

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**



## **ĐỒ ÁN MÔN HỌC**

**ĐỀ TÀI: XÂY DỰNG PHẦN MỀM CHO THUÊ XE “RENTOR”**

**Giảng viên hướng dẫn:** Trần Hồng Nghi

**Môn học:** Phát triển ứng dụng trên thiết bị di động

**Lớp:** NT117.N11

**Sinh viên thực hiện:**

<b>Họ và Tên</b>	<b>Mã số sinh viên</b>
<b>Trang Kỳ Anh</b>	20521086
<b>Bùi Nguyễn Anh Trung</b>	20520332
<b>Trần Văn Quý</b>	20521818
<b>Đương Đức Anh</b>	20520131

*TP.HCM, ngày 16 tháng 10 năm 2022*

## NHẬN XÉT CỦA GIẢNG VIÊN

This image shows a full page of a document template. It consists of approximately 28 evenly spaced horizontal dotted lines across the entire width of the page, providing a guide for handwriting or typing. The background is plain white, and there are no margins, headers, or footers visible.

## LỜI CẢM ƠN

Đồ án ‘Hệ thống cho thuê xe RENTOR’ là minh chứng cho quá trình cố gắng không ngừng trong việc tìm tòi, học hỏi trong suốt quãng thời gian vừa mà chúng em dành qua cho môn học. Thể hiện khả năng của bản thân trong việc áp dụng kiến thức chuyên môn vào giải quyết một bài toán mới mà chúng em chưa từng trải nghiệm trước đó.

Trong quá trình thực hiện đồ án, em đã nhận được sự hướng dẫn tận tình từ quý thầy cô, sự giúp đỡ và hỗ trợ từ gia đình cũng như bạn bè. Một cách đặc biệt, em xin được gửi lời cảm ơn chân thành đến:

- Các thầy cô trường Đại học Công nghệ Thông Tin nói chung và cô Trần Hồng Nghi nói riêng đã tận tình truyền đạt kiến thức tới em trong suốt thời gian học tập. Đó là tài sản quý báu đã giúp em hoàn thành đồ án và cũng là hành trang cho con đường sự nghiệp của bản thân sau này.
- Các thành viên trong nhóm đã luôn nỗ lực phấn đấu, lắng nghe và luôn sẵn sàng hỗ trợ nhau trong suốt cả quá trình học tập và thực hiện đồ án. Để đạt được kết quả như ngày hôm nay.

Lời cuối cùng, em xin dành lời cảm ơn gửi tới gia đình, anh chị, bạn bè đã luôn bên cạnh để động viên và đóng góp ý kiến trong quá trình hoàn thành đồ án này.

*Thành phố Hồ Chí Minh, ngày 16 tháng 10 năm 2022*

## DANH MỤC BẢNG

Bảng 1. Các giao đoạn thực hiện trong mô hình thác nước .....	11
Bảng 2. Usecase đăng nhập .....	18
Bảng 3. Usecase đăng ký .....	19
Bảng 4. Usecase xác thực số điện thoại .....	20
Bảng 5. Usecase thêm thông tin căn cước công dân .....	21
Bảng 6. Usecase xóa tài khoản .....	22
Bảng 7. Usecase quên mật khẩu .....	22
Bảng 8. Usecase thay đổi mật khẩu.....	24
Bảng 9. Usecase đăng xuất .....	24
Bảng 10. Usecase xem thông tin cá nhân .....	25
Bảng 11. Usecase chỉnh sửa thông tin cá nhân .....	26
Bảng 12. Usecase thêm xe .....	27
Bảng 13. Usecase xem thông tin xe có thể thuê .....	28
Bảng 14. Usecase xem thông tin xe mình cung cấp .....	29
Bảng 15. Usecase xóa xe .....	29
Bảng 16. Usecase cập nhật thông tin xe .....	30
Bảng 17. Usecase gửi yêu cầu đặt xe .....	32
Bảng 18. Usecase phản hồi yêu cầu thuê xe.....	33
Bảng 19. Usecase thanh toán.....	34
Bảng 20. Mô tả cơ sở dữ liệu.....	37
Bảng 21. Thành viên tham gia công việc phân tích thiết kế .....	42
Bảng 22. Thành viên tham gia công việc thiết kế giao diện trên Figma .....	42
Bảng 23. Thành viên tham gia công việc hiện thực hóa giao diện lên Android Studio ....	43
Bảng 24. Thành viên tham gia công việc thực hiện chức năng trên Android Studio.....	44
Bảng 25. Thành viên tham gia công việc báo cáo.....	45

## **DANH MỤC HÌNH ẢNH**

Hình 1. Quy trình thác nước .....	10
Hình 2. Sơ đồ phân rã chức năng .....	16
Hình 3. Sơ đồ usecase tổng quát.....	16
Hình 4. Mô hình ERD.....	35

## MỤC LỤC

<b>I.</b>	<b>GIỚI THIỆU TỔNG QUAN ĐỀ TÀI.....</b>	<b>8</b>
1.	Lí do chọn đề tài.....	8
2.	Mục tiêu đề tài.....	8
3.	Đối tượng và Phạm vi áp dụng.....	8
3.1	Đối tượng sử dụng.....	8
3.2	Phạm vi áp dụng.....	9
3.3	Phương pháp thực tiễn.....	9
3.4	Kết quả dự kiến.....	9
<b>II.</b>	<b>CƠ SỞ LÝ THUYẾT .....</b>	<b>10</b>
1.	<b>Lý thuyết sử dụng: Mô hình Waterfall .....</b>	<b>10</b>
1.1	Các giai đoạn thực hiện trong mô hình thác nước (Waterfall).....	10
1.2	Ưu điểm.....	11
1.3	Nhược điểm.....	12
2.	<b>Công nghệ sử dụng.....</b>	<b>12</b>
2.1	Java Delopment Kit (JDK).....	12
2.2	Android SDK.....	13
2.3	Java Virtual Machine (JVM).....	14
2.4	Firebase.....	15
<b>III.</b>	<b>PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG .....</b>	<b>16</b>
1.	<b>Sơ đồ phân rã chức năng .....</b>	<b>16</b>
2.	<b>Sơ đồ usecase.....</b>	<b>16</b>
2.1	Usecase tổng quát.....	16
2.2	Đặc tả usecase.....	17
3.	<b>Thiết kế cơ sở dữ liệu .....</b>	<b>34</b>
3.1	Sơ đồ cơ sở dữ liệu.....	35
3.2	Mô tả cơ sở dữ liệu.....	36
<b>IV.</b>	<b>THỰC HIỆN ĐỀ TÀI .....</b>	<b>37</b>
1.	<b>Phác thảo giao diện.....</b>	<b>37</b>
2.	<b>Hoàn thành giao diện .....</b>	<b>38</b>

<b>3. Công việc các thành viên.....</b>	<b>42</b>
3.1 Phân tích thiết kế .....	42
3.2 Thiết kế giao diện trên figma.....	42
3.3 Hiện thực hóa giao diện lên Android Studio .....	42
3.4 Thực hiện chức năng trên Android Studio .....	43
3.5 Báo cáo .....	44
<b>V. KẾT LUẬN.....</b>	<b>45</b>
1. Kết quả đạt được .....	45
2. Hướng phát triển .....	46
<b>TÀI LIỆU THAM KHẢO.....</b>	<b>47</b>

## I. GIỚI THIỆU TỔNG QUAN ĐỀ TÀI

### 1. Lí do chọn đề tài

Sau khi đại dịch nổ ra, mạng xã hội phát triển ngày càng mạnh mẽ, mọi hoạt động của con người dần dần được tối ưu hóa qua các ứng dụng trực tuyến. Mặc dù đại dịch đã dần kết thúc nhưng sự phát triển đó vẫn chưa có dấu hiệu dừng lại. Bên cạnh đó, các hoạt động khôi phục kinh tế đang được đẩy mạnh hơn, trong đó có cả hoạt động du lịch.

Để phục vụ những nhu cầu thiết yếu của con người, rất nhiều ứng dụng thương mại điện tử, đặt phòng, đặt xe, đặt đồ ăn,... ra đời. Trong đó, các dịch vụ đặt xe hiện nay sự chủ động nằm ở người tài xế và khách hàng chỉ là người sử dụng một cách thụ động nên nó chỉ đáp ứng được nhu cầu tức thời.

Để đáp ứng nhu cầu dài hạn cũng như tiết kiệm cho khách hàng, nhóm đã chọn đề tài “Xây dựng phần mềm cho thuê xe RENTOR” giúp cho khách hàng chủ động hơn trong việc di chuyển mỗi khi đi du lịch. Khách hàng sẽ không còn gò bó khi phải di chuyển nhiều tuyến đường khác nhau trong một lượt.

### 2. Mục tiêu đề tài

- Áp dụng được kiến thức nền tảng của môn “Phát triển ứng dụng trên thiết bị di động” vào đề tài mà nhóm thực hiện.
- Tạo nên một ứng dụng thực tế có thể phục vụ nhu cầu của cộng đồng thông qua các dịch vụ thiết thực và giao diện thân thiện, dễ thao tác.
- Có hệ thống thông báo, lưu lại lịch sử, đánh giá, xử lý vi phạm và kiểm soát thông tin từ nhà cung cấp và khách hàng.

### 3. Đối tượng và Phạm vi áp dụng

#### 3.1 Đối tượng sử dụng

- Công dân Việt Nam (có căn cước công dân) và khách du lịch tại Việt Nam có cung cấp đầy đủ thông tin cá nhân.



### 3.2 Phạm vi áp dụng

- Ứng dụng được phát triển trên nền tảng ứng dụng di động cho hệ điều hành Android.
- Ứng dụng chỉ được sử dụng ở trong khu vực lãnh thổ Việt Nam.

### 3.3 Phương pháp thực tiễn

- Tìm hiểu và phân tích quy trình hiện tại tại Trường Đại học Công nghệ Thông tin.
- Phân tích các giải pháp của các sản phẩm tương tự trên thị trường.
- Xây dựng ứng dụng theo mô hình Agile để dễ dàng kiểm soát, gia tăng chất lượng của sản phẩm đầu ra và dễ dàng xử lý tình huống thay đổi của yêu cầu.

### 3.4 Kết quả dự kiến

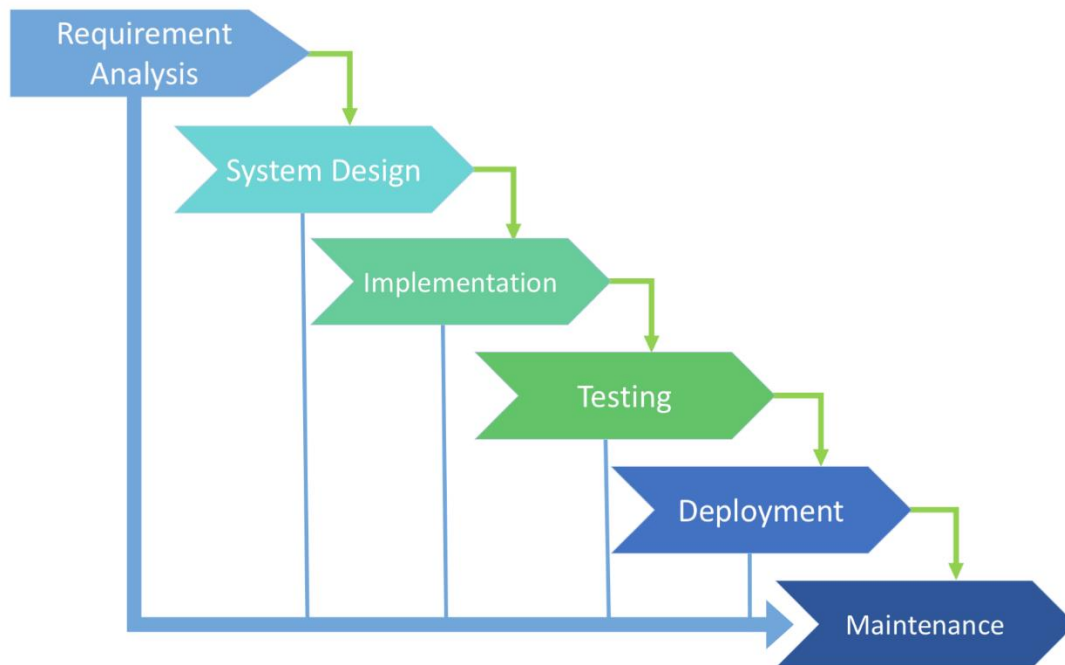
- Hiểu rõ các công nghệ mới:
  - Java Development Kit (JDK): bộ công cụ cung cấp các thư viện, trình biên dịch và công cụ quản lý mã nguồn Java để phát triển ứng dụng Android trong Android Studio.
  - Android SDK: bộ công cụ phát triển ứng dụng Android cung cấp các thư viện, API và các công cụ quản lý để phát triển ứng dụng Android.
  - Java Virtual Machine (JVM): môi trường chạy mã Java, cung cấp môi trường thực thi cho các ứng dụng Java trên nhiều nền tảng khác nhau.
  - Gradle: công cụ quản lý phụ thuộc và biên dịch mã nguồn Java để tạo ra các ứng dụng Android trong Android Studio.
  - Firebase: nền tảng dịch vụ đám mây được phát triển bởi Google, cung cấp cho các nhà phát triển các công cụ và dịch vụ để phát triển, triển khai và quản lý ứng dụng di động và web. Trong đó, nhóm sử dụng một số dịch vụ Authentication (xác thực người dùng), Cloud Firestore (cơ sở dữ liệu NoSQL), Cloud Storage (lưu trữ tệp tin trên đám mây).
- Hoàn thành sản phẩm với các chức năng đã được trình bày.
- Giao diện thân thiện, UX tốt và bám sát với kết quả khảo sát giao diện của người dùng
- Tạo ra sản phẩm mang lại giá trị của mục tiêu hướng tới

## II. CƠ SỞ LÝ THUYẾT

### 1. Lý thuyết sử dụng: Mô hình Waterfall

Mô hình Waterfall (mô hình thác nước) là một trong những mô hình quản lý dự án dễ hiểu nhất hiện nay, mô hình Waterfall là một phương pháp quản lý dự án dựa trên quy trình thiết kế tuần tự và liên tiếp.

Trong mô hình Waterfall, các giai đoạn của dự án được thực hiện lần lượt và nối tiếp nhau. Giai đoạn mới chỉ được bắt đầu khi giai đoạn trước nó đã được hoàn thành.



Hình 1. Quy trình thác nước

#### 1.1 Các giai đoạn thực hiện trong mô hình thác nước (Waterfall)

Các giai đoạn	Đặc điểm
Phân tích yêu cầu	Tất cả các yêu cầu mà hệ thống có thể được phát triển đều được nắm bắt trong giai đoạn này và được ghi lại trong tài liệu đặc tả yêu cầu.
Thiết kế hệ thống	Dựa vào các yêu cầu từ giai đoạn đầu, giai đoạn này sẽ nghiên cứu các thông số kỹ thuật liên quan và chuẩn bị cho việc thiết kế hệ thống. Việc này giúp xác định các

	yêu cầu hệ thống và phần cứng thích hợp cho kiến trúc hệ thống tổng thể.
Thực hiện	Với đầu vào từ phần thiết kế hệ thống, hệ thống sẽ được phát triển từ các chương trình nhỏ, tách biệt nhau và sẽ được tích hợp trong giai đoạn tiếp theo. Mỗi chương trình được phát triển và kiểm tra chức năng của nó, được gọi là Units Testing.
Tích hợp và Thử nghiệm	Tất cả các chương trình ở bước trên được tích hợp vào một hệ thống sau khi thử nghiệm các chương trình đơn lẻ đó. Sau khi tích hợp, toàn bộ hệ thống được kiểm tra xem có bất kỳ lỗi phát sinh nào không.
Triển khai	Sau khi hoàn thành kiểm tra chức năng và phi chức năng; sản phẩm được triển khai trong môi trường khách hàng hoặc tung ra thị trường.
Bảo trì	Có một số vấn đề xảy ra trong môi trường máy khách. Để khắc phục những sự cố đó, các bản vá được phát hành. Ngoài ra để nâng cao sản phẩm, một số phiên bản tốt hơn được phát hành. Bảo trì được thực hiện để cung cấp những thay đổi này trong môi trường khách hàng.

Bảng 1. Các giai đoạn thực hiện trong mô hình thác nước

## 1.2 Ưu điểm

- Đơn giản và dễ hiểu và dễ sử dụng, dễ sắp xếp công việc
- Dễ quản lý do tính chặt chẽ của mô hình(Mỗi giai đoạn có các sản phẩm cụ thể và một quy trình xem xét riêng)
- Các giai đoạn được xử lý và hoàn thành tại một thời điểm
- Hoạt động tốt cho các dự án nhỏ, nơi các yêu cầu được hiểu rất rõ.
- Các giai đoạn được xác định rõ ràng.
- Các mốc quan trọng được hiểu rõ.

### 1.3 Nhược điểm

- Không có phần mềm làm việc được sản xuất cho đến cuối vòng đời.
- Số lượng rủi ro và sự không chắc chắn cao.
- Không phải là một mô hình tốt cho các dự án phức tạp và hướng đối tượng.
- Mô hình kém cho các dự án dài và đang diễn ra.
- Không phù hợp với các dự án mà các yêu cầu có nguy cơ thay đổi từ trung bình đến cao. Vì vậy, rủi ro và sự không chắc chắn là cao với mô hình quy trình này.
- Rất khó để đo lường sự tiến bộ trong các giai đoạn.
- Không thể đáp ứng yêu cầu thay đổi.
- Việc điều chỉnh phạm vi trong vòng đời có thể kết thúc một dự án.

## 2. Công nghệ sử dụng

### 2.1 Java Delopment Kit (JDK)



Java Development Kit (JDK) là bộ công cụ phát triển phần mềm được sử dụng để phát triển các ứng dụng Java. Nó cung cấp các công cụ cần thiết để phát triển, kiểm tra và gỡ lỗi các chương trình Java.

JDK bao gồm một trình biên dịch chuyển đổi mã nguồn Java thành mã bytecode, có thể được thực thi bởi Java Virtual Machine (JVM). Nó cũng bao

## 2.2 Android SDK



Android Software Development Kit (SDK) là một bộ công cụ phần mềm miễn phí được cung cấp bởi Google để phát triển ứng dụng trên nền tảng Android. Bộ công cụ này bao gồm các công cụ phát triển, thư viện và tài liệu hướng dẫn cho việc phát triển ứng dụng Android.

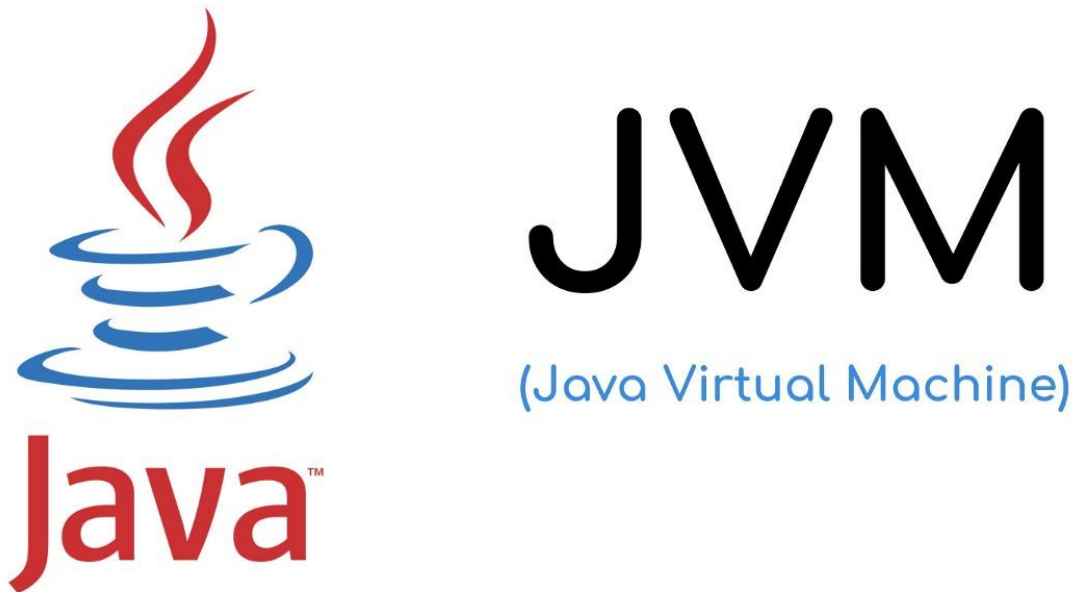
Android SDK cung cấp cho nhà phát triển một loạt các công cụ để thiết kế, phát triển và kiểm tra ứng dụng Android. Bao gồm các thành phần chính như:

- **Android Studio:** Môi trường phát triển tích hợp (IDE) cho phát triển ứng dụng Android, cung cấp một giao diện người dùng và các công cụ hỗ trợ để thiết kế giao diện người dùng, quản lý mã và triển khai ứng dụng.
- **Android Debug Bridge (ADB):** Công cụ giao tiếp giữa máy tính và thiết bị Android, cho phép nhà phát triển gửi lệnh và truyền tập vào thiết bị Android để thực hiện kiểm thử và gỡ lỗi.
- **Android Emulator:** Công cụ mô phỏng thiết bị Android trên máy tính, cho phép nhà phát triển thử nghiệm ứng dụng của mình trên nhiều thiết bị khác nhau.
- **Android SDK Platform:** Bao gồm các thư viện và API cần thiết để phát triển ứng dụng Android.

Ngoài ra, Android SDK cũng cung cấp cho nhà phát triển các công cụ hỗ trợ như trình giả lập màn hình, hỗ trợ thiết bị đa ngôn ngữ và hỗ trợ khả năng tương tác giữa ứng dụng và các dịch vụ của Google.

Android SDK là công cụ quan trọng cho các nhà phát triển ứng dụng Android, giúp họ có thể phát triển, thử nghiệm và triển khai các ứng dụng trên nền tảng Android.

### 2.3 Java Virtual Machine (JVM)



Java Virtual Machine (JVM) là một phần mềm chạy trên máy tính, được sử dụng để thực thi các chương trình viết bằng ngôn ngữ lập trình Java. JVM là một phần quan trọng của nền tảng Java và cho phép các chương trình Java được chạy trên nhiều hệ điều hành khác nhau, bao gồm Windows, Mac OS và Linux.

JVM là một môi trường chạy ảo, có khả năng đọc và thực thi mã Java được biên dịch thành bytecode. Bytecode là một mã trung gian giữa mã nguồn Java và mã máy, được tạo ra bởi trình biên dịch Java. Khi một chương trình Java được biên dịch thành bytecode, nó có thể chạy trên bất kỳ JVM nào, cho phép các chương trình Java trở thành phần mềm di động và phát triển đa nền tảng.

JVM cũng có nhiều tính năng quan trọng khác, bao gồm:

- Quản lý bộ nhớ: JVM có khả năng quản lý bộ nhớ tự động, giúp tối ưu hóa việc sử dụng bộ nhớ và tránh việc tràn bộ nhớ.
- Multithreading: JVM có khả năng thực thi các chương trình Java đồng thời trên nhiều luồng, giúp tăng tốc độ thực thi chương trình.
- Quản lý exception: JVM quản lý các exception (ngoại lệ) trong chương trình Java, giúp phát hiện và xử lý lỗi một cách hiệu quả.
- Dynamic support: JVM có khả năng tải và thực thi mã nguồn trong quá trình chạy chương trình, giúp các ứng dụng Java trở nên linh hoạt hơn.

Tổng quan, JVM là một thành phần quan trọng của nền tảng Java, cho phép các chương trình Java chạy trên nhiều hệ điều hành khác nhau, giúp tối ưu hóa việc sử dụng bộ nhớ và tăng tốc độ thực thi chương trình.

## 2.4 Firebase



# firebase

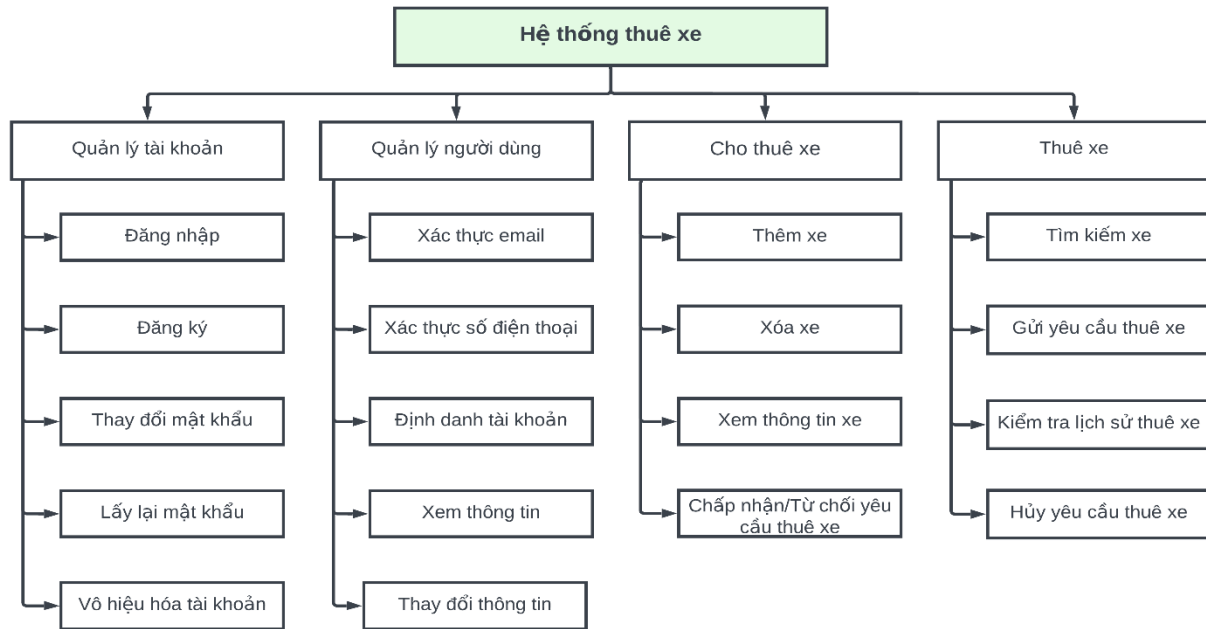
Firebase là một nền tảng phát triển ứng dụng web và di động của Google, cung cấp một loạt các công cụ và dịch vụ để giúp nhà phát triển xây dựng ứng dụng nhanh chóng và hiệu quả. Firebase cung cấp các tính năng như lưu trữ dữ liệu, xác thực người dùng, phân tích và cảnh báo, đám mây và thông báo đẩy, và nhiều tính năng khác.

Firebase cũng cung cấp các công cụ để phát triển ứng dụng thời gian thực, cho phép các ứng dụng trò chuyện, phát sóng video trực tiếp và trao đổi dữ liệu đồng bộ. Firebase cũng hỗ trợ tích hợp với các công nghệ khác như React, Vue, Angular và các thư viện khác.

Một trong những điểm mạnh của Firebase là nó cung cấp một giao diện điều khiển đồ họa dễ sử dụng cho phép quản trị viên quản lý các ứng dụng và dịch vụ của mình. Firebase cũng cung cấp các tài liệu và hướng dẫn chi tiết để giúp các nhà phát triển bắt đầu sử dụng nền tảng và phát triển các ứng dụng chất lượng cao.

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

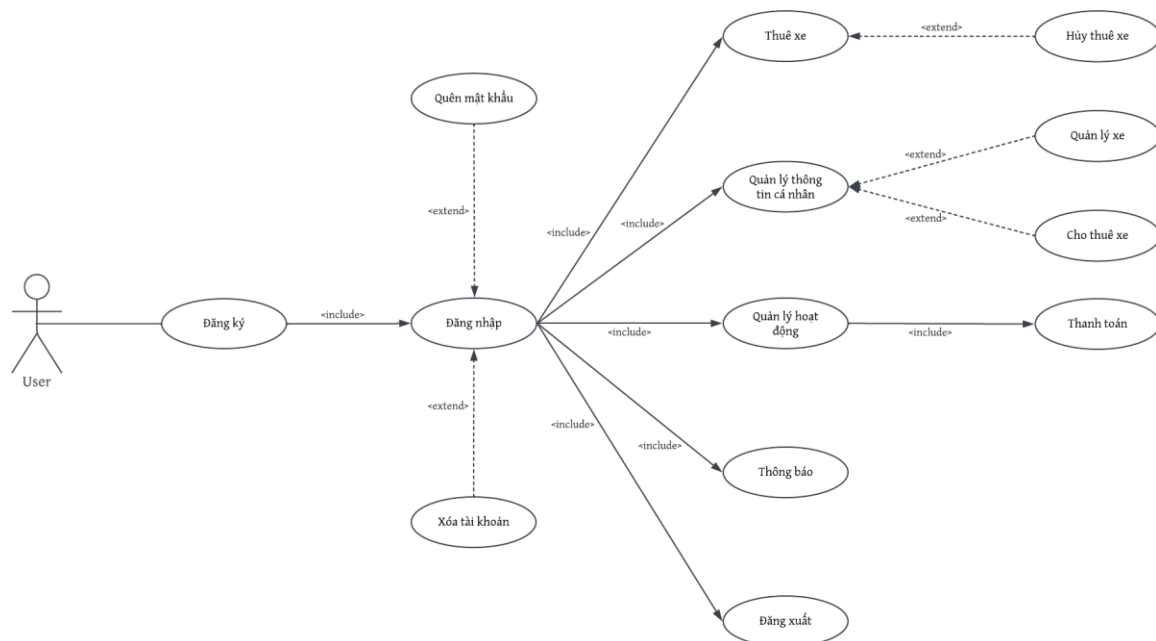
#### 1. Sơ đồ phân rã chức năng



Hình 2. Sơ đồ phân rã chức năng

#### 2. Sơ đồ usecase

##### 2.1 Usecase tổng quát



Hình 3. Sơ đồ usecase tổng quát



## 2.2 Đặc tả usecase

### a. Usecase đăng nhập

Usecase	Nội dung
Tên Usecase	Đăng nhập
Mô tả	Usecase cho phép người dùng đăng nhập vào hệ thống để thực hiện chức năng
Actor	Mọi người dùng
Điều kiện kích hoạt	Khi người dùng chức năng đăng nhập từ trang đăng nhập của hệ thống
Tiền điều kiện	Phải có tài khoản đăng nhập của hệ thống
Hậu điều kiện	Người dùng đăng nhập thành công
Luồng sự kiện chính	<ol style="list-style-type: none"> <li>1. Hệ thống hiển thị màn hình đăng nhập</li> <li>2. Người dùng nhập email và mật khẩu</li> <li>3. Hệ thống kiểm tra email và mật khẩu</li> <li>4. Nếu hợp lệ hệ thống thông báo đăng nhập thành công</li> <li>5. Kết thúc Usecase</li> </ol>
Luồng sự kiện phụ	<p><b><u>A1. Mật khẩu hoặc email sai:</u></b></p> <ol style="list-style-type: none"> <li>1. Hệ thống hiển thị lại màn hình đăng nhập để người dùng nhập lại thông tin kèm theo thông báo đăng nhập thất bại.</li> <li>2. Quay lại bước 2 của Luồng sự kiện chính.</li> </ol> <p><b><u>A2. Tài khoản đăng nhập lần đầu tiên:</u></b></p> <ol style="list-style-type: none"> <li>1. Hệ thống hiển thị màn hình gửi mã OTP về số điện thoại đã đăng ký.</li> </ol>

	<ol style="list-style-type: none"> <li>Hệ thống hiển thị màn hình yêu cầu người dùng nhập thông tin thẻ căn cước công dân mặt trước và sau.</li> <li>Điền thông tin cá nhân để hoàn thiện tài khoản</li> <li>Kết thúc Usecase.</li> </ol>
--	---

Bảng 2. Usecase đăng nhập

*b. Usecase đăng ký*

Usecase	Nội dung
Tên Usecase	Đăng ký
Mô tả	Usecase cho phép người dùng đăng ký tài khoản hệ thống để thực hiện chức năng
Actor	Mọi người dùng
Điều kiện kích hoạt	Khi người dùng nhấn vào nút Đăng ký nằm dưới cùng màn hình Đăng nhập
Tiền điều kiện	Phải có tài khoản email, số điện thoại
Hậu điều kiện	Người dùng đăng ký thành công
Luồng sự kiện chính	<ol style="list-style-type: none"> <li>Hệ thống hiển thị màn hình đăng nhập</li> <li>Người dùng chọn Đăng ký</li> <li>Nhập thông tin tài khoản (gồm email, số điện thoại và mật khẩu)</li> <li>Firebase gửi link xác thực về email mà bạn nhập</li> <li>Sau khi xác thực thông tin của bạn sẽ được lưu vào Authentication và Firestore</li> <li>Quay lại màn hình đăng nhập</li> <li>Kết thúc Usecase</li> </ol>
Luồng sự kiện phụ	<b><u>A1. Email đã tồn tại :</u></b>

	<ol style="list-style-type: none"> <li>Hệ thống hiển thị lại màn hình nhập thông tin kèm theo thông báo không thể tạo user.</li> <li>Quay lại bước 3 của Luồng sự kiện chính.</li> </ol> <p><b><u>A2. Email không tồn tại</u></b></p> <ol style="list-style-type: none"> <li>Firebase sẽ không gửi được email về để xác thực</li> <li>Quay lại màn hình đăng nhập</li> <li>Quay lại bước 2 của Luồng sự kiện chính.</li> </ol> <p><b><u>A3. Xác nhận mật khẩu sai</u></b></p> <ol style="list-style-type: none"> <li>Hệ thống hiển thị lại màn hình nhập thông tin kèm theo thông báo “Mật khẩu không khớp, mời nhập lại”</li> <li>Quay lại bước 3 của Luồng sự kiện chính.</li> </ol>
--	--

Bảng 3. Usecase đăng ký

## c. Usecase xác thực số điện thoại

Usecase	Nội dung
Tên Usecase	Xác thực số điện thoại
Mô tả	Usecase bắt buộc người dùng xác thực số điện thoại
Actor	Mọi người dùng
Điều kiện kích hoạt	Người dùng đăng nhập vào ứng dụng lần đầu tiên
Tiền điều kiện	Người dùng phải đăng nhập thành công vào hệ thống
Hậu điều kiện	Xác thực thành công số điện thoại
Luồng sự kiện chính	<ol style="list-style-type: none"> <li>Người dùng đăng nhập lần đầu tiên trên ứng dụng</li> <li>Firebase gửi OTP về số điện thoại mà người dùng đã nhập ở màn hình đăng ký</li> </ol>

	<ol style="list-style-type: none"> <li>3. Người dùng nhập OTP đã gửi về số điện thoại</li> <li>4. Thông tin xác thực sẽ được lưu vào Authentication</li> <li>5. Ứng dụng chuyển sang giao diện cập nhật căn cước công dân</li> <li>6. Kết thúc Usecase</li> </ol>
Luồng sự kiện phụ	<p><b><u>A1. Số điện thoại của người dùng không nhận được OTP</u></b></p> <ol style="list-style-type: none"> <li>1. Sau 1 phút nếu người dùng vẫn chưa nhận được OTP thì bấm vào nút gửi lại phía dưới phần nhập OTP</li> <li>2. Quay lại bước 2 của luồng sự kiện chính</li> </ol> <p><b><u>A2. Người dùng nhập sai OTP</u></b></p> <ol style="list-style-type: none"> <li>1. Ứng dụng sẽ hiển thị thông báo “OTP không đúng”</li> <li>2. Người dùng kiểm tra lại OTP được gửi về số điện thoại của mình</li> <li>3. Quay lại bước 3 ở luồng sự kiện chính</li> </ol>

Bảng 4. Usecase xác thực số điện thoại

*d. Usecase thêm thông tin căn cước công dân*

Usecase	Nội dung
Tên Usecase	Thêm thông tin căn cước công dân
Mô tả	Usecase bắt buộc người dùng thêm căn cước công dân
Actor	Mọi người dùng
Điều kiện kích hoạt	Người dùng đăng nhập vào ứng dụng lần đầu tiên
Tiền điều kiện	Người dùng đã hoàn thành xác thực số điện thoại

Hậu điều kiện	Thêm thông tin căn cước công dân thành công
Luồng sự kiện chính	<ol style="list-style-type: none"> <li>1. Người dùng đã xác thực thành công số điện thoại</li> <li>2. Ứng dụng sẽ chuyển sang màn hình để người dùng tải ảnh căn cước công dân của mình lên</li> <li>3. Người dùng bấm vào tải ảnh lên để lưu hình ảnh vào Storage và Firestore</li> <li>4. Ứng dụng chuyển sang giao diện cập nhật thông tin cá nhân</li> <li>5. Kết thúc Usecase</li> </ol>
Luồng sự kiện phụ	Không

Bảng 5. Usecase thêm thông tin căn cước công dân

## e. Usecase xóa tài khoản

Usecase	Nội dung
Tên Usecase	Xóa tài khoản
Mô tả	Usecase cho phép người dùng gửi yêu cầu xóa tài khoản của mình
Actor	Mọi người dùng
Điều kiện kích hoạt	Người dùng chọn phần xóa tài khoản trong phần setting
Tiền điều kiện	Người dùng phải đăng nhập thành công vào hệ thống
Hậu điều kiện	Không
Luồng sự kiện chính	<ol style="list-style-type: none"> <li>7. Người dùng truy cập vào màn hình setting</li> <li>8. Người dùng chọn mục xóa tài khoản</li> </ol>

	<p>9. Hệ thống hiển thị” xóa thành công” và xóa tài khoản ra khỏi Authentication</p> <p>10. Kết thúc Usecase</p>
--	--

Bảng 6. Usecase xóa tài khoản

*f. Usecase quên mật khẩu*

Usecase	Nội dung
Tên Usecase	Quên mật khẩu
Mô tả	Usecase cho phép người dùng chọn quên mật khẩu được thiết lập trên màn hình đăng nhập
Actor	Mọi người dùng
Điều kiện kích hoạt	Khi người dùng chọn chức năng quên mật khẩu
Tiền điều kiện	Phải có tài khoản đăng nhập của hệ thống
Hậu điều kiện	Người dùng tạo lại mật khẩu thành công
Luồng sự kiện chính	<ol style="list-style-type: none"> <li>1. Người dùng bấm chọn quên mật khẩu</li> <li>2. Firebase gửi email về cho người dùng</li> <li>3. Người dùng truy cập vô mail sẽ được dẫn sang một trang web khác để nhập mật khẩu mới</li> <li>4. Người dùng quay lại màn hình chính và đăng nhập bằng mật khẩu mới</li> <li>5. Kết thúc Usecase</li> </ol>
Luồng sự kiện phụ	Không

Bảng 7. Usecase quên mật khẩu

*g. Usecase thay đổi mật khẩu*

Usecase	Nội dung
Tên Usecase	Thay đổi mật khẩu
Mô tả	Usecase cho phép người dùng thay đổi mật khẩu
Actor	Mọi người dùng
Điều kiện kích hoạt	Khi người dùng chọn chức năng thay đổi mật khẩu
Tiền điều kiện	Phải có tài khoản đăng nhập của hệ thống và phải nhớ mật khẩu cũ
Hậu điều kiện	Phải đăng nhập thành công
Luồng sự kiện chính	<ol style="list-style-type: none"> <li>1. Người dùng truy cập vào màn hình setting</li> <li>2. Người dùng bấm chọn mục thay đổi mật khẩu</li> <li>3. Người dùng nhập mật khẩu cũ và mật khẩu mới. Sau đó bấm nút tiếp tục</li> <li>4. Hệ thống hiển thị “Thay đổi mật khẩu thành công” và mật khẩu mới sẽ được cập nhật vào Authentication</li> <li>5. Hệ thống trở về giao diện setting</li> <li>6. Kết thúc Usecase</li> </ol>
Luồng sự kiện phụ	<p><b><u>A1. Người dùng nhập sai mật khẩu hiện tại:</u></b></p> <ol style="list-style-type: none"> <li>1. Người dùng nhập sai mật khẩu hiện tại</li> <li>2. Hệ thống hiển thị thông báo “Mật khẩu hiện tại không chính xác”</li> <li>3. Quay lại bước 3 của luồng sự kiện chính</li> </ol>

	<p><b><u>A2. Người dùng nhập không khớp mật khẩu mới:</u></b></p> <ol style="list-style-type: none"> <li>1. Người dùng nhập không khớp mật khẩu mới</li> <li>2. Hệ thống hiển thị thông báo “Mật khẩu không khớp, mời nhập lại”</li> <li>3. Quay lại bước 3 của luồng sự kiện chính</li> </ol>
--	--

Bảng 8. Usecase thay đổi mật khẩu

#### *h. Usecase đăng xuất*

Usecase	Nội dung
Tên Usecase	Đăng xuất
Mô tả	Usecase cho phép người dùng đăng xuất
Actor	Mọi người dùng
Điều kiện kích hoạt	Khi người dùng chọn chức năng Đăng xuất trên màn hình setting
Tiền điều kiện	Người dùng đã đăng nhập thành công vào hệ thống
Hậu điều kiện	Người dùng đăng xuất thành công
Luồng sự kiện chính	<ol style="list-style-type: none"> <li>1. Người dùng truy cập vào màn hình setting</li> <li>2. Người dùng chọn nút đăng xuất</li> <li>3. Hệ thống trở về giao diện đăng nhập để người dùng có thể đăng nhập tài khoản khác</li> <li>4. Kết thúc Usecase</li> </ol>
Luồng sự kiện phụ	Không

Bảng 9. Usecase đăng xuất



*i. Usecase xem thông tin cá nhân*

Usecase	Nội dung
Tên Usecase	Xem thông tin cá nhân
Mô tả	Usecase cho phép người dùng xem thông tin cá nhân
Actor	Mọi người dùng
Điều kiện kích hoạt	Khi người dùng chọn mục thông tin tài khoản ở phần setting
Tiền điều kiện	Người dùng phải đăng nhập thành công vào hệ thống
Hậu điều kiện	Không
Luồng sự kiện chính	<ol style="list-style-type: none"> <li>1. Người dùng truy cập vào màn hình setting</li> <li>2. Người dùng chọn mục thông tin tài khoản</li> <li>3. Hệ thống hiển thị màn hình thông tin cá nhân</li> <li>4. Người dùng xem các thông tin cá nhân</li> <li>5. Kết thúc Usecase</li> </ol>
Luồng sự kiện phụ	Không

Bảng 10. Usecase xem thông tin cá nhân

*j. Usecase chỉnh sửa thông tin cá nhân*

Usecase	Nội dung
Tên Usecase	Chỉnh sửa thông tin cá nhân
Mô tả	Usecase cho phép người dùng chỉnh sửa thông tin cá nhân
Actor	Mọi người dùng
Điều kiện kích hoạt	Khi người dùng chọn nút chỉnh sửa thông tin ,tại màn hình thông tin cá nhân

Tiền điều kiện	Người dùng đã đăng nhập thành công vào hệ thống
Hậu điều kiện	Cập nhật thông tin cá nhân thành công
Luồng sự kiện chính	<ol style="list-style-type: none"> <li>1. Người dùng chọn nút chỉnh sửa thông tin cá nhân dưới màn hình xem thông tin cá nhân</li> <li>2. Hệ thống hiển thị màn hình cập nhật thông tin bao gồm các thông tin trước đó</li> <li>3. Người dùng chỉ có thể chỉnh sửa: Tên, địa chỉ, thành phố, ngày sinh, avatar</li> <li>4. Người dùng bấm nút cập nhật thông tin</li> <li>5. Hệ thống chuyển về màn hình chính và cập nhật dữ liệu vào Firestore</li> <li>6. Kết thúc Usecase</li> </ol>
Luồng sự kiện phụ	Không

Bảng 11. Usecase chỉnh sửa thông tin cá nhân

*k. Usecase thêm xe*

Usecase	Nội dung
Tên Usecase	Thêm xe
Mô tả	Usecase cho phép nhà cung cấp thêm xe
Actor	Mọi người dùng
Điều kiện kích hoạt	Nhà cung cấp chọn thêm xe trong màn hình chính
Tiền điều kiện	Người dùng phải chuyển qua giao diện nhà cung cấp
Hậu điều kiện	Người dùng phải đăng nhập thành công vào hệ thống

Luồng sự kiện chính	<ol style="list-style-type: none"> <li>1. Nhà cung cấp bấm nút thêm xe tại giao diện chính</li> <li>2. Hệ thống hiển thị màn hình thêm xe mới</li> <li>3. Nhà cung cấp điền các thông tin cần thiết</li> <li>4. Khi điền đầy đủ thông tin, nhà cung cấp bấm nút thêm.</li> <li>5. Hệ thống hiển thị thông báo thêm thành công và lưu thông tin vào Firestore</li> <li>6. Quay lại giao diện màn hình chính của nhà cung cấp</li> <li>7. Kết thúc Usecase</li> </ol>
Luồng sự kiện phụ	<p><b><u>A1. Nhà cung cấp chưa điền đầy đủ thông tin thêm xe:</u></b></p> <ol style="list-style-type: none"> <li>1. Nhà cung cấp chưa điền đầy đủ thông tin. Bấm nút thêm</li> <li>2. Hệ thống hiển thị thông báo “Vui lòng nhập đầy đủ thông tin”</li> <li>3. Quay lại bước 3 của luồng sự kiện chính</li> </ol>

Bảng 12. Usecase thêm xe

### 1. Usecase xem thông tin xe có thể thuê

Usecase	Nội dung
Tên Usecase	Xem thông tin xe có thể thuê
Mô tả	Usecase cho phép người dùng xem thông tin chi tiết của các xe do nhà cung cấp đăng tải
Actor	Mọi người dùng
Điều kiện kích hoạt	Khi người dùng chọn một xe bất kì trên màn hình chính

Tiền điều kiện	<ol style="list-style-type: none"> <li>1. Phải đăng nhập thành công vào hệ thống</li> <li>2. Phải có ít nhất một nhà cung cấp đăng tải một xe lên ứng dụng</li> </ol>
Hậu điều kiện	Người dùng xem được bài post của nhà cung cấp
Luồng sự kiện chính	<ol style="list-style-type: none"> <li>1. Hệ thống hiển thị màn hình chính</li> <li>2. Người dùng chọn bất kỳ xe trên màn hình chính</li> <li>3. Hệ thống hiển thị màn hình thông tin chi tiết của chiếc xe người dùng chọn</li> <li>4. Kết thúc Usecase</li> </ol>
Luồng sự kiện phụ	Không

Bảng 13. Usecase xem thông tin xe có thể thuê

*m. Usecase xem thông tin xe mình cung cấp*

Usecase	Nội dung
Tên Usecase	Xem thông tin xe mình cung cấp
Mô tả	Usecase cho phép nhà cung cấp xem xe đã đăng tải
Actor	Nhà cung cấp đã đăng tải xe
Điều kiện kích hoạt	Nhà cung cấp chọn một xe trong giao diện chính
Tiền điều kiện	<ol style="list-style-type: none"> <li>3. Người dùng truy cập vào màn hình setting</li> <li>4. Nhà cung cấp phải đăng tải ít nhất một xe</li> </ol>
Hậu điều kiện	Không
Luồng sự kiện chính	<ol style="list-style-type: none"> <li>1. Nhà cung cấp chọn một xe muốn xem thông tin</li> <li>2. Hệ thống hiển thị giao diện thông tin chi tiết của xe nhà cung cấp đã chọn</li> </ol>

	3. Kết thúc Usecase
Luồng sự kiện phụ	Không có

Bảng 14. Usecase xem thông tin xe mình cung cấp

*n. Usecase xóa xe*

Usecase	Nội dung
Tên Usecase	Xóa xe
Mô tả	Usecase cho phép nhà cung cấp xóa xe đã đăng tải
Actor	Nhà cung cấp đã đăng tải xe
Điều kiện kích hoạt	Nhà cung cấp chọn một xe trong giao diện chính
Tiền điều kiện	<ol style="list-style-type: none"> <li>1. Người dùng truy cập vào màn hình setting</li> <li>2. Nhà cung cấp phải đăng tải ít nhất một xe</li> </ol>
Hậu điều kiện	Không
Luồng sự kiện chính	<ol style="list-style-type: none"> <li>1. Hệ thống hiển thị giao diện thông tin xe</li> <li>2. Nhà cung cấp bấm chọn nút xóa</li> <li>3. Hệ thống hiển thị “xóa xe thành công” và xóa dữ liệu khỏi Firestore</li> </ol>
Luồng sự kiện phụ	Không có

Bảng 15. Usecase xóa xe

*o. Usecase cập nhật thông tin xe*

Usecase	Nội dung
Tên Usecase	Cập nhật thông tin xe
Mô tả	Usecase cho phép nhà cung cấp cập nhật xe đã đăng tải
Actor	Nhà cung cấp đã đăng tải xe
Điều kiện kích hoạt	Nhà cung cấp chọn một xe trong giao diện chính
Tiền điều kiện	<ol style="list-style-type: none"> <li>1. Người dùng truy cập vào màn hình setting</li> <li>2. Nhà cung cấp phải đăng tải ít nhất một xe</li> </ol>
Hậu điều kiện	Không
Luồng sự kiện chính	<ol style="list-style-type: none"> <li>1. Hệ thống hiển thị giao diện thông tin xe</li> <li>2. Nhà cung cấp sửa thông tin xe, bấm nút cập nhật</li> <li>3. Hệ thống hiển thị “Cập nhật thông tin thành công”, hệ thống cập nhật thông tin vào Firestore</li> <li>4. Quay lại màn hình chính của nhà cung cấp</li> <li>5. Kết thúc Usecase</li> </ol>
Luồng sự kiện phụ	Không có

Bảng 16. Usecase cập nhật thông tin xe

*p. Usecase gửi yêu cầu đặt xe*

Usecase	Nội dung
Tên Usecase	Gửi yêu cầu đặt xe
Mô tả	Usecase cho phép người dùng gửi yêu cầu đặt xe
Actor	Mọi người dùng

Điều kiện kích hoạt	Người dùng chọn một xe mong muốn trên màn hình chính
Tiền điều kiện	<ol style="list-style-type: none"> <li>1. Phải đăng nhập thành công vào hệ thống</li> <li>2. Phải có ít nhất một nhà cung cấp đăng tải một xe lên ứng dụng</li> </ol>
Hậu điều kiện	Người dùng xem được các bài post của nhà cung cấp
Luồng sự kiện chính	<ol style="list-style-type: none"> <li>1. Hệ thống hiển thị màn hình chính</li> <li>2. Người dùng chọn một xe trong màn hình chính</li> <li>3. Hệ thống hiển thị màn hình thông tin chi tiết của xe người dùng đã chọn</li> <li>4. Người dùng xem thông tin nhà cung cấp và thông tin xe. Nếu đồng ý thì bấm nút thuê xe</li> <li>5. Hệ thống hiển thị màn hình điền thông tin</li> <li>6. Người dùng chọn ngày, giờ nhận và trả xe</li> <li>7. Người dùng nhấn nút gửi yêu cầu thuê xe</li> <li>8. Hệ thống hiển thị màn hình gửi yêu cầu thành công và lưu thông tin vào Firestore</li> <li>9. Người dùng nhấn nút về trang chủ và chờ nhà cung cấp xác nhận yêu cầu.</li> <li>10. Kết thúc Usecase</li> </ol>
Luồng sự kiện phụ	<p><b><u>A1. Người dùng không đồng ý với thông tin của nhà cung cấp hoặc xe:</u></b></p> <ol style="list-style-type: none"> <li>1. Hệ thống hiển thị màn hình thông tin chi tiết của xe người dùng đã chọn</li> </ol>

	<ol style="list-style-type: none"> <li>Người dùng không đồng ý với thông tin nhà cung cấp hoặc xe. Người dùng bấm nút hình mũi tên back ở góc trái màn hình.</li> <li>Hệ thống hiển thị màn hình chính</li> <li>Quay lại bước 2 của luồng sự kiện chính</li> </ol>
--	--

Bảng 17. Usecase gửi yêu cầu đặt xe

*q. Usecase phản hồi yêu cầu thuê xe*

Usecase	Nội dung
Tên Usecase	Phản hồi yêu cầu thuê xe
Mô tả	Usecase cho phép nhà cung cấp phản hồi lại yêu cầu thuê xe của khách hàng
Actor	Những nhà cung cấp đã đăng tải xe và được khách hàng gửi yêu cầu thuê xe
Điều kiện kích hoạt	Nhà cung cấp chọn vào màn hình Activity
Tiền điều kiện	Phải có ít nhất một khách hàng gửi yêu cầu
Hậu điều kiện	Phải chuyển sang giao diện nhà cung cấp
Luồng sự kiện chính	<ol style="list-style-type: none"> <li>Nhà cung cấp bấm chọn vào màn hình Activity bên dưới màn hình giao diện nhà cung cấp</li> <li>Hệ thống hiển thị giao diện các yêu cầu khách hàng gửi đến cho nhà cung cấp</li> <li>Nhà cung cấp bấm chọn một yêu cầu trong danh sách</li> <li>Hệ thống hiển thị giao diện thông tin chi tiết yêu cầu gồm: Thông tin khách hàng, thông tin xe và trạng thái.</li> </ol>



	<p>5. Nhà cung cấp phản hồi yêu cầu của khách hàng</p> <p>6. Hệ thống hiển thị lại giao diện thông tin yêu cầu với trạng thái mới và cập nhật trạng thái mới vào Firestore</p> <p>7. Kết thúc Usecase</p>
Luồng sự kiện phụ	<p><b><u>A1. Nhà cung cấp xác nhận yêu cầu:</u></b></p> <ol style="list-style-type: none"> <li>1. Nhà cung cấp bấm nút xác nhận tại giao diện thông tin yêu cầu</li> <li>2. Hệ thống hiển thị thông báo “Đã xác nhận”, cập nhật lại thông tin vào Firestore</li> <li>3. Quay lại bước 6 luồng sự kiện chính</li> </ol> <p><b><u>A2. Nhà cung cấp hủy yêu cầu:</u></b></p> <ol style="list-style-type: none"> <li>1. Nhà cung cấp bấm nút hủy tại giao diện thông tin yêu cầu</li> <li>2. Hệ thống hiển thị thông báo “Đã hủy đơn hàng”, cập nhật lại thông tin vào Firestore</li> <li>3. Quay lại bước 6 luồng sự kiện chính</li> </ol>

Bảng 18. Usecase phản hồi yêu cầu thuê xe

## r. Usecase thanh toán

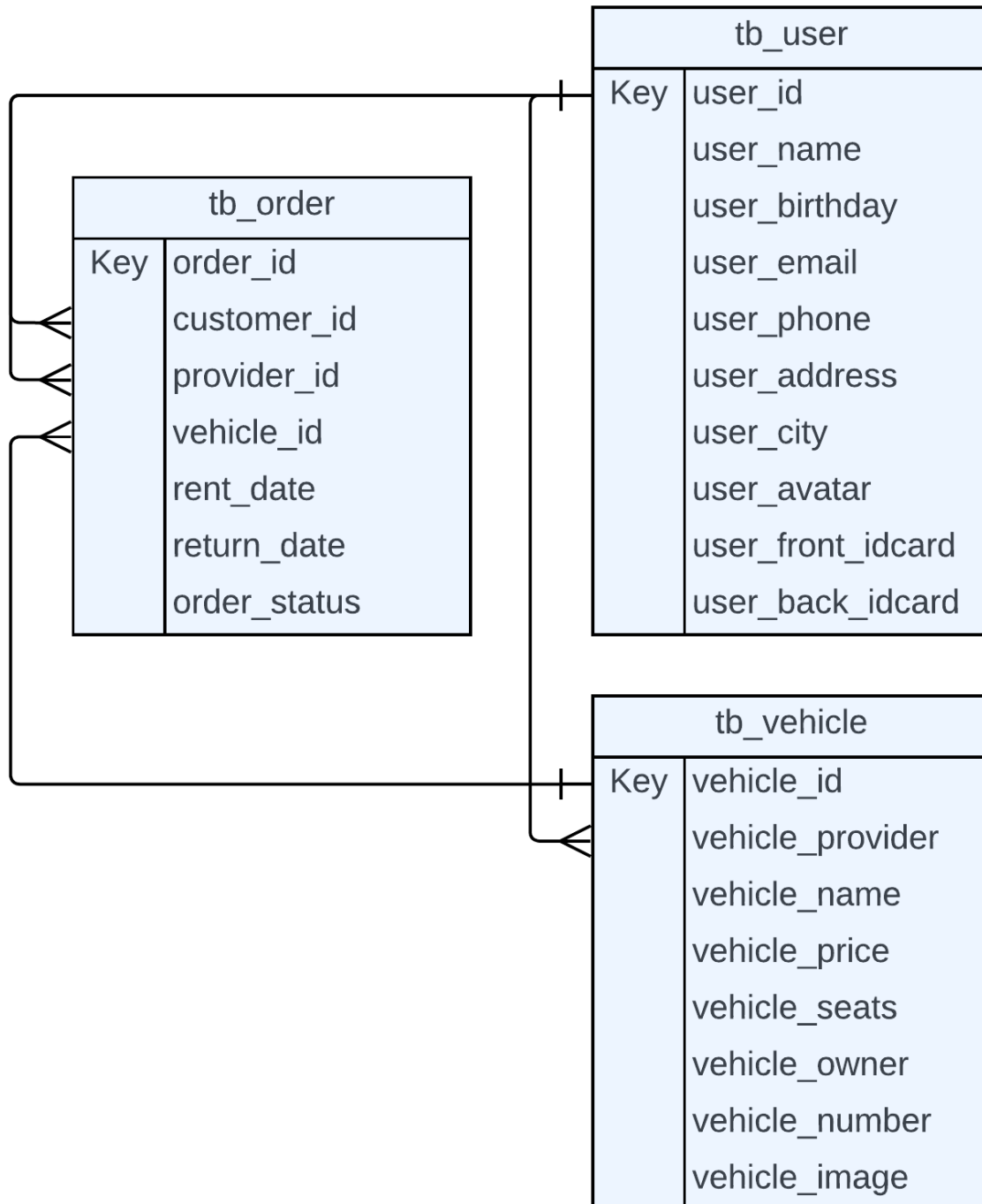
Usecase	Nội dung
Tên Usecase	Thanh toán
Mô tả	Usecase cho phép người dùng thanh toán
Actor	Người dùng đã gửi yêu cầu và được nhà cung cấp xác nhận

Điều kiện kích hoạt	Người dùng chọn trong phần Activity
Tiền điều kiện	<ol style="list-style-type: none"> <li>1. Phải có tài khoản đăng nhập của hệ thống</li> <li>2. Yêu cầu thuê xe đã được nhà cung cấp xác nhận</li> </ol>
Hậu điều kiện	Người dùng xem được các yêu cầu đã gửi cho nhà cung cấp
Luồng sự kiện chính	<ol style="list-style-type: none"> <li>1. Người dùng chọn màn hình Activity bên dưới màn hình</li> <li>2. Hệ thống hiển thị màn hình Activity chứa các yêu cầu người dùng đã gửi</li> <li>3. Người dùng bấm chọn vào một yêu cầu</li> <li>4. Hệ thống hiển thị màn hình thông tin đơn hàng người dùng đã gửi cho nhà cung cấp</li> <li>5. Người dùng bấm nút xác nhận và thanh toán</li> <li>6. Nếu nhà cung cấp đã xác nhận, hệ thống chuyển qua trang thanh toán</li> <li>7. Thanh toán thành công sẽ quay lại màn hình trước đó và cập nhật trạng thái “Đã thanh toán” vào Firestore</li> <li>8. Kết thúc Usecase</li> </ol>

Bảng 19. Usecase thanh toán

### 3. Thiết kế cơ sở dữ liệu

#### 3.1 Sơ đồ cơ sở dữ liệu



Hình 4. Mô hình ERD

## 3.2 Mô tả cơ sở dữ liệu

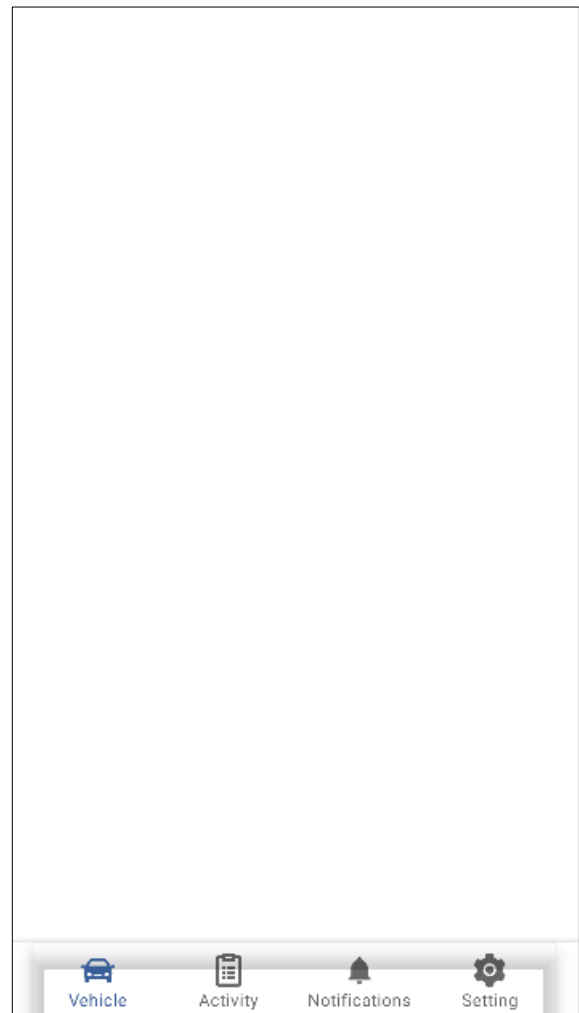
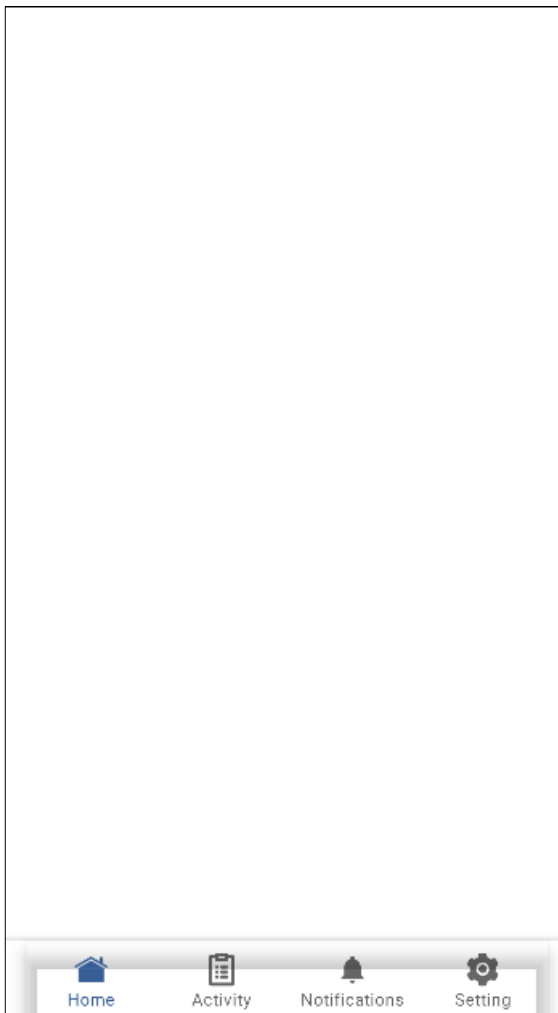
Quan hệ	Thuộc tính	Kiểu dữ liệu	Ý nghĩa	Ghi chú
<b>tb_user</b>	user_id	String	ID người dùng	khóa chính
	user_name	String	Tên người dùng	
	user_birthday	String	Ngày tháng năm sinh của người dùng	
	user_email	String	Địa chỉ email	
	user_phone	String	Số điện thoại	
	user_address	String	Địa chỉ người dùng	
	user_city	String	Thành phố	
	user_avatar	String	Avatar người dùng	
	user_front_idcard	String	Mặt trước thẻ căn cước	
	user_back_idcard	String	Mặt sau thẻ căn cước	
<b>tb_vehicle</b>	vehicle_id	String	Id xe	khóa chính
	vehicle_provider	String	Tên nhà cung cấp	khóa ngoại
	vehicle_name	String	Tên xe	
	vehicle_price	String	Gía thuê xe	
	vehicle_seats	String	Số lượng chỗ ngồi	
	vehicle_owner	String	Chủ sở hữu	
	vehicle_number	String	Biển số xe	
	vehicle_image	String	Hình ảnh xe	
<b>tb_order</b>	order_id	String	ID thuê xe	khóa chính
	customer_id	String	ID khách hàng	khóa ngoại
	provider_id	String	ID nhà cung cấp	khóa ngoại
	vehicle_id	String	ID xe	khóa ngoại
	rent_date	String	Ngày nhận xe	

	return_date	String	Ngày trả xe	
	order_status	String	Trạng thái cho thuê	

Bảng 20. Mô tả cơ sở dữ liệu

## IV. THỰC HIỆN ĐỀ TÀI

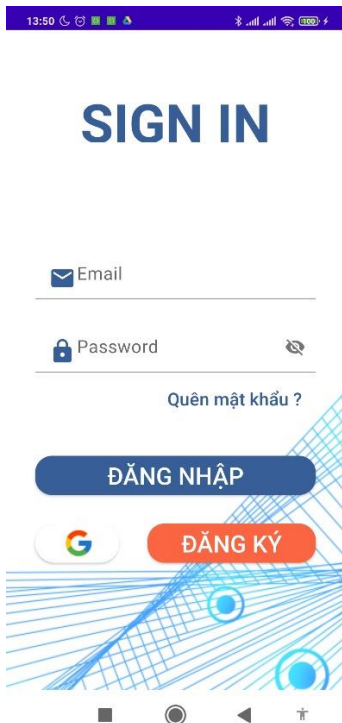
### 1. Phác thảo giao diện



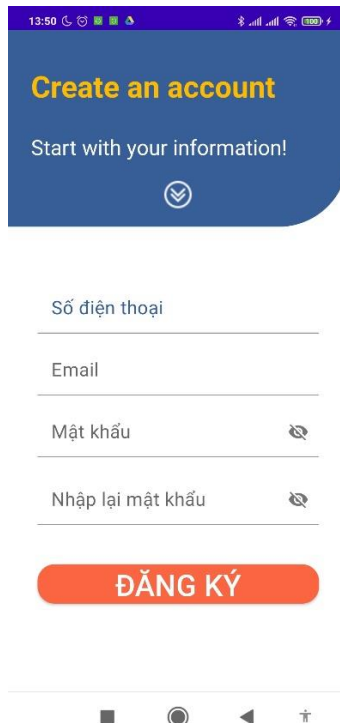
- Bản phác thảo chi tiết trên Figma:

<https://www.figma.com/file/oMM69h51KrZ75WLW64AjbK/H%E1%BB%87-th%E1%BB%91ng-thu%C3%AA-xe?node-id=0%3A1&t=bRIImF7fAe7XPRFE-1>

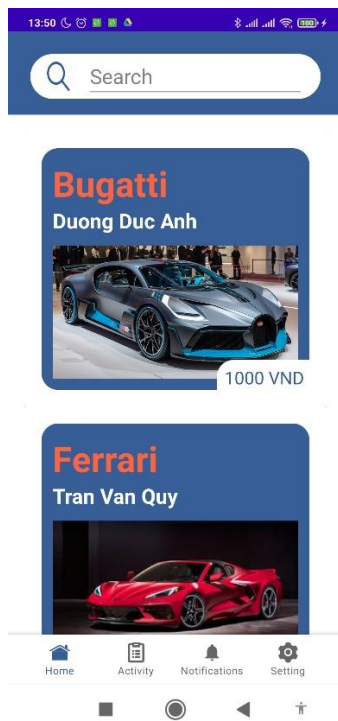
## 2. Hoàn thành giao diện



Màn hình đăng nhập



Màn hình Đăng ký

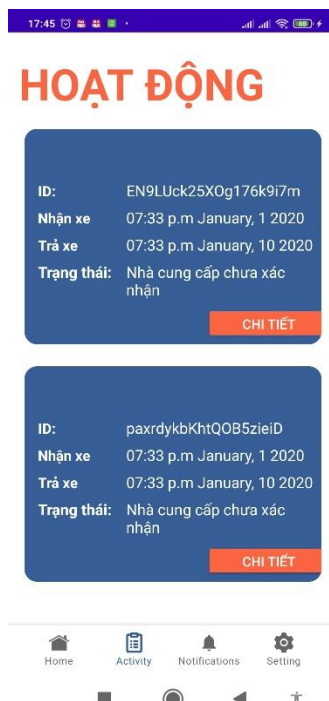


Màn hình chính



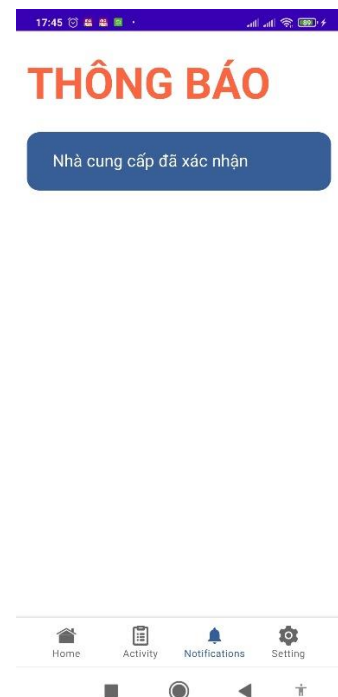
Màn hình chỉnh sửa thông tin cá nhân

Màn hình seeting

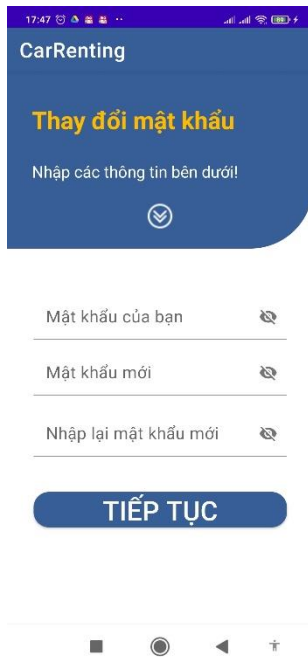


Màn hình Activity

Màn hình thông tin cá nhân



Màn hình thông báo của khách hàng



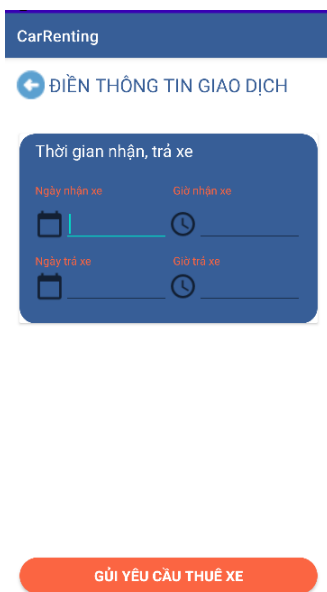
Màn hình thay đổi mật khẩu



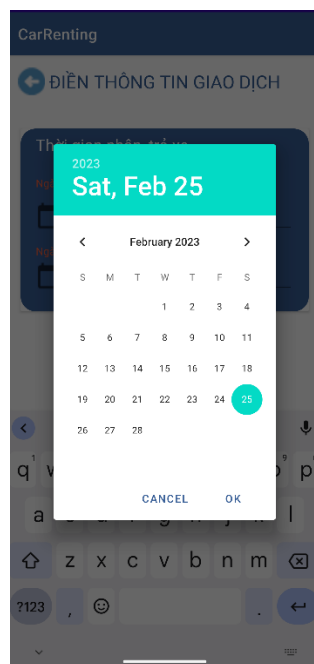
Màn hình lấy lại mật khẩu



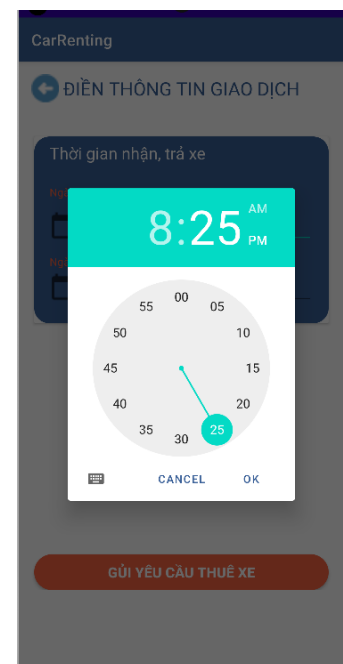
Màn hình thông tin



Màn hình điền thông tin



Màn hình điền thông tin  
(Chọn ngày)

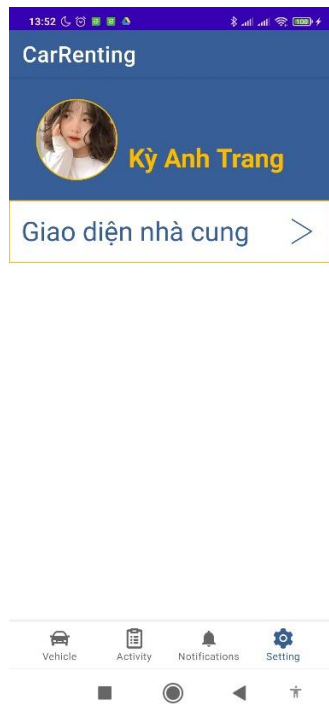


Màn hình điền thông tin  
(Chọn thời gian)





Màn hình thông báo bên  
luồng nhà cung cấp



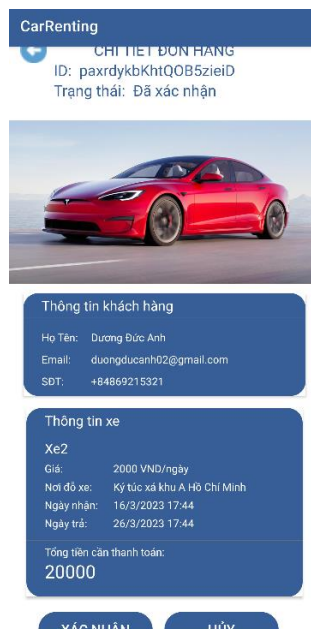
Màn hình setting bên luồng  
nhà cung cấp



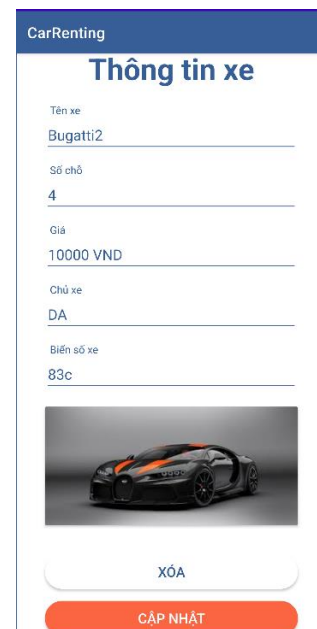
Màn hình Thêm xe



Màn hình gửi yêu cầu  
thành công



Màn hình nhà cung cấp  
phản hồi yêu cầu



Màn hình cập nhật và  
xóa xe

### 3. Công việc các thành viên

#### 3.1 Phân tích thiết kế

Công việc	Kỳ Anh	Quý	Đức Anh	Trung
Vẽ mô hình phân rã chức năng	x			
Vẽ sơ đồ usecase tổng quát			x	
Chỉnh sửa, hoàn thiện sơ đồ usecase tổng quát	x			
Đặc tả usecase		x		
Chỉnh sửa, bổ sung đặc tả usecase	x			x
Vẽ mô hình cơ sở dữ liệu ERD				
Chỉnh sửa, hoàn thiện mô hình ERD	x			
Mô tả cơ sở dữ liệu	x			

Bảng 21. Thành viên tham gia công việc phân tích thiết kế

#### 3.2 Thiết kế giao diện trên Figma

Công việc	Kỳ Anh	Quý	Đức Anh	Trung
Phác thảo các màn hình chính	x			
Phác thảo các màn hình phụ		x		

Bảng 22. Thành viên tham gia công việc thiết kế giao diện trên Figma

#### 3.3 Hiện thực hóa giao diện lên Android Studio

Công việc	Kỳ Anh	Quý	Đức Anh	Trung
Thiết lập navigation và menu khách hàng	x			
Thiết lập navigation và menu nhà cung cấp	x			
Màn hình chờ				x
Màn hình đăng nhập				x
Chỉnh sửa màn hình đăng nhập	x			
Màn hình đăng ký				x
Màn hình lấy lại mật khẩu				x
Màn hình thay đổi mật khẩu	x			

Màn hình nhập số điện thoại để xác thực khi đăng nhập bằng google	X			
Màn hình chỉnh sửa thông tin cá nhân				X
Thay đổi màn hình chỉnh sửa thông tin cá nhân	X			
Màn hình xem thông tin cá nhân	X			
Màn hình thêm xe	X			
Màn hình chỉnh sửa thông tin xe	X			
Màn hình cập nhật căn cước công dân				X
Màn hình hiển thị xe cả 2 phía	X			
Màn hình thông tin chi tiết xe		X		
Màn hình hoàn tất gửi yêu cầu đặt xe		X		
Màn hình xác nhận thông tin và thanh toán		X		
Màn hình chọn thời gian đặt xe		X		
Màn hình hiển thị hoạt động đặt xe 2 phía		X		
Màn hình thông báo		X		
Màn hình setting 2 phía	X			

Bảng 23. Thành viên tham gia công việc hiện thực hóa giao diện lên Android Studio

### 3.4 Thực hiện chức năng trên Android Studio

Công việc	Kỳ Anh	Quý	Đức Anh	Trung
Chức năng đăng nhập				X
Chỉnh sửa chức năng đăng nhập			X	
Chức năng đăng ký				X
Chỉnh sửa chức năng đăng ký	X			
Chức năng quên mật khẩu				X
Chức năng đăng nhập bằng goggle	X			
Chức năng thay đổi mật khẩu	X			
Chức năng xác thực số điện thoại				X

Chức năng cập nhật căn cước công dân				X
Chức năng xóa tài khoản	X			
Chức năng xem thông tin cá nhân			X	
Chức năng chỉnh sửa thông tin cá nhân				X
Xử lý ngày, tháng trong thông tin cá nhân			X	
Chức năng thêm, xóa xe	X			
Chức năng chỉnh sửa thông tin xe	X			
Chức năng hiển thị xe 2 phía	X			
Chức năng xem thông tin chi tiết xe			X	
Chức năng gửi yêu cầu đặt xe		X		
Chức năng xác nhận yêu cầu đặt xe		X		
Chức năng thực hiện thanh toán			X	
Chức năng hiển thị thông báo		X		
Xử lý thời gian nhận xe và trả xe			X	
Xử lý tổng tiền thuê xe			X	
Xử lý các thao tác qua lại với zalopay			X	
Xử lý luồng thay đổi fragment giữa các màn hình qua navigation và giữa giao diện khách hàng với nhà cung cấp	X			
Nổi các màn hình khi đặt xe		X		
Xử lý đăng nhập lần đầu	X			
Kiểm thử, debug và sửa lỗi	X		X	

Bảng 24. Thành viên tham gia công việc thực hiện chức năng trên Android Studio

### 3.5 Báo cáo

Công việc	Kỳ Anh	Quý	Đức Anh	Trung
Trình bày báo cáo	X			
Làm slide báo cáo	X			
Quay demo	X			

Viết báo cáo phần đặc tả usecase		X		
Viết hoàn chỉnh báo cáo	X			

Bảng 25. Thành viên tham gia công việc báo cáo

- Quản lí source code tại:  
[https://github.com/AllieInJune04/RENTOR\\_NHOM1\\_NT118.N11\\_2023](https://github.com/AllieInJune04/RENTOR_NHOM1_NT118.N11_2023) (đã đính kèm link quản lý source code cũ trong file Readme)
- Repo mới sẽ không có bạn Trung vì bạn không muốn vào
- Repo cũ sẽ có source code hoàn chỉnh của source code vì bạn Trung yêu cầu phải up lên
- Tài khoản git của các bạn:
  - Trang Kỳ Anh: <https://github.com/AllieInJune04/>
  - Dương Đức Anh: <https://github.com/ashentray>
  - Trần Văn Quý: <https://github.com/quy-k15>
  - Bùi Nguyễn Anh Trung: <https://github.com/AnhTrung732>

## V. KẾT LUẬN

### 1. Kết quả đạt được

Thông qua việc nghiên cứu và tìm hiểu công nghệ mới, nhóm chúng em đã phát triển hoàn thiện ứng dụng cho thuê xe RENTOR. Do thời gian có hạn nên ứng dụng nhiều thiếu sót và chưa thực hiện được các chức năng đặc biệt nổi trội. Tuy nhiên, kết quả chạy thử cho thấy hệ thống hoạt động tốt và có thể mở rộng, phát triển thêm trong tương lai. Đặc biệt, đối với chức năng thanh toán vẫn có nhiều nhóm không thể hoàn thiện vì vấn đề đăng ký tài khoản nhưng cuối cùng nhóm chúng em vẫn có thể hoàn thành tốt đó là một thành công lớn của nhóm.

Về mặt kỹ thuật, nhóm chính em đã đạt được những kết quả như sau:

- Biết cách tạo và sắp xếp project để dễ dàng thực hiện đồ án cũng như debug trên thiết bị
- Kết hợp thiết bị thực và thiết bị ảo hợp lí để có thể hoàn thiện tốt các chức năng
- Tìm hiểu và tận dụng tốt các phần hỗ trợ của Firebase một cách hợp lí

- Học được quy trình thực hiện một ứng dụng hoàn chỉnh từ lên kế hoạch, hiện thực, kiểm thử, deploy,...
- Rèn luyện tư duy, kỹ năng phân tích các luồng logic một cách hợp lí
- Nâng cao khả năng tìm hiểu và nghiên cứu công nghệ, đặc biệt là công nghệ mới

## **2. Hướng phát triển**

Đề tài còn có thể mở rộng và phát triển thêm nhiều tính năng với phạm vi rộng hơn như:

- Nhà cung cấp và khách hàng có thể liên lạc trực tiếp với nhau để trao đổi thông tin
- Nhà cung cấp có thể theo dõi xe của họ trong khi cho thuê xe
- Đặt ra các điều khoản sử dụng, công bố hợp đồng giữa nhà cung cấp và khách hàng
- Thêm nhiều hình thức thanh toán hơn
- Thêm nhiều loại phương tiện để phù hợp với nhu cầu khách hàng

## TÀI LIỆU THAM KHẢO

- [1] Google Services. (2023, 2 20). Firebase Document. Retrieved from Firebase:  
<https://firebase.google.com/docs>
- [2] StackOverflow. (2023, 2 20). StackOverflow. Retrieved from StackOverflow:  
<https://stackoverflow.com/>