi tài khoản ví .....	33
3.3.8 Xem lịch sử chuyển xe .....	33
3.3.9 Quản lý khách hàng.....	34
3.3.10 Quản lý tài xế .....	35
3.4 Đặc tả yêu cầu .....	36
3.4.1 Yêu cầu giao diện bên ngoài .....	36

3.4.2 Yêu cầu chức năng .....	36
3.5 Thiết kế cơ sở dữ liệu .....	48
3.5.1 Danh sách các bảng trong Cơ sở dữ liệu .....	48
3.5.2 Biểu đồ lớp phân tích .....	55
3.5.3 Mô hình thực thể liên kết .....	55
3.6 Kết luận Chương III .....	56
<b>CHƯƠNG IV. CÀI ĐẶT HỆ THỐNG .....</b>	<b>57</b>
4.1 Cài đặt môi trường làm việc .....	57
4.1.1 Cài đặt Visual Studio Code .....	57
4.1.2 Cài đặt Androi Studio.....	57
4.1.3 Cài đặt NodeJS và khởi tạo Dự án .....	58
4.2 Kiểm thử hệ thống .....	66
4.2.1 Kiểm thử đơn vị (Unit Testing).....	66
4.2.2 Kiểm thử tích hợp (Integration Testing): .....	67
4.2.3 Kiểm thử hệ thống (System Testing): .....	67
4.2.3.1 Kiểm thử chức năng: .....	67
4.2.3.2 Kiểm thử cài đặt: .....	67
4.2.3.3 Kiểm thử hiệu năng: .....	67
4.2.4 Các kịch bản kiểm thử (Test cases):.....	67
4.2.5 Công cụ và môi trường kiểm thử: .....	68
4.2.6 Kết quả kiểm thử .....	70
4.2.6.1 Kết quả kiểm thử chức năng.....	70
4.2.6.2 Kết quả kiểm thử kết hợp .....	70
4.2.6.3 Kết quả kiểm thử hệ thống .....	70
4.2.6.4 Kết luận .....	70
4.3 Các màn hình triển khai.....	71
4.4 Kết luận chương IV .....	76
<b>KẾT LUẬN .....</b>	<b>77</b>
1. Kết quả đạt được .....	77
2. Hạn chế của hệ thống .....	77
3. Định hướng phát triển hệ thống .....	77
<b>DANH MỤC TÀI LIỆU THAM KHẢO .....</b>	<b>78</b>

**BẢNG VIẾT TẮT VÀ THUẬT NGỮ**

<b>TỪ VIẾT TẮT VÀ THUẬT NGỮ</b>	<b>Ý NGHĨA</b>
API	Application Programming Interface
MQTT	Message Queuing Telemetry Transport
IoT	Internet of Things
Client	Người dùng
Server	Máy chủ
OTP	One Time Password



## **DANH SÁCH HÌNH VẼ**

Hình 1.1 Mức độ phổ biến các thương hiệu gọi xe tại Việt Nam Tham khảo Q&M.....	4
Hình 2.1 Mô hình Waterfall .....	10
Hình 2.2 Sơ đồ kiến trúc hệ thống.....	15
Hình 2.3 Phân phối tập dữ liệu huấn luyện của mô hình 5CD-ViSoBERT .....	18
Hình 3.1 Sơ đồ môi trường hệ thống .....	21
Hình 3.2 Usecase Tổng quát.....	22
Hình 3.3 Usecase Đăng nhập.....	23
Hình 3.4 Usecase Quản lý thông tin cá nhân .....	23
Hình 3.5 Usecase Xem lịch sử chuyến đi.....	24
Hình 3.6 Usecase Nạp tiền vào ví ứng dụng .....	25
Hình 3.7 Usecase Đăng ký .....	25
Hình 3.8 Usecase Đặt xe .....	25
Hình 3.9 Usecase Thực hiện chuyển xe .....	26
Hình 3.10 Usecase Tìm kiếm chuyển xe .....	26
Hình 3.11 Usecase Chọn chuyển xe .....	27
Hình 3.12 Usecase Đánh giá chuyển xe .....	27
Hình 3.13 Usecase Thanh toán.....	28
Hình 3.14 Usecase Tạo yêu cầu rút tiền.....	28
Hình 3.15 Usecase Quản lý khách hàng.....	28
Hình 3.16 Usecase Quản lý tài xế .....	29
Hình 3.17 Usecase Thống kê.....	29
Hình 3.18 Biểu đồ hoạt động Đăng nhập .....	30
Hình 3.19 Biểu đồ hoạt động Đăng ký .....	30
Hình 3.20 Biểu đồ hoạt động Khách hàng đặt xe.....	31
Hình 3.21 Biểu đồ hoạt động Tài xế nhận chuyển xe .....	31
Hình 3.22 Biểu đồ hoạt động Thực hiện chuyển xe .....	32
Hình 3.23 Biểu đồ hoạt động Nạp tiền vào tài khoản ví .....	32
Hình 3.24 Biểu đồ hoạt động Tài xế rút tiền khỏi tài khoản ví.....	33
Hình 3.25 Biểu đồ hoạt động Xem lịch sử chuyến xe.....	33
Hình 3.26 Biểu đồ hoạt động Quản lý khách hàng .....	34

Hình 3.27 Biểu đồ hoạt động Quản lý tài xế .....	35
Hình 3.28 Biểu đồ lớp phân tích.....	55
Hình 3.29 Mô hình thực thể liên kết.....	55
Hình 4.1 Trang chủ Visual Studio Code .....	57
Hình 4.2 Trang chủ Android Studio .....	57
Hình 4.3 Cài đặt NodeJS #1 .....	58
Hình 4.4 Cài đặt NodeJS #2 .....	59
Hình 4.5 Cài đặt NodeJS #3 .....	60
Hình 4.6 Cài đặt NodeJS #4 .....	61
Hình 4.7 Cài đặt NodeJS #5 .....	62
Hình 4.8 Cài đặt NodeJS #6 .....	63
Hình 4.9 Màn hình Visual Studio Code .....	63
Hình 4.10 Chạy câu lệnh npm install -g create-react-app .....	64
Hình 4.11 Chạy câu lệnh create-react-app admin .....	64
Hình 4.12 Cài đặt Flutter và Dart Plugin cho Android Studio .....	65
Hình 4.13 Tạo Dự án Flutter mới .....	66
Hình 4.14 Giao diện Postman.....	69
Hình 4.15 Máy ảo Android và IOS .....	69
Hình 4.16 Màn hình thống kê.....	75
Hình 4.17 Màn hình Quản lý khách hàng.....	75
Hình 4.18 Màn hình Quản lý vị trí tài xế .....	76
Hình 4.19 Màn hình Quản lý giao dịch .....	76

## DANH SÁCH BẢNG

Bảng 2.1 Bảng giá cước cơ bản.....	19
Bảng 2.2 Bảng các loại phụ phí.....	19
Bảng 3.1 Đăng nhập .....	37
Bảng 3.2 Đăng ký .....	38
Bảng 3.3 Đặt xe .....	39
Bảng 3.4 Tìm kiếm chuyển xe.....	40
Bảng 3.5 Xem lịch sử chuyển xe.....	40
Bảng 3.6 Quản lý thông tin cá nhân .....	41
Bảng 3.7 Đánh giá chuyển xe.....	42
Bảng 3.8 Hủy chuyển xe .....	43
Bảng 3.9 Thanh toán.....	43
Bảng 3.10 Nạp tiền vào tài khoản ví .....	44
Bảng 3.11 Thực hiện chuyển xe .....	46
Bảng 3.12 Tạo yêu cầu rút tiền.....	46
Bảng 3.13 Quản lý khách hàng (tài xế) .....	47
Bảng 3.14 Thống kê .....	48
Bảng 3.15 Bảng customer.....	48
Bảng 3.16 Bảng driver.....	49
Bảng 3.17 Bảng admin .....	49
Bảng 3.18 Bảng ride .....	50
Bảng 3.19 Bảng transaction.....	50
Bảng 3.20 Bảng customer_transaction .....	51
Bảng 3.21 Bảng driver_transaction .....	51
Bảng 3.22 Bảng payment .....	51
Bảng 3.23 Bảng promotion.....	52
Bảng 3.24 Bảng request_ride .....	53
Bảng 3.25 Bảng review .....	53
Bảng 3.26 Bảng vehicle.....	53
Bảng 3.27 Bảng vehicle_type.....	54
Bảng 3.28 Bảng notification.....	54

Bảng 3.29 Bảng customer_notification .....	54
Bảng 3.30 Bảng driver_notification .....	55
Bảng 4.1 Kịch bản Test Case .....	68
Bảng 4.2 Màn hình ứng dụng .....	75

## 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**

Chương I giới thiệu lý do chọn đề tài, mục tiêu, phạm vi nghiên cứu và tính cấp thiết của việc phát triển hệ thống đặt xe trực tuyến. Mục tiêu chính là xây dựng ứng dụng di động cho phép đặt xe, tích hợp thanh toán điện tử và theo dõi GPS thời gian thực. Phạm vi nghiên cứu bao gồm phát triển cho iOS và Android, sử dụng công nghệ GPS và tích hợp các phương thức thanh toán phổ biến. Hệ thống này đáp ứng nhu cầu hiện đại và nâng cao chất lượng dịch vụ di chuyển.

### **Chương II: PHƯƠNG PHÁP VÀ CÔNG NGHỆ SỬ DỤNG**

Chương II bao trình bày về phương pháp phát triển phần mềm với mô hình Waterfall, tiến hành các giai đoạn phát triển 1 cách chặt chẽ và xuyên suốt. Thêm vào đó, chương II còn trình bày cụ thể các công nghệ sử dụng trong hệ thống một cách khá chi tiết và những phần mở rộng sử dụng trong hệ thống.

### **Chương III: PHÂN TÍCH & THIẾT KẾ HỆ THỐNG**

Chương III trình bày nội dung bao gồm tập trung phân tích dựa trên các yêu cầu đã nêu ra như xây dựng usecase tổng quan, chi tiết, các kịch bản chính của hệ thống, đồng thời thiết kế cơ sở dữ liệu cho hệ thống

#### **Chương IV: CÀI ĐẶT HỆ THỐNG**

Chương IV trình bày về cách thức cài đặt hệ thống, cài đặt môi trường để phát triển hệ thống. Thêm vào đó, Chương IV còn trình bày phần kiểm thử hệ thống, đưa ra kết luận về những ưu điểm cũng như tồn tại của hệ thống sau kiểm thử. Và cuối cùng, chương này đưa ra các màn hình hiển thị để có một cái nhìn chi tiết nhất về hệ thống cũng như cách vận hành của hệ thống.

#### **TỔNG KẾT**

##### **Kết luận**

Đề tài "Hệ thống đặt xe trực tuyến tích hợp thanh toán điện tử và theo dõi GPS thời gian thực trên ứng dụng di động" đã hoàn thành các mục tiêu đề ra, đáp ứng nhu cầu di chuyển cơ bản với các tính năng đặt xe, thanh toán tự động và theo dõi GPS thời gian thực. Hệ thống cải thiện trải nghiệm người dùng với giao diện thân thiện, minh bạch giá cả và tích hợp công nghệ tiên tiến. Mặc dù đạt được nhiều kết quả tích cực, hệ thống vẫn còn một số hạn chế, như thiếu tính năng phân phối xe tự động, hạn chế về công nghệ hoặc phân tích dữ liệu nâng cao. Đây sẽ là định hướng cho các nghiên cứu và phát triển tiếp theo. Nhìn chung, đồ án đã chứng minh tính khả thi và ý nghĩa thực tiễn, mở ra tiềm năng ứng dụng rộng rãi và phát triển các dịch vụ vận tải thông minh trong tương lai.

## CHƯƠNG I. GIỚI THIỆU

Chương này sẽ trình bày tổng quan về dịch vụ đặt xe trực tuyến, 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 Tổng quan về dịch vụ đặt xe trực tuyến

#### 1.1.1 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 thô 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.

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

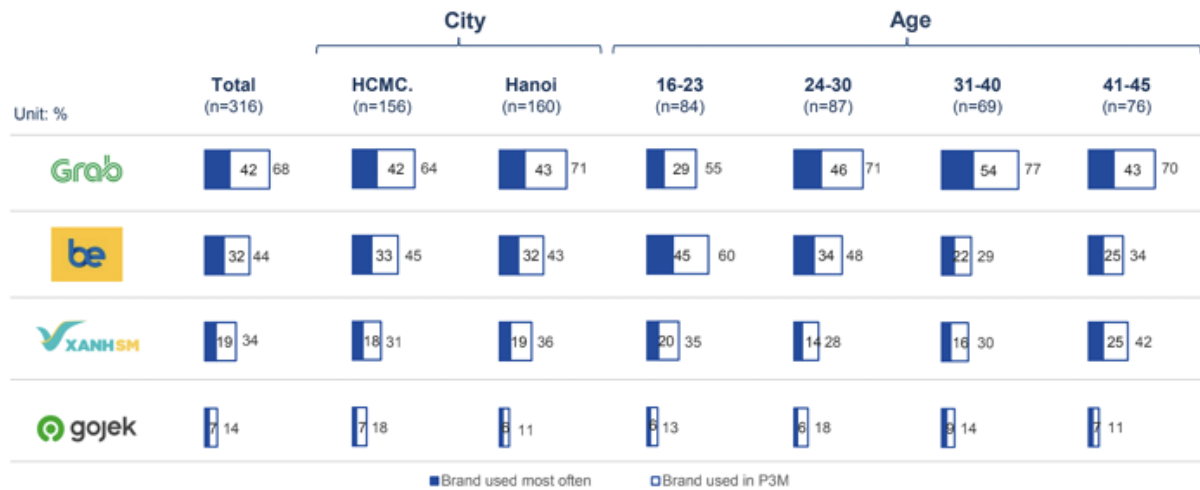
Grab: Là một sản phẩm của công ty Grab Holdings Inc, ra đời từ năm 2012, và hiện đang là hệ thống đặt xe trực tuyến có thị phần cao nhất tại Việt Nam.

Gojex là sản phẩm của PT Aplikasi Karya Anak Bangsa (Indonesia), được ra mắt vào năm 2010, tuy dần thâm 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, chủ 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 1.1 Mức độ phổ biến các thương hiệu gọi xe tại Việt Nam

Tham khảo Q&M

## 1.2 Hệ thống thanh toán điện tử

### 1.2.1 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....

### 1.2.2 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 cổng 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.

### 1.2.3 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.

## 1.3 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.

### 1.3.1 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).

### 1.3.2 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.

## 1.4 Mục tiêu nghiên cứu

Phát triển hệ thống đặt xe trực tuyến: Mục tiêu chính của đồ án là xây dựng một ứng dụng di động cho phép khách hàng dễ dàng đặt xe, tài xế dễ dàng nhận chuyển 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.

## 1.5 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 tài xế, vì vậy việc kiểm soát GPS hiện tại chỉ kiểm soát thông qua thiết bị di động.
- Hạn chế về chức năng thanh toán: chưa thể thanh toán tự động hoặc nhúng các phần mềm ví điện tử để thanh toán vì chưa có đăng ký doanh nghiệp.

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.

## 1.6 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 tài xế để thanh toán, gây bất cập nếu khách hàng không mang tiền mặt hay tài xế 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 tài xế 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 đàm để nhận chuyển xe. Điều này vẫn làm một cách thủ công và không khoa học, cũng 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.

## 1.7 Kết luận Chương I

Trong chương I, 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. PHƯƠNG PHÁP VÀ CÔNG NGHỆ SỬ DỤNG

Chương II sẽ đưa ra tổng quan về phương pháp và các công nghệ được sử dụng trong đề tài cũng như thiết kế tổng quát của hệ thống.

### 2.1 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)*

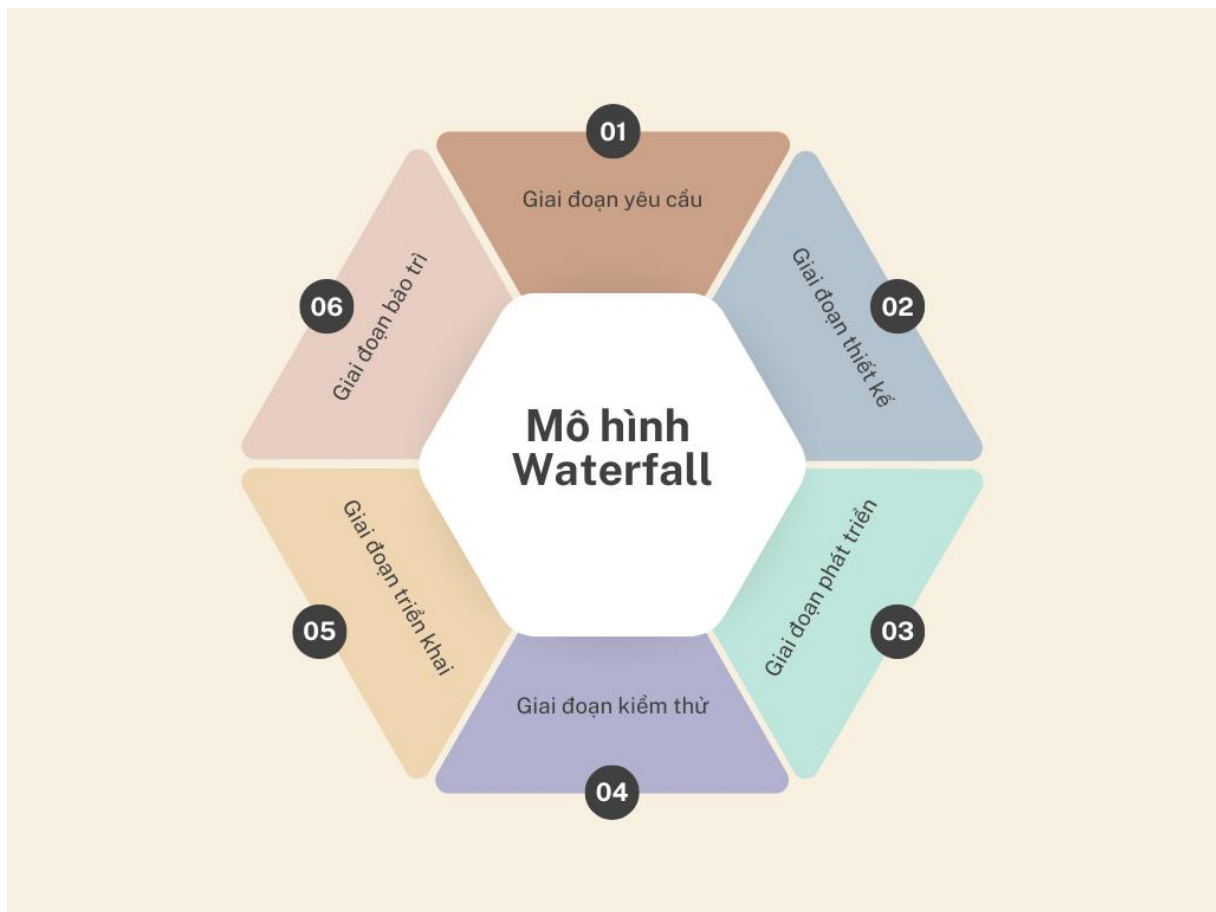
Sau khi giai đoạn phát triển thực hiện xong, đội ngũ phát triển sẽ phải tiếp tục tiến hành hoạt động phát hiện và sửa các lỗi trong phần mềm. Các loại kiểm thử như Kiểm thử đơn vị (Unit testing), Kiểm thử tích hợp (Integration testing) và kiểm thử hệ thống (System testing) sẽ được thực hiện để đảm bảo 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 2.1 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.

## 2.2 Công nghệ sử dụng

### 2.2.1 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.

### 2.2.2 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 để 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.



### 2.2.3 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.

### 2.2.4 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 thông, 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 tuyên 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.

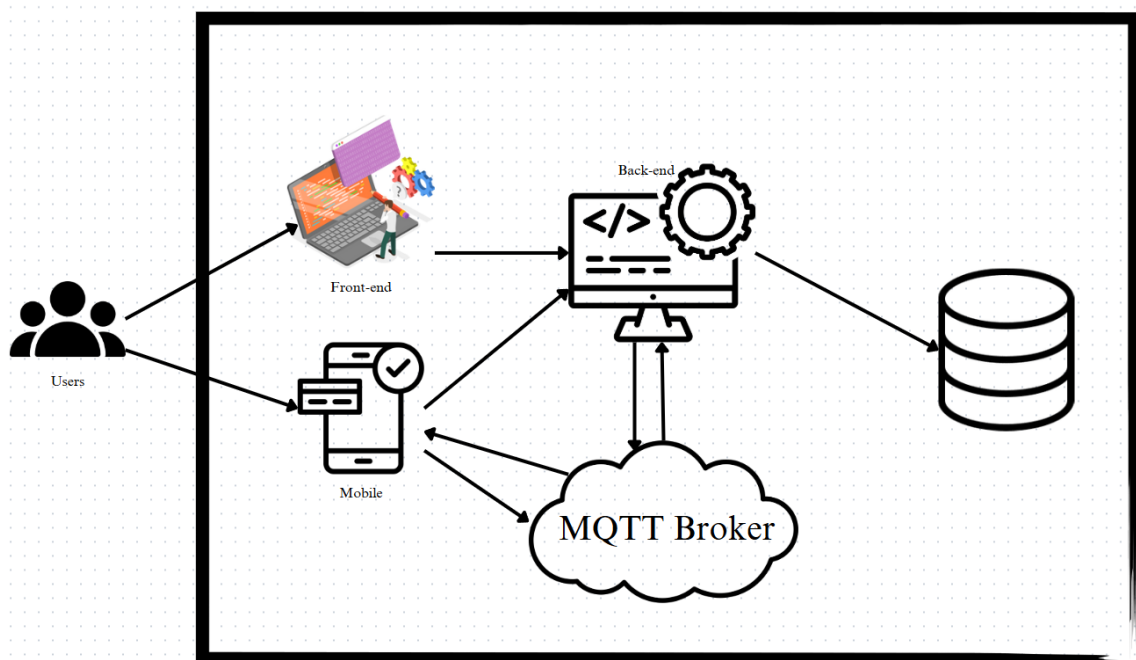
## 2.3 Tổng quan hệ thống

### 2.3.1 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 (Giao tiếp thời gian thực): Dùng MQTT để hỗ trợ truyền tin thời gian thực cho việc cập nhật vị trí GPS và thông báo

### 2.3.2 Sơ đồ kiến trúc hệ thống



Hình 2.2 Sơ đồ kiến trúc hệ thống

### 2.3.3 Các công nghệ hỗ trợ và tích hợp

#### 2.3.3.1 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 tọa 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
  1. Vị trí điểm xuất phát
  2. Vị trí điểm đích
  3. Phương thức di chuyển (DRIVE, BICYCLE, WALK, TRANSIT)
  4. 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
  5. 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
  6. 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.

### 2.3.3.2 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.

#### 2.3.3.3 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.

#### 2.3.4 Mô hình phân loại đánh giá

Sử dụng mô hình **5CD-ViSoBERT**. Đây là mô hình được thiết kế để phân tích cảm xúc (sentiment analysis) trên nội dung tiếng Việt, đánh giá từ các mạng xã hội, đánh giá sản phẩm hay bình luận có biểu tượng cảm xúc.

Kiến trúc mô hình: Mô hình được xây dựng trên nền tảng Transformer của HuggingFace - một mô hình xây dựng để xử lý ngôn ngữ tự nhiên (NLP). Cụ thể mô hình sử dụng mô hình BERT (Bidirectional Encoder Representations from Transformers), một mô hình ngôn ngữ do pre-trained do Google phát triển, với khả năng xử lý 2 chiều, cho phép khi phân tích 1 từ, mô hình sẽ xem xét cả từ phía trước và phía sau trong câu để hiểu ý nghĩa đầy đủ. Và mô hình được mở rộng sang mô hình PhoBERT, một mô hình được thiết kế đặc biệt dành cho tiếng Việt, do VinAI Research phát triển. Dựa vào các mô hình trên, ViSoBERT ra đời với mục đích phân tích cảm xúc trên nội dung tiếng Việt.

Tập dữ liệu huấn luyện: Tiềm huấn luyện trên tập dữ liệu lớn (14GB) của nội dung tiếng Việt từ mạng xã hội. Các tập dữ liệu sử dụng để xây dựng mô hình:

- SA-VLSP2016: Tập dữ liệu từ cuộc thi VLSP 2016 về phân tích cảm xúc
- AIVIVN-2019: Dữ liệu đánh giá từ nền tảng e-commerce Việt Nam
- Tiki-reviews: Bình luận từ khách hàng trên nền tảng Tiki
- VOZ-HSD (Gemini-label): Bình luận trên diễn đàn VOZ với nhãn cảm xúc được làm sạch và tinh chỉnh
- ...

Dataset	Train			Test			Val		
	Neg	Pos	Neu	Neg	Pos	Neu	Neg	Pos	Neu
All-filtered	62708	41400	11593	-	-	-	5079	3724	638
SA-VLSP2016	4759	4798	4459	1180	1190	1114	-	-	-
UIT-VSFC	5325	5643	458	1409	1590	167	-	-	-
UIT-VSMEC (Gemini-label)	3219	1665	594	458	407	210	71	388	239
AIVIVN-2019	6776	7879	-	4770	5168	-	-	-	-
UIT-VICTSD (Gemini-label)	3370	2615	933	3370	2615	933	3370	2615	933
UIT-ViHSD	4162	19886	-	1132	5548	-	482	2190	-
UIT-VISFD	2850	3670	1266	827	1000	397	409	515	188
UIT-ViOCD	2292	2095	-	279	270	-	283	265	-
Tiki-reviews	20093	6669	4698	-	-	-	-	-	-
VOZ-HSD (Gemini-label)	2676	1213	1071	-	-	-	-	-	-
Vietnamese-amazon-polarity	2559	2441	-	1017	983	-	523	477	-

Hình 2.3 Phân phối tập dữ liệu huấn luyện của mô hình 5CD-ViSoBERT

Sử dụng:

- **Đầu vào:** "Miếng dán dễ xước, ko khít với dt 11prm"
- **Đầu ra:** "[{'label': 'NEG', 'score': 0.998149037361145}]"
- Giải thích: 'label': nhãn của câu đầu vào, **NEG** (negative): tiêu cực, **POS** (positive): tích cực, **NEU** (Neutral): trung lập, 'score' là điểm đánh giá với label đó. Đầu ra đầy đủ có thể như sau

```

Sentence: Đây là mô hình rất hay, đáp ứng tốt nhu cầu của nhiều doanh nghiệp
### Sentiment score ###
1) POS: 0.9995
2) NEG: 0.0003
3) NEU: 0.0003

```

Cách sử dụng cho hệ thống:

Mô hình được áp dụng vào hệ thống bằng cách viết một API bằng ngôn ngữ Python với thư viện Flask, sau đó bên phía Back-end sẽ sử dụng API này khi khách hàng đánh giá chuyển xe với các bước như sau:

- Bước 1: Khách hàng gửi đánh giá tới hệ thống, hệ thống sẽ đưa đánh giá qua mô hình để phân loại xem đánh giá thuộc loại nào.

- Bước 2: Sau khi đã có phân loại, hệ thống sẽ lưu vào thuộc tính “sentiment” của đánh giá với tên của phân loại đó

Sau khi đã có đánh giá, bên phía quản trị viên sẽ có thể thống kê và Xếp hạng tài xế theo phân loại đánh giá đó.

### 2.3.5 Các nội dung khác

#### 2.3.5.1 Bảng tính giá chuyển 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

*Bảng 2.1 Bảng giá cước cơ bản*

- **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	10%/Tổng tiền
Phí cầu đường, vé xe	Chi phí khi đi đường gập ghềnh	Thoả thuận giữa tài xế và người đặt xe

*Bảng 2.2 Bảng các loại phụ phí*

**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:** *Chuyến 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*  
 $\Rightarrow$  *Giá cước:*  $29.000 + 8 * 10000 + 30 * 450 + 50000 + 4000 = 135.500đ$

## **2.4 Kết luận Chương II**

Như vậy, chương II đã đưa ra rất chi tiết về những công nghệ sử dụng và thiết kế tổng quát hệ thống. 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.

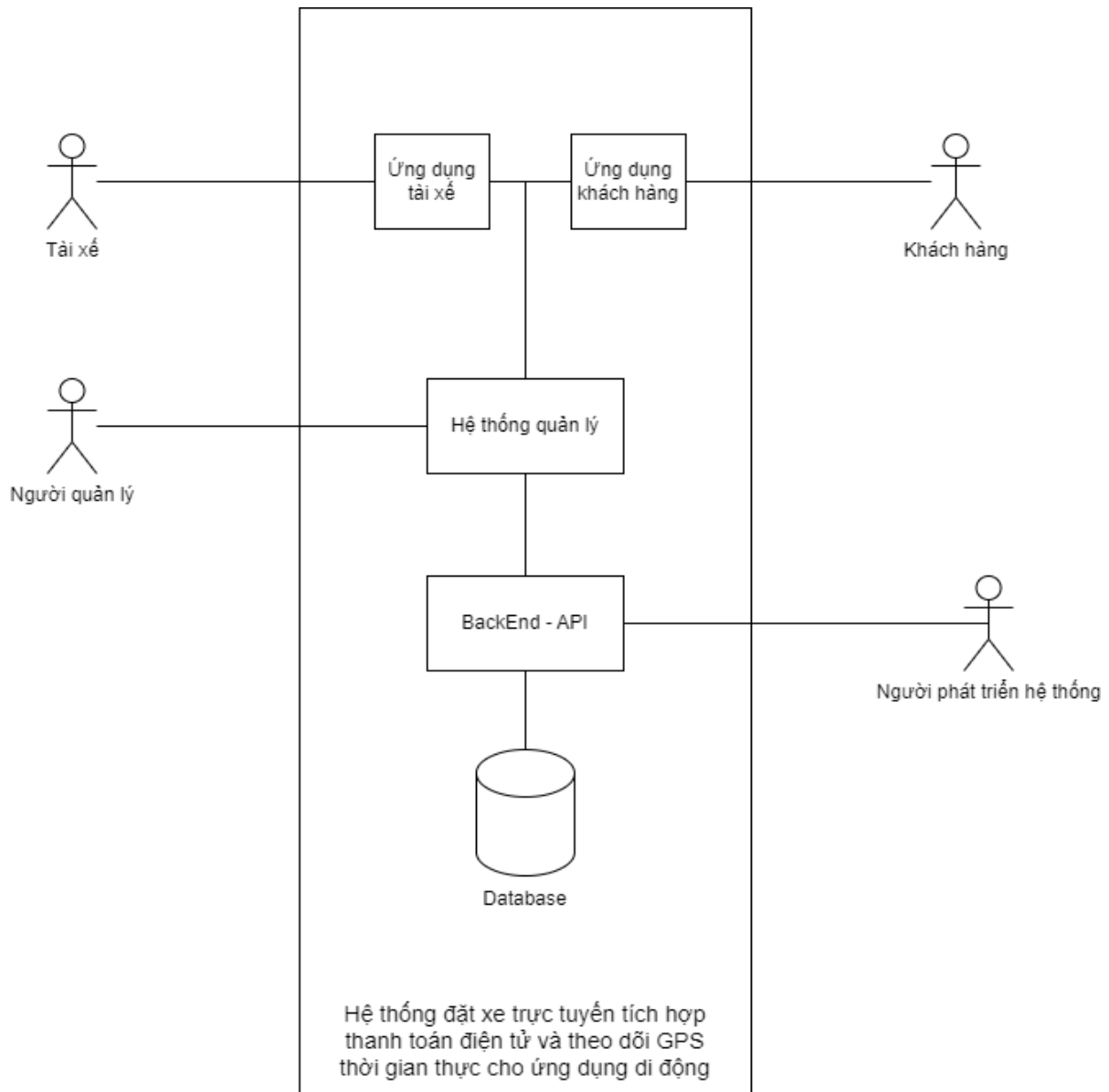


## CHƯƠNG III. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

Chương III 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 chức năng của hệ thống cũng như thiết kế cơ sở dữ liệu cho hệ thống.

### 3.1 Tổng quan hệ thống

#### 3.1.1 Môi trường hệ thống

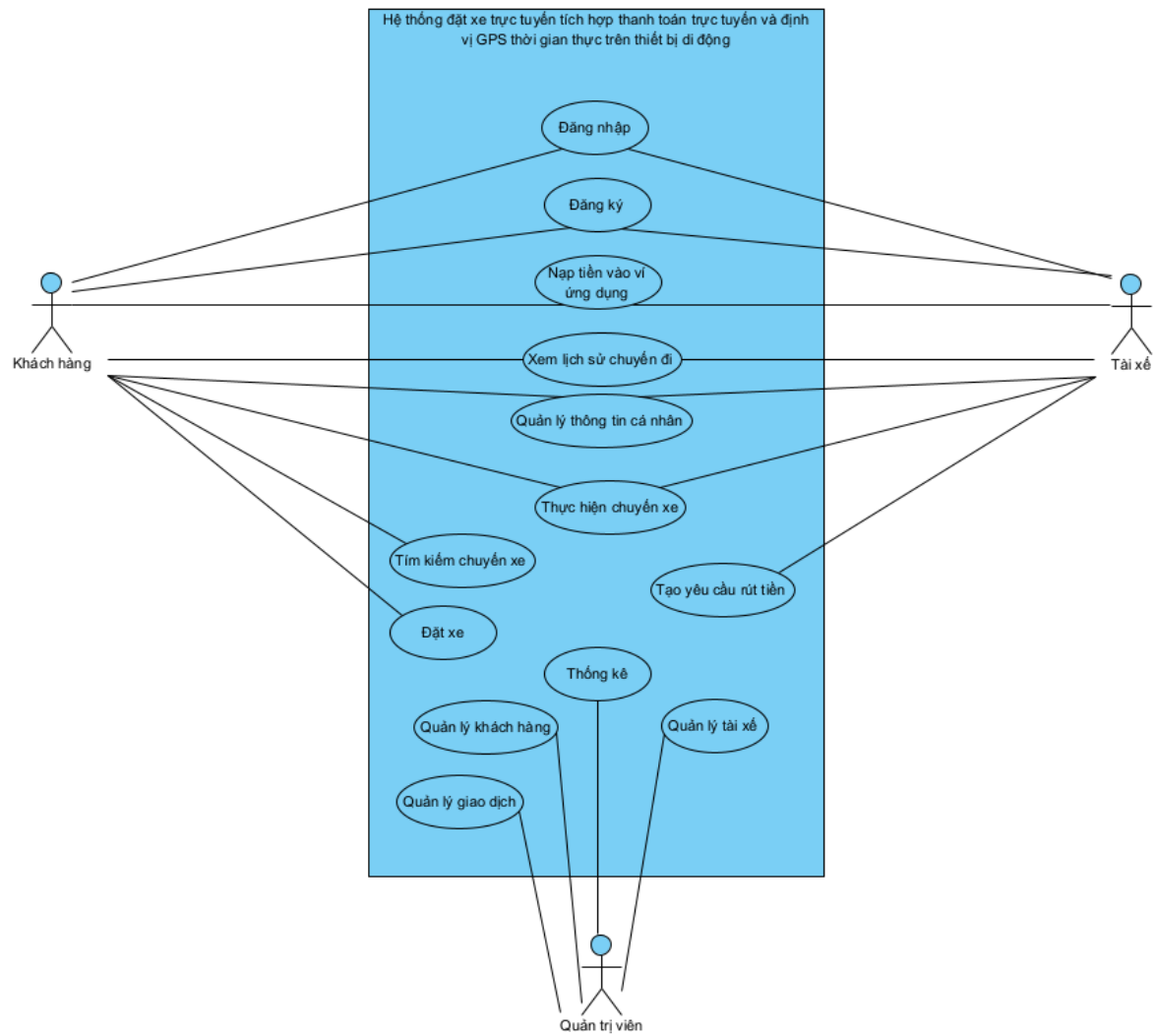


Hình 3.1 Sơ đồ môi trường hệ thống

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

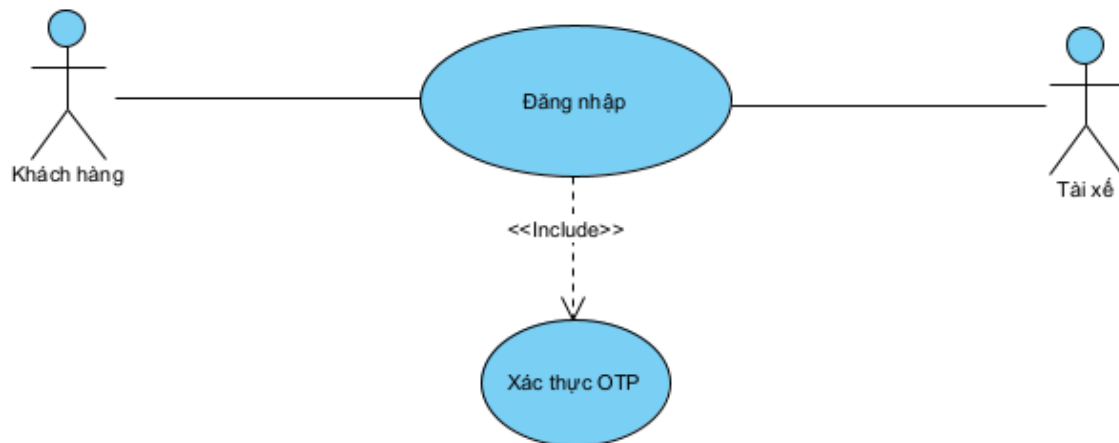
## 3.2 Sơ đồ Usecase

### 3.2.1 Usecase Tổng quát



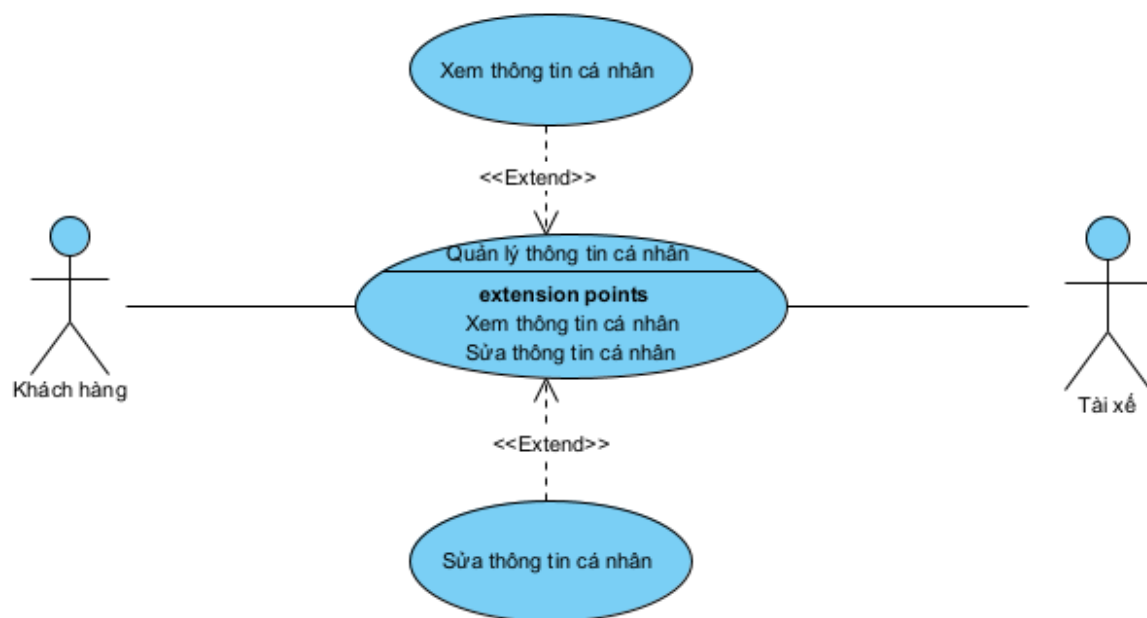
Hình 3.2 Usecase Tổng quát

### 3.2.2 Usecase Đăng nhập



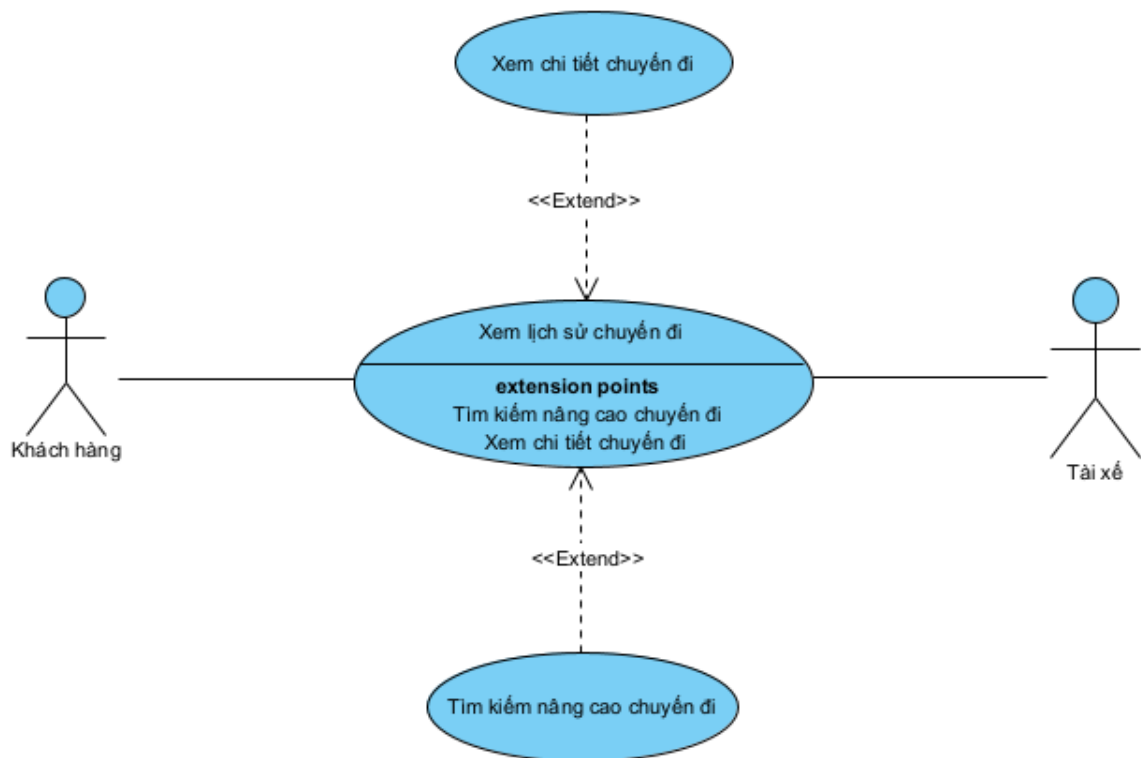
Hình 3.3 Usecase Đăng nhập

### 3.2.3 Usecase Quản lý thông tin cá nhân



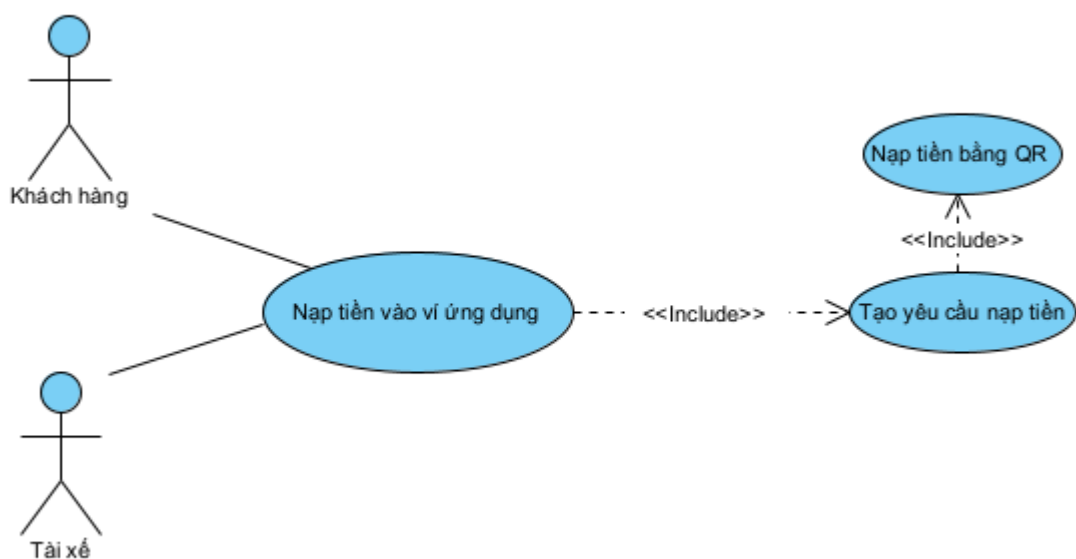
Hình 3.4 Usecase Quản lý thông tin cá nhân

### 3.2.4 Usecase Xem lịch sử chuyến đi



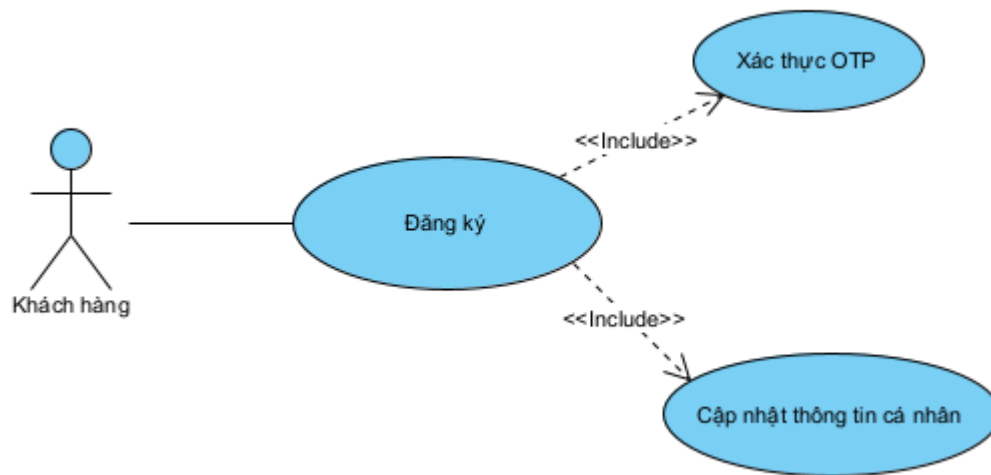
Hình 3.5 Usecase Xem lịch sử chuyến đi

### 3.2.5 Usecase Nạp tiền vào



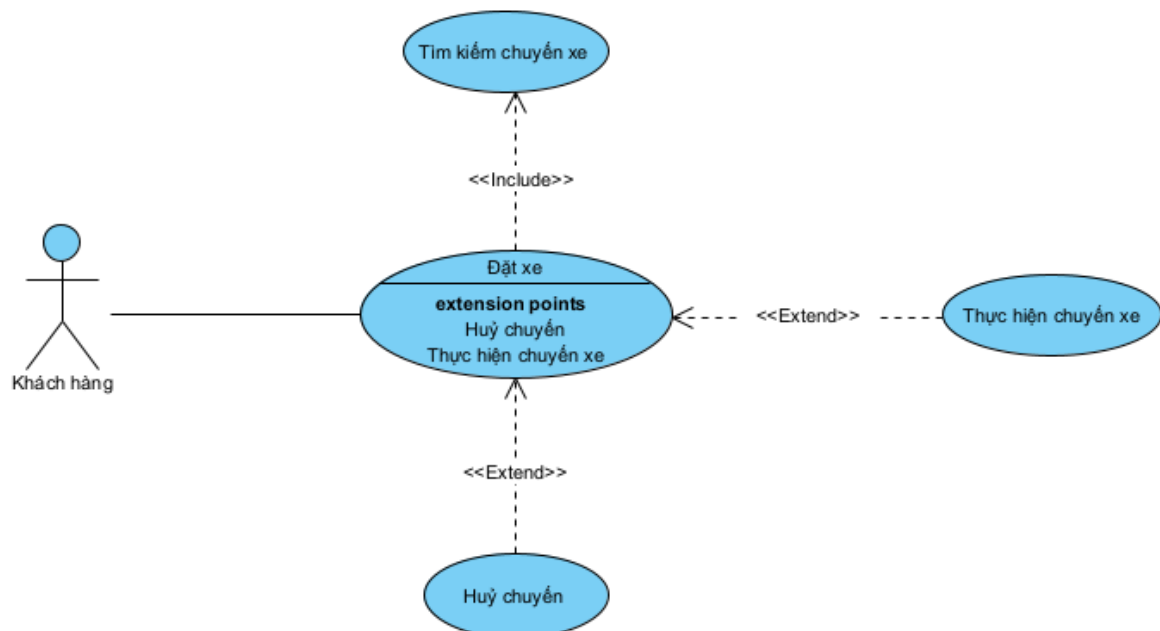
Hình 3.6 Usecase Nạp tiền vào ví ứng dụng

### 3.2.6 Usecase Đăng ký



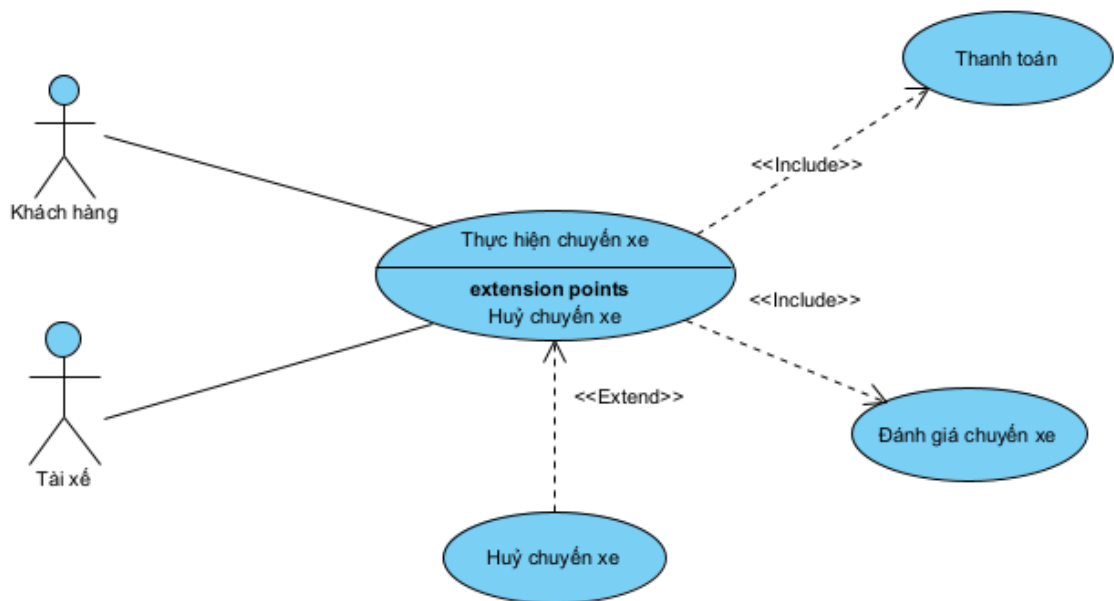
Hình 3.7 Usecase Đăng ký

### 3.2.7 Usecase Đặt xe



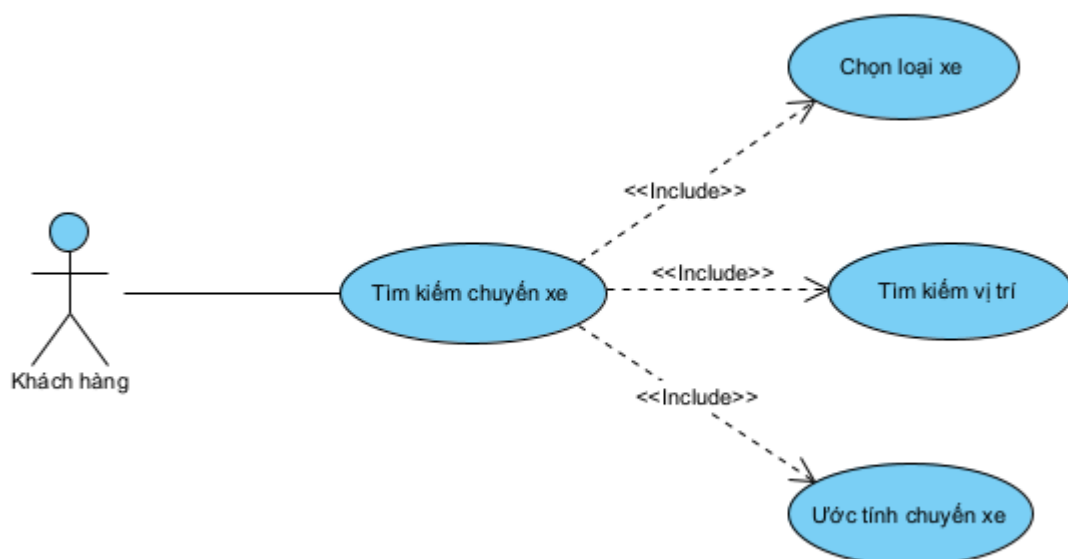
Hình 3.8 Usecase Đặt xe

### 3.2.8 Usecase Thực hiện chuyến xe



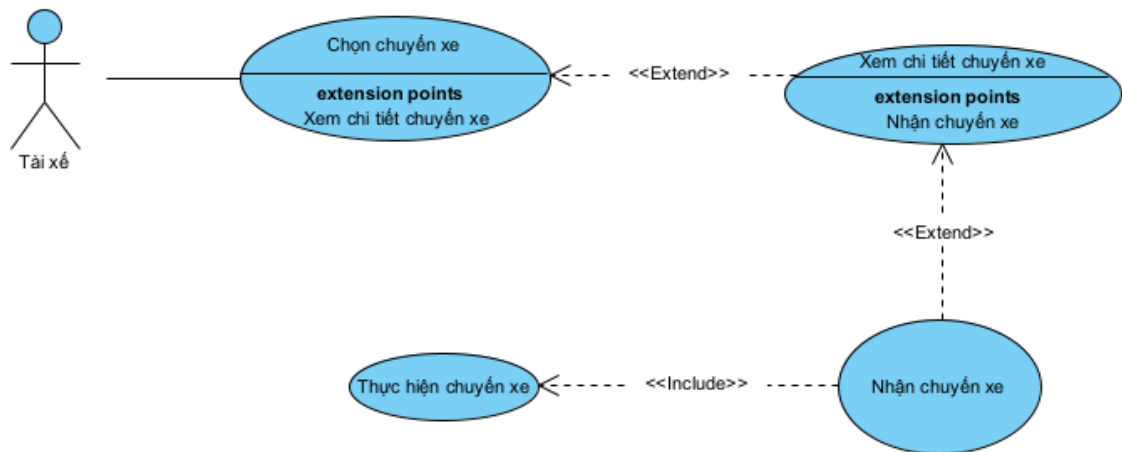
Hình 3.9 Usecase Thực hiện chuyến xe

### 3.2.9 Usecase Tìm kiếm chuyến xe



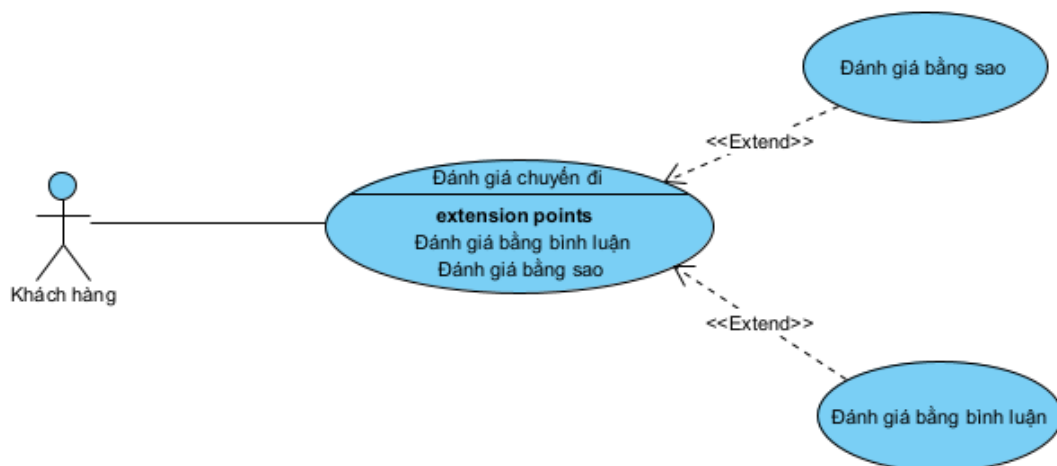
Hình 3.10 Usecase Tìm kiếm chuyến xe

### 3.2.10 Usecase Chọn chuyển xe



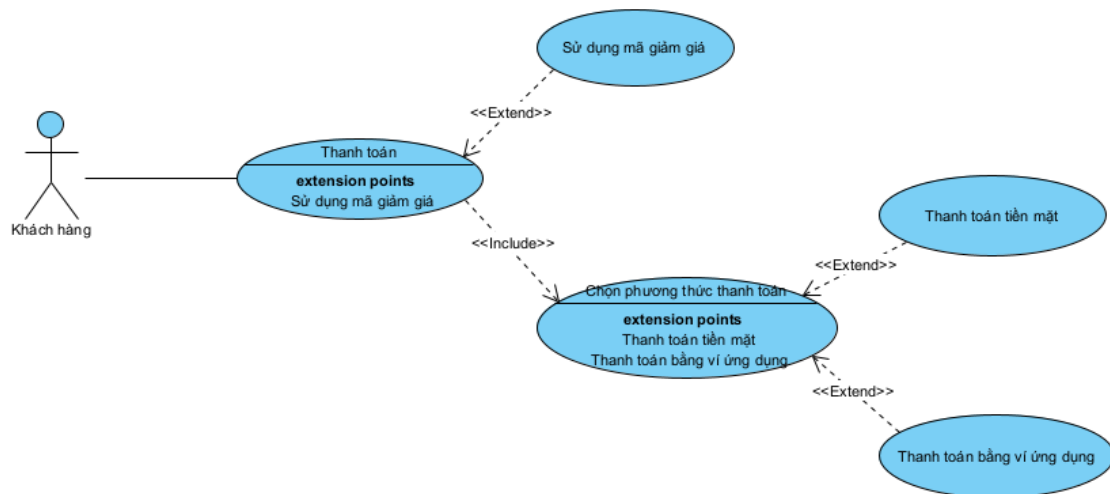
Hình 3.11 Usecase Chọn chuyển xe

### 3.2.11 Usecase Đánh giá chuyển xe



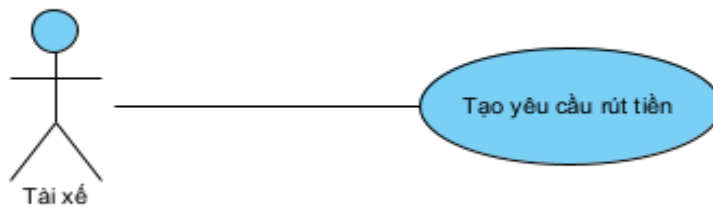
Hình 3.12 Usecase Đánh giá chuyến xe

### 3.2.12 Usecase Thanh toán



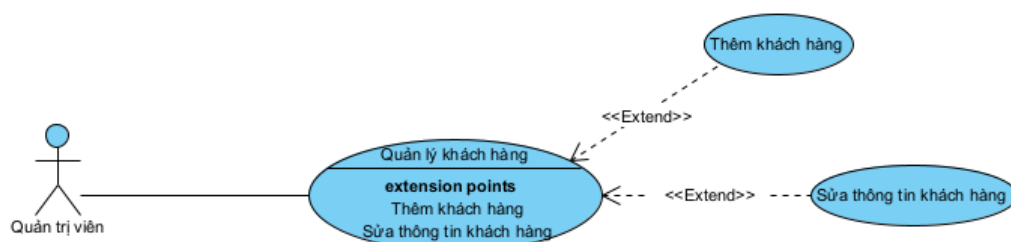
Hình 3.13 Usecase Thanh toán

### 3.2.13 Usecase Tạo yêu cầu rút tiền



Hình 3.14 Usecase Tạo yêu cầu rút tiền

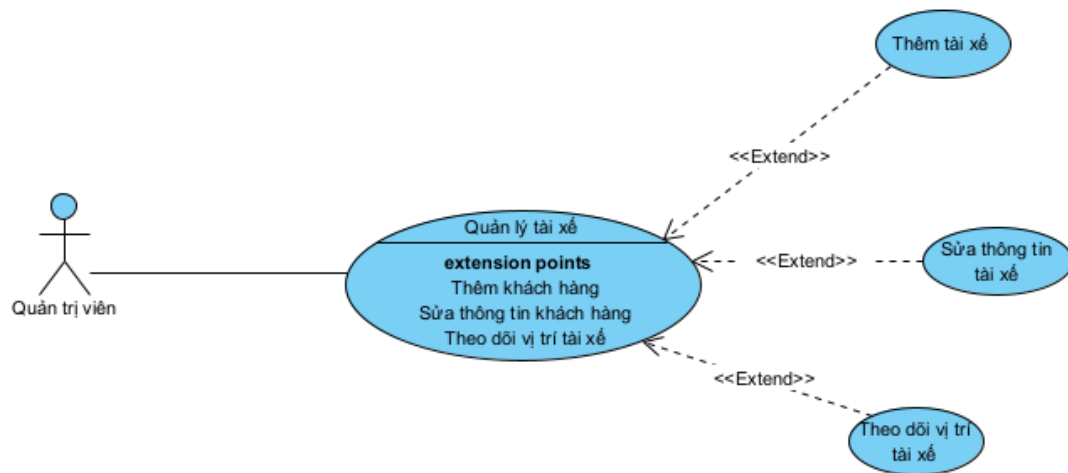
### 3.2.14 Usecase Quản lý khách hàng



Hình 3.15 Usecase Quản lý khách hàng

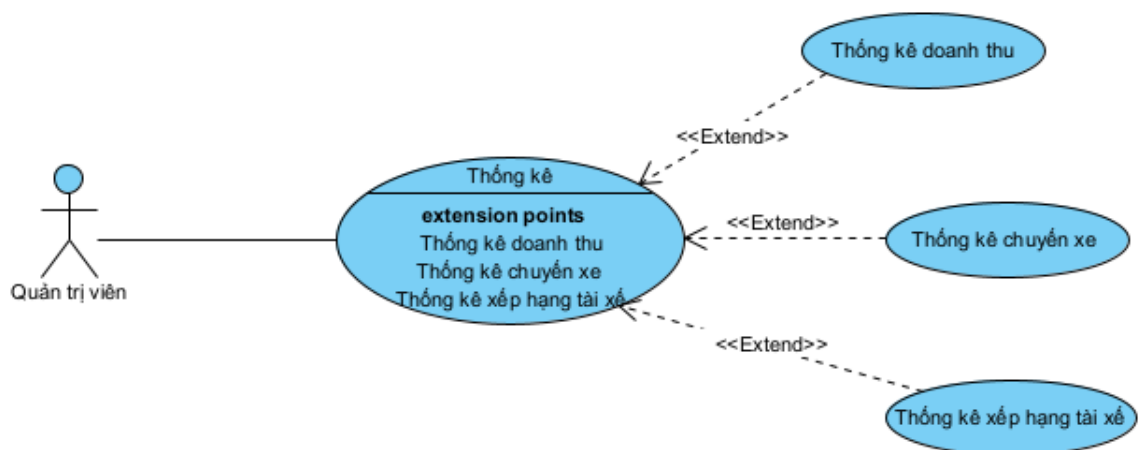


### 3.2.15 Usecase Quản lý tài xế



Hình 3.16 Usecase Quản lý tài xế

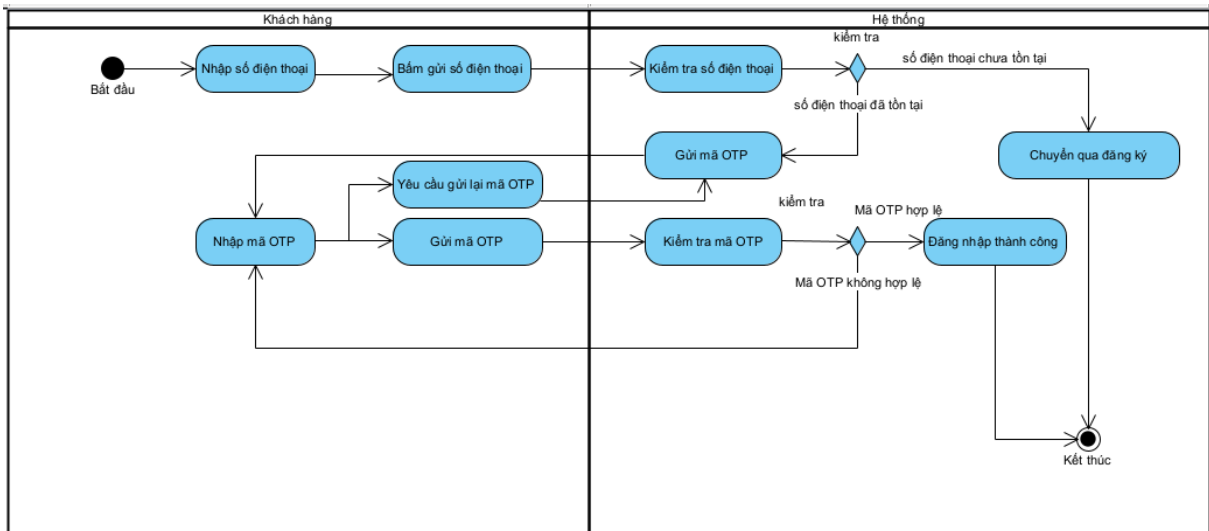
### 3.2.16 Usecase Thống kê



Hình 3.17 Usecase Thống kê

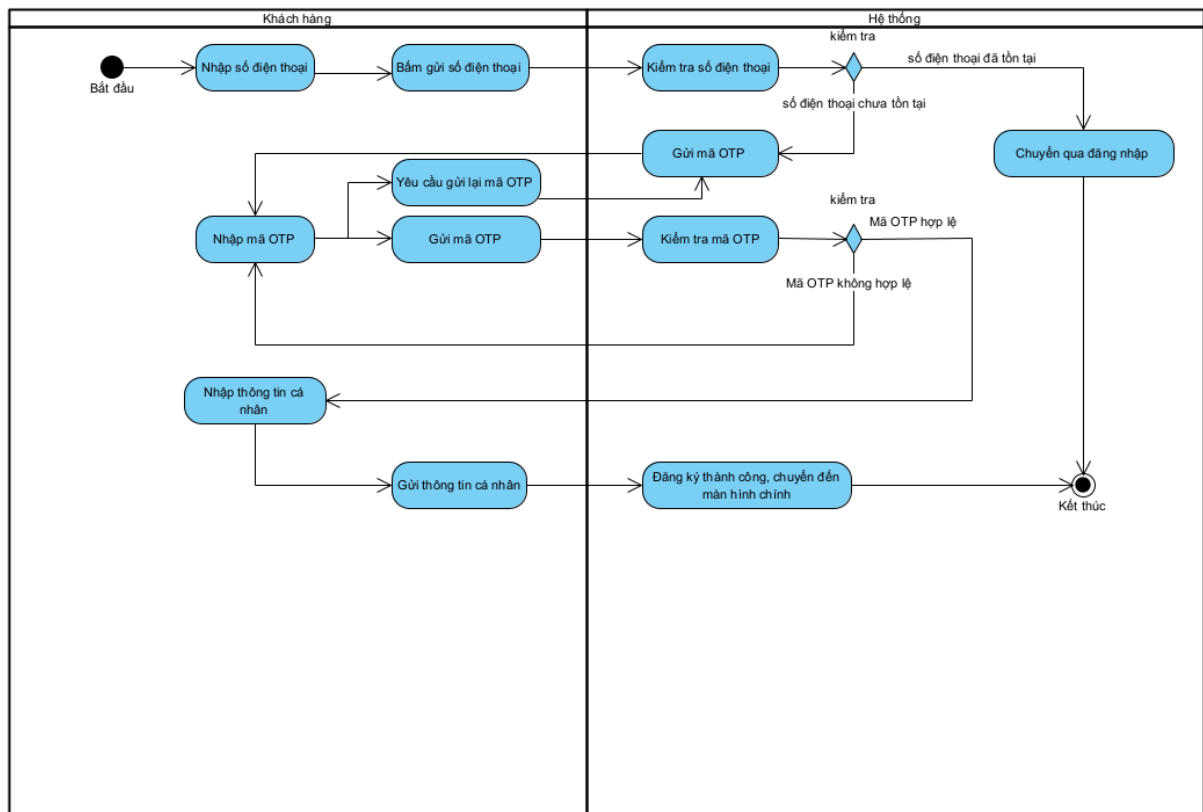
### 3.3 Biểu đồ hoạt động (Activity Diagram)

#### 3.3.1 Đăng nhập



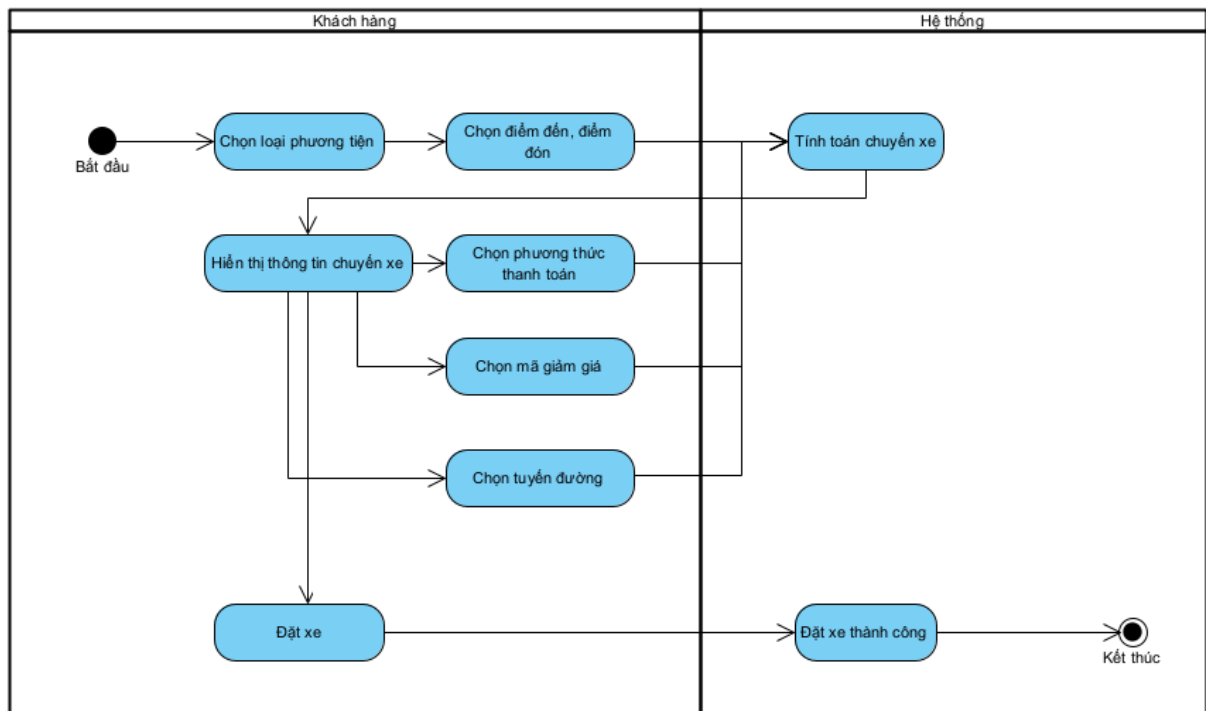
Hình 3.18 Biểu đồ hoạt động Đăng nhập

#### 3.3.2 Đăng ký



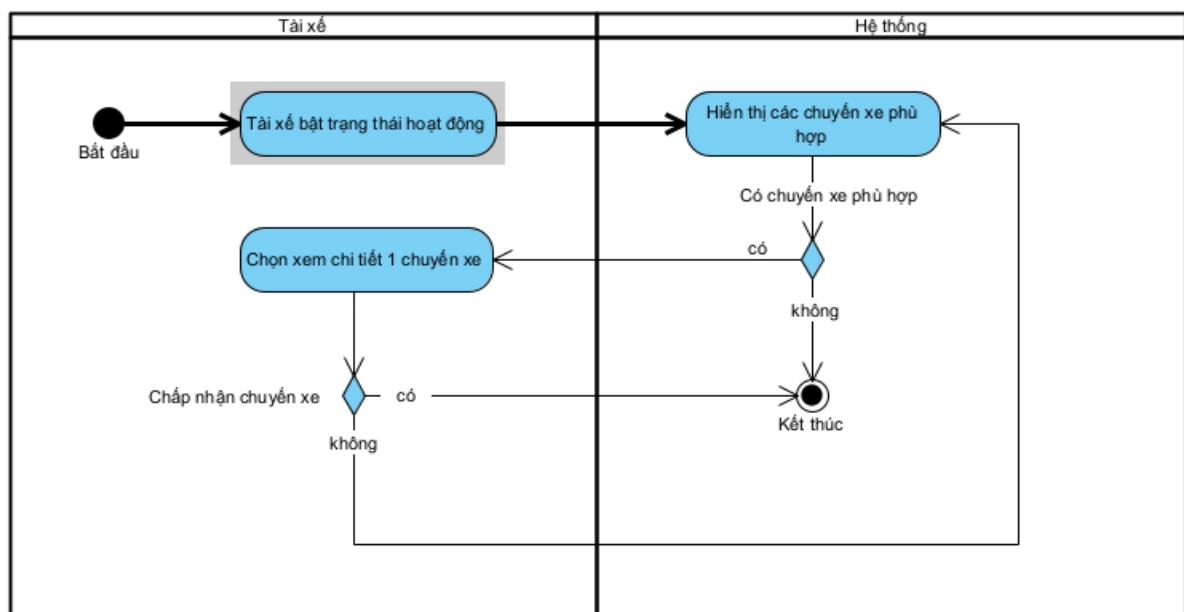
Hình 3.19 Biểu đồ hoạt động Đăng ký

### 3.3.3 Khách hàng đặt xe



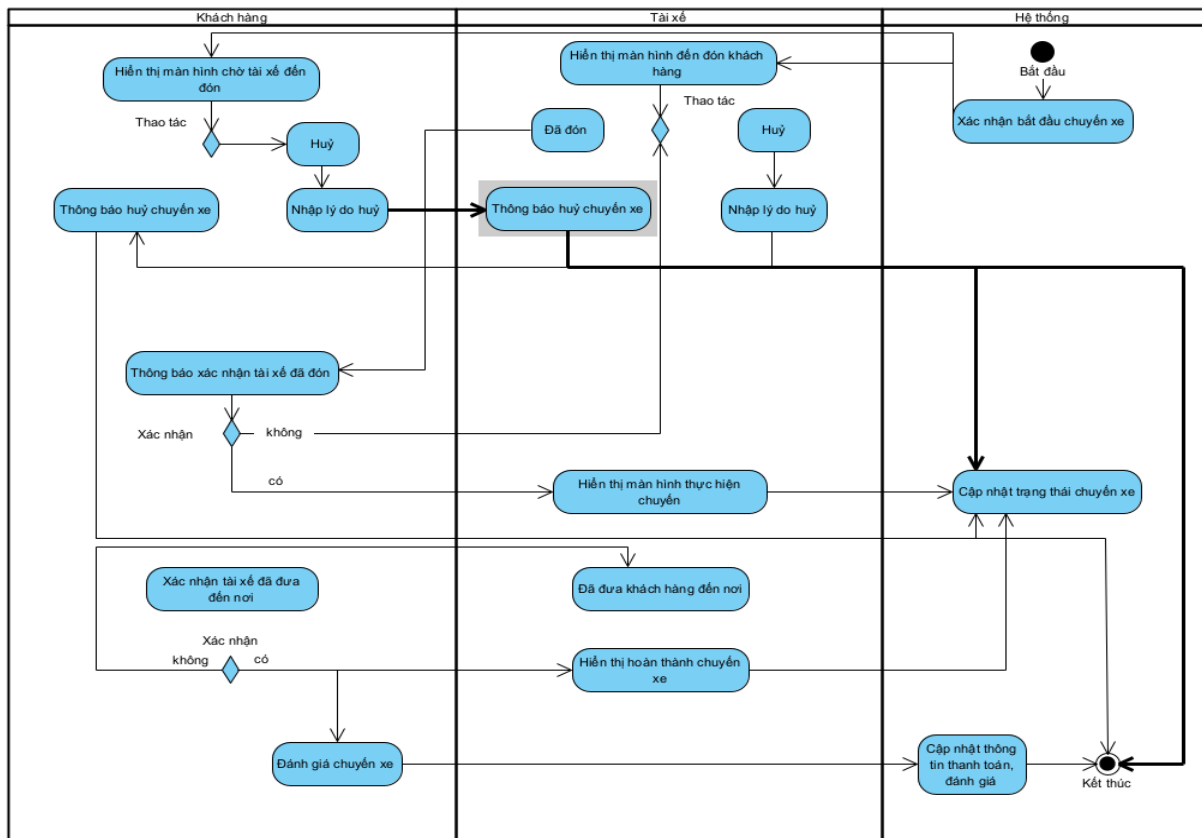
Hình 3.20 Biểu đồ hoạt động Khách hàng đặt xe

### 3.3.4 Tài xế nhận chuyến xe



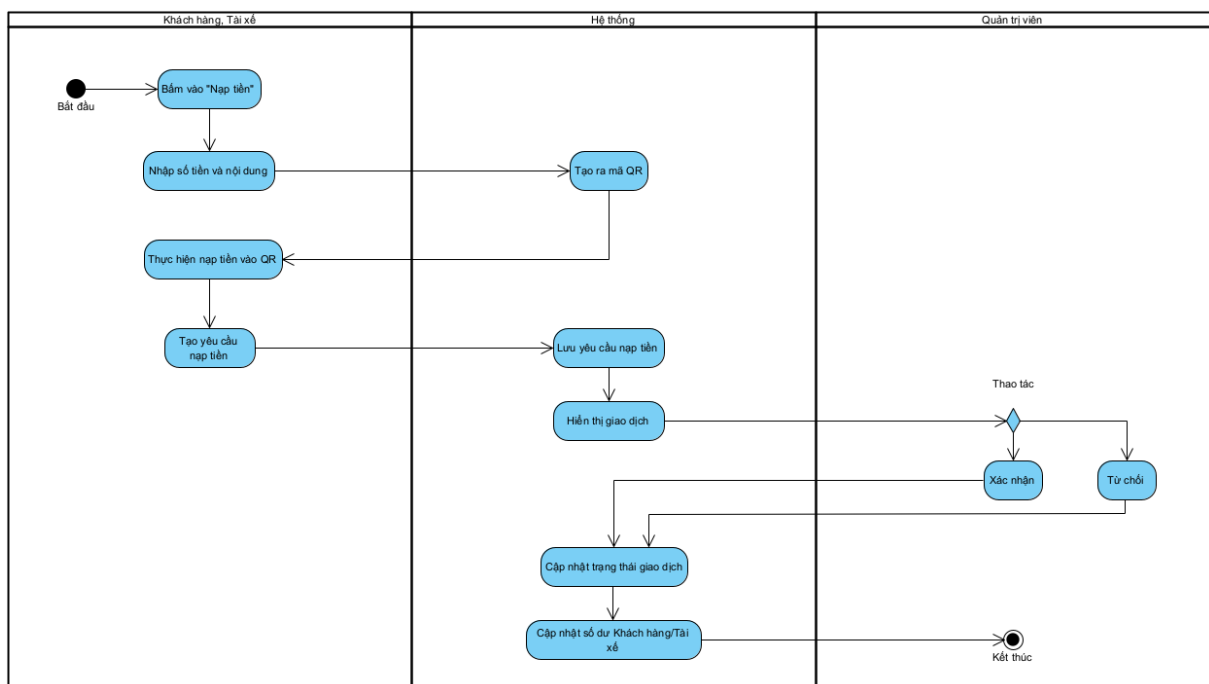
Hình 3.21 Biểu đồ hoạt động Tài xế nhận chuyến xe

### 3.3.5 Thực hiện chuyển xe



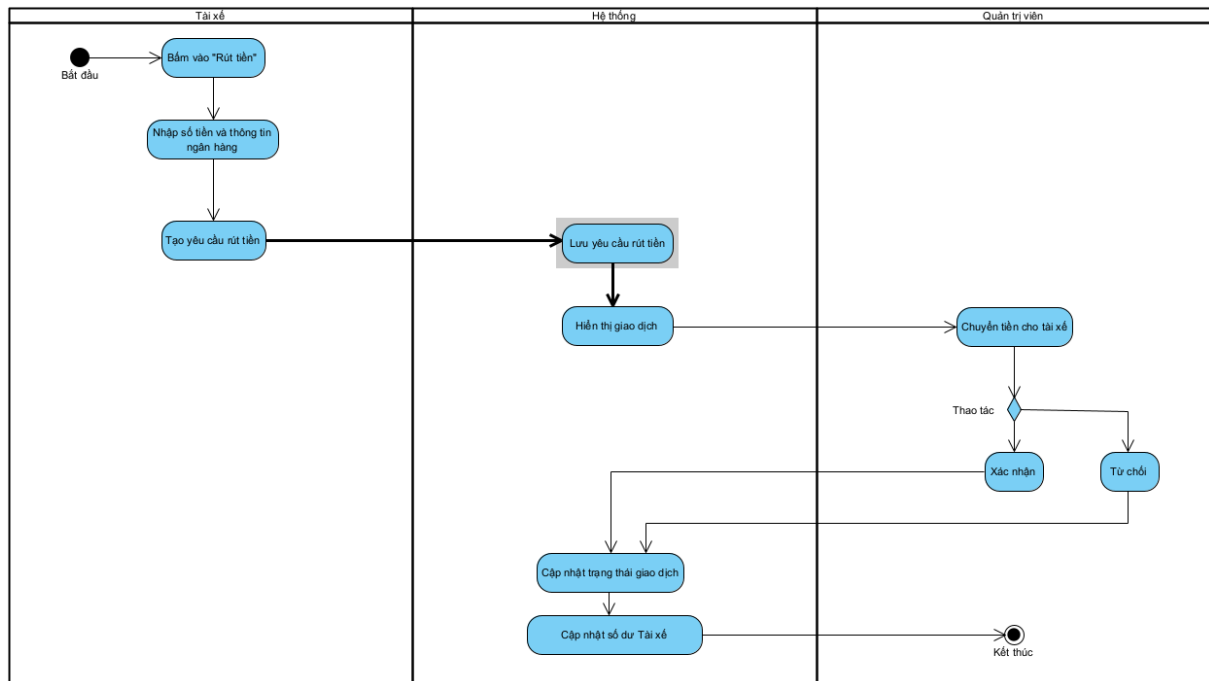
Hình 3.22 Biểu đồ hoạt động Thực hiện chuyển xe

### 3.3.6 Nạp tiền vào tài khoản ví



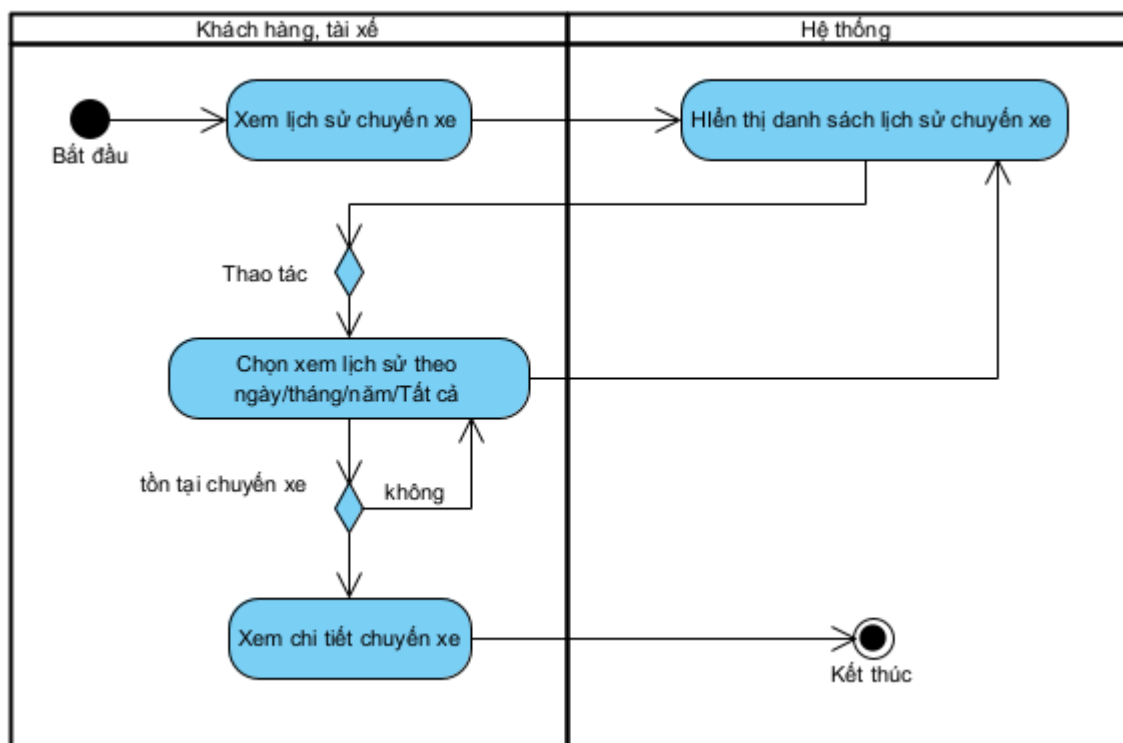
Hình 3.23 Biểu đồ hoạt động Nạp tiền vào tài khoản ví

### 3.3.7 Rút tiền khởi tài khoản ví



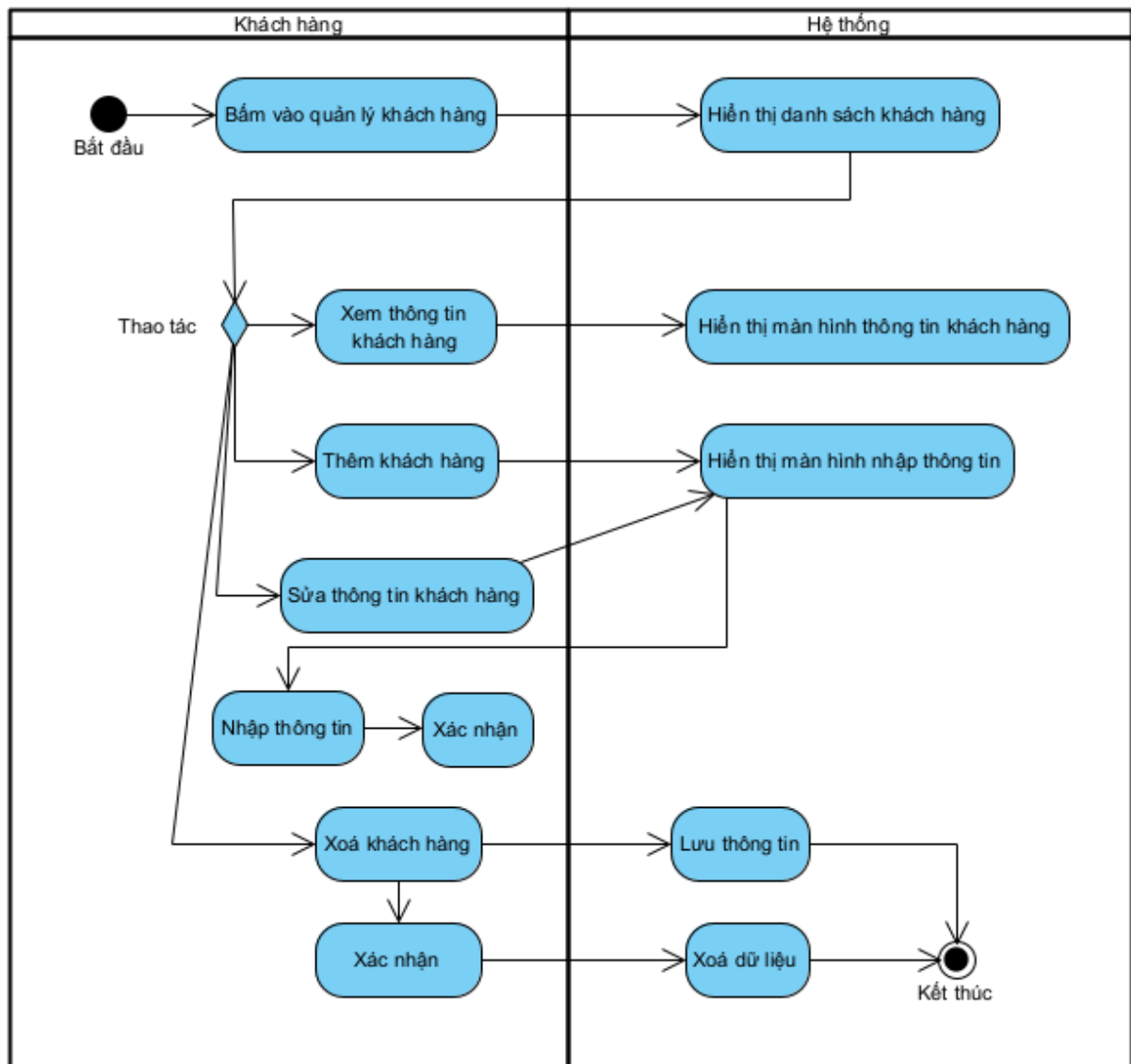
Hình 3.24 Biểu đồ hoạt động Tài xế rút tiền khởi tài khoản ví

### 3.3.8 Xem lịch sử chuyến xe



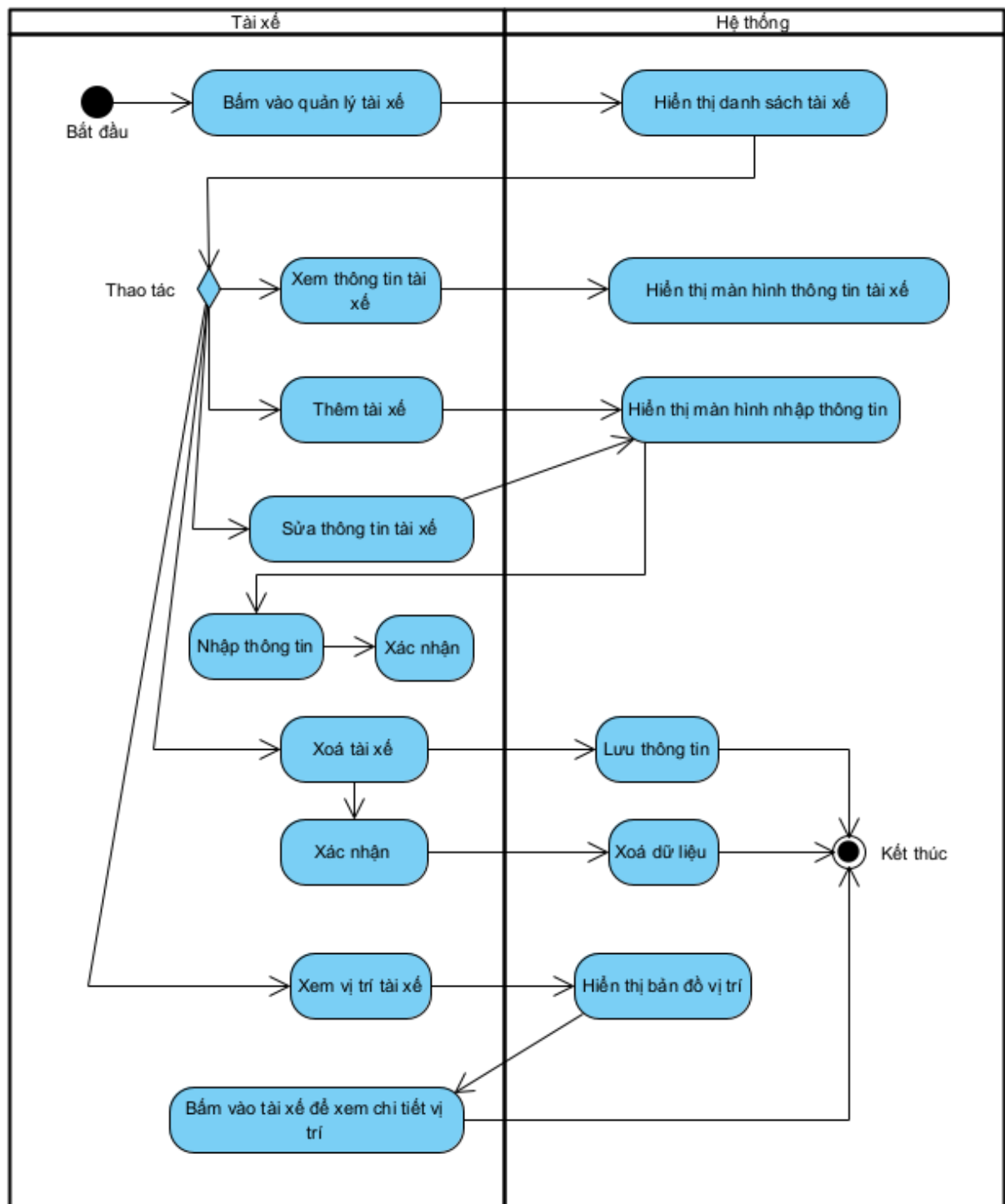
Hình 3.25 Biểu đồ hoạt động Xem lịch sử chuyến xe

### 3.3.9 Quản lý khách hàng



Hình 3.26 Biểu đồ hoạt động Quản lý khách hàng

### 3.3.10 Quản lý tài xế



Hình 3.27 Biểu đồ hoạt động Quản lý tài xế

### 3.4 Đặc tả yêu cầu

#### 3.4.1 Yêu cầu giao diện bên ngoài

##### 3.4.1.1 Ứng dụng khách hàng

- Màn hình đăng ký/đăng nhập: Cho phép khách hàng đăng ký tài khoản mới hoặc đăng nhập vào tài khoản hiện có.
- Màn hình đặt xe: Cho phép khách hàng chọn điểm đón, điểm đến, và loại xe.
- Màn hình theo dõi chuyến đi: Hiển thị vị trí tài xế và thông tin chuyến đi.
- Màn hình lịch sử: Hiển thị lịch sử các chuyến đi đã thực hiện.
- Màn hình ví: Cho phép khách hàng xem số dư, xem lịch sử giao dịch và có thể tạo giao dịch nạp tiền vào ví.

##### 3.4.1.2 Ứng dụng tài xế

- Màn hình đăng nhập: Cho phép tài xế đăng nhập vào tài khoản hiện có.
- Màn hình nhận chuyến: Hiển thị các yêu cầu chuyến đi từ khách hàng và cho phép tài xế chấp nhận chuyến đi.
- Màn hình chi tiết chuyến đi: Hiển thị chi tiết chuyến đi và dẫn đường cho tài xế.
- Màn hình lịch sử: Hiển thị lịch sử các chuyến đi đã thực hiện.
- Màn hình ví: Cho phép khách hàng xem số dư, xem lịch sử giao dịch và có thể tạo giao dịch nạp tiền vào ví cũng như tạo yêu cầu rút tiền.

##### 3.4.1.3 Ứng dụng quản trị

- Màn hình quản lý người dùng: Quản lý thông tin tài xế và khách hàng.
- Màn hình theo dõi thời gian thực: Hiển thị vị trí của tất cả tài xế trên bản đồ.
- Màn hình thống kê: Hiển thị các báo cáo về hoạt động của hệ thống.

#### 3.4.2 Yêu cầu chức năng

##### 3.4.2.1 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



<b>Sự kiện chính</b>	<ol style="list-style-type: none"> <li>1. Khách hàng (Tài xế) mở ứng dụng</li> <li>2. Màn hình nhập số điện thoại của ứng dụng hiện lên</li> <li>3. Khách hàng (Tài xế) nhập số điện thoại và bấm nút Xác nhận</li> <li>4. Màn hình xác nhận mã OTP hiện ra</li> <li>5. Khách hàng (Tài xế) nhập mã OTP được gửi về điện thoại và bấm nút xác nhận</li> <li>6. Hệ thống thông báo Đăng nhập thành công</li> </ol>
<b>Sự kiện thay thế</b>	
<b>Hậu điều kiện</b>	
<b>Ngoại lệ</b>	<ol style="list-style-type: none"> <li>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.</li> <li>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</li> <li>6. Khách hàng (Tài xế) nhập sai mã OTP, hệ thống yêu cầu khách hàng nhập lại.</li> </ol>
<b>Ngoài ra</b>	

Bảng 3.1 Đăng nhập

## 3.4.2.2 Đăng ký

<b>Tên Use case</b>	Đăng ký
<b>Actor</b>	Khách hàng
<b>Điều kiện tiên quyết</b>	Khách hàng đã tải về ứng dụng và chưa có tài khoản
<b>Sự kiện chính</b>	<ol style="list-style-type: none"> <li>1. Khách hàng mở ứng dụng</li> <li>2. Màn hình nhập số điện thoại của ứng dụng hiện lên</li> <li>3. Khách hàng nhập số điện thoại và bấm nút Xác nhận</li> <li>4. Màn hình xác nhận mã OTP hiện ra</li> <li>5. Khách hàng nhập mã OTP được gửi về điện thoại và bấm nút xác nhận</li> </ol>

	6. Hệ thống thông báo Xác thực số điện thoại thành công và màn hình hiển thị 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
<b>Sự kiện thay thế</b>	
<b>Hậu điều kiện</b>	Thông tin tài khoản khách hàng sẽ được lưu vào hệ thống
<b>Ngoại lệ</b>	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
<b>Ngoài ra</b>	

Bảng 3.2 Đăng ký

## 3.4.2.3 Đặt xe

<b>Tên Use case</b>	Đặt xe
<b>Actor</b>	Khách hàng
<b>Tham chiếu</b>	Tìm kiếm chuyển xe ở phần 3.2.4
<b>Điều kiện tiên quyết</b>	Khách hàng đã đăng nhập vào ứng dụng
<b>Sự kiện chính</b>	1. Khách hàng tiến hành Tìm kiếm chuyển xe

	2. Khách hàng chọn phương thức thanh toán, chọn mã giảm giá (nếu có) và chọn tuyến đường mong muốn 3. Khách hàng bấm vào nút Đặt xe 4. Thông báo “Đặt xe thành công” hiện ra.
<b>Luồng thay thế</b>	
<b>Hậu điều kiện</b>	
<b>Ngoại lệ</b>	2. Nếu số dư trong ví ứng dụng của khách hàng không còn đủ để thanh toán cho chuyến xe, khách hàng không thể lựa chọn thanh toán bằng phương thức “Ví ứng dụng”
<b>Ngoài ra</b>	

Bảng 3.3 Đặt xe

## 3.4.2.4 Tìm kiếm chuyến xe

<b>Tên Use case</b>	Tìm kiếm chuyến xe
<b>Actor</b>	Khách hàng
<b>Điều kiện tiên quyết</b>	Khách hàng đã đăng nhập vào ứng dụng
<b>Sự kiện chính</b>	1. Khách hàng bấm vào ô “Bạn muốn đi đâu?” 2. Màn hình chọn loại xe hiện lên 3. Khách hàng chọn loại xe và bấm Tiếp 4. Màn hình nhập điểm đón và điểm đến hiện ra 5. Khách hàng nhập vào điểm đón và điểm đến 6. Màn hình bản đồ với các thông tin điểm đón, điểm đến, đường đi, thời gian, giá tiền ước tính hiện ra
<b>Luồng thay thế</b>	- 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 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 6 ở trên</p> <p>- Ở bước 2, khách hàng chọn vào biểu tượng xe tương ứng mà khách hàng muốn đi</p> <p>2. Màn hình nhập vào điểm đón điểm đến hiện ra</p> <p>Các bước tiếp theo sẽ bắt đầu từ bước 5 ở trên</p>
<b>Hậu điều kiện</b>	<p>1. Thông tin chuyến xe sẽ được lưu vào hệ thống</p> <p>2. Khách hàng tới đúng địa chỉ Điểm đến</p>
<b>Ngoại lệ</b>	3. Hệ thống không xác nhận được địa điểm mà khách hàng nhập
<b>Ngoài ra</b>	

Bảng 3.4 Tìm kiếm chuyến xe

## 3.4.2.5 Xem lịch sử chuyến xe

<b>Tên Use case</b>	Xem lịch sử chuyến xe
<b>Actor</b>	Khách hàng, Tài xế
<b>Điều kiện tiên quyết</b>	Khách hàng, tài xế đã đăng nhập vào ứng dụng
<b>Sự kiện chính</b>	<p>1. Khách hàng, tài xế bấm vào Xem lịch sử chuyến xe</p> <p>2. Màn hình lịch sử chuyến xe hiện ra</p> <p>3. Khách hàng, tài xế lựa chọn cách hiển thị lịch sử chuyến xe thông qua bộ lọc</p> <p>4. Lịch sử chuyến xe hiện ra</p> <p>5. Khách hàng bấm vào để xem chi tiết chuyến xe</p>
<b>Luồng thay thế</b>	
<b>Hậu điều kiện</b>	
<b>Ngoại lệ</b>	4. Khách hàng, tài xế không có chuyến xe nào, hệ thống sẽ hiển thị là chưa có chuyến xe
<b>Ngoài ra</b>	

Bảng 3.5 Xem lịch sử chuyến xe

## 3.4.2.6 Quản lý thông tin cá nhân

<b>Tên Use case</b>	Quản lý thông tin cá nhân
---------------------	---------------------------

<b>Actor</b>	Khách hàng, tài xế
<b>Điều kiện tiên quyết</b>	Khách hàng (tài xế) đã đăng nhập vào ứng dụng
<b>Sự kiện chính</b>	<ol style="list-style-type: none"> <li>1. Khách hàng (tài xế) bấm vào phần Tài khoản</li> <li>2. Màn hình tài khoản khách hàng hiện lên</li> <li>3. Khách hàng (tài xế) bấm vào thông tin cá nhân</li> <li>4. Màn hình thông tin cá nhân của khách hàng hiện lên</li> <li>5. Khách hàng (tài xế) chọn sửa thông tin cá nhân</li> <li>6. Màn hình sửa thông tin cá nhân của khách hàng hiện lên</li> <li>7. Khách hàng (tài xế) chọn sửa các thông tin Tên, email, giới tính, số điện thoại và bấm nút Xác nhận</li> <li>8. Hệ thống thông báo khách hàng sửa thông tin cá nhân thành công</li> </ol>
<b>Luồng thay thế</b>	<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"> <li>8. Màn hình nhập mã OTP hiện ra</li> <li>9. Khách hàng (tài xế) nhập vào mã OTP được gửi về số điện thoại mới thay đổi và bấm xác nhận</li> <li>10. Hệ thống thông báo khách hàng sửa thông tin cá nhân thành công</li> </ol>
<b>Hậu điều kiện</b>	Thông tin cá nhân khách hàng được cập nhật vào hệ thống
<b>Ngoại lệ</b>	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
<b>Ngoài ra</b>	

Bảng 3.6 Quản lý thông tin cá nhân

## 3.4.2.7 Đánh giá chuyển xe

<b>Tên Use case</b>	Đánh giá chuyển xe
<b>Actor</b>	Khách hàng

<b>Điều kiện tiên quyết</b>	Khách hàng đã hoàn thành chuyến xe
<b>Sự kiện chính</b>	<ol style="list-style-type: none"> <li>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 và ghi vào đánh giá rồi bấm Xác nhận</li> <li>2. Hệ thống xác nhận khách hàng đánh giá tài xế thành công và chuyển tới màn hình chính</li> </ol>
<b>Luồng thay thế</b>	
<b>Hậu điều kiện</b>	Thông tin đánh giá sẽ được lưu vào hệ thống, đánh giá được phân loại thông qua mô hình
<b>Ngoại lệ</b>	
<b>Ngoài ra</b>	

*Bảng 3.7 Đánh giá chuyến xe*

## 3.4.2.8Huỷ chuyến xe

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

*Bảng 3.8 Huỷ chuyến xe*

## 3.4.2.9 Thanh toán

<b>Tên Use case</b>	Thanh toán
<b>Actor</b>	Khách hàng
<b>Điều kiện tiên quyết</b>	Khách hàng đã hoàn thành chuyến xe
<b>Sự kiện chính</b>	<ol style="list-style-type: none"> <li>1. Sau khi đã hoàn thành chuyến xe, kết thúc chuyến xe hiện lên</li> <li>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"> <li>1. Khách hàng chọn thanh toán tiền mặt thì sẽ đưa tiền mặt cho tài xế</li> <li>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 thông qua ứng dụng ngân hàng của tài xế</li> <li>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</li> </ol> </li> <li>3. Hệ thống hiển thị thông báo chuyến xe hoàn thành.</li> </ol>
<b>Luồng thay thế</b>	
<b>Hậu điều kiện</b>	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.
<b>Ngoại lệ</b>	
<b>Ngoài ra</b>	

*Bảng 3.9 Thanh toán*

## 3.4.2.10 Nạp tiền vào tài khoản ví

<b>Tên Use case</b>	Nạp tiền vào tài khoản ví
<b>Actor</b>	Khách hàng (Tài xế)

<b>Điều kiện tiên quyết</b>	Khách hàng (Tài xế) đã đăng nhập vào ứng dụng
<b>Sự kiện chính</b>	<ol style="list-style-type: none"> <li>1. Khách hàng (tài xế) bấm vào nút “Nạp tiền”</li> <li>2. Màn hình nạp tiền hiện ra</li> <li>3. Khách hàng (tài xế) nhập vào số tiền cần nạp, nội dung nạp tiền và bấm Tạo QR nạp tiền</li> <li>4. QR nạp tiền hiện ra, khách hàng tiến hành chuyển tiền theo QR đã có và bấm nút “Yêu cầu nạp tiền”</li> <li>5. Thông báo xác nhận Yêu cầu nạp tiền hiện ra</li> <li>6. Khách hàng (tài xế) bấm vào nút “Xác nhận”</li> <li>7. Hệ thống thông báo Tạo yêu cầu nạp tiền thành công, Khách hàng (tài xế) đợi tới khi người quản trị xác nhận giao dịch thì tiền sẽ được thêm vào ví.</li> </ol>
<b>Luồng thay thế</b>	
<b>Hậu điều kiện</b>	Thông tin yêu cầu thanh toán sẽ được lưu vào hệ thống và hiển thị cho người quản trị xem
<b>Ngoại lệ</b>	7. Người quản trị từ chối yêu cầu nạp tiền của khách hàng, Giao dịch sẽ có trạng thái là Thất bại và được hiển thị trong Lịch sử giao dịch của khách hàng
<b>Ngoài ra</b>	

Bảng 3.10 Nạp tiền vào tài khoản ví

## 3.4.2.11 Thực hiện chuyển xe

<b>Tên Use case</b>	Thực hiện chuyển xe
<b>Actor</b>	Khách hàng (tài xế)
<b>Điều kiện tiên quyết</b>	Khách hàng đã đặt yêu cầu chuyển xe, Tài xế đã xem được chi tiết chuyến đi của khách hàng
<b>Sự kiện chính</b>	<ol style="list-style-type: none"> <li>1. Tài xế bấm vào nút “Nhận chuyển xe”</li> <li>2. Màn hình thực hiện chuyển xe bao gồm vị trí tài xế, vị trí điểm đón, đường đi từ điểm đến tới điểm đón và thông tin chi tiết chuyến xe hiện lên ở cả màn hình của</li> </ol>



	<p>Tài xế và khách hàng, thông báo “Tài xế đã nhận chuyến xe” hiển thị bên phía màn hình khách hàng</p> <p>3. Sau khi tài xế đã tới điểm đón, tài xế bấm nút “Đã đón”</p> <p>4. Bên phía màn hình Tài xế, thông báo “Chờ khách hàng xác nhận” hiện lên. Bên phía màn hình khách hàng, Thông báo yêu cầu “Xác nhận tài xế đã đến đón” hiện lên.</p> <p>5. Khách hàng bấm vào nút “Xác nhận”</p> <p>6. Màn hình trong chuyến xe hiện ra, lúc này hệ thống sẽ hiển thị đường đi từ vị trí hiện tại đến điểm đến.</p> <p>7. Sau khi đến điểm đến, tài xế chọn Hoàn thành chuyến xe</p> <p>8. Bên phía màn hình Tài xế, thông báo “Chờ khách hàng xác nhận” hiện lên. Bên phía màn hình khách hàng, Thông báo yêu cầu “Xác nhận tài xế đưa bạn đến điểm đến” hiện lên.</p> <p>9. Khách hàng bấm vào nút “Xác nhận”</p> <p>10. Màn hình kết thúc chuyến đi hiện lên, khách hàng tiến hành thanh toán và đánh giá cho tài xế.</p>
<b>Luồng thay thế</b>	<p>Ở bước 3, Tài xế hoặc khách hàng bấm vào nút “Huỷ”</p> <p>4. Màn hình nhập lý do huỷ hiện ra</p> <p>5. Khách hàng (tài xế) nhập lý do huỷ và bấm xác nhận</p> <p>6. Thông báo “Bạn có chắc chắn muốn huỷ chuyến xe” hiện lên</p> <p>7. Khách hàng (tài xế) bấm vào xác nhận</p> <p>8. Màn hình kết thúc chuyến đi hiện ra</p>
<b>Hậu điều kiện</b>	Thông tin chuyến xe được lưu lại trong hệ thống, nếu tài xế huỷ chuyến thì điểm của tài xế sẽ bị giảm
<b>Ngoại lệ</b>	Tất cả các trường hợp thông báo hiển thị chọn “Huỷ” hoặc “Xác nhận”, nếu Khách hàng (tài xế) chọn Huỷ thì sẽ quay trở lại màn hình trước đó
<b>Ngoài ra</b>	

*Bảng 3.11 Thực hiện chuyển xe*

## 3.4.2.12 Tạo yêu cầu rút tiền

<b>Tên Use case</b>	Tạo yêu cầu rút tiền
<b>Actor</b>	Tài xế
<b>Điều kiện tiên quyết</b>	Tài xế đã đăng nhập và số dư tài khoản của tài xế > 0đ
<b>Sự kiện chính</b>	<ol style="list-style-type: none"> <li>1. Tài xế vào ví, chọn phần “Rút tiền”</li> <li>2. Màn hình tạo yêu cầu rút tiền hiện ra</li> <li>3. Tài xế nhập vào số tiền muốn rút và bấm xác nhận</li> <li>4. Hệ thống thông báo “Tạo yêu cầu rút tiền thành công”</li> </ol>
<b>Luồng thay thế</b>	
<b>Hậu điều kiện</b>	Thông tin yêu cầu rút tiền sẽ được lưu vào hệ thống
<b>Ngoại lệ</b>	3. Nếu số tiền tài xế muốn rút lớn hơn số dư hiện tại của tài xế, hệ thống sẽ thông báo không đủ số dư và bắt tài xế nhập lại
<b>Ngoài ra</b>	

*Bảng 3.12 Tạo yêu cầu rút tiền*

## 3.4.2.13 Quản lý khách hàng

<b>Tên Use case</b>	Quản lý khách hàng (tài xế)
<b>Actor</b>	Quản trị viên
<b>Điều kiện tiên quyết</b>	Quản trị viên đã đăng nhập vào hệ thống
<b>Sự kiện chính</b>	<ol style="list-style-type: none"> <li>1. Quản trị viên bấm vào Quản lý khách hàng (tài xế)</li> <li>2. Màn hình quản lý khách hàng (tài xế) hiện lên</li> <li>3. Quản trị viên lựa chọn các Thêm khách hàng (tài xế)</li> <li>4. Màn hình thêm khách hàng (tài xế) hiện ra</li> <li>5. Quản trị viên nhập các thông tin của khách hàng (tài xế) mới và bấm Thêm khách hàng</li> <li>6. Hệ thống thông báo thêm khách hàng (tài xế) thành công và trở về trang trước</li> </ol>

<b>Luồng thay thế</b>	<ul style="list-style-type: none"> <li>- Ở bước 3, Quản trị viên chọn Sửa</li> <li>4. Màn hình Sửa thông tin khách hàng (tài xế) hiện ra</li> <li>5. Quản trị viên nhập các thông tin cần sửa và bấm Xác nhận</li> <li>6. Hệ thống thông báo sửa thông tin khách hàng thành công và trở về trang trước</li> <li>- Ở bước 3, Quản trị viên chọn Xóa</li> <li>4. Thông báo Xác nhận xóa khách hàng (tài xế) hiện ra</li> <li>5. Khách hàng bấm vào Xác nhận</li> <li>6. Hệ thống thông báo Xóa khách hàng (tài xế) thành công và trở về trang trước</li> </ul>
<b>Hậu điều kiện</b>	Thông tin khách hàng (tài xế) sẽ được cập nhật vào hệ thống
<b>Ngoại lệ</b>	Ở những màn nhập thông tin, nếu Quản trị viên để trống hoặc không nhập đúng định dạng, hệ thống sẽ thông báo nhập đúng định dạng và yêu cầu Quản trị viên nhập lại
<b>Ngoài ra</b>	

Bảng 3.13 Quản lý khách hàng (tài xế)

## 3.4.2.14 Thống kê

<b>Tên Use case</b>	Thống kê
<b>Actor</b>	Quản trị viên
<b>Điều kiện tiên quyết</b>	Quản trị viên đã đăng nhập vào hệ thống
<b>Sự kiện chính</b>	<ol style="list-style-type: none"> <li>1. Quản trị viên bấm vào Thống kê</li> <li>2. Màn hình thống kê hiện ra</li> <li>3. Quản trị viên lựa chọn Nội dung thống kê bao gồm: Thống kê doanh thu, thống kê chuyến xe, Xếp hạng tài xế, tiếp đến Quản trị viên lựa chọn bộ lọc cho Thống kê</li> <li>4. Màn hình thống kê hiện ra với các thông tin chi tiết</li> </ol>
<b>Luồng thay thế</b>	

<b>Hậu điều kiện</b>	
<b>Ngoại lệ</b>	
<b>Ngoài ra</b>	

*Bảng 3.14 Thống kê*

### 3.5 Thiết kế cơ sở dữ liệu

#### 3.5.1 Danh sách các bảng trong Cơ sở dữ liệu

##### 3.5.1.1 customer (Khách hàng)

Tên trường	Thuộc tính	Ghi chú
id	varchar(10)	khoá chính, id
name	varchar(255)	tên khách hàng
phone_number	varchar(25)	số điện thoại khách hàng
gender	varchar(10)	giới tính
passcode	varchar(6)	mã 6 số
balance	long	số dư tài khoản
created_date	timestamp	thời gian tạo tài khoản
email	varchar(255)	email khách hàng

*Bảng 3.15 Bảng customer*

##### 3.5.1.2 driver (Tài xế)

Tên trường	Thuộc tính	Ghi chú
id	varchar(10)	khoá chính, id
name	varchar(255)	tên tài xế
phone_number	varchar(25)	số điện thoại tài xế
email	varchar(255)	email tài xế
gender	varchar(25)	giới tính
passcode	varchar(6)	mã 6 số

balance	long	số dư tài khoản
created_date	timestamp	thời gian tạo tài khoản
license_number	varchar(255)	mã bằng lái xe
vehicle_id	varchar(255)	khoá ngoại, mã phương tiện
lastest_location_lat	double	vĩ độ địa điểm cuối cùng tài xế gửi vị trí
lastest_location_lng	double	kinh độ địa điểm cuối cùng tài xế gửi vị trí
rating	double	đánh giá tài xế
status	varchar(25)	trạng thái tài xế

*Bảng 3.16 Bảng driver*

## 3.5.1.3 admin (Quản trị viên)

Tên trường	Thuộc tính	Ghi chú
id	varchar(10)	khoá chính, id
name	varchar(255)	tên Quản trị viên
phone_number	varchar(25)	số điện thoại Quản trị viên
username	varchar(255)	tài khoản
password	varchar(255)	mật khẩu

*Bảng 3.17 Bảng admin*

## 3.5.1.4 ride (Chuyến xe)

Tên trường	Thuộc tính	Ghi chú
id	varchar(10)	khoá chính, id
driver_id	varchar(255)	khoá ngoại, id tài xế
customer_id	varchar(25)	khoá ngoại, id khách hàng
pickup_location	varchar(255)	địa điểm đón

dropoff_location	varchar(255)	địa điểm đến
pickup_time	timestamp	thời gian đón
dropoff_time	timestamp	thời gian đến
status	varchar(50)	trạng thái
fare	long	số tiền chuyển xe
note	varchar(255)	ghi chú
created_time	timestamp	thời gian tạo chuyển xe
payment_method	varchar(255)	phương thức thanh toán chuyển xe
vehicle_type	varchar(255)	loại xe

*Bảng 3.18 Bảng ride*

## 3.5.1.5 transaction (Giao dịch)

Tên trường	Thuộc tính	Ghi chú
id	varchar(10)	khoá chính, id
amount	long	số tiền giao dịch
transaction_type	varchar(25)	loại giao dịch
timestamp	timestamp	thời gian xác nhận giao dịch
status	varchar(255)	trạng thái
created_time	timestamp	thời gian tạo giao dịch
description	varchar(255)	mô tả giao dịch
actor	varchar(255)	đối tượng giao dịch

*Bảng 3.19 Bảng transaction*

## 3.5.1.6 customer\_transaction (giao dịch của khách hàng)

Tên trường	Thuộc tính	Ghi chú
id	varchar(10)	khoá chính, id

customer_id	varchar(10)	khoá ngoại, id khách hàng
transaction_id	varchar(10)	khoá ngoại, id giao dịch

*Bảng 3.20 Bảng customer\_transaction*

## 3.5.1.7 driver\_transaction (giao dịch của tài xế)

Tên trường	Thuộc tính	Ghi chú
id	varchar(10)	khoá chính, id
driver_id	varchar(10)	khoá ngoại, id tài xế
transaction_id	varchar(10)	khoá ngoại, id giao dịch

*Bảng 3.21 Bảng driver\_transaction*

## 3.5.1.8 payment (thanh toán)

Tên trường	Thuộc tính	Ghi chú
id	varchar(10)	khoá chính, id
payment_method	varchar(50)	phương thức thanh toán
payment_status	varchar(50)	trạng thái
timestamp	timestamp	thời gian thanh toán
ride_id	varchar(10)	khoá ngoại, mã chuyến xe
transaction_id	varchar(10)	khoá ngoại, mã giao dịch

*Bảng 3.22 Bảng payment*

## 3.5.1.9 promotion (giảm giá)

Tên trường	Thuộc tính	Ghi chú
id	varchar(10)	khoá chính, id
code	varchar(50)	mã giảm giá
discount_amount	double	phần trăm giảm giá
customer_id	varchar(10)	khoá ngoại, mã khách hàng

expired_date	timestamp	thời gian hết hạn
min_price_validated	double	giá tiền tối thiểu áp dụng giảm giá
description	varchar(50)	mô tả giảm giá
max_discount	double	số tiền lớn nhất có thể được giảm giá
usage_limit	int	số lần sử dụng

*Bảng 3.23 Bảng promotion*

## 3.5.1.10 request\_ride (chuyến xe yêu cầu)

Tên trường	Thuộc tính	Ghi chú
id	varchar(10)	khoá chính, id
customer_id	varchar(10)	khoá ngoại, mã khách hàng
pickup_lat	double	vĩ độ điểm đón
pickup_lng	double	kinh độ điểm đón
dropoff_lat	double	vĩ độ điểm đến
dropoff_lng	double	kinh độ điểm đến
distance	double	khoảng cách
final_price	long	số tiền chuyến xe
coordinates	varchar	vị trí các điểm trên đường đi
duration	double	thời gian ước tính
payment_method	varchar(50)	phương thức thanh toán
vehicle_type	varchar(50)	loại xe
pickup_address	varchar(255)	địa chỉ điểm đón
dropoff_address	varchar(255)	địa chỉ điểm đến



*Bảng 3.24 Bảng request\_ride*

## 3.5.1.11 review (đánh giá)

Tên trường	Thuộc tính	Ghi chú
id	varchar(10)	khoá chính, id
ride_id	varchar(10)	khoá ngoại, mã chuyến xe
rating	double	đánh giá
comment	varchar(255)	đánh giá qua bình luận
sentiment	varchar(3)	phân loại cảm xúc bình luận

*Bảng 3.25 Bảng review*

## 3.5.1.12 vehicle (phương tiện)

Tên trường	Thuộc tính	Ghi chú
id	varchar(10)	khoá chính, id
name	varchar(255)	tên phương tiện
vehicle_type_id	varchar(10)	khoá ngoại, mã loại phương tiện
description	varchar(255)	mô tả phương tiện
code	varchar(10)	biển số xe

*Bảng 3.26 Bảng vehicle*

## 3.5.1.13 vehicle\_type (loại phương tiện)

Tên trường	Thuộc tính	Ghi chú
id	varchar(10)	khoá chính, id
name	varchar(255)	tên loại phương tiện
price_start	long	giá tiền khởi đầu 2km
price_per_km	long	giá tiền trên mỗi kilomet tính từ kilomet thứ 3

price_per_time	long	giá trên trên mỗi phút
----------------	------	------------------------

*Bảng 3.27 Bảng vehicle\_type*

## 3.5.1.14 notification (thông báo)

Tên trường	Thuộc tính	Ghi chú
id	varchar(10)	khoá chính, id
message	varchar(255)	nội dung thông báo
timestamp	timestamp	thời gian thông báo
title	varchar(255)	tiêu đề thông báo
type	varchar(10)	loại thông báo
is_read	bool	trạng thái đã đọc chưa

*Bảng 3.28 Bảng notification*

## 3.5.1.15 customer\_notification (thông báo của khách hàng)

Tên trường	Thuộc tính	Ghi chú
id	varchar(10)	khoá chính, id
customer_id	varchar(10)	khoá ngoại, id khách hàng
notification_id	varchar(10)	khoá ngoại, id thông báo

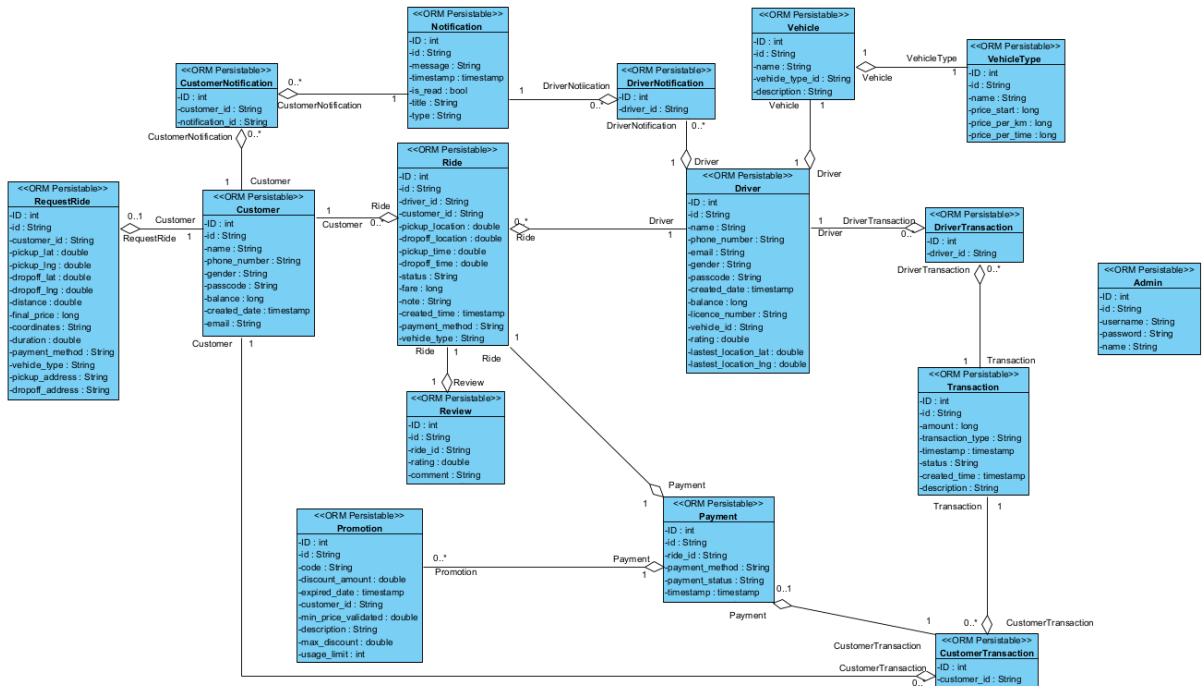
*Bảng 3.29 Bảng customer\_notification*

## 3.5.1.16 driver\_notification (thông báo của tài xế)

Tên trường	Thuộc tính	Ghi chú
id	varchar(10)	khoá chính, id
driver_id	varchar(10)	khoá ngoại, id tài xế
notification_id	varchar(10)	khoá ngoại, id thông báo

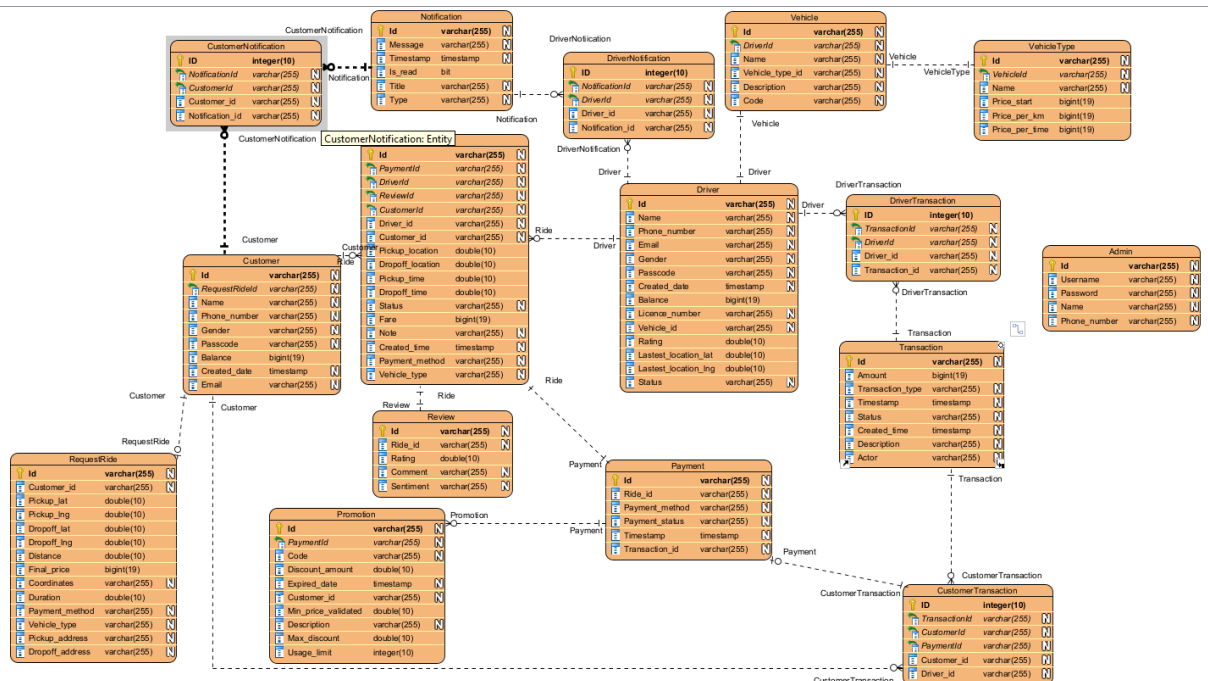
Bảng 3.30 Bảng driver\_notification

## 3.5.2 Biểu đồ lớp phân tích



Hình 3.28 Biểu đồ lớp phân tích

## 3.5.3 Mô hình thực thể liên kết



Hình 3.29 Mô hình thực thể liên kết

### **3.6 Kết luận Chương III**

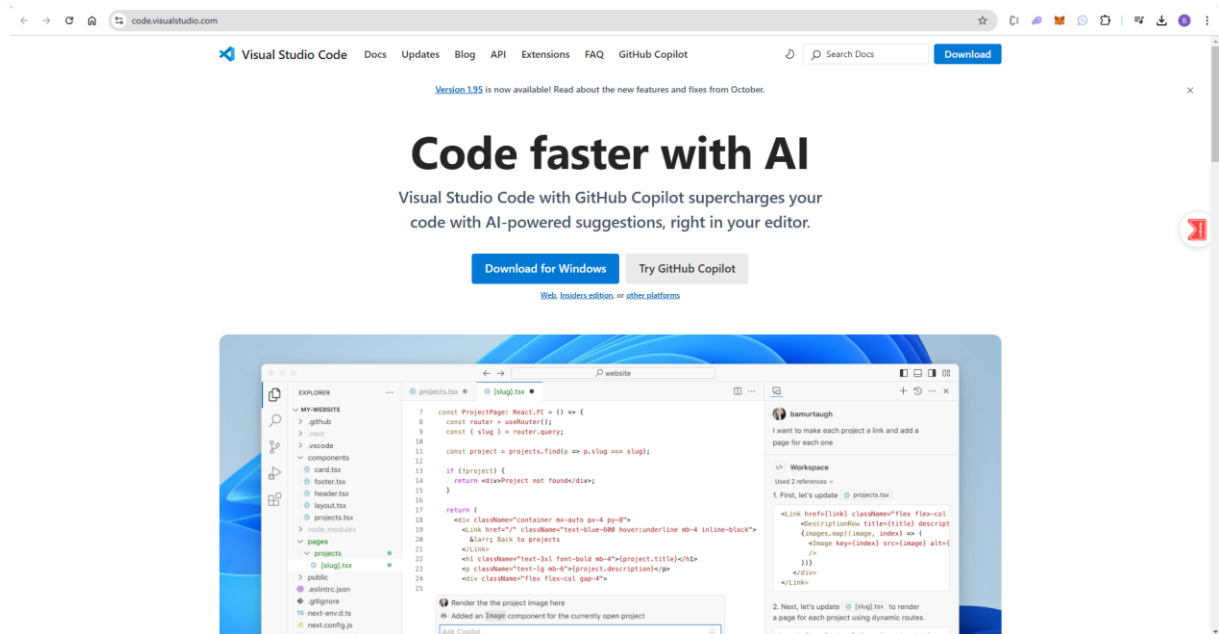
Như vậy, Chương III đã phân tích chi tiết các yêu cầu chức năng, xây dựng sơ đồ hoạt động và thiết kế cơ sở dữ liệu cho hệ thống đặt xe trực tuyến. Các biểu đồ Usecase và Activity đã được triển khai để mô phỏng các quy trình chính của hệ thống như đặt xe, thanh toán và quản lý thông tin. Thiết kế cơ sở dữ liệu đảm bảo khả năng lưu trữ và truy xuất thông tin một cách hiệu quả, đáp ứng các nhu cầu vận hành thực tế. Phân tích và thiết kế trong chương này sẽ là nền tảng vững chắc cho việc triển khai và kiểm thử hệ thống trong các chương tiếp theo.

## CHƯƠNG IV. CÀI ĐẶT HỆ THỐNG

### 4.1 Cài đặt môi trường làm việc

#### 4.1.1 Cài đặt Visual Studio Code

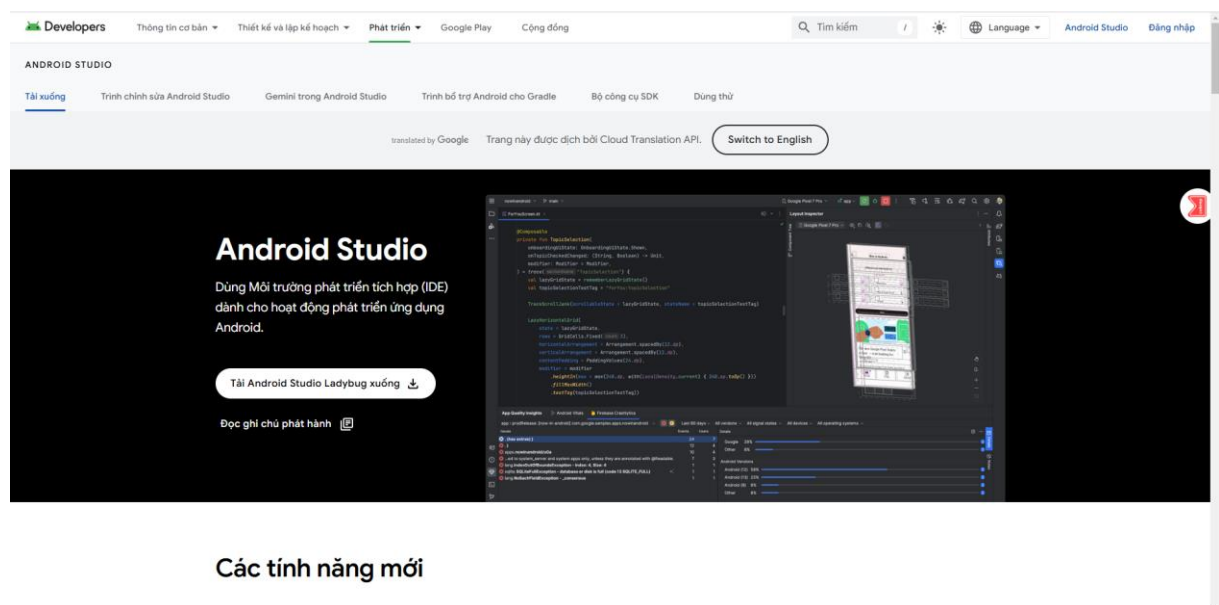
Vào [trang chủ](#) của Visual Studio Code và cài đặt theo hướng dẫn



Hình 4.1 Trang chủ Visual Studio Code

#### 4.1.2 Cài đặt Android Studio

Vào [trang chủ](#) của Android studio và cài đặt theo hướng dẫn



Các tính năng mới

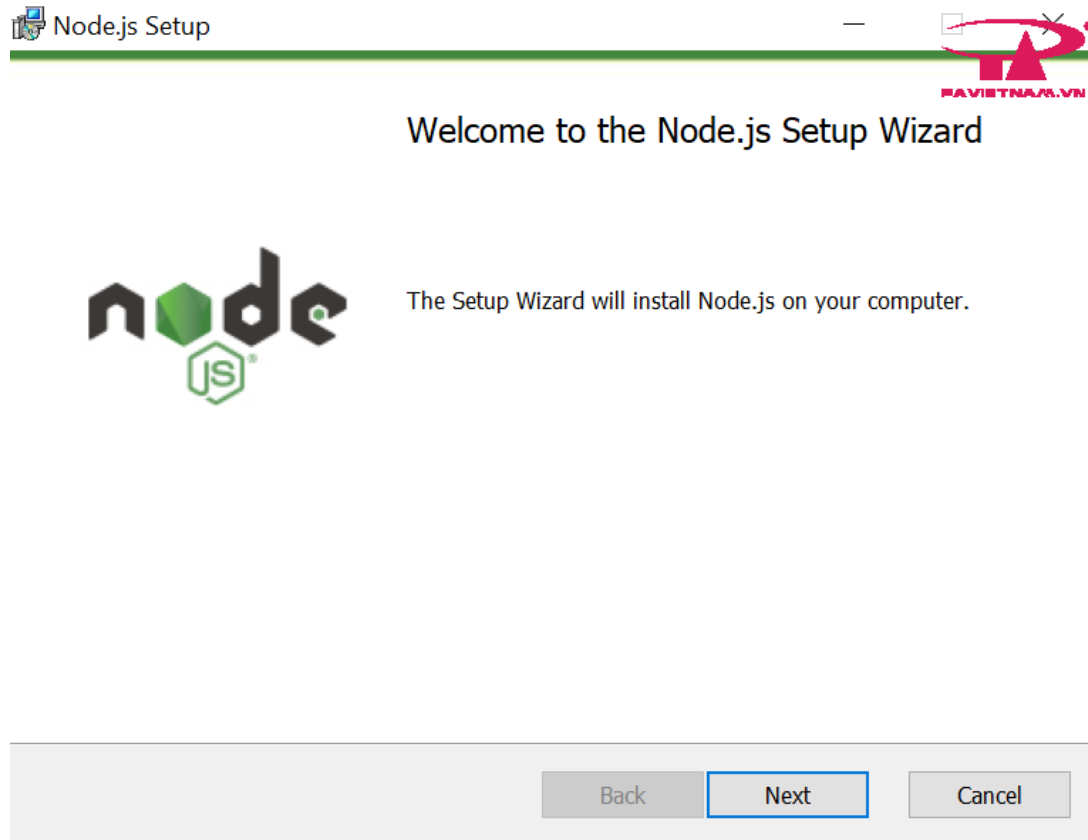
Hình 4.2 Trang chủ Android Studio

### 4.1.3 Cài đặt NodeJS và khởi tạo Dự án

1. Tải nodeJS tại đường dẫn ở [đây](#)

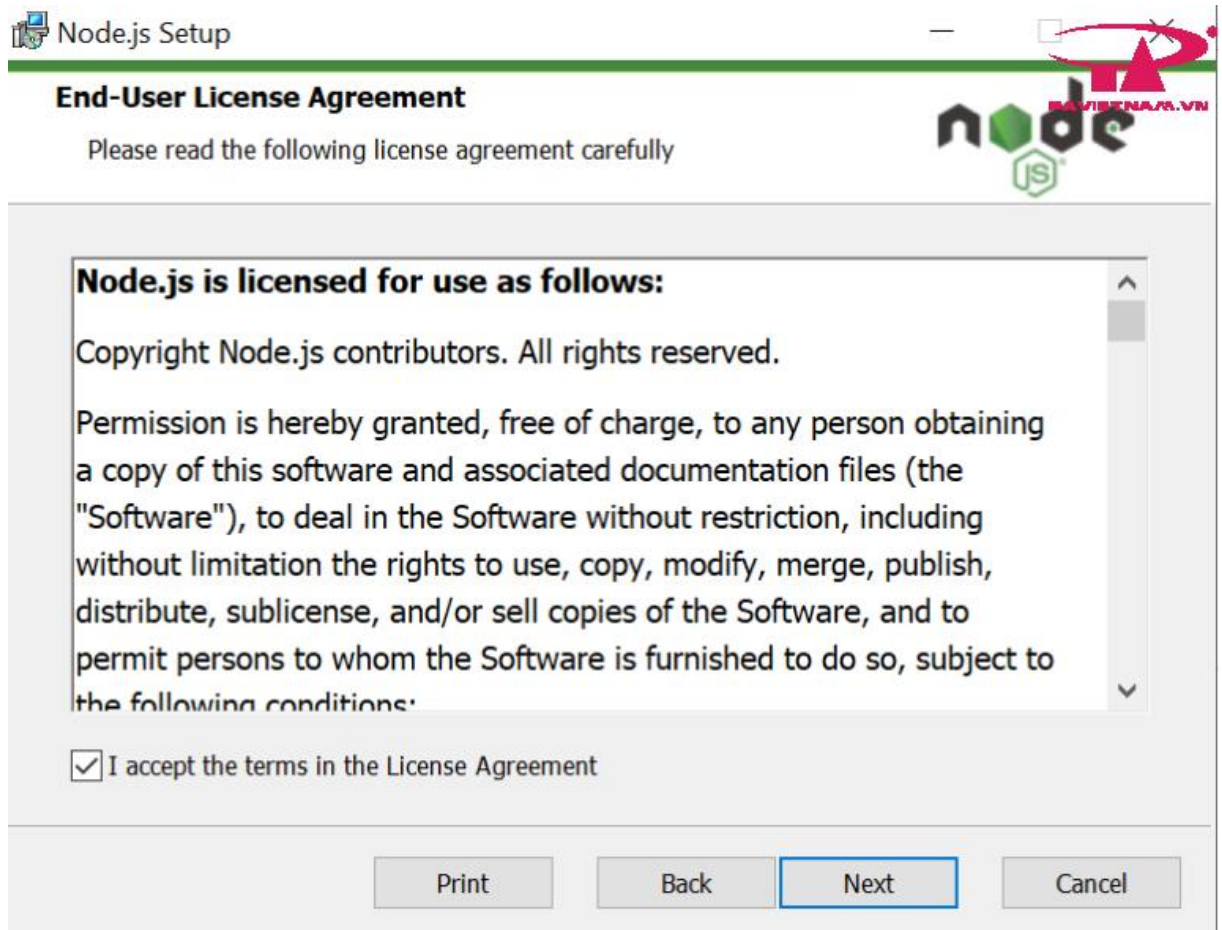
2. Cài đặt NodeJS

- Tại màn hình cài đặt bạn bấm Next



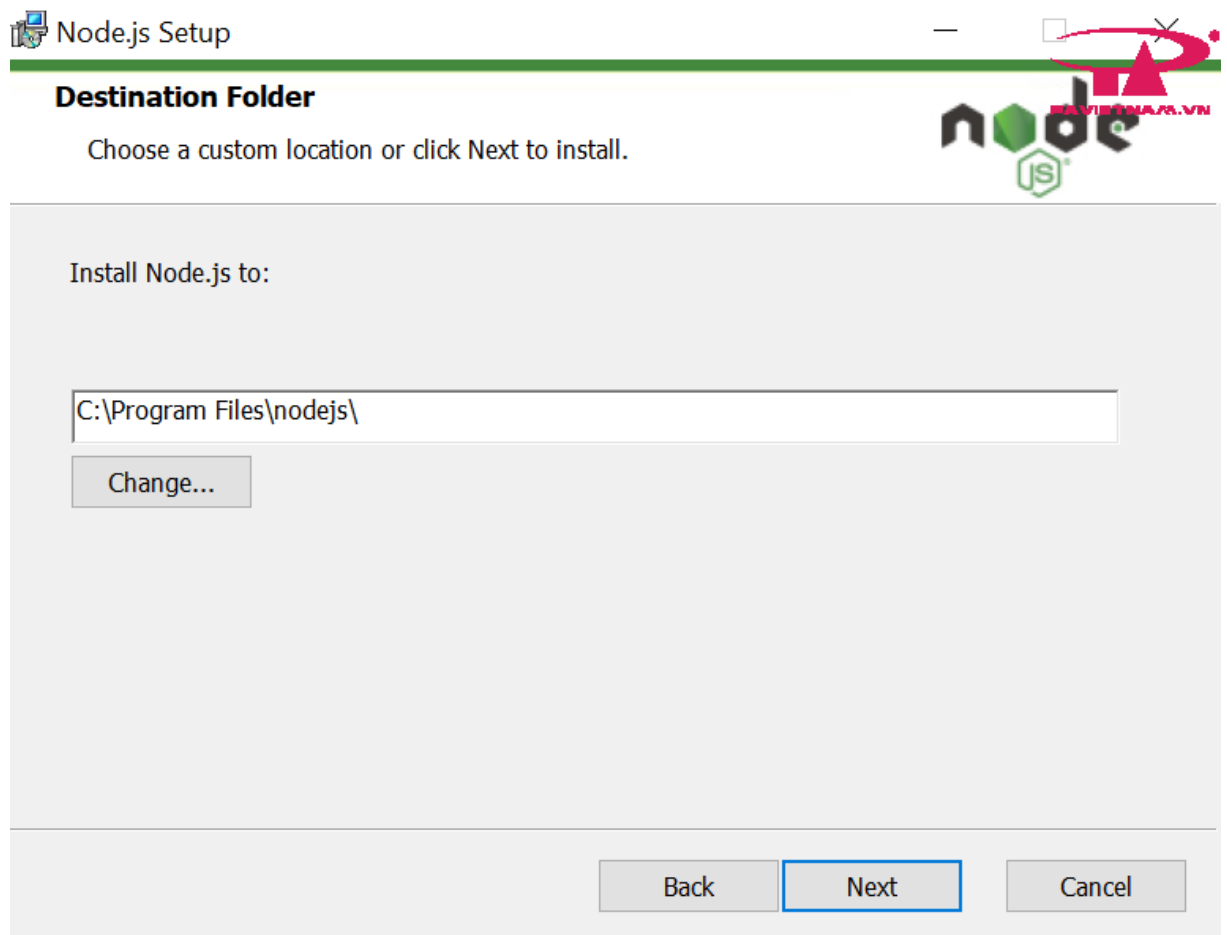
Hình 4.3 Cài đặt NodeJS #1

- Tích chọn “I accept...” và tiếp tục bấm Next



Hình 4.4 Cài đặt NodeJS #2

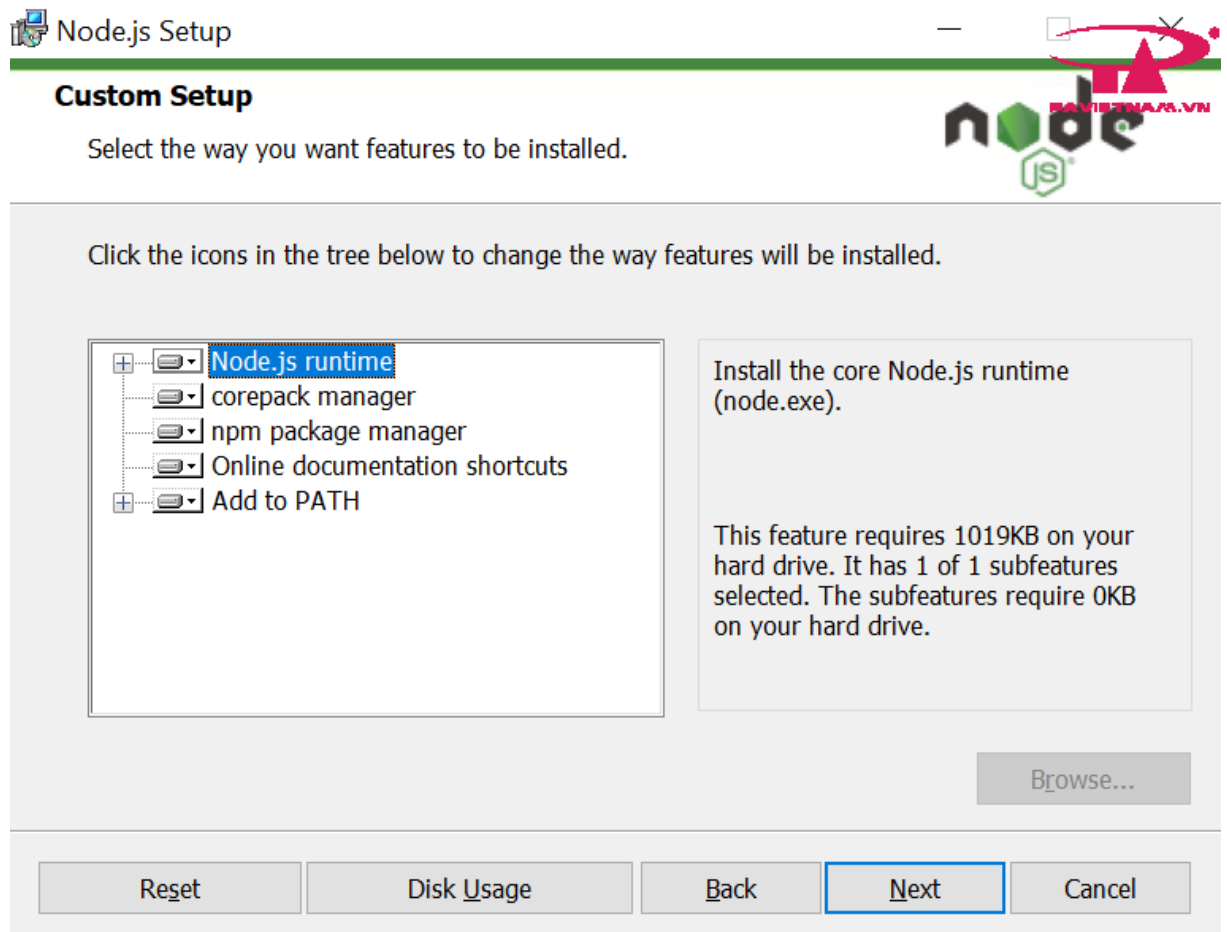
- Chọn nơi lưu trữ, bạn có thể bỏ qua và bấm Next.



Hình 4.5 Cài đặt NodeJS #3

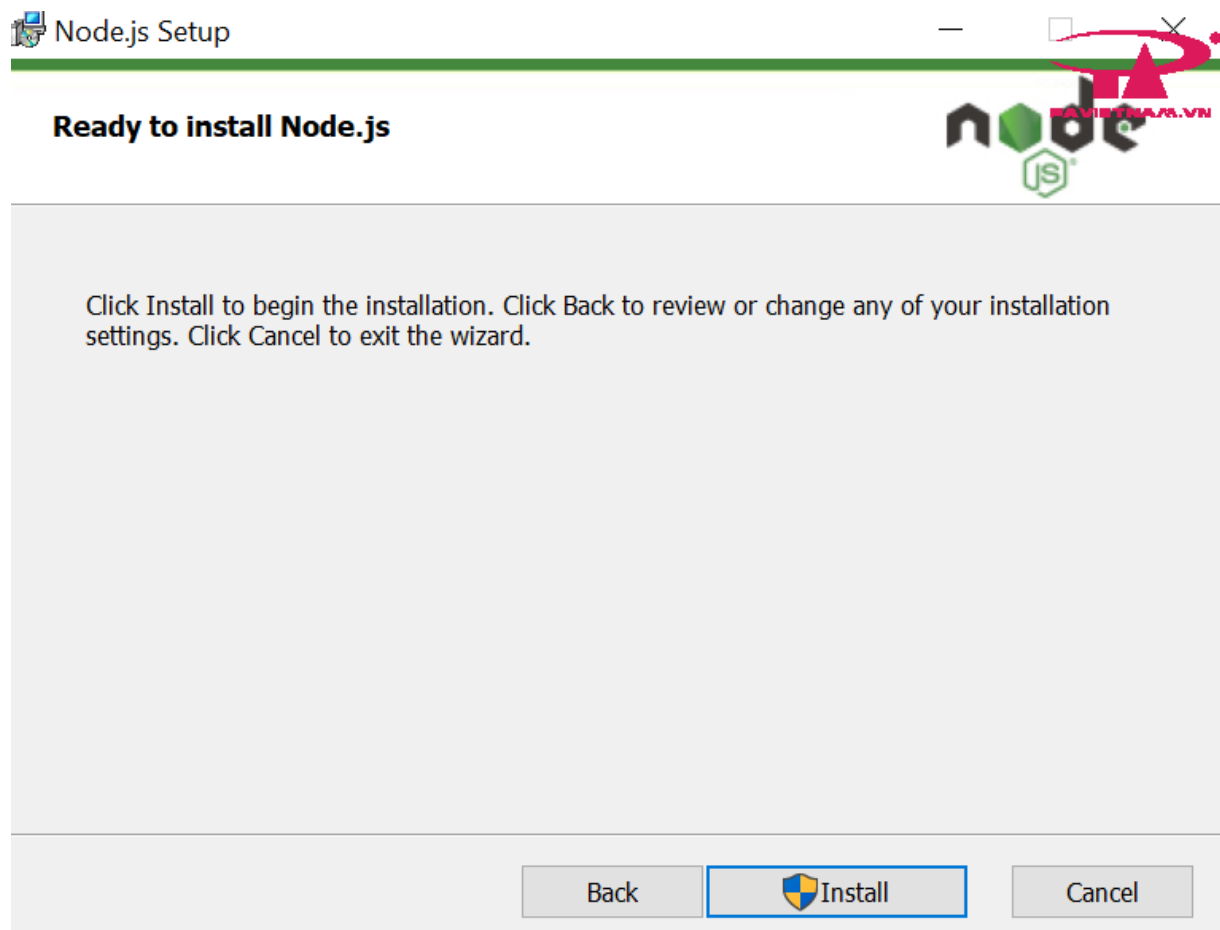


- Bước này chọn các tính năng trong NodeJS như NPM... Bạn có thể để mặc định và bấm Next.

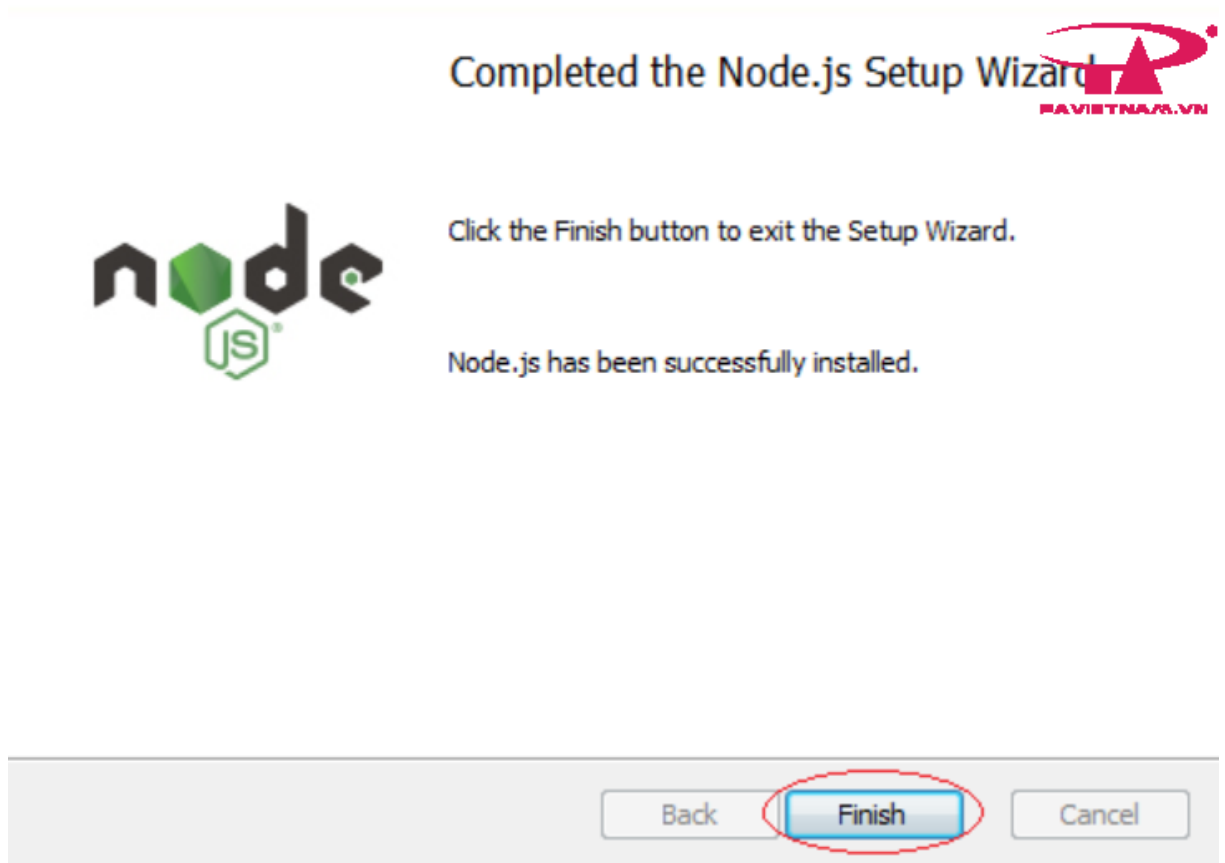


Hình 4.6 Cài đặt NodeJS #4

- Tiếp tục bấm Next đến bước Install thì bấm Install để bắt đầu cài đặt NodeJS, Cài đặt thành công bấm Finish để hoàn tất.



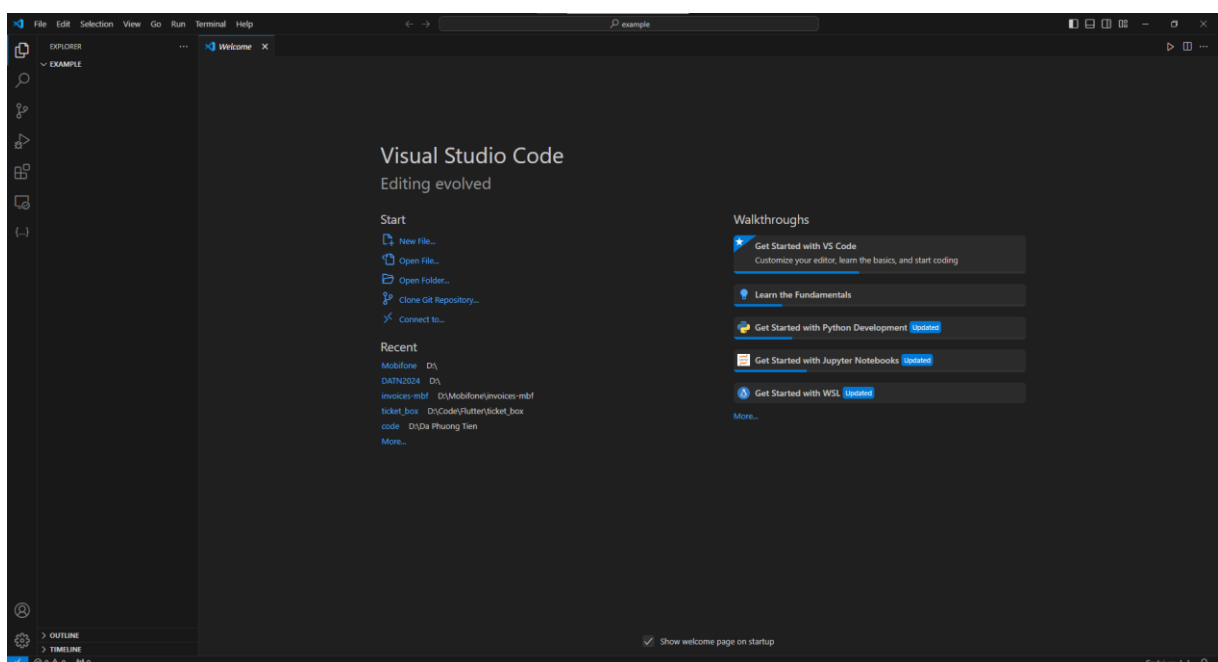
Hình 4.7 Cài đặt NodeJS #5



Hình 4.8 Cài đặt NodeJS #6

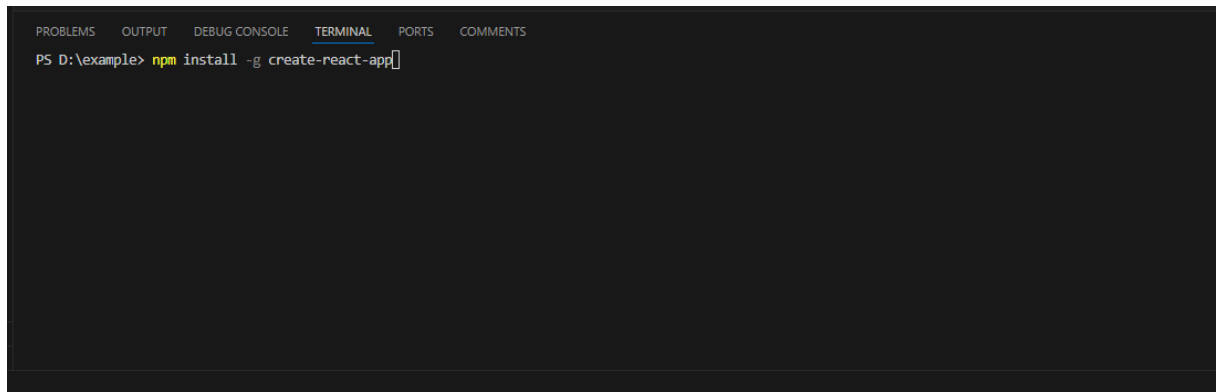
### 3. Khởi tạo dự án

- Mở Visual Studio Code, chọn vào địa chỉ sẽ khởi tạo dự án



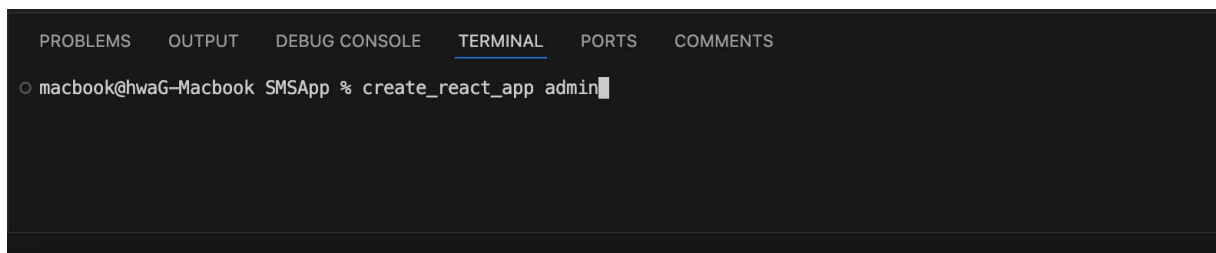
Hình 4.9 Màn hình Visual Studio Code

- Cài đặt Create React App bằng câu lệnh ***npm install -g create-react-app***



Hình 4.10 Chạy câu lệnh *npm install -g create-react-app*

- Tạo dự án React: sử dụng câu lệnh ***create-react-app admin***



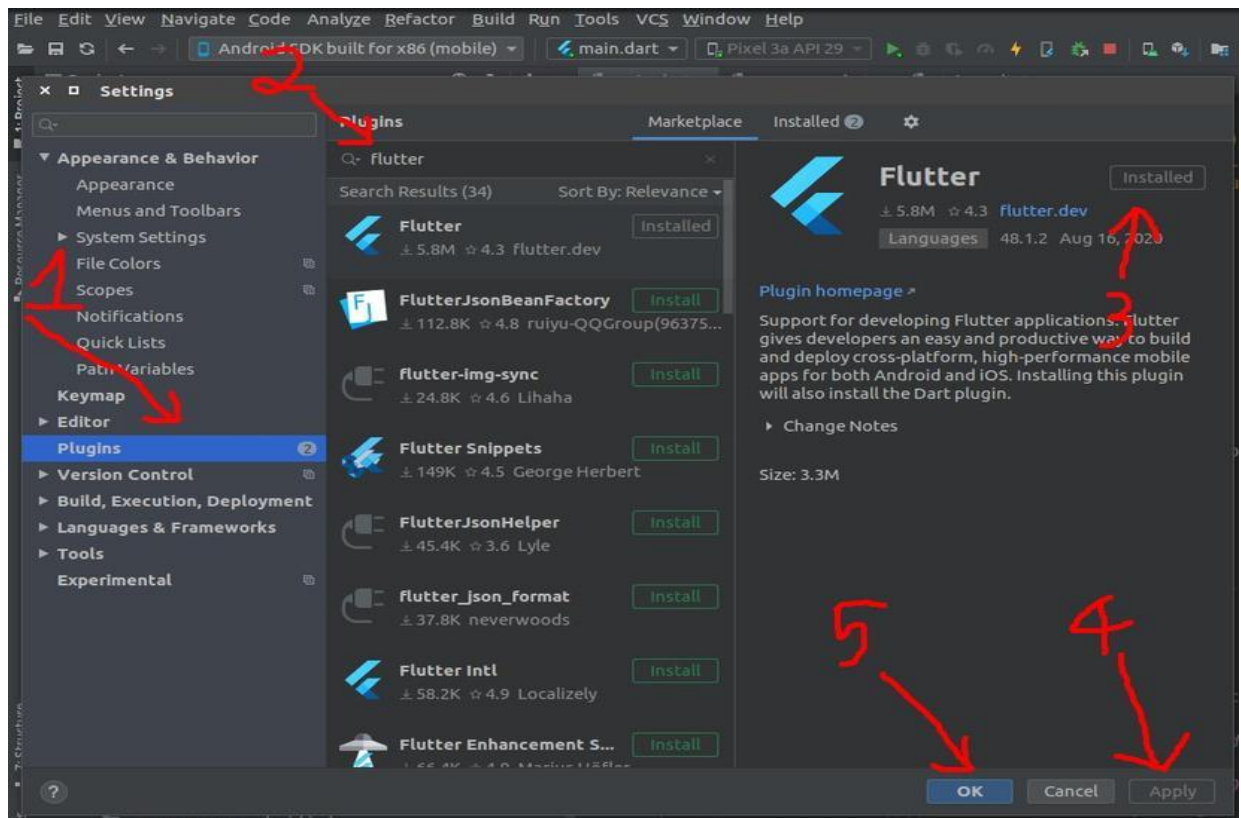
Hình 4.11 Chạy câu lệnh *create-react-app admin*

#### 4.1.4 Cài đặt Flutter

1. ***Cài đặt Flutter SDK tại đường dẫn ở đây (phiên bản 3.24)***
2. ***Setup Môi trường***

Với Android studio thì tiến hành cài đặt như bình thường, còn Sdk của Flutter thì chỉ cần giải nén vào một file nào đó chúng ta sẽ dùng sau. Sau khi download và cài đặt giải nén các thứ thì bây giờ hãy mở Android Studio lên và tiến hành các config dưới đây:

- Vào File -> Settings...
- Chọn Plugin -> search với từ khóa "flutter" và install như hình dưới đây

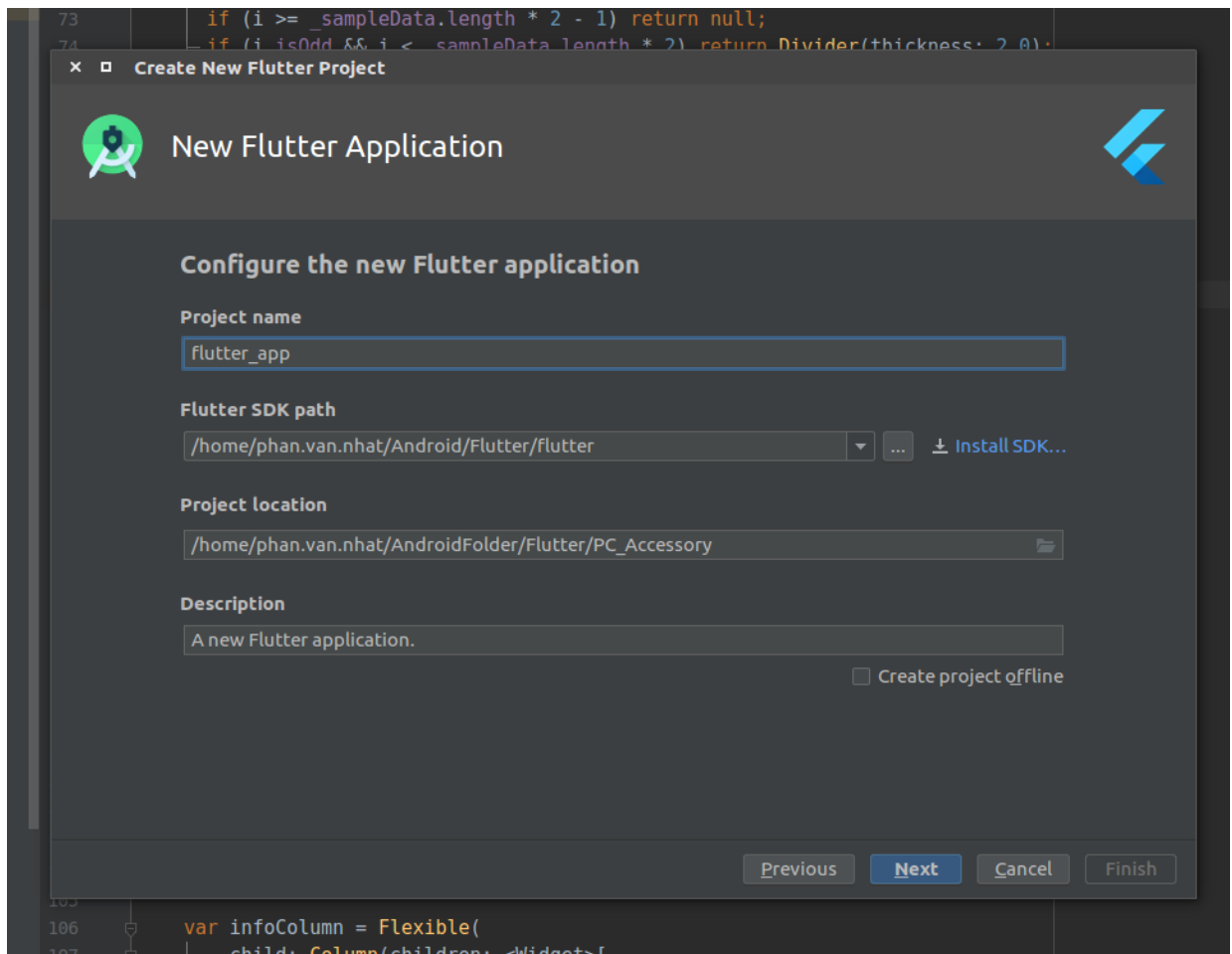


Hình 4.12 Cài đặt Flutter và Dart Plugin cho Android Studio

- Search thêm "dart" tại mục số (2) trong hình và install plug Dart tương tự.
- Chọn Apply -> OK
- Khởi động lại Android Studio.

### 3. Tạo mới Project Flutter

- Mở Android Studio Code
- Vào File -> New -> New Flutter Project (nó ở dòng thứ 2) -> Chọn cái đầu tiên là Flutter Application -> flutter\_app, chúng ta sẽ được một bảng tùy chọn như này



Hình 4.13 Tạo Dự án Flutter mới

## 4.2 Kiểm thử hệ thống

Trong quá trình phát triển hệ thống đặt xe trực tuyến, việc kiểm thử là bước quan trọng nhằm đảm bảo tính ổn định, an toàn và đáp ứng các yêu cầu đã đặt ra. Hoạt động kiểm thử được chia thành ba loại chính: kiểm thử đơn vị, kiểm thử tích hợp và kiểm thử hệ thống. Dưới đây là trình bày về các cách kiểm thử cho hệ thống.

### 4.2.1 Kiểm thử đơn vị (Unit Testing)

Kiểm thử từng chức năng nhỏ lẻ trong hệ thống, các chức năng kiểm thử

- Đăng nhập, đăng ký xác thực thông tin OTP
- Các API của Google Map: tìm kiếm địa chỉ, chỉ đường
- Tính toán giá cước dựa trên địa điểm (tính toán theo khoảng cách, thời gian, thời tiết...)
- Gửi và nhận thông tin thời gian thực thông qua MQTT
- Xem lịch sử chuyến xe, xem lịch sử giao dịch
- Tạo yêu cầu nạp tiền, tạo yêu cầu rút tiền

- Cập nhật thông tin cá nhân
- Đánh giá chuyến xe
- Thống kê dữ liệu

#### 4.2.2 Kiểm thử tích hợp (Integration Testing):

Kiểm thử các chức năng khi kết hợp với nhau

- Kết hợp giữa GPS và hiển thị vị trí trên bản đồ
- Kết hợp API chỉ đường của Google Map và hiển thị đường đi trên bản đồ
- Kết hợp chức năng tạo yêu cầu nạp tiền với API tạo mã QR thanh toán của VietQR
- Kết nối dữ liệu từ Back-end và cơ sở dữ liệu

#### 4.2.3 Kiểm thử hệ thống (System Testing):

Kiểm thử toàn bộ hệ thống trong môi trường giả lập gần giống với thực tế nhất

##### 4.2.3.1 Kiểm thử chức năng:

- Chức năng đặt xe
- Chức năng thực hiện chuyến xe
- Chức năng tạo yêu cầu nạp tiền, tạo yêu cầu rút tiền

##### 4.2.3.2 Kiểm thử cài đặt:

- Thử nghiệm cài đặt trên môi trường IOS và Android

##### 4.2.3.3 Kiểm thử hiệu năng:

- Kiểm thử khả năng hiển thị vị trí của tài xế khi di chuyển
- Kiểm tra tốc độ truyền tin của MQTT

#### 4.2.4 Các kịch bản kiểm thử (Test cases):

Kịch bản	Mô tả	Kỳ vọng
Kịch bản 1	Khách hàng đặt xe với thông tin hợp lệ	Hệ thống xác nhận đặt xe thành công và thông tin đặt xe được hiển thị trên màn hình của tài xế
Kịch bản 2	Tài xế bấm nhận chuyến xe	Hệ thống thông báo cho khách hàng đã có tài xế nhận chuyến

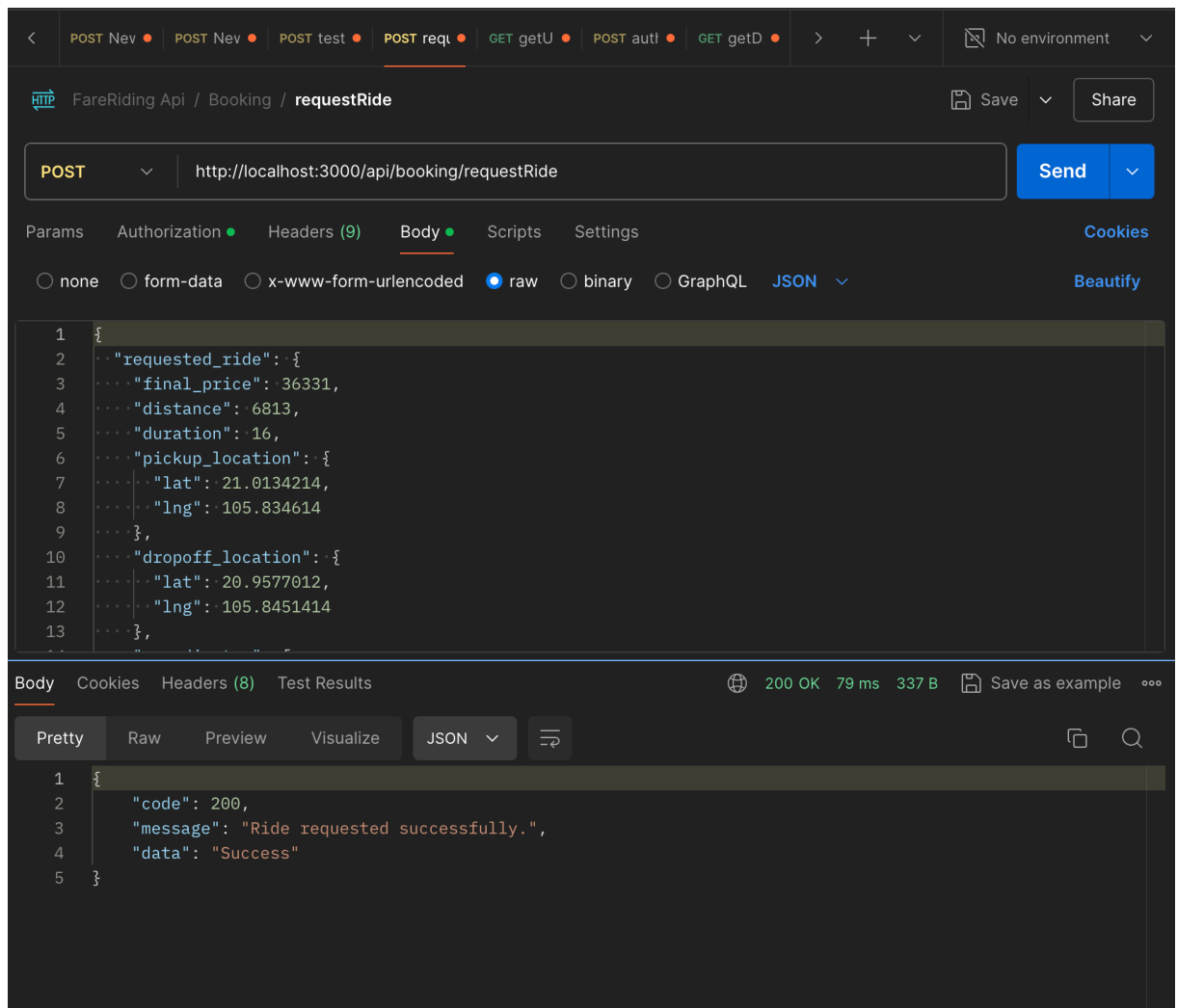
		xe và chuyển tới màn hình thực hiện chuyển xe
Kịch bản 3	Trong khi thực hiện chuyển xe, khi tài xế đang đến đón khách hàng, khách hàng hoặc tài xế bấm Huỷ chuyển xe và Xác nhận lý do	Cả 2 ứng dụng chuyển tới màn hình kết thúc chuyển đi với thông báo chuyển xe bị Huỷ
Kịch bản 4	Tài xế bấm vào "Hoàn thành chuyển xe", khách hàng bấm vào xác nhận trong thông báo	Cả 2 ứng dụng chuyển tới màn hình hoàn thành chuyển xe, khách hàng sẽ có phần đánh giá chuyển xe
Kịch bản 5	Tài xế hoặc khách hàng bấm tạo yêu cầu nạp tiền	Yêu cầu nạp tiền hiển thị trên hệ thống với các thông tin hợp lệ
Kịch bản 6	Tài xế tạo yêu cầu rút tiền nhiều hơn số tiền tài xế hiện có	Hệ thống thông báo số tiền rút không vượt quá số tiền hiện có
Kịch bản 7	Tài xế đang di chuyển thì mất mạng, sau đó có mạng trở lại	Hệ thống cập nhật ngay lập tức vị trí và đường đi của tài xế khi có mạng trở lại
Kịch bản 8	Khách hàng đánh giá chuyến xe	Hệ thống sẽ cập nhật trạng thái của đánh giá sau khi chạy qua model phân loại, bên phía Quản trị viên cũng sẽ được cập nhật trong bảng xếp hạng tài xế.

Bảng 4.1 Kịch bản Test Case

#### 4.2.5 Công cụ và môi trường kiểm thử:

- Postman: dùng để kiểm thử các API liên quan





Hình 4.14 Giao diện Postman

- Máy ảo: Sử dụng Android Emulator và IOS Simulator để giả lập các thiết bị di động trên các hệ điều hành Android 8.0 và IOS 13 trở lên



Hình 4.15 Máy ảo Android và IOS

## 4.2.6 Kết quả kiểm thử

### 4.2.6.1 Kết quả kiểm thử chức năng

- Chức năng Đặt xe: Hệ thống hiển thị chính xác thông tin đặt xe, thỉnh thoảng thời gian thực hiện tìm đường đi xe bị lâu do khoảng cách tìm kiếm quá xa
- Chức năng thực hiện chuyển xe: Hệ thống đồng bộ tốt vị trí ở cả 3 phía, thực hiện chuyển xe trơn tru, nhưng còn lỗi khi kết nối MQTT (rất ít khi nhưng vẫn bị mất kết nối giữa chừng)
- Chức năng Nạp/Rút tiền: Thực hiện chức năng tốt, thỉnh thoảng mã QR thanh toán sẽ xuất hiện lâu do mạng hoặc do phía VietQR
- Các API của Google Map: các API sử dụng tốt, nhưng sẽ vẫn có thiếu sót bị mất kết nối trong lúc thực hiện, nhất là API chỉ đường (vì gọi API chỉ đường rất nhiều)
- Chức năng Đánh giá tài xế thông qua AI: Vì sử dụng mô hình của bên thứ 3 (thư viện Transformer) nên thời gian thực hiện sẽ không ổn định, nhưng việc hoàn thiện công việc lại ổn định, không bị ngắt kết nối giữa chừng, công việc kiểm thử hoàn thiện 50/50 lần thử. Việc hiển thị xếp hạng trên trang Quản trị viên không có vấn đề.

### 4.2.6.2 Kết quả kiểm thử kết hợp

- Kết hợp giữa GPS và bản đồ Google Map: Hệ thống hiển thị chính xác đường đi và vị trí của tài xế khi di chuyển. Nhưng do vấn đề về mạng và layer miễn phí của Google Map nên nhiều khi bản đồ không được hiển thị đúng cách, nhưng đường đi vẫn chuẩn xác.
- Các chức năng còn lại thực hiện một cách tốt, có thì cũng chỉ là lỗi do mạng hoặc bên thứ 3.

### 4.2.6.3 Kết quả kiểm thử hệ thống

- Các Test Case thực hiện đúng quy trình và đều đạt được kỳ vọng của hệ thống.


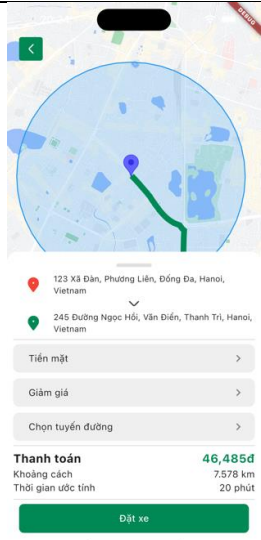
### 4.2.6.4 Kết luận

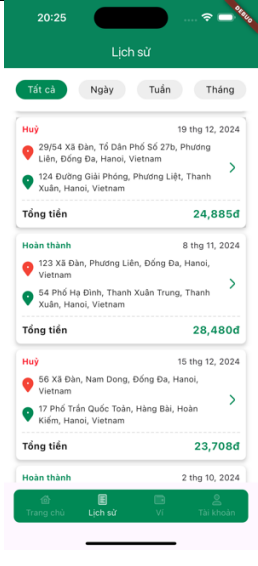
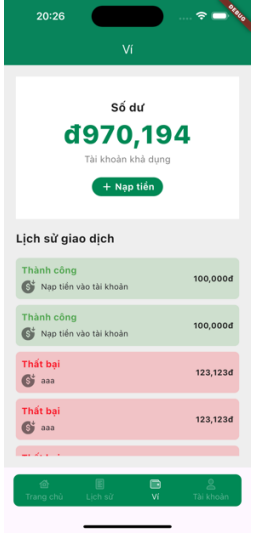

- Thành công: Tất cả các chức năng hoạt động ổn định, đáp ứng yêu cầu. Hầu hết các API của Goolge Map đều chạy trơn tru, chưa bị vượt khỏi ngân sách. Giao diện ưa nhìn, dễ sử dụng. Hệ thống truyền tin khá ổn định và đáng tin cậy


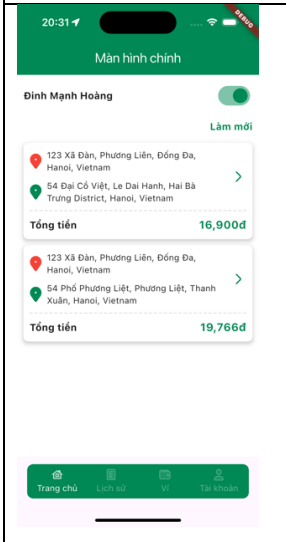
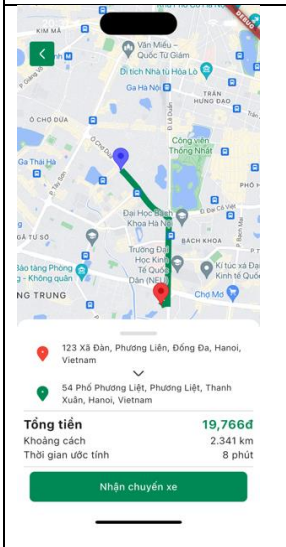
- Hạn chế còn tồn tại: Vấn đề về kết nối trong MQTT vẫn còn bị mất kết nối, vẫn còn bị trì hoãn một vài tác vụ như đặt xe, tạo mã QR do phụ thuộc vào bên thứ 3. Chưa thể cài đặt app trên môi trường thật IOS vì hạn chế về tài nguyên. Giao diện người quản trị vẫn còn sơ sài, chưa bắt mắt.

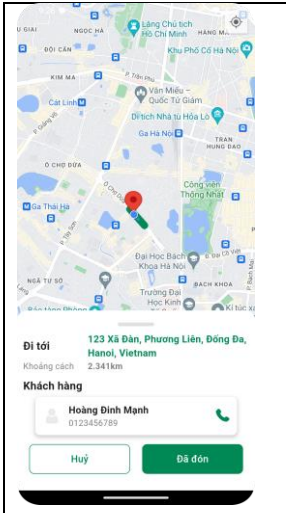
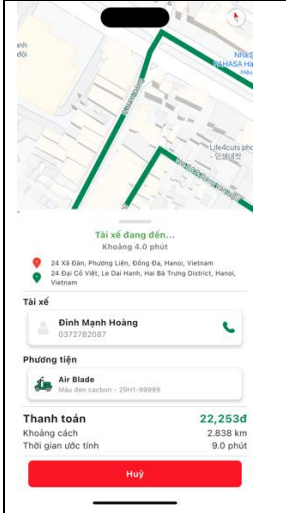
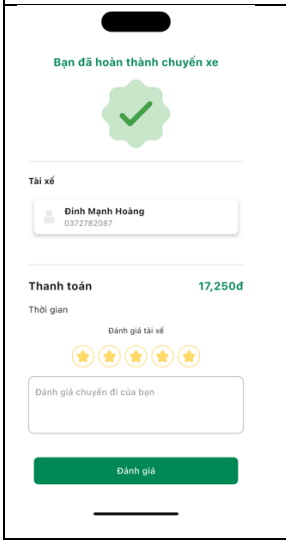
Kết quả kiểm thử cho thấy hệ thống đã đạt được các yêu cầu cơ bản và đảm bảo chất lượng hoạt động trong môi trường thực tế. Mặc dù vẫn còn một số hạn chế nhỏ, chúng không ảnh hưởng nghiêm trọng đến chức năng chính của hệ thống và có thể khắc phục trong các phiên bản cập nhật tiếp theo.

### 4.3 Các màn hình triển khai

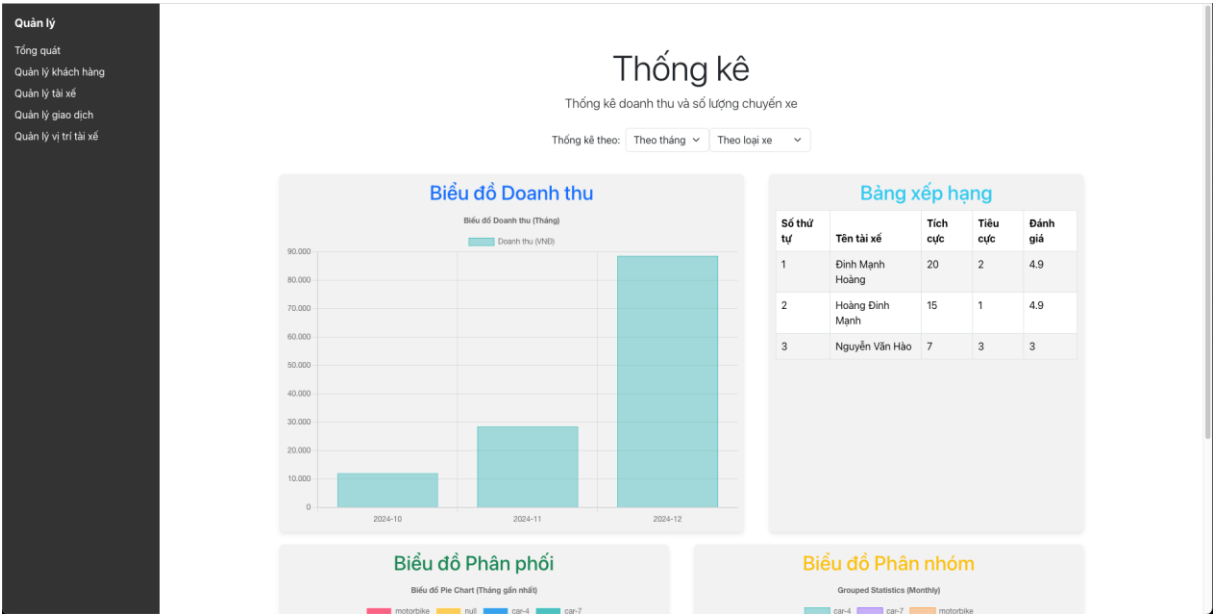
Màn hình	Tên màn hình
	Màn hình chính Khách hàng
	Màn hình đặt xe

	Màn hình xem lịch sử chuyến xe
	Màn hình ví
	Màn hình tài khoản

	Màn hình tạo yêu cầu nạp tiền
	Màn hình chính của Tài xế
	Màn hình xem chi tiết chuyến xe

	Màn hình thực hiện chuyến xe của Tài xế
	Màn hình thực hiện chuyến xe của Khách hàng
	Màn hình hoàn thành chuyến xe của Khách hàng

Bảng 4.2 Màn hình ứng dụng



Hình 4.16 Màn hình thống kê

Quản lý

Tổng quát

Quản lý khách hàng

Quản lý tài xế

Quản lý giao dịch

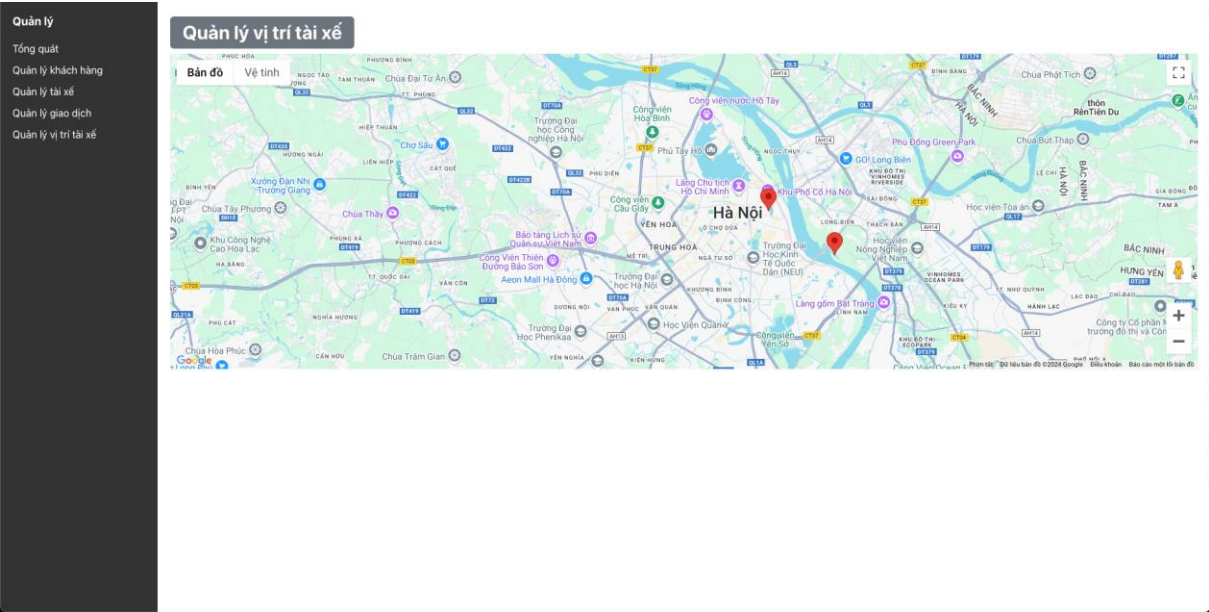
Quản lý vị trí tài xế

Quản lý khách hàng

Thêm người dùng

ID	Tên	Email	Số điện thoại	Hành động
3	Nguyễn Văn A	a@gmail.com	01234567899	<div>SửaXóa</div>
10	aa	a@gmail.com	0123456776	<div>SửaXóa</div>
1	Đinh Mạnh Hoàng	a@gmail.com	0123456798	<div>SửaXóa</div>
2	Hoàng Đình Mạnh	a@gmail.com	0123456789	<div>SửaXóa</div>

Hình 4.17 Màn hình Quản lý khách hàng



Hình 4.18 Màn hình Quản lý vị trí tài xế

Quản lý giao dịch							
STT	Mã giao dịch	Loại giao dịch	Số tiền	Thời gian tạo	Trạng thái	Lời nhắn	Hành động
1	7337442752	Nạp tiền	100.000 đ	19:03:33 9/12/2024	Thành công		
2	1861734976	Rút tiền	14.903 đ	10:54:41 15/12/2024	Thành công	Thanh toán chuyển xe 20241215T01554097824D84	
3	9525280284	Rút tiền	14.903 đ	10:53:59 15/12/2024	Thành công	Thanh toán chuyển xe 20241215T01554097824D84	
4	1859495163	Rút tiền	100.000 đ	19:08:41 9/12/2024	Đang xử lý	Ngân hàng: aa Số tài khoản: 0372782211 Người thụ hưởng: aaa Chi nhánh: aaa	Chấp nhận Hủy bỏ
5	7661128725	Nạp tiền	100.000 đ	19:07:31 9/12/2024	Đang xử lý		Chấp nhận Hủy bỏ
6	7764868473	Nạp tiền	100.000 đ	19:02:00 9/12/2024	Đã hủy		
7	4682380153	Nạp tiền	100.000 đ	19:05:14 9/12/2024	Đang xử lý		Chấp nhận Hủy bỏ
8	5453537689	Nạp tiền	100.000 đ	19:05:06 9/12/2024	Đang xử lý		Chấp nhận Hủy bỏ
9	8403882936	Rút tiền	100.000 đ	15:30:00 1/12/2024	Đang xử lý	Ngân hàng: MBBank Số tài khoản: 0372782087 Người thụ hưởng: DINH MANH HOANG Chi nhánh: Chi nhánh Nam Hà Nội	Chấp nhận Hủy bỏ
10	4616182599	Rút tiền	100.000 đ	15:30:00 1/12/2024	Đang xử lý	Ngân hàng: MBBank Số tài khoản: 0372782087 Người thụ hưởng: DINH MANH HOANG Chi nhánh: branch	Chấp nhận Hủy bỏ
11	4107927402	Rút tiền	100.000 đ	15:30:00 1/12/2024	Đang xử lý	Ngân hàng: MBBank Số tài khoản: 0372782087 Người thụ hưởng: DINH MANH HOANG Chi nhánh: branch	Chấp nhận Hủy bỏ

Hình 4.19 Màn hình Quản lý giao dịch

4.4 Kết luận chương IV

Chương IV đã chỉ ra cách thức cài đặt hệ thống và các màn hình của hệ thống để có được sự hình dung cụ thể nhất về giao diện cũng như cách vận hành của hệ thống. Những màn hình này không chỉ thể hiện khả năng kỹ thuật mà còn chứng minh tính thực tiễn và tiềm năng ứng dụng của hệ thống trong môi trường thực tế. Nhìn chung, việc hoàn thiện chương này đóng vai trò quan trọng trong việc tổng kết toàn bộ quá trình nghiên cứu và phát triển, đồng thời làm nổi bật sự thành công của hệ thống trong việc giải quyết các vấn đề đặt ra từ đầu đề tài.



## KẾT LUẬN

### 1. Kết quả đạt được

Đồ án đã xây dựng thành công hệ thống đặt xe trực tuyến tích hợp thanh toán tự động và theo dõi vị trí thời gian thực thông qua hệ thống GPS. Ngoài ra, hệ thống đã áp dụng được mô hình đánh giá cảm xúc để phân loại cảm xúc đánh giá của khách hàng về chuyến xe để có thể phân tích và đánh giá về những đánh giá đó của khách hàng, nhằm xếp hạng, thống kê một cách tốt hơn

### 2. Hạn chế của hệ thống

Hệ thống vẫn còn một số hạn chế, bao gồm:

- Chưa triển khai đầy đủ các tính năng phân tích dữ liệu nâng cao.
- Thiếu khả năng tự động phân phối chuyến xe.
- Khả năng mở rộng tích hợp thêm nhiều phương thức thanh toán và tối ưu hiệu năng hệ thống vẫn cần được cải thiện.
- Chưa tích hợp được thanh toán tự động.
- Hệ thống chỉ đường vẫn phải phụ thuộc hoàn toàn vào Google.
- Phân tích cảm xúc đánh giá đang sử dụng mô hình có sẵn, chưa thiết kế được mô hình.
- Thiết kế UI/UX chưa được tối ưu.

### 3. Định hướng phát triển hệ thống

Trong tương lai, hệ thống sẽ tiếp tục cần được cải thiện về hiệu năng sử dụng, cải thiện thêm các dịch vụ để làm hài lòng người dùng. Tiếp đến là phát triển thêm hệ thống nạp tiền tự động, tích hợp thêm các ứng dụng thanh toán như VNPAY, Momo,... Hơn hết, cần tự phát triển hệ thống chỉ đường, hệ thống phân phối chuyến xe cũng như hệ thống phân loại cảm xúc đánh giá.

## DANH MỤC TÀI LIỆU THAM KHẢO

### Tài liệu, giáo trình:

[1] 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](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>
- [4] <https://kb.pavietnam.vn/cai-dat-nodejs-tren-windows.html>
- [5] <https://viblo.asia/p/hoc-flutter-phan-1-tao-ung-dung-dau-tien-hello-world-RnB5pMADKPG>
- [7] <https://huggingface.co/5CD-AI/Vietnamese-Sentiment-visober>

