

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA: HỆ THỐNG THÔNG TIN



BÁO CÁO ĐỒ ÁN PHÁT TRIỂN ỦNG DỤNG
THƯƠNG MẠI DI ĐỘNG
**PHÁT TRIỂN ỦNG DỤNG DI ĐỘNG
CHO THUÊ XE “T9 CAR RENTAL”**

Lớp: EC402.P21

Giảng viên hướng dẫn: LÝ ĐOÀN DUY KHÁNH

Sinh viên thực hiện

TRƯƠNG ĐĂNG KHÔI	21521019
NGUYỄN VIỆT QUANG	21522516
NGUYỄN TRẦN YẾN MY	22520902
PHẠM NGỌC ÁNH HỒNG	22520492
NGUYỄN THỊ CẨM LY	22520838
TRẦN GIA MINH	22520888

TP. Hồ Chí Minh, tháng 6 năm 2025

LỜI CẢM ƠN

Lời đầu tiên, Nhóm 9 chúng em xin gửi lời biết ơn sâu sắc tới Thầy Lý Đoàn Duy Khánh về sự hướng dẫn và sự hỗ trợ quý báu trong suốt quá trình thực hiện đồ án môn "Phát triển ứng dụng thương mại di động". Trong thời gian tham gia lớp học EC402.P21 của thầy, chúng em đã có thêm cho mình những kiến thức bổ ích với một tinh thần học tập năng suất, hiệu quả. Đó chắc chắn sẽ là những hành trang quý báu để chúng em vững bước sau này.

Mặc dù đã rất cố gắng nhưng do chưa có nhiều kinh nghiệm làm đề tài cũng như hạn chế về mặt chuyên môn, nên trong quá trình hoàn thành bài đồ án nhóm chúng em sẽ không tránh khỏi những thiếu sót. Vì vậy, những lời nhận xét, chỉ bảo của thầy chính là những góp ý cực kỳ quý giá để nhóm có thể hoàn thiện kiến thức và bản thân mình tốt hơn trong các đồ án sau.

Cuối cùng, chúng em xin gửi lời chúc tốt đẹp nhất tới Thầy Lý Đoàn Duy Khánh, kính chúc thầy sẽ tiếp tục nhận được nhiều sức khỏe và thành công trong sự nghiệp giảng dạy và nghiên cứu của mình.

Thành phố Hồ Chí Minh, tháng 6 năm 2025

Nhóm sinh viên thực hiện

NHẬN XÉT CỦA GIẢNG VIÊN

MỤC LỤC

I. GIỚI THIỆU ĐỀ TÀI.....	10
1. Bối cảnh hình thành ứng dụng.....	10
2. Mục tiêu phát triển ứng dụng.....	11
3. Đối tượng sử dụng ứng dụng.....	12
4. Phạm vi thực hiện.....	13
4.1. Phạm vi.....	13
4.2. Phương pháp thực tiễn.....	14
4.3. Kết quả dự kiến.....	15
II. CƠ SỞ LÝ THUYẾT.....	17
1. Tổng quan về thương mại di động.....	17
1.1. Khái niệm.....	17
1.2. Đặc điểm nổi bật.....	17
1.3. Xu hướng.....	17
2. Mô hình ứng dụng TMĐT trên mobile - C2C.....	18
2.1. Định nghĩa.....	18
2.2. Đặc điểm của mô hình C2C trên mobile.....	18
3. Tổng quan về công nghệ sử dụng.....	18
3.1. Java Development Kit (JDK).....	18
3.2. Android SDK.....	19
3.3. Java Virtual Machine (JVM).....	20
3.4. Firebase.....	21
4. UX/UI trong thiết kế ứng dụng di động.....	22
4.1. Khái niệm:.....	22
4.2. Tầm quan trọng trong ứng dụng thuê xe.....	23
III. PHÂN TÍCH YÊU CẦU.....	23
1. Phân tích người dùng mục tiêu.....	23
1.1. Người dùng (Người thuê xe).....	23
1.2. Người cho thuê xe.....	25
1.3. Admin quản lý hệ thống.....	26

2. Tính năng cần có (chức năng chính).....	28
3. Lưu đồ quy trình nghiệp vụ.....	29
3.1. Quy trình nghiệp vụ đối với người dùng (user).....	29
Quy trình đăng ký (Tạo hồ sơ khách hàng).....	29
Quy trình đăng nhập.....	30
Quy trình đặt xe.....	31
Quy trình thanh toán.....	33
Quy trình Nhận xe và Trả xe.....	34
Quy trình quản lý đơn đặt xe.....	35
Quy trình đánh giá và góp ý.....	36
3.2. Quy trình nghiệp vụ đối với quản trị viên (admin).....	37
Quy trình tổng quát của admin.....	37
Quy trình quản lý người dùng.....	38
Quy trình quản lý danh mục thuê xe.....	39
Quy trình báo cáo và thống kê hệ thống.....	40
Quy trình cài đặt hệ thống.....	41
4. Use-case diagram.....	44
4.1. Tổng quát.....	44
4.2. Đặc tả.....	44
IV. THIẾT KẾ ỨNG DỤNG.....	46
1. Giao diện người dùng (UI): sơ đồ màn hình, bô cục.....	46
1.1. Admin.....	46
1.1.1. Quản lý người dùng.....	46
1.1.2. Quản lý danh mục thuê xe.....	46
1.1.2.1. Quản lý xe chờ duyệt.....	46
1.1.2.2. Danh sách và tìm kiếm xe.....	47
1.1.2.3. Dashboard quản lý xe.....	48
1.1.2.4. Quản lý xe đang hoạt động.....	49
1.1.3. Báo cáo & thống kê hệ thống.....	50
1.1.4. Cài đặt hệ thống.....	51

1.1.4.1. Quản lý tài khoản.....	51
1.1.4.2. Quản lý hóa đơn sản phẩm.....	51
1.1.4.3. Cấu hình hệ thống thông báo.....	52
1.1.4.4. Lịch sử thao tác admin.....	53
1.2. Account.....	54
1.2.1. Splash.....	54
1.2.2. Sign in.....	54
1.2.3. Sign up.....	55
1.2.4. Account.....	56
1.2.5. Main screen.....	57
1.2.6. Car Details.....	58
1.2.7. Select Date & Time.....	59
1.2.8. Request Success.....	60
1.2.9. Car Owner.....	61
1.2.9.1 Order request.....	61
1.2.9.2 Payment confirmation.....	62
2. Thiết kế cơ sở dữ liệu.....	63
2.1. Sơ đồ cơ sở dữ liệu.....	63
2.2. Mô tả cơ sở dữ liệu.....	63
2.2.1 Bảng: users.....	63
2.2.2. Bảng: vehicles.....	64
2.2.3. Bảng: rentals.....	65
2.2.4. Bảng: payments.....	65
2.2.5. Bảng: reviews.....	66
2.2.6. Bảng: vehicle_categories.....	66
2.2.7. Bảng: locations.....	66
2.2.8. Bảng: vehicle_images.....	66
2.2.9. Bảng: favorites.....	67
2.2.10. Bảng: notifications.....	67
2.2.11. Bảng: admins.....	67

2.2.12. Bảng: login_history.....	67
2.2.13. Bảng: user_verifications.....	68
2.2.14. Bảng: vehicle_availability.....	68
2.2.15. Bảng: access_tokens.....	68
2.2.16. Bảng: roles.....	68
2.2.17. Bảng: user_roles.....	69
3. Kiến trúc tổng thể của ứng dụng.....	69
V. TRIỂN KHAI.....	69
1. Công cụ, công nghệ đã sử dụng.....	69
2. Từng chức năng đã lập trình.....	70
2.1. Chức năng dành cho khách hàng (Customer).....	70
2.2. Chức năng dành cho chủ xe (Owner).....	71
2.3. Chức năng dành cho quản trị viên (Admin).....	72
2.4. Chức năng chung.....	73
3. Giao diện thực tế.....	74
3.1. Màn hình giới thiệu (Splash Screen):.....	74
3.2. Danh sách xe (Customer Home):.....	74
3.3. Chi tiết xe (Vehicle Detail):.....	75
3.4. Chọn thời gian thuê xe (Schedule Select):.....	76
3.5. Chi tiết đơn đặt xe và thanh toán (Booking Detail):.....	78
3.6. Lịch sử hoạt động (Customer Activity):.....	79
3.7. Quản lý thông báo (Customer Notifications):.....	80
3.8. Quản lý hồ sơ cá nhân (Customer Profile):.....	81
3.9. Thêm xe mới (Add Vehicle):.....	83
3.10. Cập nhật xe (Update Vehicle):.....	84
3.11. Chấp nhận yêu cầu đặt xe (Owner Booking):.....	85
3.12. Quản lý xe của chủ xe (Owner Vehicles):.....	87
3.13. Quản lý thông báo của chủ xe (Owner Notifications):.....	87
3.14. Quản lý xe của admin (Admin Vehicles):.....	88
3.15. Quản lý người dùng của admin (Admin Users):.....	89

3.16. Thống kê của admin (Admin Statistics):.....	91
3.17. Đăng nhập/Đăng ký:.....	91
3.18. Chat:.....	92
4. Đoạn code mẫu minh họa các chức năng chính.....	94
4.1. Thuê xe (Customer):.....	94
4.2. Quản lý xe (Admin):.....	94
4.3. Quản lý User (Admin):.....	95
4.4. Thanh toán (Customer):.....	96
4.5. Chọn thời gian thuê xe (ScheduleSelect, Customer):.....	98
4.6. Thêm xe mới (AddVehicle, Owner):.....	100
4.7. Cập nhật xe (UpdateVehicle, Owner):.....	100
4.8. Chấp nhận yêu cầu đặt xe (Owner):.....	101
VI. KIỂM THỬ.....	102
 1. Cách kiểm thử từng tính năng.....	102
1.1. Chức năng dành cho khách hàng (Customer).....	102
1.2. Chức năng dành cho chủ xe (Owner).....	107
1.3 Chức năng dành cho quản trị viên (Admin).....	109
1.4 Chức năng chung.....	111
1.5. Flow kiểm thử cụ thể.....	112
 2. Ghi nhận lỗi và cách xử lý.....	114
2.1. Lỗi thực tế: Truy vấn ràng buộc Firestore phức tạp gây không hiển thị dữ liệu:.....	114
2.2. Lỗi giả định: Tải ảnh xe thất bại do URL không hợp lệ trong Cloudinary:..	115
2.3. Lỗi giả định: Thanh toán ZaloPay thất bại do token hết hạn:.....	116
2.4. Lỗi giả định: Thông báo không gửi được đến khách hàng/chủ xe:.....	116
 3. Đánh giá mức độ hoàn thiện.....	117
VII. KẾT QUẢ VÀ ĐÁNH GIÁ.....	118
 1. Tính khả thi của ứng dụng.....	118
 2. Mức độ hoàn thành các tính năng so với kế hoạch.....	119
 3. Khả năng triển khai thực tế (thị trường, đối tượng mục tiêu).....	119

VIII. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....	120
1. Tổng kết những gì đạt được.....	120
2. Hướng cải tiến về mặt tính năng/ giao diện/ công nghệ.....	120
3. Tiềm năng thương mại hóa.....	121
IX. TÀI LIỆU THAM KHẢO.....	121
X. PHỤ LỤC.....	122
1. Business Rule Catalogue.....	122
2. Data Dictionary.....	131
3. Glossary.....	132

I. GIỚI THIỆU ĐỀ TÀI

1. Bối cảnh hình thành ứng dụng

Trong giai đoạn hiện nay, công nghệ thông tin và truyền thông phát triển nhanh chóng, đặc biệt là sự phổ cập của thiết bị di động thông minh (smartphone) và mạng Internet. Một báo cáo của Rakuten Viber cho thấy Việt Nam hiện có hơn 100 triệu thuê bao sử dụng smartphone; trung bình một người dùng di động ở Việt Nam tải về 3 tỷ ứng dụng trong năm 2024, xếp thứ 11 toàn cầu (theo [vnexpress.net](#)). Những con số này chứng tỏ người tiêu dùng ngày càng chuyển sang trải nghiệm trên nền tảng di động, thậm chí nhiều phần mềm hiện chỉ cung cấp dưới dạng ứng dụng di động (mobile-only) (theo [vnexpress.net](#)). Trong bối cảnh đó, việc ứng dụng công nghệ di động vào các dịch vụ truyền thống ngày càng trở thành xu hướng tất yếu.

Song song với xu hướng chung của thị trường di động, nhu cầu về dịch vụ vận tải và cho thuê phương tiện cá nhân cũng ngày càng tăng cao. Tại Việt Nam, thị trường cho thuê xe (ô tô, xe tự lái,...) đang phát triển nhanh chóng. Theo số liệu từ Statista, thị trường cho thuê xe tại Việt Nam dự kiến tăng trưởng với tốc độ khoảng 4,05% mỗi năm giai đoạn 2024–2029, đạt doanh thu xấp xỉ 1,11 tỷ USD vào năm 2029 (theo [thitruong24h.vtv.vn](#)). Mức độ sở hữu ô tô tại Việt Nam hiện vẫn khá thấp (khoảng 63 xe trên 1.000 người) (theo [thitruong24h.vtv.vn](#)), trong khi du lịch nội địa và quốc tế đang tăng trưởng mạnh mẽ (năm 2023 Việt Nam đón hơn 12 triệu lượt khách quốc tế, dự kiến 17 triệu lượt vào 2025) (theo [thitruong24h.vtv.vn](#)). Điều này đồng nghĩa với nhu cầu cao về phương tiện di chuyển linh hoạt: nhiều người muốn tự thuê xe ô tô để chủ động trong việc đi lại. Đồng thời, các ứng dụng gọi xe công nghệ như Grab hay Be với tỷ lệ thị phần lớn (Grab chiếm trên 80% gọi xe công nghệ) đã thay đổi thói quen di chuyển của người dân, khởi dậy xu hướng sử dụng dịch vụ vận tải trên di động (theo [thitruong24h.vtv.vn](#)).

Tuy nhiên, hiện nay vẫn thiếu một nền tảng tập trung và tiện lợi cho dịch vụ cho thuê xe cá nhân. Người muốn cho thuê phương tiện gặp khó khăn trong việc quảng bá và quản lý thông tin xe, trong khi khách thuê thường phải tìm kiếm thông tin rải rác, gọi điện hoặc nhắn tin nhiều nơi mới thu thập được đủ lựa chọn. Mặt khác, các đơn vị cho thuê nhỏ lẻ chưa có công cụ tối ưu để kết nối với khách hàng tiềm năng. Từ thực tế đó, việc xây dựng một ứng dụng di động cho thuê xe sẽ giúp kết nối liền mạch giữa người cho thuê và người thuê, đẩy mạnh khả năng tiếp cận thông tin, rút ngắn quy trình liên hệ, đồng thời cập nhật nhanh chóng các thông tin về phương tiện sẵn có. Vì những lý do trên, đề tài “Xây dựng ứng dụng di động cho thuê xe” trở nên cần thiết và có tính thực tiễn cao trong bối cảnh phát triển hiện nay.

2. Mục tiêu phát triển ứng dụng

Mục tiêu chung của đề tài là phát triển một ứng dụng di động cho phép người dùng thuê và cho thuê xe ô tô một cách tiện lợi, an toàn và hiệu quả. Ứng dụng này sẽ tập trung vào cung cấp trải nghiệm thân thiện, dễ sử dụng cho cả hai bên tham gia: chủ xe và người thuê xe. Cụ thể, để đạt được mục tiêu chung, đề tài đặt ra các mục tiêu thành phần như sau:

- Nghiên cứu nhu cầu và quy trình cho thuê xe: Thu thập yêu cầu chức năng và nghiệp vụ của cả người cho thuê lẫn người thuê thông qua khảo sát và tham khảo mô hình các dịch vụ hiện có (như thuê xe tự lái, ứng dụng gọi xe). Từ đó xác định các tính năng cần thiết cho ứng dụng (đăng ký tài khoản, đăng tin xe cho thuê, tìm kiếm xe, đặt xe, liên hệ, thanh toán, v.v.).
- Thiết kế giao diện và kiến trúc hệ thống: Thiết kế giao diện người dùng (UI/UX) phù hợp với điện thoại di động, đảm bảo trực quan, dễ sử dụng. Xây dựng kiến trúc hệ thống gồm ứng dụng di động và cơ sở dữ liệu, xác định các thành phần chính như mô-đun đăng nhập, quản lý người dùng, quản lý tin cho thuê, tìm kiếm và đặt xe, bộ phận xử lý thanh toán và thông báo.

- Phát triển ứng dụng chức năng: Triển khai các chức năng chính cho thuê xe trên nền tảng di động (Android và/hoặc iOS): cho phép chủ xe đăng tin giới thiệu phương tiện kèm thông số, giá thuê và hình ảnh; cho phép người thuê tìm kiếm và lọc danh sách xe theo yêu cầu (loại xe, ngày thuê, địa điểm,...), xem chi tiết xe, liên hệ đặt thuê thông qua ứng dụng (qua gọi điện, chat hoặc hẹn giờ), cũng như thực hiện thanh toán (có thể mua phỏng hoặc tích hợp công thanh toán).
- Đảm bảo an toàn và hiệu suất: Nghiên cứu và áp dụng các biện pháp bảo mật dữ liệu người dùng (ví dụ mã hóa thông tin cá nhân, bảo vệ thông tin liên lạc) và tối ưu hiệu suất ứng dụng (tốc độ tải trang, xử lý tìm kiếm nhanh). Đảm bảo ứng dụng có thể vận hành ổn định trên các thiết bị di động thông thường.
- Triển khai thử nghiệm và đánh giá: Thực hiện kiểm thử hệ thống (kiểm thử chức năng, kiểm thử giao diện) và tổ chức thử nghiệm với người dùng thực tế (định công ngắn) để thu thập ý kiến phản hồi. Điều chỉnh, hoàn thiện ứng dụng dựa trên kết quả thử nghiệm để đảm bảo sản phẩm đáp ứng được yêu cầu đề ra.

Qua những mục tiêu trên, đề tài nhằm xây dựng được một ứng dụng mẫu (prototype) cho thuê xe trên di động, đảm bảo đầy đủ chức năng cốt lõi và trải nghiệm người dùng. Ngoài ra, kết quả của đề tài sẽ bao gồm tài liệu hướng dẫn, mã nguồn, báo cáo chi tiết quá trình thực hiện và đánh giá kết quả.

3. Đối tượng sử dụng ứng dụng

Đối tượng tiềm năng chính của ứng dụng là người cho thuê xe và người có nhu cầu thuê xe. Cụ thể:

- Chủ xe cho thuê: Là các cá nhân hoặc doanh nghiệp sở hữu ô tô muốn cho người khác thuê để kiếm thu nhập. Họ cần một kênh thuận tiện để đăng tải thông tin, quản lý lịch sử cho thuê và tương tác với khách hàng. Trên ứng dụng, chủ xe có thể tạo tài khoản,

đăng thông tin xe (gồm hình ảnh, mô tả, giá thuê, điều kiện thuê), theo dõi yêu cầu đặt thuê và quản lý các giao dịch.

- Khách hàng thuê xe: Bao gồm cá nhân đi công tác, du lịch, sinh viên hoặc người đi du lịch ngắn ngày (trong nước hoặc quốc tế) muốn thuê phương tiện tự lái. Họ sử dụng ứng dụng để tìm kiếm xe phù hợp theo yêu cầu (về loại xe, thời gian và địa điểm thuê), xem chi tiết thông tin xe, liên hệ và tiến hành đặt thuê. Ứng dụng sẽ giúp khách thuê dễ dàng so sánh các lựa chọn, liên hệ nhanh với chủ xe và thanh toán thuận tiện.

Ngoài ra, đối tượng gián tiếp bao gồm nhân viên quản trị hệ thống (admin) nếu có, để giám sát cơ sở dữ liệu và hỗ trợ khi cần.

4. Phạm vi thực hiện

4.1. Phạm vi

Để tài được thực hiện trong khuôn khổ môn Thiết kế Ứng dụng Di động, do đó phạm vi dự án có những giới hạn cụ thể sau:

- Phạm vi về loại phương tiện: Ứng dụng tập trung vào việc cho thuê xe cá nhân cụ thể là xe ô tô. Các loại phương tiện đặc thù khác (xe tải, xe bus, tàu thuyền...) không nằm trong phạm vi hiện tại.
- Phạm vi địa lý và thị trường: Ứng dụng hướng đến thị trường Việt Nam, ưu tiên các thành phố lớn nơi nhu cầu thuê xe cao (như Hà Nội, TP. Hồ Chí Minh, Đà Nẵng). Trong giai đoạn đầu, dữ liệu thử nghiệm có thể chỉ mô phỏng khu vực cụ thể (ví dụ một hoặc hai tỉnh/thành phố) để dễ dàng quản lý. Ứng dụng có thể mở rộng cho thị trường khác trong tương lai nhưng hiện tại chỉ xem xét điều kiện và luật pháp trong nước.
- Phạm vi chức năng và kỹ thuật: Đề tài xây dựng nguyên mẫu (prototype) của ứng dụng, do đó không nhất thiết tích hợp đầy đủ mọi thành phần thương mại như cổng thanh toán thực tế, hệ thống định vị toàn cầu (GPS) chính xác, hay dịch vụ xử lý phí và bảo hiểm xe. Một số tính năng phức tạp (ví dụ bảo hiểm giao dịch, giải

pháp thanh toán đa dạng) có thể được giả lập hoặc triển khai dạng đơn giản. Ngoài ra, ứng dụng chủ yếu phát triển trên nền tảng Android (có thể mở rộng sang iOS nếu có thời gian và nguồn lực).

- Phạm vi thời gian và dữ liệu: Đề tài giới hạn trong thời gian thực hiện của đồ án (thường một kỳ học hoặc vài tháng). Dữ liệu sử dụng chủ yếu là dữ liệu mẫu (test data) và trường hợp giả lập. Việc thu thập dữ liệu thực từ người dùng hoặc doanh nghiệp cho thuê xe sẽ ở mức sơ bộ (thông qua điều tra, phỏng vấn để lấy yêu cầu) nhưng không mang tính chất triển khai thương mại đại trà.

Tóm lại, phạm vi áp dụng của đề tài này là xây dựng và thử nghiệm một ứng dụng di động cho thuê xe trong môi trường giả lập và quy mô giới hạn, tập trung vào yếu tố chức năng cốt lõi và trải nghiệm người dùng trong bối cảnh thị trường Việt Nam.

4.2. Phương pháp thực tiễn

Đề tài áp dụng phương pháp phân tích và thiết kế hệ thống kết hợp với quy trình phát triển phần mềm chuẩn để thực hiện các nội dung nghiên cứu và triển khai. Các bước thực hiện cụ thể bao gồm:

- **Thu thập và phân tích yêu cầu:** Xác định yêu cầu người dùng thông qua việc khảo sát, phỏng vấn nhóm đối tượng tiềm năng (mục 3.1), tham khảo các ứng dụng tương tự. Thu thập các yêu cầu chức năng (đăng nhập, đăng tin xe, tìm kiếm, đặt xe, thanh toán,...) và yêu cầu phi chức năng (giao diện, bảo mật, hiệu năng).
- **Thiết kế hệ thống:** Dựa trên yêu cầu đã phân tích, xây dựng mô hình hệ thống. Thiết kế sơ đồ luồng nghiệp vụ (Flowchart), mô hình Use-case, sơ đồ lớp (Class Diagram), và thiết kế cơ sở dữ liệu (ERD). Song song đó, thiết kế giao diện đồ họa (wireframe/UI mockup) cho các màn hình chính của ứng dụng.
- **Phát triển và hiện thực:** Lập trình ứng dụng di động sử dụng ngôn ngữ và công nghệ phù hợp (ví dụ: Kotlin/Java cho Android, hoặc Flutter/React Native cho đa nền tảng). Chia công việc theo từng

mô-đun: xử lý đăng nhập, quản lý người dùng, quản lý đăng tin xe, tìm kiếm và lọc xe, quản lý đơn hàng, thông báo, v.v. Tích hợp cơ sở dữ liệu (có thể dùng SQLite cục bộ hoặc kết nối đến cơ sở dữ liệu từ xa) và API/Server nếu cần. Áp dụng phương pháp Agile hoặc phân chia sprint để triển khai theo từng giai đoạn và kiểm soát tiến độ.

- **Kiểm thử và hiệu chỉnh:** Tiến hành kiểm thử chức năng (đơn vị và hệ thống) để đảm bảo mỗi tính năng hoạt động đúng theo yêu cầu. Tổ chức thử nghiệm với người dùng thực tế (phản hồi từ bạn bè, thầy cô hoặc nhóm thử nghiệm) để đánh giá mức độ thân thiện và sửa lỗi. Dựa trên kết quả kiểm thử, điều chỉnh, hoàn thiện phần mềm và giao diện.
- **Triển khai và đánh giá kết quả:** Cài đặt ứng dụng trên thiết bị di động thử nghiệm, thực hiện đánh giá tính đáp ứng của hệ thống qua các tiêu chí kỹ thuật (tốc độ, ổn định) và phi kỹ thuật (mức độ hài lòng của người dùng thử nghiệm). Chuẩn bị báo cáo kết quả, bao gồm tài liệu hướng dẫn sử dụng, tài liệu thiết kế và mã nguồn.

Các phương pháp trên đảm bảo tính khoa học và thực tiễn cho quá trình phát triển ứng dụng. Đồng thời, việc thử nghiệm trực tiếp với người dùng giúp đánh giá sát thực nhu cầu và cải thiện chất lượng sản phẩm cuối cùng.

4.3. Kết quả dự kiến

Cuối cùng, đề tài kỳ vọng mang lại những kết quả thực tế như sau:

- **Ứng dụng di động mẫu chạy ổn định:** Hoàn thành một ứng dụng di động (prototype) cho thuê xe chạy được trên thiết bị Android với các chức năng chính. Người dùng có thể tạo tài khoản, đăng nhập, tìm kiếm danh sách phương tiện cho thuê, xem thông tin chi tiết (loại xe, hình ảnh, giá thuê), và liên hệ đặt thuê qua ứng dụng. Chủ xe có thể đăng tin quảng cáo phương tiện, quản lý các bài đăng và trạng thái cho thuê. Nếu có, các chức năng hỗ trợ như thanh toán

trực tuyến (hoặc mô phỏng thanh toán), thông báo cho người thuê về trạng thái đơn hàng cũng sẽ được tích hợp. Giao diện của ứng dụng được thiết kế thân thiện, đáp ứng tốt trải nghiệm di động.

- **Hệ thống cơ sở dữ liệu và quản lý:** Xây dựng cơ sở dữ liệu chứa thông tin về người dùng, phương tiện, đơn đặt xe, và tương tác đặt thuê. Nếu có phân hệ quản trị (dành cho admin hoặc người phát triển), có thể quản lý được người dùng, duyệt/kiểm duyệt tin đăng và xem báo cáo thống kê đơn giản.
- **Tăng tính hiệu quả và tiện lợi:** Dự kiến, ứng dụng cho phép rút ngắn quy trình cho thuê xe so với phương thức thủ công. Người thuê không còn phải gọi điện hoặc đến trực tiếp nơi thuê để xem xe; chủ xe cũng dễ dàng quảng bá thông tin đến người có nhu cầu. Kết quả này được đánh giá qua phản hồi của người dùng thử nghiệm cho thấy thời gian tìm kiếm và đặt thuê giảm đáng kể, mức độ hài lòng tăng lên.
- **Tài liệu và báo cáo hoàn chỉnh:** Hoàn thiện báo cáo đồ án bao gồm phần giới thiệu, phân tích yêu cầu, thiết kế hệ thống, kết quả thử nghiệm và đánh giá. Cùng với đó, cung cấp mã nguồn của ứng dụng và các tài liệu hướng dẫn cài đặt, sử dụng. Đây sẽ là cơ sở để chứng minh kết quả đề tài đáp ứng được mục tiêu đã đề ra.

Như vậy, sau khi hoàn thành đề tài, nhóm em mong muốn có được một sản phẩm mẫu hữu ích, đồng thời rút ra các bài học kinh nghiệm về quy trình thiết kế và phát triển ứng dụng di động. Kết quả đề tài nếu thành công sẽ góp phần minh chứng tính khả thi của giải pháp công nghệ trong lĩnh vực cho thuê xe và làm cơ sở để phát triển các sản phẩm tương tự ở quy mô lớn hơn.

II. CƠ SỞ LÝ THUYẾT

1. Tổng quan về thương mại di động

1.1. Khái niệm

Thương mại di động (m-commerce) là quá trình mua và bán hàng hóa thông qua các thiết bị cầm tay không dây như điện thoại thông minh và máy tính bảng. Các giao dịch thương mại di động có thể dành cho các sản phẩm hoặc dịch vụ bao gồm các mặt hàng thời trang, phần mềm kinh doanh hoặc hàng tiêu dùng.

M-commerce là một loại thương mại điện tử cho phép người mua sắm trực tuyến mua hàng mà không cần sử dụng máy tính.

Ví dụ về thương mại di động bao gồm:

- Mua hàng dựa trên trình duyệt web
- Mua hàng trong ứng dụng
- Ứng dụng thị trường ảo
- Ví điện tử như Apple Pay, Google Pay và Samsung Wallet

1.2. Đặc điểm nổi bật

- Truy cập mọi lúc mọi nơi.
- Cá nhân hóa theo hành vi người dùng.
- Tích hợp thanh toán di động (e-wallet, QR code...).
- Hỗ trợ định vị và tương tác theo thời gian thực (real-time).

1.3. Xu hướng

Tăng trưởng mạnh: Theo báo cáo của Insider Intelligence, đến năm 2025, hơn 10% tổng giao dịch bán lẻ toàn cầu được thực hiện qua thiết bị di động, tăng từ 8% vào năm 2023.

Sự hỗ trợ từ công nghệ 5G: Kết nối nhanh hơn giúp trải nghiệm mua sắm di động mượt mà hơn, đặc biệt trong livestream hoặc AR/VR.

Phát triển của Social Commerce: Facebook Shop, TikTok Shop, Instagram Shopping trở thành kênh bán hàng chủ đạo.

Trí tuệ nhân tạo và dữ liệu lớn: Được ứng dụng mạnh để tự động hóa tiếp thị, cá nhân hóa đề xuất sản phẩm và phân tích hành vi.

Thời lượng sử dụng di động tăng: Trung bình mỗi người dành hơn 3 giờ mỗi ngày cho điện thoại, mở thiết bị mỗi 10 phút/lần, tạo ra tiềm năng lớn cho thương mại di động.

2. Mô hình ứng dụng TMĐT trên mobile - C2C

2.1. Định nghĩa

C2C (Consumer-to-Consumer) là mô hình thương mại trong đó **người dùng là bên trực tiếp bán và mua sản phẩm/dịch vụ lẫn nhau**, thông qua nền tảng trung gian (ứng dụng mobile). Nền tảng cung cấp công cụ hỗ trợ đăng tin, xác minh, thanh toán, và giao hàng an toàn.

2.2. Đặc điểm của mô hình C2C trên mobile

- Người dùng hai chiều: Một tài khoản có thể vừa là người bán (chủ xe), vừa là người mua (người thuê xe).
- Tập trung vào xác minh và đánh giá uy tín: Do không có bên doanh nghiệp bảo đảm, nên hệ thống cần có cơ chế xác minh giấy tờ, đánh giá sao, phản hồi người dùng.
- Thanh toán linh hoạt: Hỗ trợ nhiều hình thức thanh toán di động như ví điện tử, QR Code, chuyển khoản.
- Tính năng báo cáo, xử lý tranh chấp: Quan trọng để kiểm soát rủi ro do sự đa dạng người dùng.
- Định vị & bản đồ: Cho phép người dùng tìm xe gần mình, tra vị trí xe trong thời gian thực.

3. Tổng quan về công nghệ sử dụng

3.1. Java Development Kit (JDK)

Mô tả: (JDK) là một trong ba gói công nghệ cốt lõi được sử dụng trong lập trình Java, cùng với JVM (Máy ảo Java - Java Virtual Machine) và JRE (Java Runtime Environment - Môi trường Java Runtime). Việc phân biệt giữa ba công nghệ này, cũng như hiểu được cách chúng kết nối với nhau là rất quan trọng.

JVM là thành phần Java nền tảng để chạy các chương trình.

JRE là phần on-disk của Java, phần tạo ra JVM.

JDK cho phép các developer tạo các chương trình Java, trong đó các chương trình có thể được JVM và JRE xử lý và chạy

Java Development Kit (JDK) là bộ công cụ phát triển phần mềm chính thức cho ngôn ngữ lập trình Java, được cung cấp bởi Oracle. JDK bao gồm trình biên dịch, công cụ gỡ lỗi, trình tạo tài liệu, và các thư viện chuẩn để hỗ trợ phát triển ứng dụng.

Vai trò trong dự án:

- Cung cấp môi trường để viết và biên dịch mã nguồn Java cho ứng dụng Android.
- Hỗ trợ các tính năng lập trình hướng đối tượng như kế thừa, đóng gói, và đa hình, giúp tổ chức mã nguồn hiệu quả.
- Quản lý bộ nhớ tự động thông qua Garbage Collector, giảm thiểu lỗi liên quan đến quản lý tài nguyên.

Phiên bản sử dụng: JDK 21, đảm bảo tương thích với Android Studio và các yêu cầu của Android SDK.

Lợi ích:

- Tăng tính di động của mã nguồn, cho phép chạy trên nhiều nền tảng.
- Cộng đồng hỗ trợ lớn, nhiều tài liệu và thư viện sẵn có.
- Hỗ trợ các công cụ phát triển tích hợp như Android Studio.

Ứng dụng cụ thể: Sử dụng JDK để lập trình các thành phần logic chính của ứng dụng, bao gồm quản lý danh mục xe, xử lý giao dịch, và tương tác với cơ sở dữ liệu.

3.2. Android SDK

Mô tả: Android Software Development Kit (SDK) là bộ công cụ và thư viện do Google cung cấp để phát triển ứng dụng trên hệ điều hành Android. Android SDK bao gồm các API, trình giả lập (emulator), và các công cụ quản lý như Android Debug Bridge (ADB).

Vai trò trong dự án:

- Thiết kế giao diện: Sử dụng XML để tạo giao diện người dùng (UI) thân thiện, hỗ trợ hiển thị danh sách xe, chi tiết sản phẩm, và các form nhập liệu.
- Quản lý tài nguyên: Xử lý hình ảnh xe, biểu tượng, và các tài nguyên đa phương tiện khác.
- Kiểm thử: Sử dụng emulator và ADB để kiểm tra ứng dụng trên các phiên bản Android khác nhau (từ Android 8.0 trở lên).

Công cụ liên quan:

- Android Studio: Môi trường phát triển tích hợp (IDE) chính, hỗ trợ viết mã, thiết kế UI, và kiểm thử.
- Gradle: Công cụ tự động hóa xây dựng, quản lý thư viện và cấu hình dự án.

Lợi ích:

- Cung cấp các API phong phú để truy cập phần cứng (camera, GPS) và các tính năng hệ thống.
- Hỗ trợ phát triển nhanh với các mẫu giao diện và thư viện sẵn có.
- Tương thích với nhiều thiết bị Android, từ điện thoại đến tablet.

Ứng dụng cụ thể:

- Xây dựng các Activity và Fragment để quản lý luồng giao diện (màn hình danh sách xe, chi tiết xe, giỏ hàng).
- Tích hợp RecyclerView để hiển thị danh sách xe dưới dạng cuộn, tối ưu hiệu suất.

3.3. Java Virtual Machine (JVM)

Mô tả: Java Virtual Machine (JVM) là một máy ảo thực thi mã bytecode Java, được tạo ra từ quá trình biên dịch mã nguồn Java. JVM hoạt động như một tầng trung gian, cho phép mã Java chạy trên nhiều nền tảng mà không cần sửa đổi.

Vai trò trong dự án:

- Chuyển đổi bytecode thành mã máy phù hợp với thiết bị Android, đảm bảo ứng dụng chạy mượt mà.
- Tối ưu hóa hiệu suất thông qua Just-In-Time (JIT) compilation, giúp tăng tốc độ thực thi.
- Quản lý bộ nhớ và tài nguyên, giảm thiểu rò rỉ bộ nhớ trong ứng dụng.

Lợi ích:

- Tính di động: Mã Java viết một lần có thể chạy trên bất kỳ thiết bị nào có JVM.
- Độ tin cậy cao nhờ cơ chế xử lý ngoại lệ và kiểm tra lỗi thời gian chạy.
- Hỗ trợ đa luồng, phù hợp cho các tác vụ như tải dữ liệu xe hoặc xử lý thông báo đẩy.

Ứng dụng cụ thể:

- Thực thi các tác vụ nền như đồng bộ dữ liệu với Firebase hoặc tải hình ảnh xe từ cơ sở dữ liệu.
- Hỗ trợ các thư viện Java trong Android SDK, chẳng hạn như xử lý JSON hoặc kết nối mạng.
- Đảm bảo tính ổn định khi ứng dụng chạy trên các thiết bị Android với cấu hình khác nhau.

3.4. Firebase

Mô tả: Firebase là một nền tảng phát triển ứng dụng di động của Google, cung cấp bộ công cụ và dịch vụ dựa trên đám mây để hỗ trợ xây dựng, quản lý, và mở rộng ứng dụng. Firebase tích hợp các tính năng như cơ sở dữ liệu, xác thực, thông báo, và phân tích.

Vai trò trong dự án:

- **Firebase Authentication:**
 - + Hỗ trợ đăng nhập/đăng ký qua email, Google, hoặc số điện thoại.

- + Quản lý thông tin người dùng an toàn, tích hợp với cơ sở dữ liệu để lưu trữ hồ sơ.
- **Cloudinary:**
 - + Lưu trữ và quản lý hình ảnh xe với dung lượng lớn, hỗ trợ tối ưu hóa và phân phối nhanh thông qua CDN.
 - + Hỗ trợ xử lý ảnh như nén, thay đổi kích thước, chuyển định dạng trực tiếp trên nền tảng.
 - + Tích hợp với Firestore để liên kết ảnh với thông tin xe.

Lợi ích:

- Giảm thời gian phát triển nhờ các dịch vụ sẵn có, không cần xây dựng backend từ đầu.
- Bảo mật cao với các giao thức mã hóa và xác thực người dùng.
- Khả năng mở rộng linh hoạt, phù hợp với lượng người dùng tăng dần.
- Tích hợp dễ dàng với Android SDK và các công cụ Google khác.
- Cloudinary giúp tối ưu hóa hình ảnh để tăng tốc độ tải trang và cải thiện trải nghiệm người dùng.

Ứng dụng cụ thể:

- Tích hợp Firebase Authentication để yêu cầu người dùng đăng nhập trước khi thực hiện giao dịch.

4. UX/UI trong thiết kế ứng dụng di động

4.1. Khái niệm:

Trong thiết kế ứng dụng di động, UX (User Experience) và UI (User Interface) là hai yếu tố quan trọng quyết định chất lượng sản phẩm.

- Giao diện người dùng (UI) là phần hiển thị trực quan trên ứng dụng, bao gồm bố cục, màu sắc, phông chữ, biểu tượng,

nút bấm... UI tốt giúp ứng dụng đẹp mắt, dễ nhìn và thu hút người dùng ngay từ lần đầu sử dụng.

- Trải nghiệm người dùng (UX) đề cập đến cảm giác và mức độ hài lòng của người dùng khi thao tác trên ứng dụng. UX tốt đảm bảo người dùng dễ dàng hiểu, thao tác thuận tiện, thực hiện nhanh các hành động cần thiết như tìm kiếm, đặt xe, liên hệ chủ xe...

4.2. Tầm quan trọng trong ứng dụng thuê xe

Với một ứng dụng thuê xe giữa người dùng với nhau (C2C), UX/UI đóng vai trò then chốt trong việc:

- Tăng trải nghiệm tin cậy: Thiết kế rõ ràng, minh bạch giúp người thuê tin tưởng thông tin xe và chủ xe.
- Rút ngắn hành trình người dùng: Giảm số bước để thực hiện các thao tác chính như đặt xe, gửi yêu cầu, thanh toán.
- Tối ưu hóa thao tác di động: Đảm bảo người dùng có thể thao tác chỉ bằng một tay, kể cả khi đang di chuyển.
- Tạo lợi thế cạnh tranh: UX/UI đẹp và hiệu quả tạo cảm giác chuyên nghiệp, tăng khả năng giữ chân người dùng so với các ứng dụng đối thủ.

III. PHÂN TÍCH YÊU CẦU

1. Phân tích người dùng mục tiêu

1.1. Người dùng (Người thuê xe)

Đặc điểm nhân khẩu học:

- Độ tuổi: Chủ yếu từ 18-50 tuổi, tập trung nhiều vào nhóm 25-40 tuổi, những người có nhu cầu di chuyển linh hoạt, đi công tác, du lịch, hoặc cần xe cho các dịp đặc biệt.
- Giới tính: Cả nam và nữ, nhưng nam giới có thể chiếm tỷ lệ cao hơn do thói quen lái xe và nhu cầu sử dụng xe cá nhân.

- Thu nhập: Từ trung bình đến cao, vì họ sẵn sàng chi trả cho dịch vụ thuê xe thay vì mua xe riêng.
- Khu vực địa lý: Chủ yếu ở các thành phố lớn (Hà Nội, TP.HCM, Đà Nẵng, v.v.) hoặc khu vực du lịch (Nha Trang, Đà Lạt, Phú Quốc, v.v.), nơi nhu cầu di chuyển cao và dịch vụ thuê xe phổ biến.
- Trình độ công nghệ: Thành thạo hoặc cơ bản về sử dụng ứng dụng di động, quen thuộc với các nền tảng như Grab, Traveloka, hoặc các ứng dụng đặt dịch vụ trực tuyến.

Nhu cầu và mong đợi:

- Tiện lợi: Dễ dàng tìm kiếm, so sánh và đặt xe nhanh chóng qua ứng dụng.
- Đa dạng lựa chọn: Muốn có nhiều loại xe (xe 4 chỗ, 7 chỗ, xe sang, xe tiết kiệm, v.v.) để phù hợp với mục đích (du lịch, công việc, sự kiện).
- Giá cả minh bạch: Mong muốn biết rõ giá thuê, chi phí phát sinh (nhiên liệu, bảo hiểm, phí trễ hạn, v.v.).
- An toàn: Xe được bảo dưỡng tốt, có bảo hiểm, và người cho thuê đáng tin cậy.
- Hỗ trợ khách hàng: Cần hỗ trợ nhanh chóng từ ứng dụng (chat, gọi điện) khi gặp sự cố như hỏng xe, tai nạn, hoặc tranh chấp.
- Thanh toán linh hoạt: Hỗ trợ nhiều phương thức (tiền mặt, thẻ tín dụng, ví điện tử như Momo, ZaloPay).

Hành vi:

- Tìm kiếm xe dựa trên vị trí, giá cả, loại xe, và đánh giá từ người dùng trước.
- Thường đặt xe ngắn hạn (vài giờ, 1-2 ngày) cho các chuyến đi công tác hoặc du lịch.
- Quan tâm đến đánh giá, nhận xét về xe và người cho thuê để đảm bảo chất lượng.

- Có thể so sánh giá và dịch vụ trên nhiều nền tảng trước khi quyết định.

Thách thức:

- Lo ngại về chất lượng xe (cũ, không sạch sẽ, không an toàn).
- Lo lắng về chi phí ăn hoặc tranh chấp với người cho thuê.
- Khó khăn trong việc tìm xe phù hợp vào giờ cao điểm hoặc mùa du lịch.

1.2. Người cho thuê xe

Đặc điểm nhân khẩu học:

- Độ tuổi: Thường từ 25-55 tuổi, là những người sở hữu xe cá nhân hoặc kinh doanh dịch vụ cho thuê xe.
- Giới tính: Cả nam và nữ, nhưng nam giới có thể chiếm ưu thế do sở hữu xe và kinh nghiệm kinh doanh.
- Thu nhập: Từ trung bình đến cao, vì họ đầu tư vào xe hơi và mong muốn tạo thu nhập thụ động hoặc kinh doanh chính.
- Khu vực địa lý: Tập trung ở khu vực đô thị hoặc gần điểm du lịch, nơi nhu cầu thuê xe cao.
- Trình độ công nghệ: Có thể từ cơ bản đến trung bình, cần hướng dẫn đơn giản để đăng ký, đăng xe, và quản lý đơn hàng qua ứng dụng.

Nhu cầu và mong đợi:

- Tăng thu nhập: Muốn kiếm tiền từ xe nhàn rỗi hoặc mở rộng kinh doanh cho thuê.
- Dễ sử dụng: Giao diện ứng dụng thân thiện, dễ đăng ký xe, cập nhật tình trạng xe, và quản lý lịch đặt.
- Bảo vệ tài sản: Đảm bảo xe được sử dụng đúng cách, có hợp đồng rõ ràng, và bảo hiểm cho các rủi ro (tai nạn, hư hỏng, trộm cắp).
- Thanh toán an toàn: Muốn nhận tiền nhanh chóng, minh bạch, và an toàn qua ứng dụng.

- Hỗ trợ: Cần hỗ trợ từ admin khi có tranh chấp với người thuê (trễ hạn, hỏng xe, không thanh toán, v.v.).
- Quảng bá: Mong muốn ứng dụng giúp tiếp cận nhiều khách hàng hơn.

Hành vi:

- Đăng ký xe với thông tin chi tiết (loại xe, năm sản xuất, tình trạng, giá thuê).
- Theo dõi lịch đặt xe, xác nhận hoặc từ chối đơn đặt.
- Kiểm tra đánh giá từ người thuê để cải thiện dịch vụ.
- Có thể cạnh tranh bằng cách giảm giá hoặc cung cấp xe chất lượng cao.

Thách thức:

- Lo ngại về rủi ro hư hỏng xe, mất mát, hoặc khách thuê không đáng tin cậy.
- Khó khăn trong việc quản lý nhiều đơn đặt cùng lúc nếu không có công cụ hỗ trợ.
- Cạnh tranh với các cá nhân/doanh nghiệp cho thuê xe khác trên ứng dụng hoặc nền tảng khác.

1.3. Admin quản lý hệ thống

Đặc điểm nhân khẩu học:

- Độ tuổi: Thường từ 22-40 tuổi, là nhân viên công nghệ hoặc quản lý được đào tạo để vận hành hệ thống.
- Giới tính: Cả nam và nữ.
- Trình độ: Có kiến thức về công nghệ, quản lý dữ liệu, và chăm sóc khách hàng.
- Vai trò: Là đội ngũ nội bộ của công ty, chịu trách nhiệm duy trì hoạt động trơn tru của ứng dụng.

Nhu cầu và mong đợi:

- Quản lý người dùng: Xác minh danh tính, giấy tờ của người thuê và người cho thuê để đảm bảo an toàn.
- Giám sát giao dịch: Theo dõi đơn đặt xe, thanh toán, và đảm bảo không có gian lận.
- Hỗ trợ giải quyết tranh chấp: Xử lý khiếu nại giữa người thuê và người cho thuê (hư hỏng xe, chi phí phát sinh, v.v.).
- Bảo mật: Đảm bảo dữ liệu cá nhân, giao dịch được bảo vệ, tuân thủ quy định pháp luật (VD: Luật Bảo vệ Dữ liệu Cá nhân tại Việt Nam).
- Phân tích dữ liệu: Cân công cụ để theo dõi số liệu (số lượng xe, đơn đặt, doanh thu, đánh giá) để tối ưu hóa hệ thống.
- Cập nhật hệ thống: Quản lý danh sách xe, giá cả, và cập nhật chính sách, khuyến mãi.

Hành vi:

- Kiểm tra và phê duyệt hồ sơ của người cho thuê xe (bằng lái, đăng ký xe, bảo hiểm).
- Theo dõi hoạt động bất thường (gian lận, đánh giá tiêu cực, khiếu nại liên tục).
- Cung cấp hỗ trợ qua chat, email, hoặc điện thoại cho cả hai nhóm người dùng.
- Phân tích báo cáo để đề xuất cải tiến ứng dụng (tăng loại xe, điều chỉnh giá, v.v.).

Thách thức:

- Xử lý khối lượng lớn dữ liệu và yêu cầu hỗ trợ khi ứng dụng mở rộng.
- Đảm bảo tính công bằng khi giải quyết tranh chấp giữa người thuê và người cho thuê.
- Đối mặt với rủi ro pháp lý nếu không tuân thủ quy định về bảo mật hoặc giao dịch.

Điểm chung và khác biệt giữa các nhóm

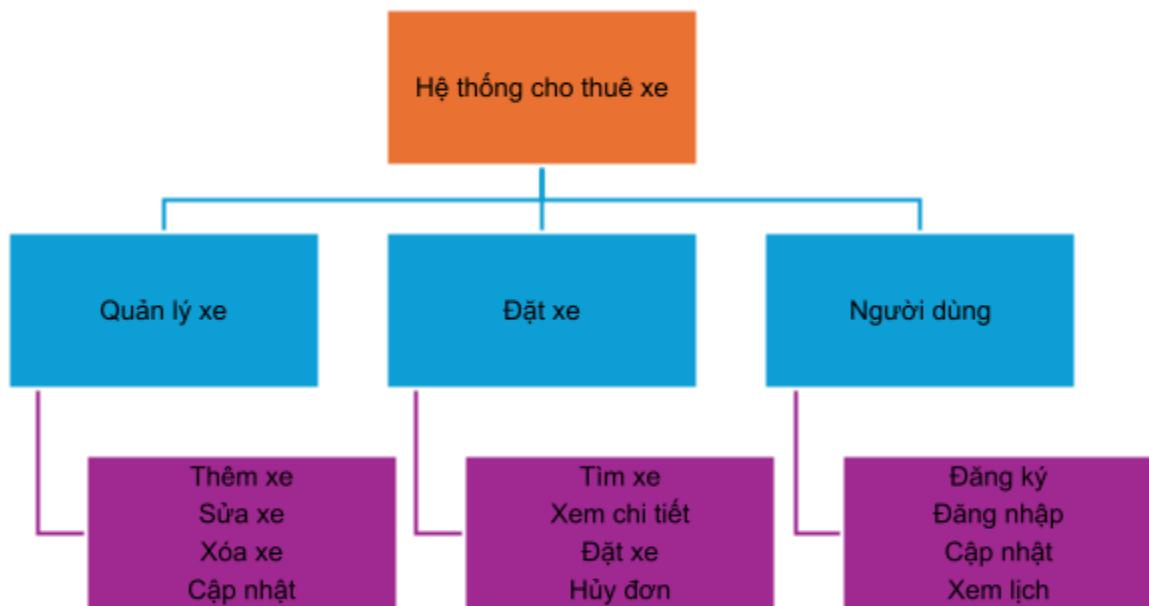
Điểm chung:

- Cần ứng dụng dễ sử dụng, giao diện thân thiện, và hỗ trợ nhanh chóng.
- Mong muốn giao dịch an toàn, minh bạch, và đáng tin cậy.
- Điều cần bảo mật thông tin cá nhân và dữ liệu giao dịch.

Điểm khác biệt:

- **Người thuê xe:** Tập trung vào giá cả, chất lượng xe, và trải nghiệm thuận tiện.
- **Người cho thuê xe:** Quan tâm đến bảo vệ tài sản, thu nhập, và khả năng tiếp cận khách hàng.
- **Admin:** Chịu trách nhiệm quản lý, giám sát, và tối ưu hóa toàn bộ hệ thống.

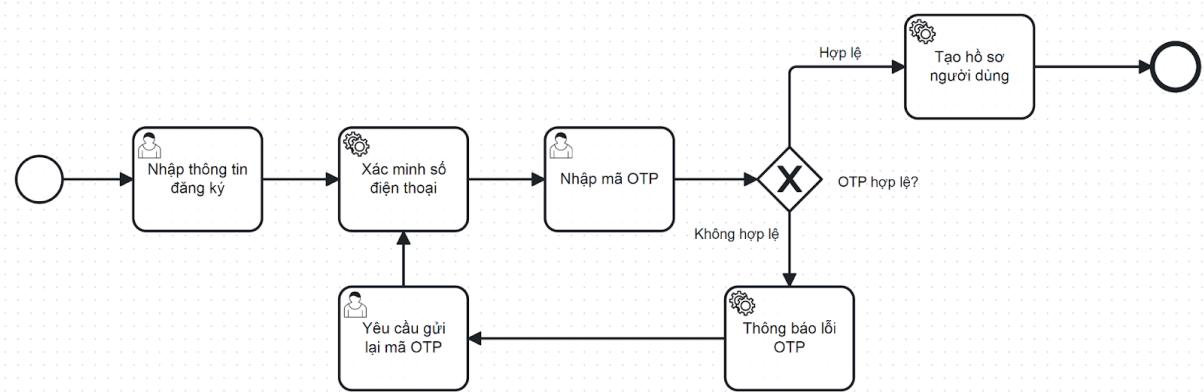
2. Tính năng cần có (chức năng chính)



3. Lưu đồ quy trình nghiệp vụ

3.1. Quy trình nghiệp vụ đối với người dùng (user)

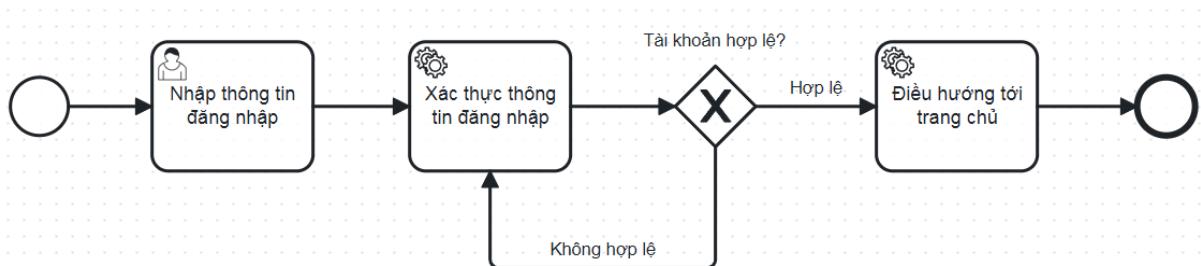
Quy trình đăng ký (Tạo hồ sơ khách hàng)



Bước	Hành động	Đối tượng thực hiện	Mô tả
1	Nhập thông tin đăng ký	Người dùng	Người dùng điền họ tên, số điện thoại, email, mật khẩu,... vào biểu mẫu đăng ký.
2	Xác minh số điện thoại	Hệ thống	Hệ thống tự động gửi mã OTP đến số điện thoại người dùng thông qua SMS hoặc dịch vụ OTP API.
3	Nhập mã OTP	Người dùng	Người dùng nhập mã OTP đã nhận để xác minh số điện thoại.
4	Gateway: OTP hợp lệ?	Hệ thống	Hệ thống kiểm tra OTP: nếu đúng → chuyển tiếp, nếu sai → thông báo lỗi.
5	Thông báo lỗi OTP	Service task	Hệ thống hiển thị thông báo OTP sai hoặc hết hạn, yêu cầu người dùng nhập lại hoặc gửi lại mã.

6	Yêu cầu gửi lại mã OTP	Người dùng	Người dùng chọn chức năng "Gửi lại mã OTP", hệ thống thực hiện lại bước gửi mã.
7	Tạo hồ sơ người dùng	Hệ thống	Khi OTP hợp lệ, hệ thống tạo hồ sơ tài khoản mới cho người dùng, lưu thông tin vào cơ sở dữ liệu.

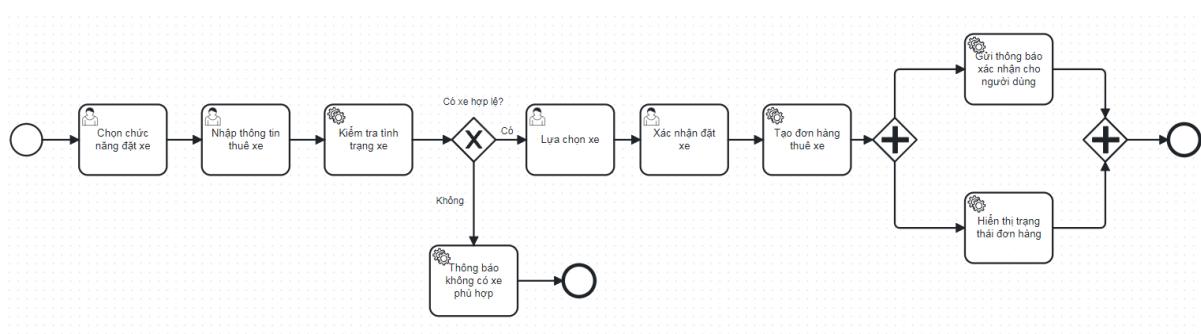
Quy trình đăng nhập



Bước	Hành động	Đối tượng thực hiện	Mô tả
1	Nhập thông tin đăng nhập	Người dùng	Input: Số điện thoại + Mật khẩu Output: Gửi thông tin đến hệ thống để xác thực
2	Xác minh thông tin đăng nhập	Hệ thống	Hệ thống tự động gửi mã OTP đến số điện thoại người dùng thông qua SMS hoặc dịch vụ OTP API.
3	Gateway(Kiểm tra tài khoản hợp lệ)	Hệ thống	Hệ thống kiểm tra OTP: nếu đúng → chuyển tiếp, nếu sai → thông báo lỗi.
4	Điều hướng tới	Hệ thống	Nếu thông tin hợp lệ, hệ thống

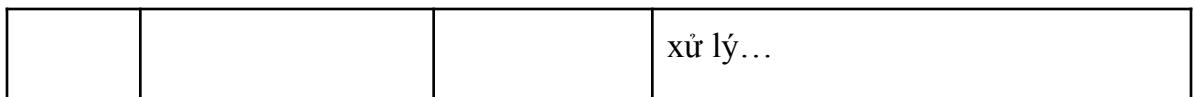
	trang chủ		chuyển người dùng đến màn hình chính.
--	-----------	--	---------------------------------------

Quy trình đặt xe

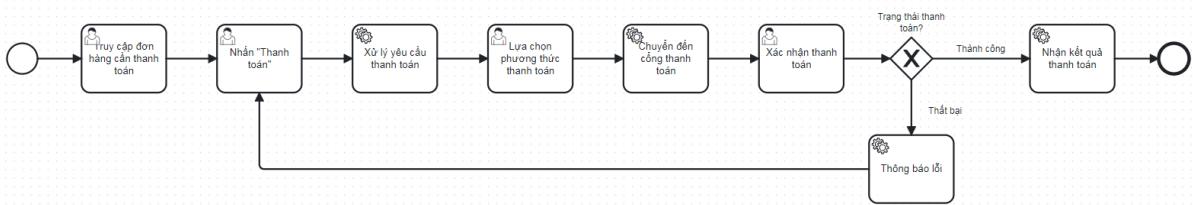


Bước	Hành động	Đối tượng thực hiện	Mô tả
1	Chọn chức năng đặt xe	Người dùng	Input: Người dùng chọn chức năng "Đặt xe" Output: Giao diện đặt xe được hiển thị với danh sách loại xe, thời gian, địa điểm
2	Nhập thông tin thuê xe	Người dùng	Input: Loại xe, ngày thuê, giờ bắt đầu, điểm đón, điểm trả Output: Thông tin được ghi nhận và chuẩn bị gửi đến hệ thống
3	Kiểm tra tình trạng xe	Hệ thống	Input: Thông tin yêu cầu thuê xe từ người dùng Output: Danh sách xe khả dụng phù hợp với yêu cầu
4	Gateway(Có xe)	Hệ thống	Hệ thống kiểm tra xem có xe phù

	phù hợp không?)		hợp với yêu cầu người dùng không
5	Thông báo không có xe phù hợp	Hệ thống	<p>Nếu không có xe khả dụng, chuyển sang nhánh thông báo "Không có xe phù hợp, vui lòng thử lại".</p> <p>Input: Không tìm thấy xe phù hợp.</p> <p>Output: Hiển thị thông báo lỗi, kết thúc quy trình.</p>
6	Lựa chọn xe	Người dùng	<p>Nếu có xe phù hợp:</p> <p>Input: Danh sách xe có sẵn</p> <p>Output: Người dùng chọn xe cụ thể để thuê</p>
7	Xác nhận đặt xe	Người dùng	<p>Input: Thông tin xe, thời gian, giá thuê</p> <p>Output: Gửi xác nhận đặt xe đến hệ thống</p>
8	Tạo đơn hàng thuê xe	Hệ thống	<p>Input: Dữ liệu xác nhận từ người dùng</p> <p>Output: Tạo đơn hàng (order ID), lưu thông tin đặt xe vào hệ thống</p>
9	Gửi thông báo xác nhận cho người dùng	Hệ thống	<p>Input: Đơn hàng vừa tạo</p> <p>Output: Hiển thị thông báo “Đặt xe thành công” + gửi chi tiết đơn hàng đến email/app của người dùng</p>
10	Hiển thị trạng thái đơn hàng	Hệ thống	<p>Input: Đơn hàng vừa tạo</p> <p>Output: Người dùng thấy trạng thái: Đã đặt – Chờ xác nhận – Đang</p>



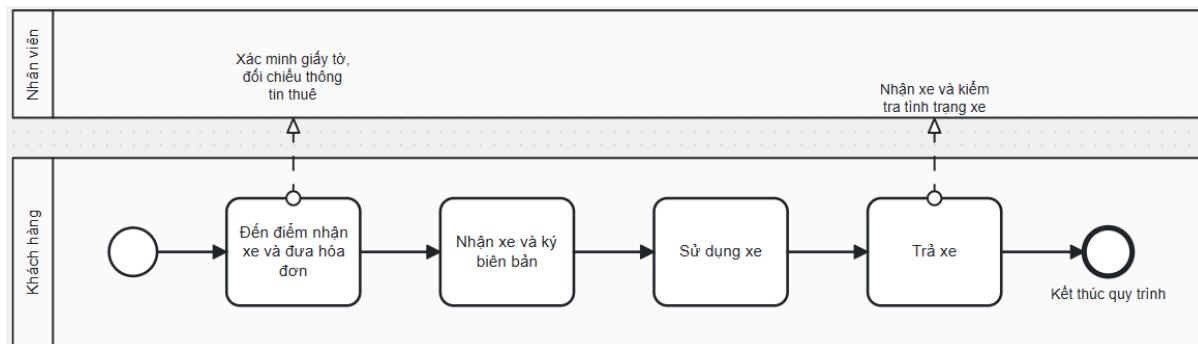
Quy trình thanh toán



Bước	Hành động	Đối tượng thực hiện	Mô tả
1	Truy cập đơn hàng cần thanh toán	Người dùng	Input: Người dùng vào mục "Đơn hàng" chọn đơn chưa thanh toán Output: Thông tin chi tiết đơn hàng hiển thị
2	Nhấn "Thanh toán"	Người dùng	Input: Đơn hàng hợp lệ Output: Gửi yêu cầu khởi tạo phiên thanh toán đến hệ thống
3	Xử lý yêu cầu thanh toán	Hệ thống	Input: Yêu cầu thanh toán từ người dùng Output: Gửi thông tin đến cổng thanh toán và khởi tạo phiên giao dịch
4	Lựa chọn phương thức thanh toán	Người dùng	Input: Các phương thức: Ví điện tử, thẻ ngân hàng, QR code... Output: Người dùng chọn phương thức cụ thể
5	Chuyển đến cổng	Hệ thống	Input: Phương thức được chọn +

	thanh toán		thông tin giao dịch Output: Giao diện thanh toán từ bên thứ ba (Momo, VNPay, ZaloPay...)
6	Xác nhận thanh toán	Người dùng	Input: Tùy theo phương thức (OTP, mã QR, số thẻ...) Output: Gửi xác nhận thanh toán
7	Gateway (Trạng thái thanh toán?)	Hệ thống	Hệ thống kiểm tra xem giao dịch có thực hiện thành công hay không
8	Thông báo lỗi	Hệ thống	Input: Giao dịch thất bại Output: Thông báo lỗi, vui lòng kiểm tra lại thông tin thanh toán
9	Nhận kết quả thanh toán	Hệ thống	Input: Trạng thái giao dịch từ cổng thanh toán Output: Trạng thái "Thành công"

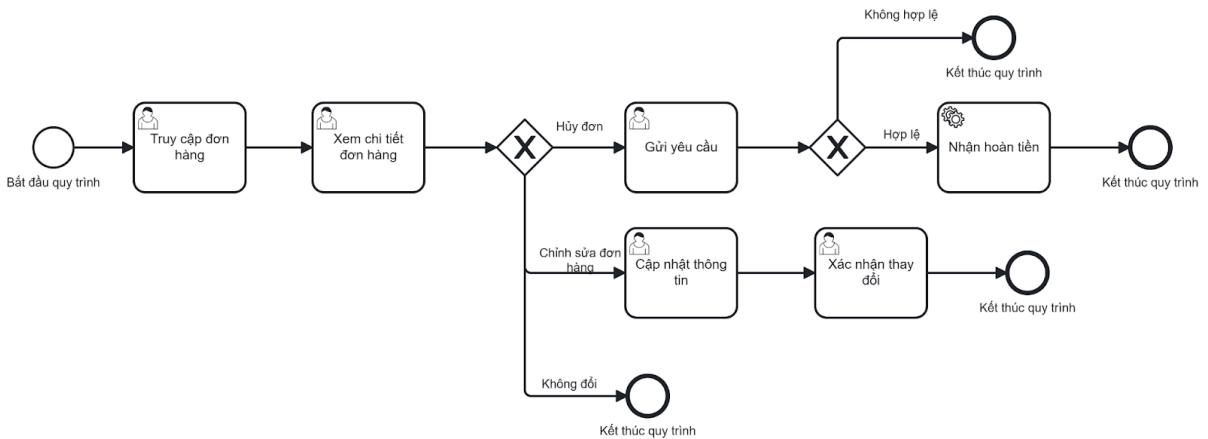
Quy trình Nhận xe và Trả xe



Bước	Hành động	Đối tượng thực hiện	Mô tả

1	Đến điểm nhận xe và đưa hóa đơn	Người dùng	Người dùng đến điểm nhận xe và xuất trình những giấy tờ liên quan
2	Xác minh giấy tờ, đổi chiểu thông tin thuê	Quản trị viên	Nhân viên xác minh giấy tờ, đổi chiểu thông tin và giao xe cho khách
3	Nhận xe và ký biên bản	Người dùng	Người dùng nhận xe và ký những biên bản liên quan và để lại cẩn cước
4	Sử dụng xe	Người dùng	Người dùng sử dụng xe trong thời gian thuê xe
5	Trả xe	Người dùng	Người dùng trả xe và nhận lại cẩn cước

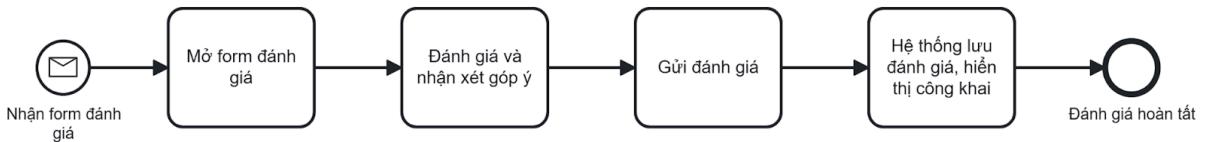
Quy trình quản lý đơn đặt xe



Bước	Hành động	Đối tượng thực hiện	Mô tả
1	Truy cập đơn hàng	Người dùng	Người dùng truy cập đơn hàng của mình để kiểm tra thông tin

2	Xem chi tiết đơn hàng	Người dùng	Người dùng chọn xem chi tiết đơn hàng
3	Giveaway (Hành động)		
4	Gửi yêu cầu	Người dùng	Nếu muốn hủy đơn hàng, chọn gửi yêu cầu hủy đơn hàng
5	Nhận hoàn tiền	Người dùng	Nếu muốn hủy đơn hàng và đã thanh toán, nhận hoàn tiền
6	Cập nhật thông tin	Người dùng	Nếu muốn cập nhật thông tin đơn hàng, chọn cập nhật thông tin
7	Xác nhận thay đổi	Người dùng	Chọn xác nhận thay đổi sau khi cập nhật thông tin đơn hàng

Quy trình đánh giá và góp ý

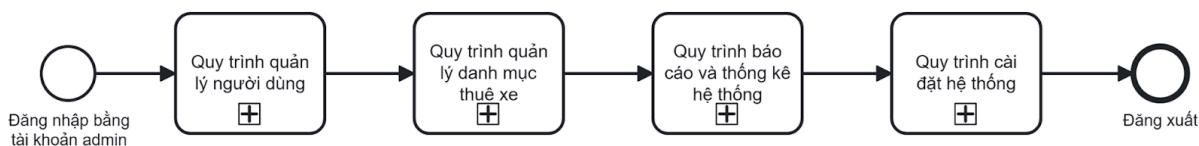


Bước	Hành động	Đối tượng thực hiện	Mô tả
1	Mở form đánh giá	Người dùng	Người dùng truy cập vào form đánh giá của hệ thống
2	Đánh giá và nhận xét góp ý	Người dùng	Người dùng nêu đánh giá, nhận xét và những góp ý
3	Gửi đánh giá	Người dùng	Xác nhận gửi đánh giá

4	Hệ thống lưu đánh giá, hiển thị công khai	Hệ thống	Hệ thống lưu đánh giá và quản trị viên sẽ duyệt đánh giá và hiển thị công khai.
---	---	----------	---

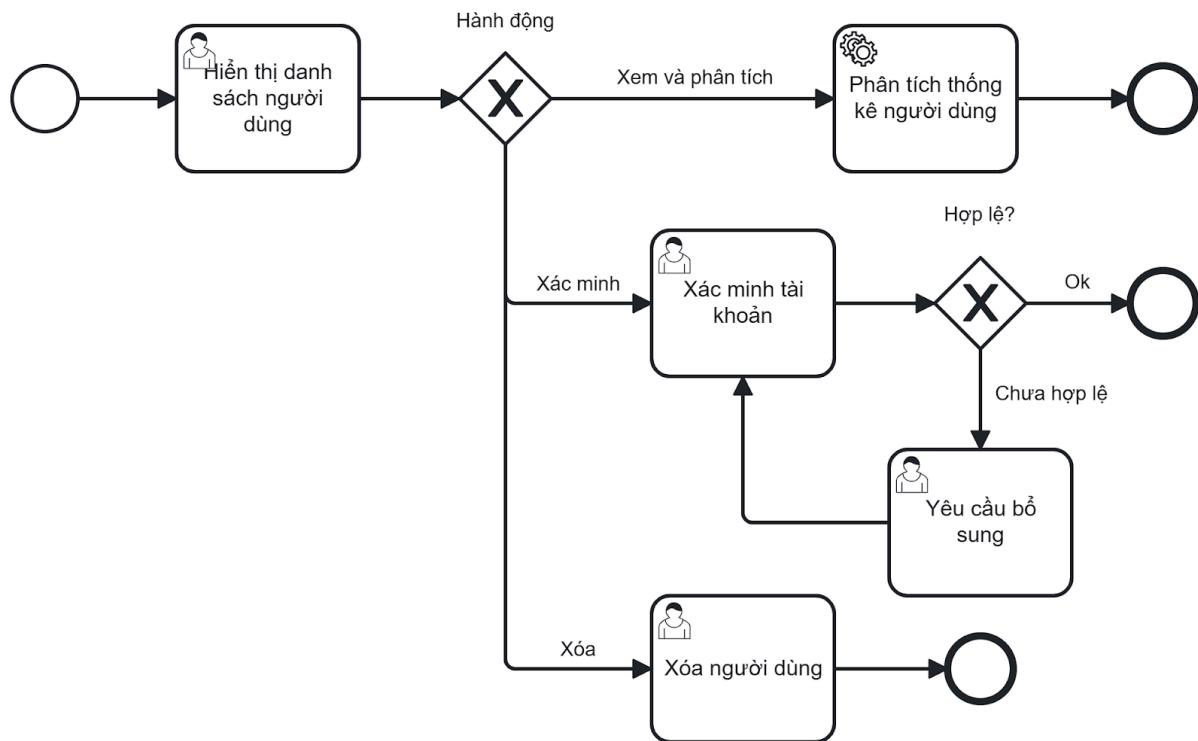
3.2. Quy trình nghiệp vụ đối với quản trị viên (admin)

Quy trình tổng quát của admin



Bước	Hành động	Đối tượng thực hiện	Mô tả
1	Đăng nhập bằng tài khoản admin	Quản trị viên	
2	Quy trình quản lý người dùng	Quản trị viên	Nhân viên chịu trách nhiệm tiến hành Quy trình quản lý người dùng
3	Quy trình quản lý danh mục thuê xe	Quản trị viên	Nhân viên chịu trách nhiệm tiến hành Quy trình quản lý danh mục thuê xe
4	Quy trình báo cáo và thống kê hệ thống	Quản trị viên	Nhân viên chịu trách nhiệm tiến hành Quy trình báo cáo và thống kê hệ thống
5	Quy trình cài đặt hệ thống	Quản trị viên	Nhân viên chịu trách nhiệm tiến hành Quy trình cài đặt hệ thống

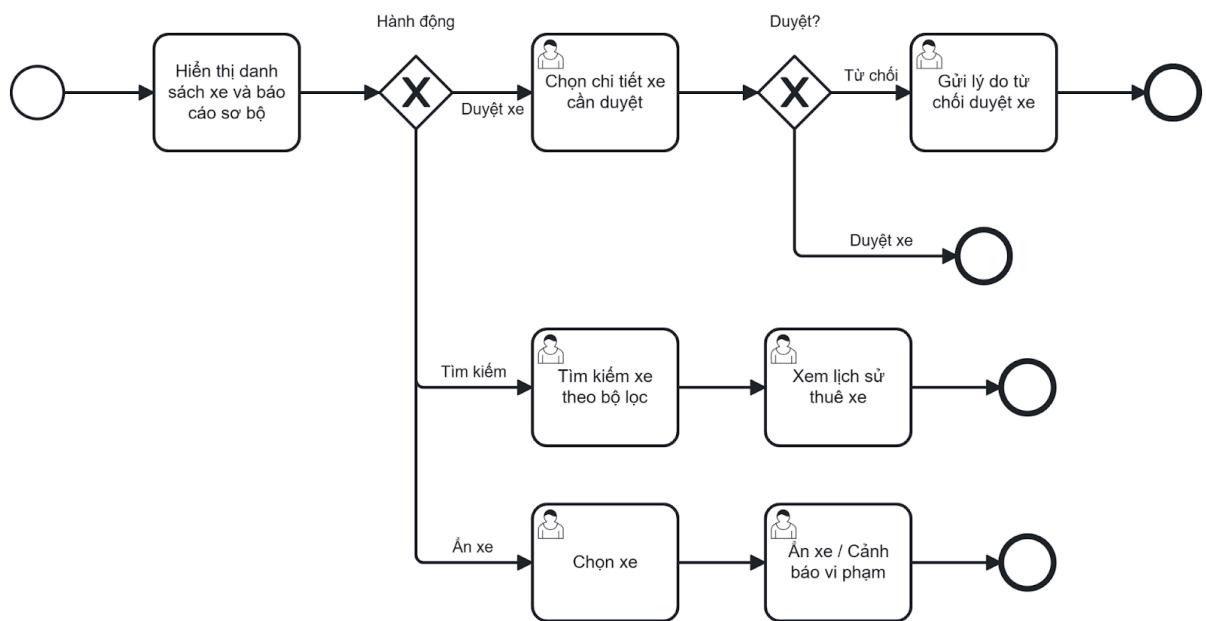
Quy trình quản lý người dùng



Bước	Hành động	Đối tượng thực hiện	Mô tả
1	Hiển thị danh sách người dùng	Quản trị viên	Nhân viên truy cập vào trang quản lý người dùng, trang sẽ hiển thị danh sách người dùng
2	Gateway (Hành động)	Quản trị viên	Xem và phân tích / Xác minh tài khoản / Xóa
3	Phân tích thống kê người dùng	Hệ thống	Xem và phân tích thống kê người dùng
4	Xác minh tài khoản	Quản trị viên	Xác minh tài khoản cho người dùng qua gateway hợp lệ
5	Gateway (Hợp lệ?)	Quản trị viên	Xét xem tài khoản có hợp lệ hay không. Nếu hợp lệ thì chọn xác

			minh và kết thúc quy trình
6	Yêu cầu bổ sung	Quản trị viên	Nếu không hợp lệ, yêu cầu người dùng bổ sung thông tin để xác minh tài khoản lại
7	Xóa người dùng	Quản trị viên	Xác nhận xóa người dùng

Quy trình quản lý danh mục thuê xe



Bước	Hành động	Đối tượng thực hiện	Mô tả
1	Hiển thị danh sách xe và báo cáo sơ bộ	Hệ thống	Nhân viên truy cập vào trang quản lý danh mục thuê xe, trang sẽ hiển thị danh sách xe và báo cáo tổng quát xe
2	Gateway (Hành động)	Quản trị viên	Duyệt xe / Tìm kiếm / Án xe

3	Chọn chi tiết xe cần duyệt	Quản trị viên	Chọn chi tiết xe cần duyệt và xem xét theo gateway duyệt xe
4	Gateway (Duyệt?)	Quản trị viên	Kiểm tra đầy đủ thông tin và ra quyết định duyệt xe hay từ chối. Nếu chọn duyệt xe, quy trình kết thúc
5	Gửi lý do từ chối duyệt xe	Quản trị viên	Nếu chọn từ chối, gửi lý do từ chối duyệt xe chi tiết cho người dùng
6	Tìm kiếm xe theo bộ lọc	Quản trị viên	Tìm kiếm xe theo các bộ lọc (loại xe, trạng thái, đánh giá)
7	Xem lịch sử thuê xe	Quản trị viên	Nhân viên có thể xem chi tiết lịch sử thuê xe
8	Chọn xe	Quản trị viên	Nếu muốn ấn xe hoặc cảnh báo vi phạm, chọn xe cần hành động trên
9	Ấn xe / Cảnh báo vi phạm	Quản trị viên	Chọn ấn xe / cảnh báo vi phạm

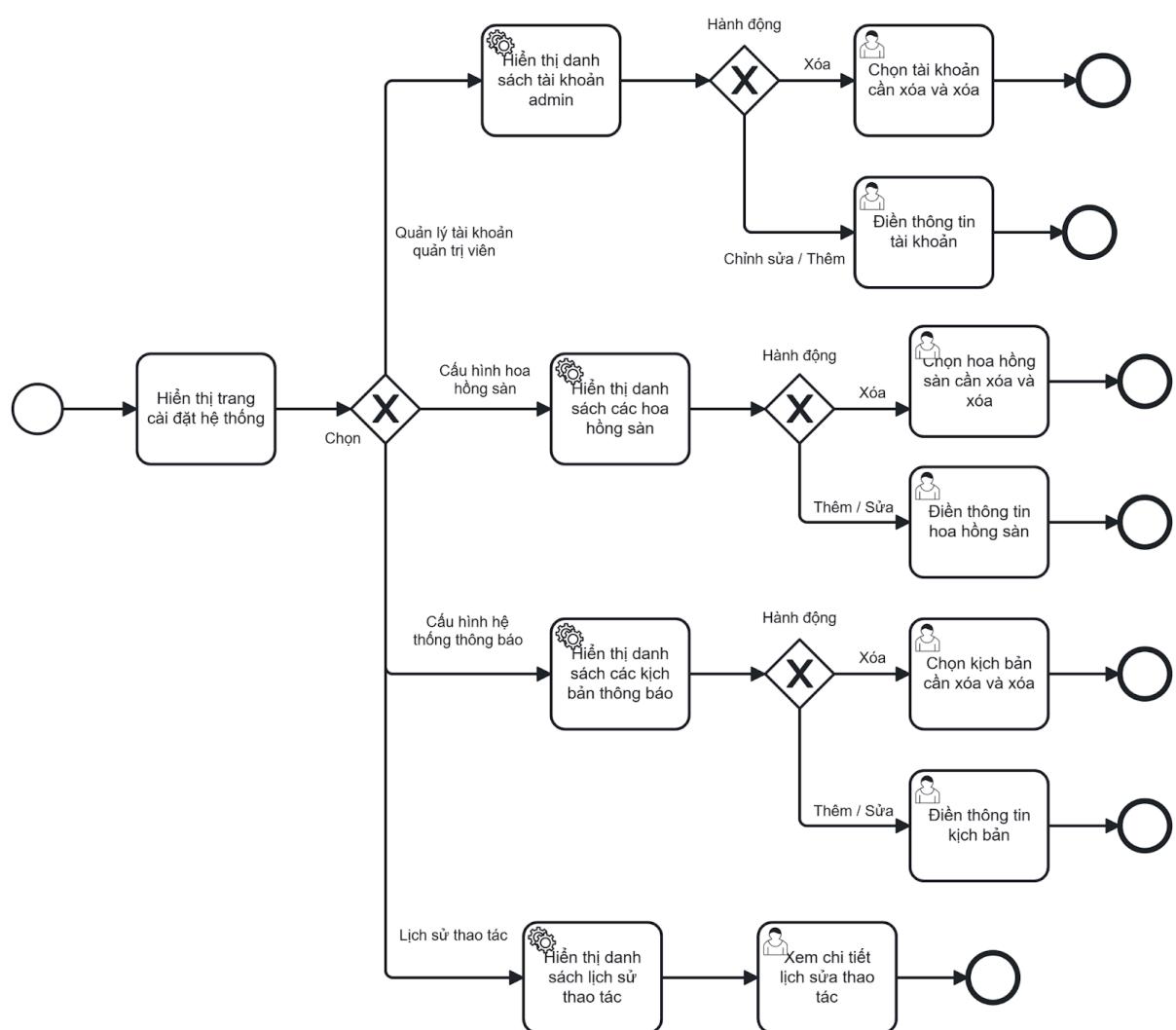
Quy trình báo cáo và thống kê hệ thống



Bước	Hành động	Đối tượng thực hiện	Mô tả
1	Hiển thị tổng quan báo cáo thống kê	Hệ thống	Hệ thống hiển thị tổng quan báo cáo thống kê

2	Thống kê người dùng	Quản trị viên	Chọn thống kê người dùng và xem chi tiết thống kê, có bộ lọc thời gian
3	Thống kê doanh thu	Quản trị viên	Chọn thống kê doanh thu và xem chi tiết thống kê, có bộ lọc thời gian và khu vực

Quy trình cài đặt hệ thống



Bước	Hành động	Đối tượng thực hiện	Mô tả

1	Hiển thị trang cài đặt hệ thống	Hệ thống	Hiển thị trang chính của cài đặt hệ thống, bao gồm các trang con liên quan
2	Gateway (Chọn)	Quản trị viên	Quản lý tài khoản quản trị viên / Cấu hình hoa hồng sàn / Cấu hình hệ thống thông báo / Lịch sử thao tác
3	Hiển thị danh sách tài khoản admin	Hệ thống	Nhân viên có trách nhiệm quản lý tài khoản admin khi chọn Quản lý tài khoản quản trị viên, trang sẽ hiển thị danh sách tài khoản admin
4	Gateway (Hành Động)	Quản trị viên	Xóa / Chính sửa / Thêm
5	Chọn tài khoản cần xóa và xóa	Quản trị viên	Nhân viên chịu trách nhiệm có thể chọn tài khoản cần xóa và xóa ra khỏi hệ thống
6	Điền thông tin tài khoản	Quản trị viên	Nhân viên chịu trách nhiệm có thể chỉnh sửa hoặc thêm tài khoản admin qua việc điền / sửa thông tin tài khoản đó
7	Hiển thị danh sách các hoa hồng sàn	Hệ thống	Nhân viên có trách nhiệm quản lý Cấu hình hoa hồng sàn, chọn và trang sẽ hiển thị danh sách các hoa hồng sàn
8	Gateway (Hành Động)	Quản trị viên	Xóa / Chính sửa / Thêm
9	Chọn hoa hồng sàn	Quản trị viên	Nhân viên chịu trách nhiệm có thể

	cần xóa và xóa		chọn hoa hồng sàn cần xóa và xóa ra khỏi hệ thống
10	Điền thông tin hoa hồng sàn	Quản trị viên	Nhân viên chịu trách nhiệm có thể chỉnh sửa hoặc thêm thông tin hoa hồng sàn qua việc điền / sửa thông tin hoa hồng sàn đó
11	Hiển thị danh sách các kịch bản thông báo	Hệ thống	Nhân viên có trách nhiệm quản lý cấu hình kịch bản thông báo, khi chọn, màn hình sẽ hiển thị danh sách các kịch bản thông báo đã được cài
12	Gateway (Hành Động)	Quản trị viên	Xóa / Chính sửa / Thêm
13	Chọn kịch bản cần xóa và xóa	Quản trị viên	Nhân viên chịu trách nhiệm có thể chọn kịch bản thông báo cần xóa và xóa ra khỏi hệ thống
14	Điền thông tin kịch bản	Quản trị viên	Nhân viên chịu trách nhiệm có thể chỉnh sửa hoặc thêm kịch bản thông báo qua việc điền / sửa thông tin kịch bản theo form trên màn hình
15	Hiển thị danh sách lịch sử thao tác	Hệ thống	Tài khoản admin có quyền, có thể xem lịch sử thao tác của các tài khoản admin để dễ dàng trong việc quản lý. Khi chọn sẽ hiển thị danh sách các lịch sử thao tác, có thể lọc theo tài khoản admin, hành động và thời gian
16	Xem chi tiết lịch sử	Quản trị viên	Có thể xem lịch sử thao tác chi tiết của

	thao tác		1 tài khoản admin
--	----------	--	-------------------

4. Use-case diagram

4.1. Tổng quát



4.2. Đặc tả

Tên chức năng	Chức năng
Đăng ký	Cho phép người dùng tạo tài khoản mới bằng email, mật khẩu và thông tin cá nhân.
Đăng nhập	Xác thực tài khoản người dùng để truy cập vào hệ thống.
Tìm kiếm xe	Cho phép người dùng tìm xe theo vị trí, ngày giờ, loại xe.
Xem chi tiết xe	Hiển thị thông tin chi tiết của xe bao gồm hình ảnh, mô tả, giá thuê.
Đặt xe	Người dùng chọn xe, chọn thời gian thuê, địa điểm nhận/trả và đặt xe.

Chọn phương thức thanh toán	Cung cấp các phương thức như ví điện tử, thẻ ngân hàng, COD nếu có.
Hủy đặt xe	Hủy đơn thuê xe trước thời gian thuê nếu cần thiết.
Đánh giá / chấm điểm	Sau khi thuê xe, người dùng có thể để lại nhận xét và đánh giá chất lượng.
Xem lịch sử đặt xe	Hiển thị danh sách các đơn thuê xe trước đó của người dùng.
Thêm / xóa xe (Admin)	Admin có quyền thêm hoặc xóa phương tiện khỏi hệ thống.
Sửa xe (Admin)	Chỉnh sửa thông tin về xe như giá, hình ảnh, tình trạng.
Cập nhật trạng thái xe (Admin)	Cập nhật tình trạng xe: đang cho thuê, sẵn sàng, bảo trì.
Kiểm duyệt đơn đặt xe (Admin)	Admin phê duyệt hoặc từ chối các yêu cầu thuê xe (nếu cần quy trình duyệt).
Xác thực người dùng (System)	Hệ thống tự động xác thực tài khoản khi đăng ký/đăng nhập.
Xử lý thanh toán (System)	Kết nối với cổng thanh toán để xử lý các giao dịch đặt xe.
Gửi email (System)	Gửi email xác nhận, thông báo đơn đặt xe, hoặc thay đổi trạng thái đơn hàng.

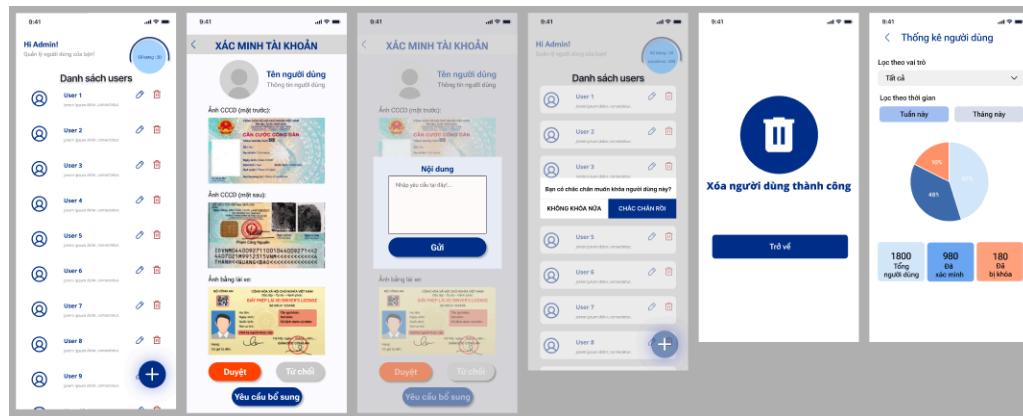
IV. THIẾT KẾ ỨNG DỤNG

1. Giao diện người dùng (UI): sơ đồ màn hình, bộ cục

1.1. Admin

1.1.1. Quản lý người dùng

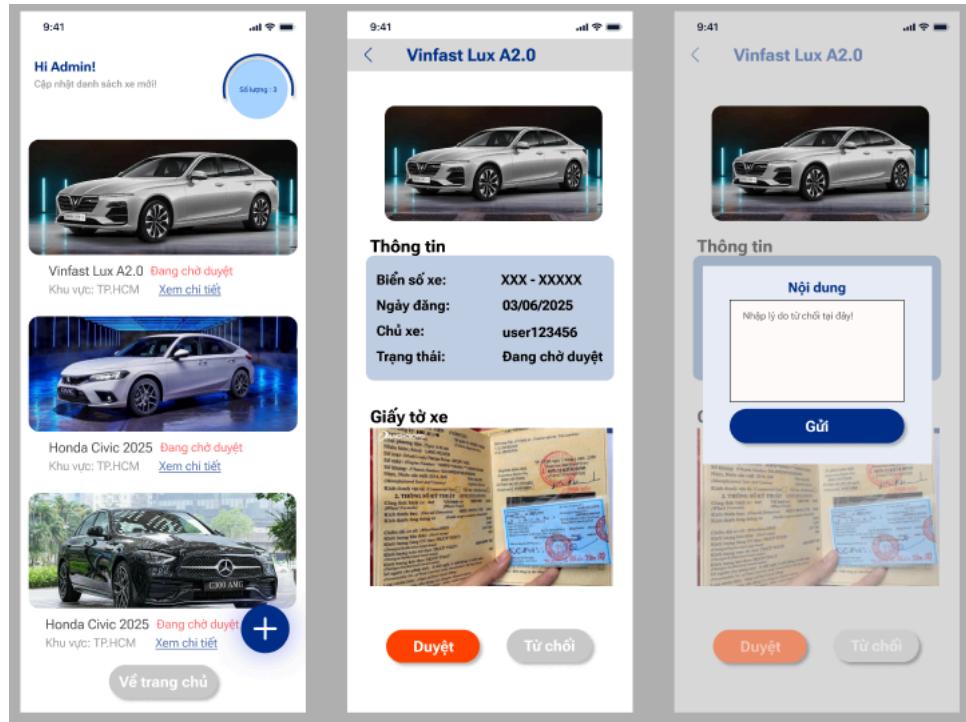
Giao diện quản lý người dùng dành cho admin cho phép theo dõi danh sách tài khoản, xác minh thông tin cá nhân, yêu cầu bổ sung giấy tờ, cũng như thực hiện thao tác khóa hoặc xóa người dùng. Admin có thể dễ dàng kiểm tra ảnh giấy tờ tùy thân, phê duyệt hoặc từ chối xác minh chỉ với một nút bấm. Giao diện cũng hỗ trợ hiển thị thống kê tổng quan về số lượng người dùng, trạng thái xác minh và tài khoản bị khóa, giúp quản lý hệ thống hiệu quả và minh bạch.



1.1.2. Quản lý danh mục thuê xe

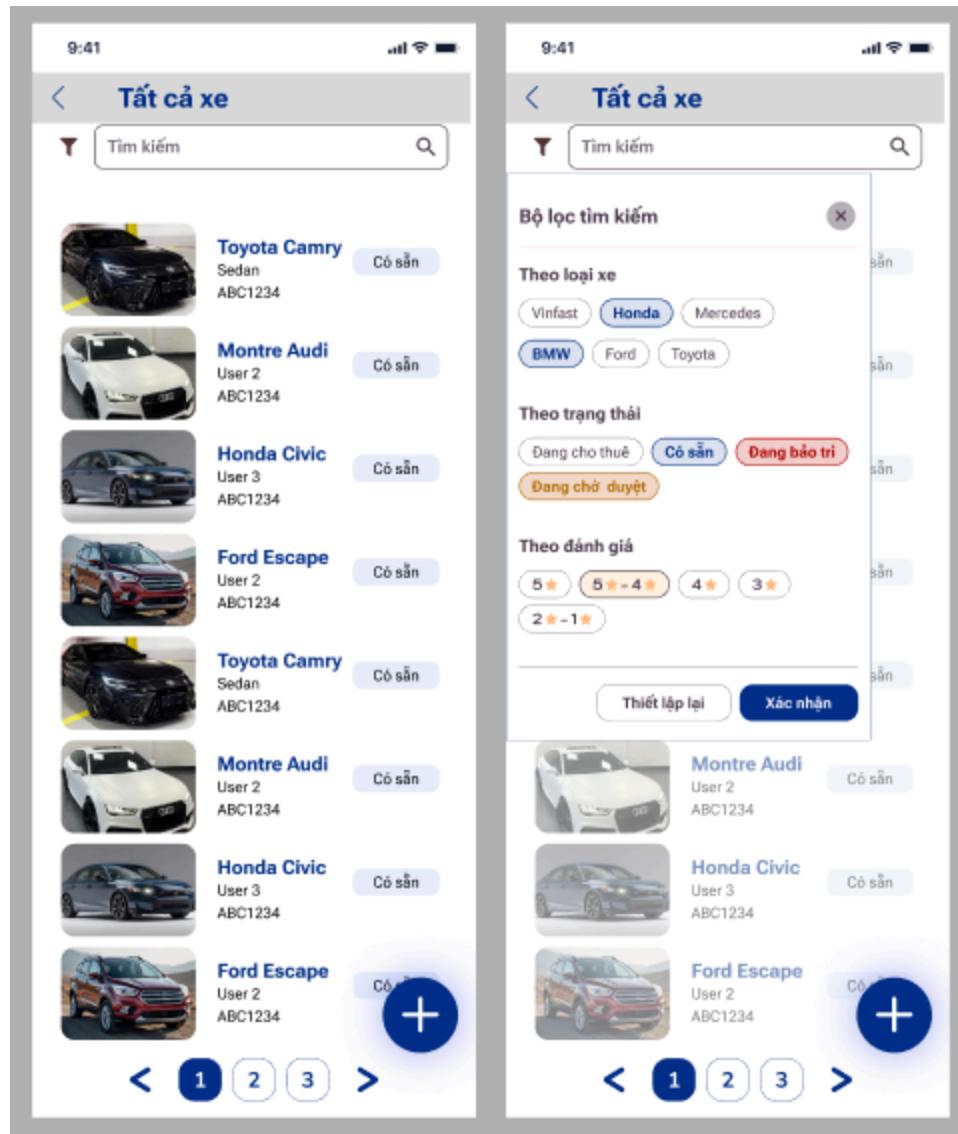
1.1.2.1. Quản lý xe chờ duyệt

Giao diện giúp admin kiểm tra thông tin xe mới đăng ký và thực hiện duyệt hoặc từ chối.



1.1.2.2. Danh sách và tìm kiếm xe

Cho phép admin theo dõi toàn bộ xe trên hệ thống, tìm kiếm, lọc và phân trang để quản lý dễ dàng.



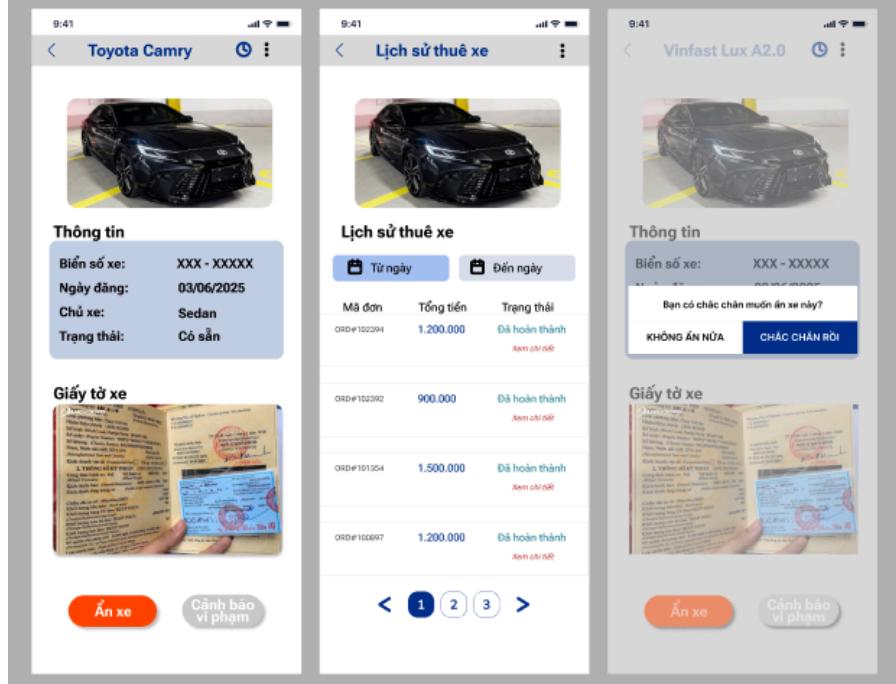
1.1.2.3. Dashboard quản lý xe

Giao diện tổng quan giúp admin nắm được số lượng xe theo trạng thái, tần suất thuê xe, và danh sách xe mới cập nhật.



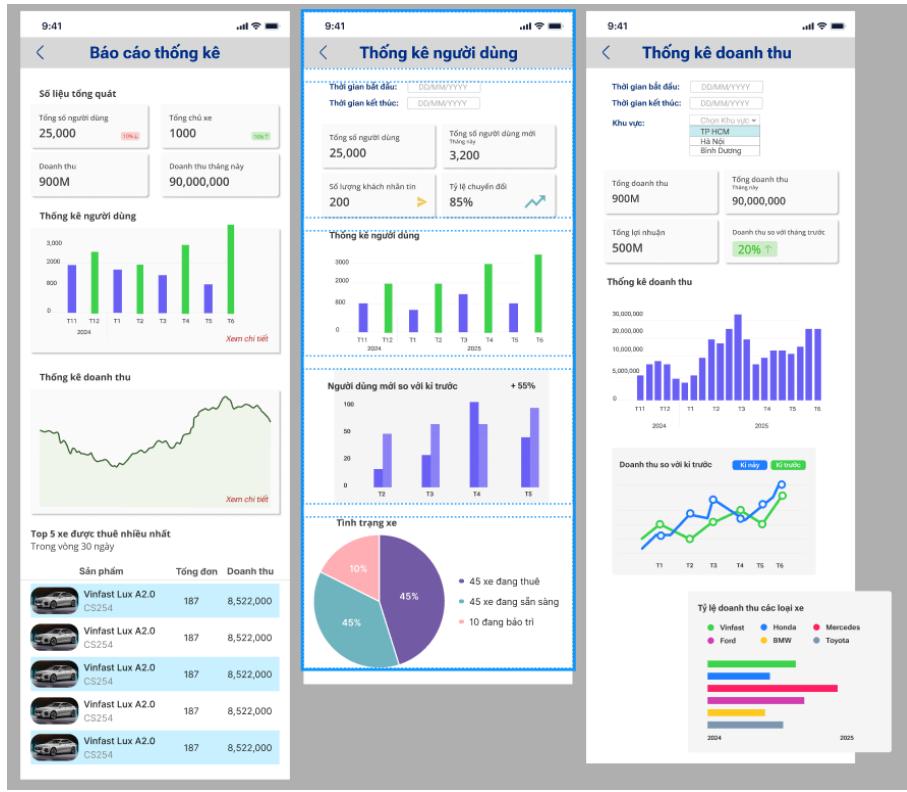
1.1.2.4. Quản lý xe đang hoạt động

Giao diện chi tiết cho từng xe đã được duyệt, cho phép theo dõi giấy tờ và thực hiện các thao tác xử lý vi phạm. Theo dõi các giao dịch thuê xe của từng phương tiện, giúp đảm bảo minh bạch và kiểm soát lịch sử sử dụng.



1.1.3. Báo cáo & thống kê hệ thống

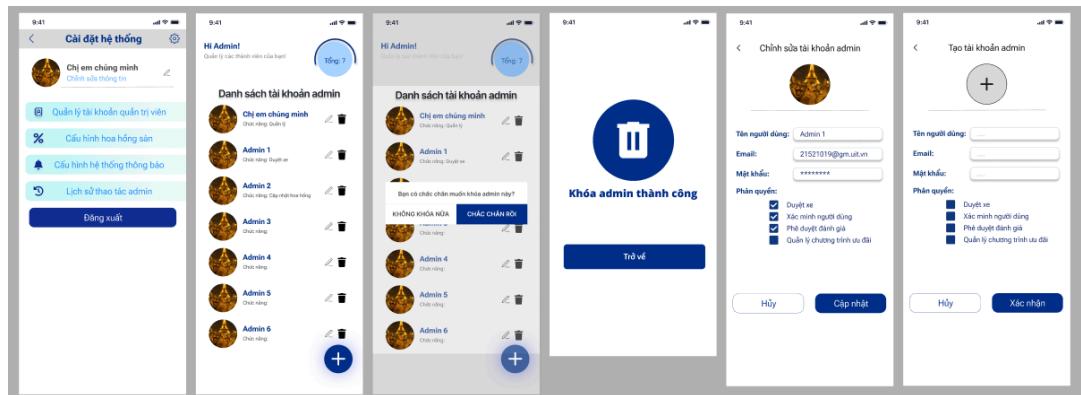
Giao diện báo cáo và thống kê giúp quản trị viên theo dõi toàn diện hiệu suất hoạt động của hệ thống. Trang "Báo cáo thống kê" hiển thị tổng quan các chỉ số chính như số người dùng, số lượng xe, tổng doanh thu và doanh thu theo tháng. Các biểu đồ cột và đường cung cấp cái nhìn rõ ràng về xu hướng người dùng, doanh thu theo thời gian và tỷ lệ tăng trưởng. Ngoài ra, hệ thống còn hỗ trợ phân tích theo khu vực, so sánh dữ liệu với kỳ trước và theo dõi tình trạng xe hiện tại. Thống kê về xe được thuê nhiều nhất trong 30 ngày cũng giúp định hướng chiến lược kinh doanh hiệu quả hơn. Tất cả thông tin được thể hiện dưới dạng đồ họa sinh động, dễ theo dõi và hỗ trợ ra quyết định nhanh chóng.



1.1.4. Cài đặt hệ thống

1.1.4.1. Quản lý tài khoản

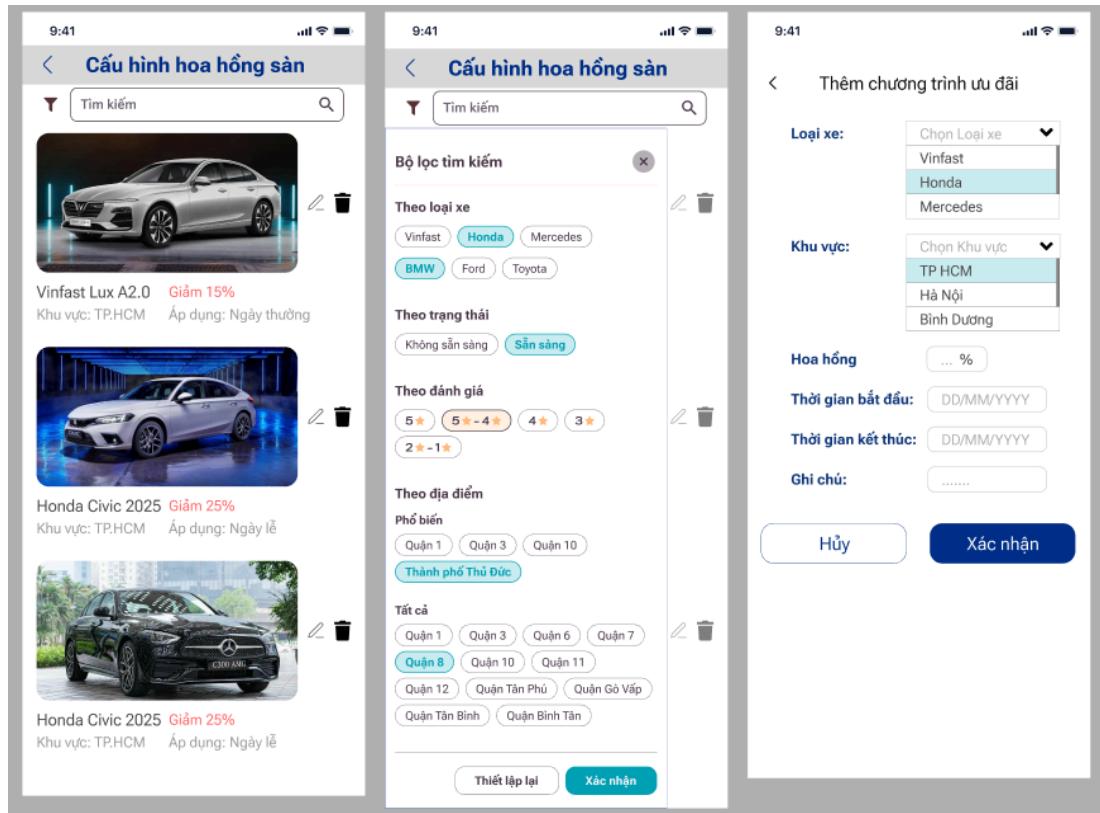
Giao diện này cho phép quản trị viên xem danh sách admin, thêm mới, chỉnh sửa thông tin, phân quyền và khóa tài khoản. Các chức năng được bố trí rõ ràng, hỗ trợ quản lý người dùng quản trị hiệu quả và an toàn.



1.1.4.2. Quản lý hoa hồng sản phẩm

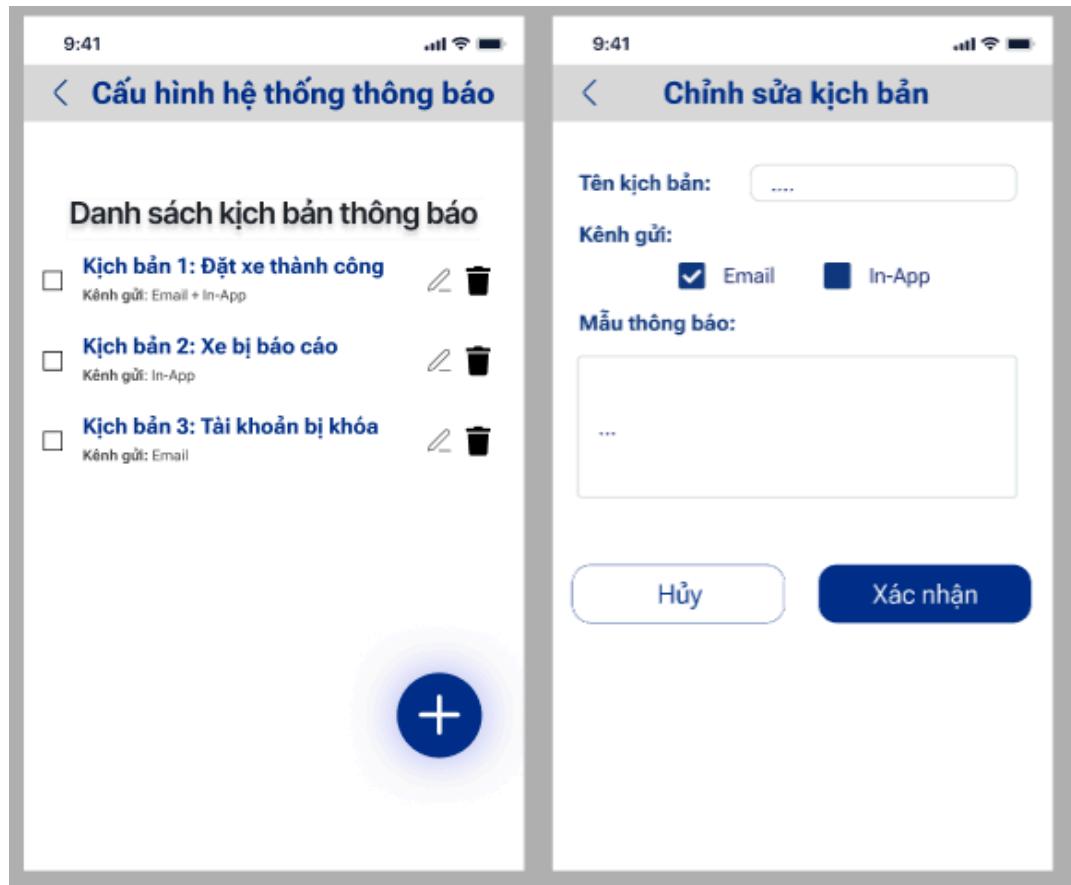
Cho phép admin cấu hình mức hoa hồng cho từng loại xe, lọc theo các tiêu chí như khu vực, thời gian hiệu lực và hãng xe.

Ngoài ra, admin cũng có thể thêm các chương trình ưu đãi với thời gian và phạm vi áp dụng cụ thể.



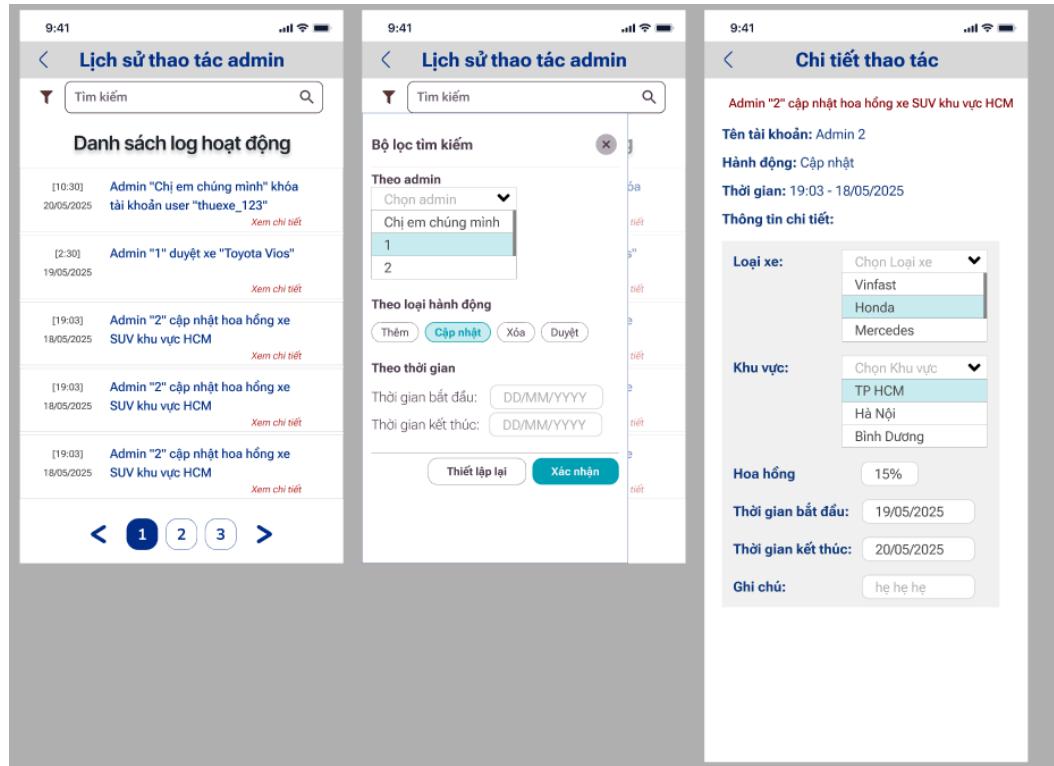
1.1.4.3. Cấu hình hệ thống thông báo

Giúp admin quản lý các kịch bản gửi thông báo đến người dùng thông qua email hoặc thông báo trong app. Các kịch bản có thể được thêm mới hoặc chỉnh sửa để phù hợp với từng tình huống cụ thể như ưu đãi mới, thay đổi dịch vụ,...



1.1.4.4. Lịch sử thao tác admin

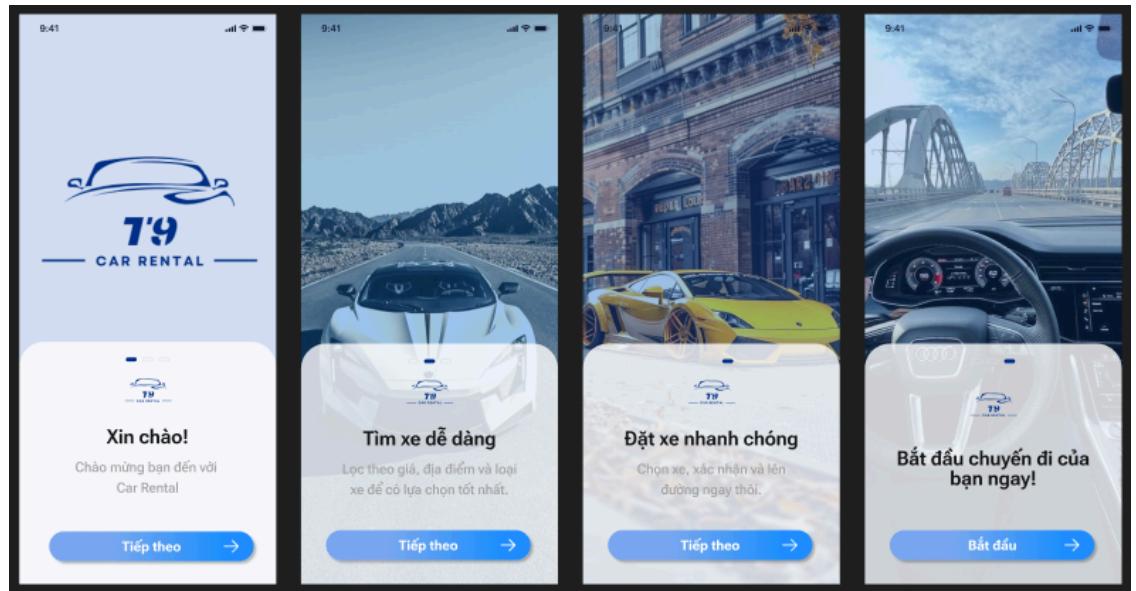
Cho phép theo dõi toàn bộ hành động của các admin trên hệ thống, từ việc thêm ưu đãi đến chỉnh sửa thông báo. Hệ thống ghi nhận chi tiết từng thao tác, thời gian, và thông tin cụ thể liên quan, giúp đảm bảo tính minh bạch và kiểm soát tốt hơn.



1.2. Account

1.2.1. Splash

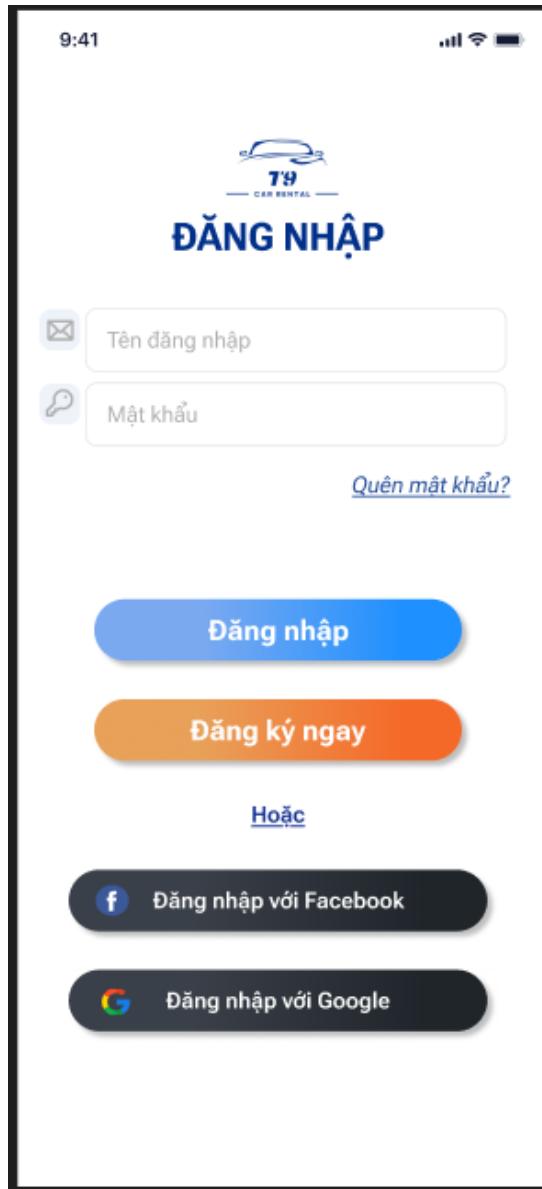
Giới thiệu về T9 Car Rental, chức năng nổi bật và bắt đầu di chuyển đến màn hình đăng nhập



1.2.2. Sign in

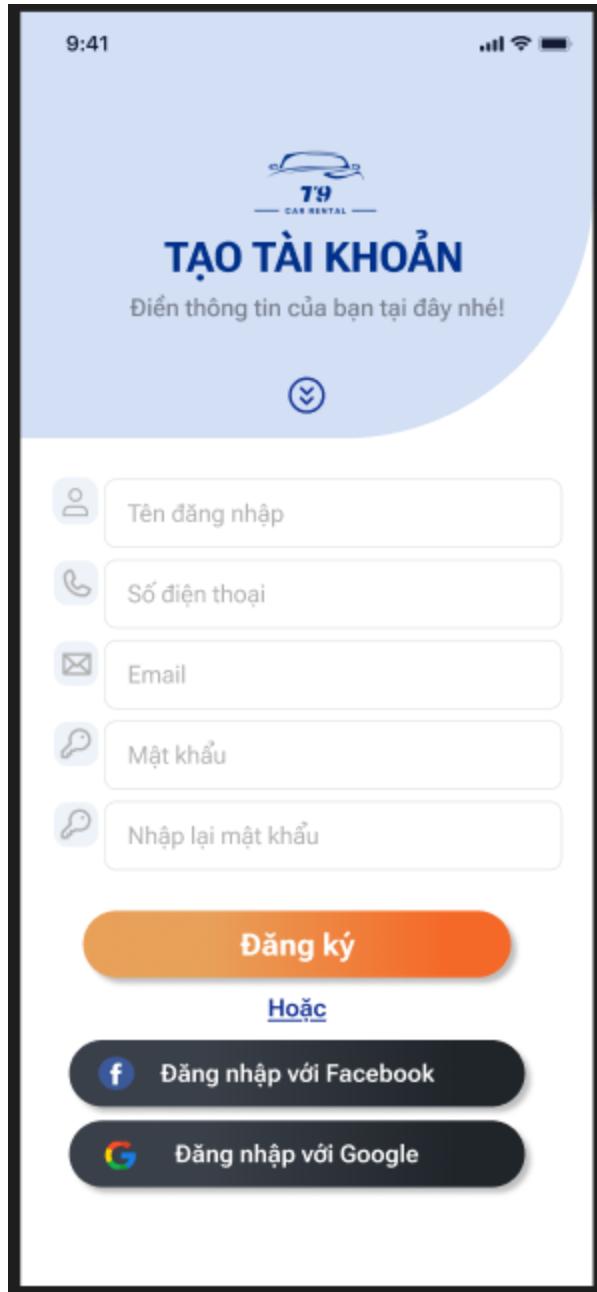
Cho phép người dùng đăng nhập vào hệ thống. Có thể đăng

nhập bằng Google hoặc Facebook. Ngoài ra, có thêm liên kết “Quên mật khẩu?” và “Tạo tài khoản mới” để hỗ trợ người dùng.



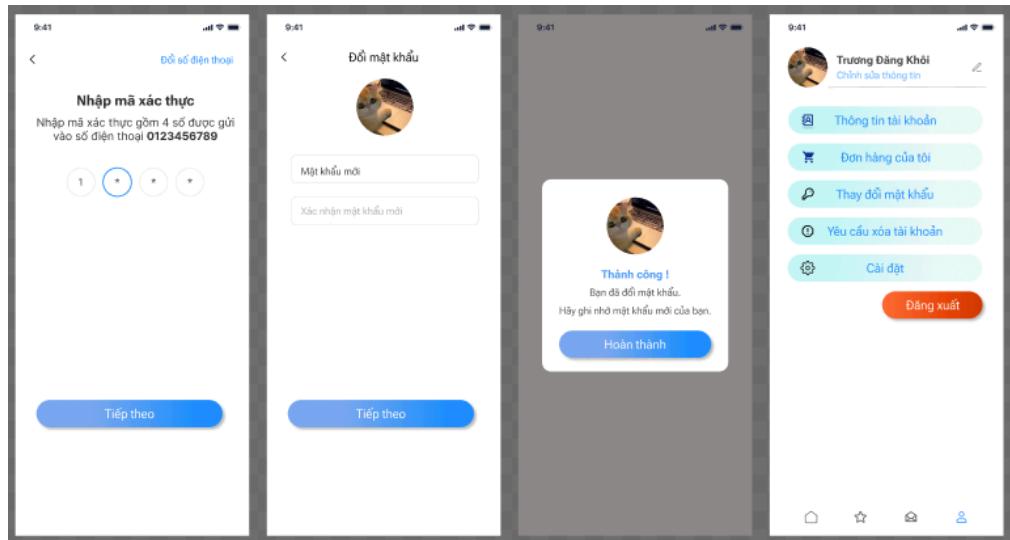
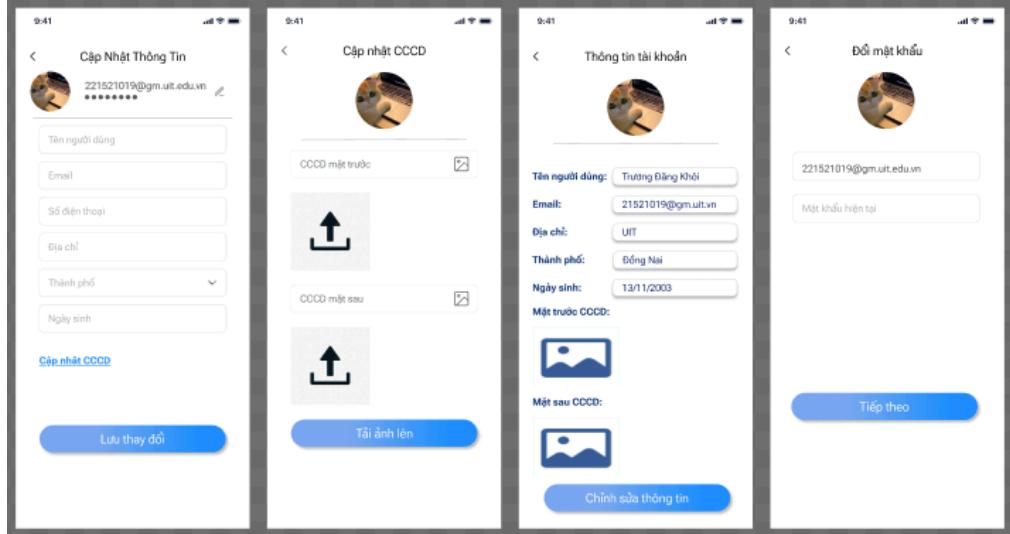
1.2.3. Sign up

Cho phép người dùng mới tạo tài khoản để thuê xe trên nền tảng T9 Car Rental. Màn hình Đăng ký cho phép người dùng nhập thông tin cơ bản như họ tên, email, số điện thoại và mật khẩu. Sau khi bấm “Đăng ký”, hệ thống sẽ gửi mã OTP để xác minh



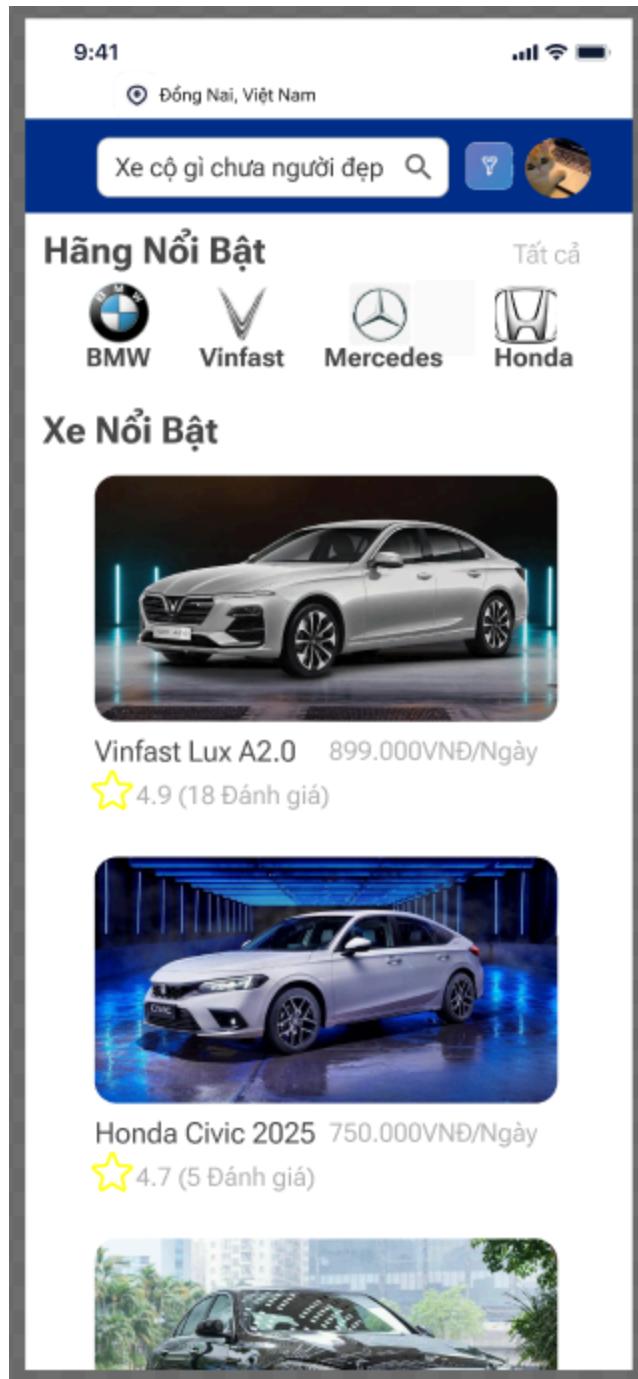
1.2.4. Account

Người dùng có thể dễ dàng theo dõi thông tin hồ sơ và cập nhật hồ sơ cho phép chỉnh sửa linh hoạt các thông tin cá nhân, hỗ trợ upload ảnh đại diện



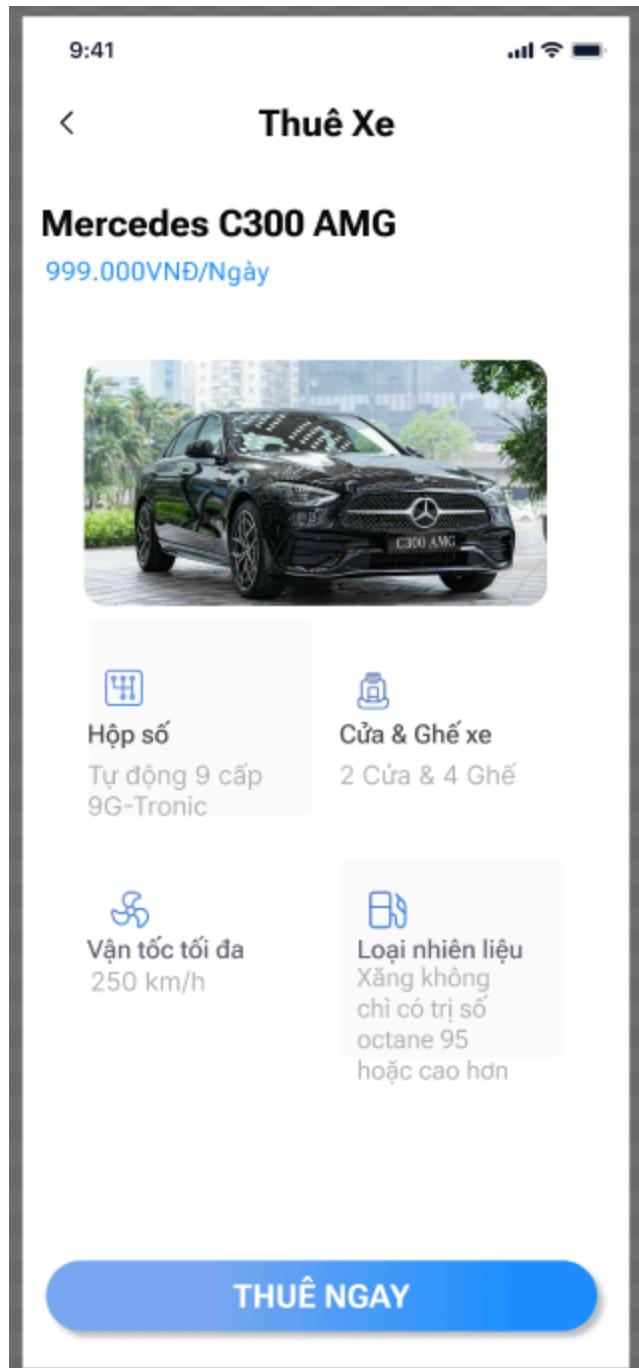
1.2.5. Main screen

Người dùng có thể nhanh chóng tìm và đặt xe thông qua thanh tìm kiếm hoặc duyệt danh sách xe nổi bật.



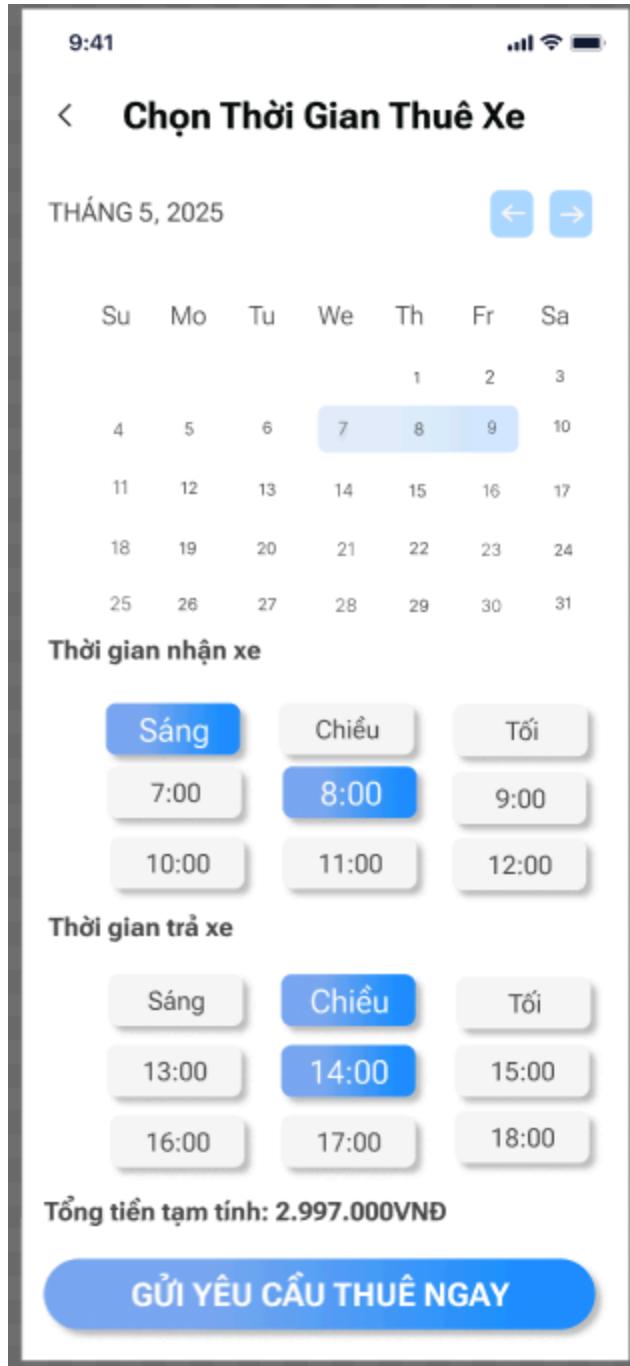
1.2.6. Car Details

Màn hình Chi tiết xe của T9 Car Rental hiển thị đầy đủ hình ảnh, thông tin kỹ thuật, tiện nghi, giá thuê. Người dùng có thể nhấn “THUÊ NGAY” để thuê xe



1.2.7. Select Date & Time

Người dùng đặt xe trong khoảng thời gian mong muốn, đồng thời kiểm tra nhanh tình trạng xe còn trống. Hệ thống cũng ước tính chi phí dựa trên thời gian đã chọn.



1.2.8. Request Success

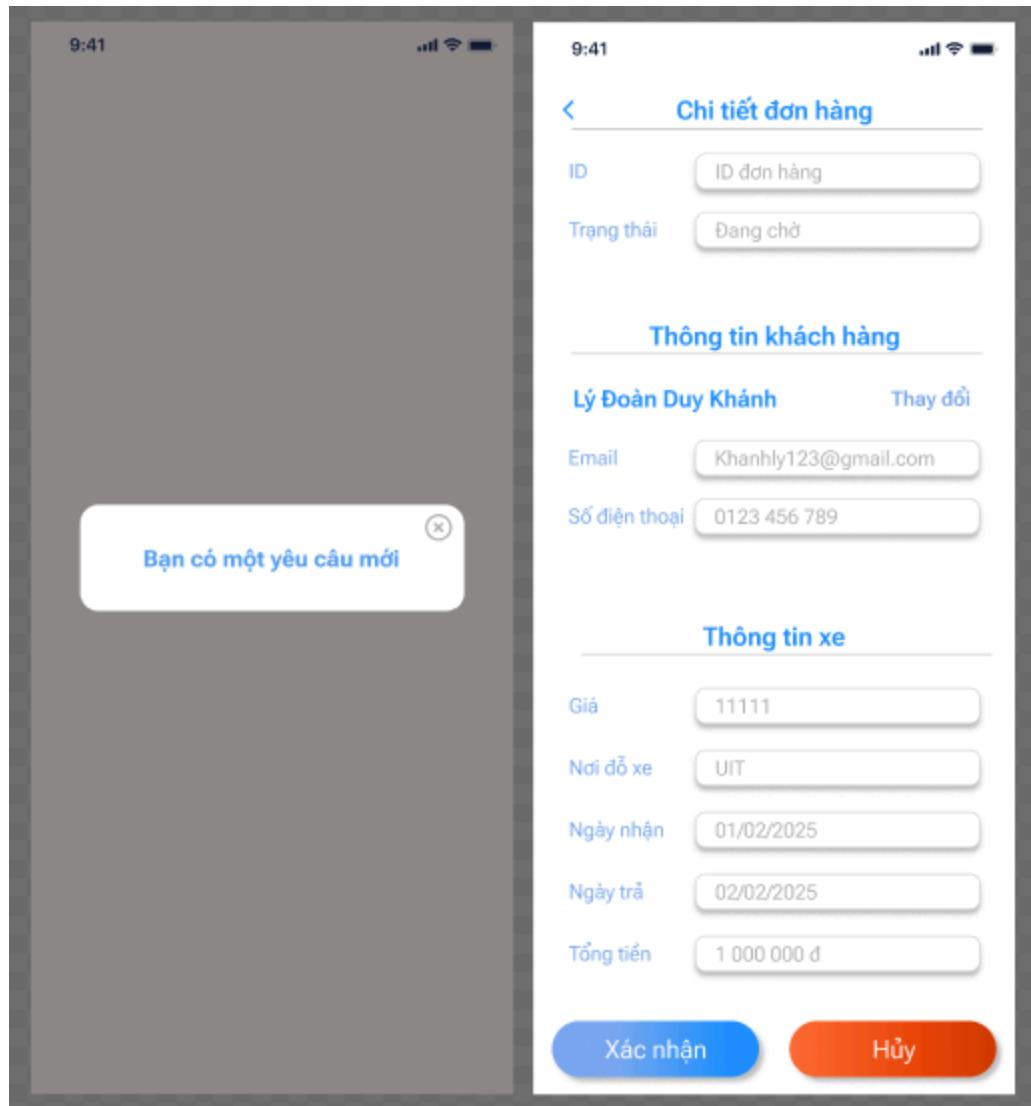
Sau khi hoàn tất yêu cầu đặt xe, người dùng sẽ thấy màn hình thông báo yêu cầu đang được phê duyệt



1.2.9. Car Owner

1.2.9.1 Order request

Người dùng có thể kiểm tra lại toàn bộ thông tin về xe, thời gian thuê, địa điểm và chi phí trước khi gửi yêu cầu cho chủ xe.



1.2.9.2 Payment confirmation

Sau khi nhà cung cấp xác nhận, người dùng kiểm tra lại thông tin và tiến hành thanh toán

2. Thiết kế cơ sở dữ liệu

2.1. Sơ đồ cơ sở dữ liệu



Link database đã thiết kế:

<https://dbdiagram.io/d/Rental-Car-DB-681b231a5b2fc4582f90d161>

2.2. Mô tả cơ sở dữ liệu

2.2.1 Bảng: users

Tên cột	Kiểu dữ liệu	Ràng buộc	Mô tả
id	int	PK, auto inc	Mã người dùng

name	varchar	not null	Tên người dùng
email	varchar	unique, not null	Email đăng nhập
password	varchar	not null	Mật khẩu đã mã hóa
phone	varchar	optional	Số điện thoại người dùng
role	varchar	default 'user'	Vai trò: user, admin
avatar_url	varchar	optional	URL ảnh đại diện
created_at	datetime	default now()	Thời gian tạo tài khoản

2.2.2. Bảng: vehicles

Tên cột	Kiểu dữ liệu	Ràng buộc	Mô tả
id	int	PK, auto inc	Mã xe
name	varchar	not null	Tên xe
brand	varchar	optional	Hãng xe
license_plate	varchar	unique	Biển số xe
description	text	optional	Mô tả xe
price_per_day	decimal	not null	Giá thuê theo ngày
status	varchar	not null	Trạng thái xe
image_url	varchar	optional	URL ảnh xe
category_id	int	FK	Loại xe
location_id	int	FK	Vị trí xe hiện tại
created_by	int	FK	Người đăng xe
created_at	datetime	default now()	Ngày thêm xe

2.2.3. Bảng: rentals

Tên cột	Kiểu dữ liệu	Ràng buộc	Mô tả
id	int	PK, auto inc	Mã đơn thuê
user_id	int	FK	Người thuê
vehicle_id	int	FK	Xe được thuê
start_date	datetime	not null	Ngày bắt đầu thuê
end_date	datetime	not null	Ngày kết thúc thuê
pickup_location	varchar	optional	Địa điểm nhận xe
dropoff_location	varchar	optional	Địa điểm trả xe
status	varchar	default 'pending'	Trạng thái đơn thuê
created_at	datetime	default now()	Ngày đặt xe

2.2.4. Bảng: payments

Tên cột	Kiểu dữ liệu	Ràng buộc	Mô tả
id	int	PK, auto inc	Mã thanh toán
rental_id	int	FK	Liên kết đơn thuê xe
amount	decimal	not null	Số tiền
payment_method	varchar	not null	Hình thức: thẻ, ví điện tử
payment_status	varchar	default 'unpaid'	Trạng thái thanh toán
payment_date	datetime	optional	Ngày thanh toán

2.2.5. Bảng: reviews

Tên cột	Kiểu dữ liệu	Ràng buộc	Mô tả
id	int	PK, auto inc	Mã đánh giá
user_id	int	FK	Người đánh giá
vehicle_id	int	FK	Xe được đánh giá
rating	int	not null	Số sao (1–5)
comment	text	optional	Nhận xét
created_at	datetime	default now()	Thời gian đánh giá

2.2.6. Bảng: vehicle_categories

Tên cột	Kiểu dữ liệu	Ràng buộc	Mô tả
id	int	PK, auto inc	Mã loại xe
name	varchar	not null	Tên loại xe
description	text	optional	Mô tả

2.2.7. Bảng: locations

Tên cột	Kiểu dữ liệu	Ràng buộc	Mô tả
id	int	PK, auto inc	Mã vị trí
city	varchar	not null	Thành phố
district	varchar	optional	Quận/Huyện
address	varchar	optional	Địa chỉ cụ thể

2.2.8. Bảng: vehicle_images

Tên cột	Kiểu dữ liệu	Ràng buộc	Mô tả
id	int	PK, auto inc	Mã ảnh
vehicle_id	int	FK	Xe liên quan
image_url	varchar	not null	Đường dẫn ảnh

2.2.9. Bảng: favorites

Tên cột	Kiểu dữ liệu	Ràng buộc	Mô tả
id	int	PK, auto inc	Mã yêu thích
user_id	int	FK	Người dùng
vehicle_id	int	FK	Xe được yêu thích
created_at	datetime	default now()	Thời điểm thêm

2.2.10. Bảng: notifications

Tên cột	Kiểu dữ liệu	Ràng buộc	Mô tả
id	int	PK, auto inc	Mã thông báo
user_id	int	FK	Người nhận
title	varchar	not null	Tiêu đề
message	text	not null	Nội dung
is_read	boolean	default false	Đã đọc hay chưa
created_at	datetime	default now()	Thời điểm gửi

2.2.11. Bảng: admins

Tên cột	Kiểu dữ liệu	Ràng buộc	Mô tả
id	int	PK, auto inc	Mã quản trị viên
user_id	int	FK	Liên kết bảng users
permissions	text	optional	Quyền nâng cao

2.2.12. Bảng: login_history

Tên cột	Kiểu dữ liệu	Ràng buộc	Mô tả
id	int	PK, auto inc	Mã bản ghi
user_id	int	FK	Người dùng
login_time	datetime	not null	Thời điểm đăng nhập
ip_address	varchar	optional	Địa chỉ IP

2.2.13. Bảng: user_verifications

Tên cột	Kiểu dữ liệu	Ràng buộc	Mô tả
id	int	PK, auto inc	Mã xác minh
user_id	int	FK	Người dùng
token	varchar	not null	Mã xác minh
expired_at	datetime	not null	Thời hạn
is_used	boolean	default false	Đã sử dụng

2.2.14. Bảng: vehicle_availability

Tên cột	Kiểu dữ liệu	Ràng buộc	Mô tả
id	int	PK, auto inc	Mã lịch
vehicle_id	int	FK	Xe
date	date	not null	Ngày
is_available	boolean	default true	Trạng thái có sẵn

2.2.15. Bảng: access_tokens

Tên cột	Kiểu dữ liệu	Ràng buộc	Mô tả
id	int	PK, auto inc	Mã token
user_id	int	FK	Người dùng
token	varchar	not null	Access token
expired_at	datetime	not null	Ngày hết hạn

2.2.16. Bảng: roles

Tên cột	Kiểu dữ liệu	Ràng buộc	Mô tả
id	int	PK, auto inc	Mã vai trò
name	varchar	not null	Tên vai trò
description	text	optional	Mô tả vai trò

2.2.17. Bảng: user_roles

Tên cột	Kiểu dữ liệu	Ràng buộc	Mô tả
id	int	PK, auto inc	Mã gán vai trò
user_id	int	FK	Người dùng
role_id	int	FK	Vai trò

3. Kiến trúc tổng thể của ứng dụng

Mô hình Client-Server-DB đáp ứng nhu cầu người thuê xe, người cho thuê xe, và admin; đảm bảo hiệu suất, bảo mật, khả năng mở rộng.

Thành phần chính:

- Client: Ứng dụng di động cho người thuê (tìm, đặt xe), người cho thuê (quản lý xe) và admin (quản trị viên).
- Server: Xử lý logic, quản lý tài khoản, đặt xe, thanh toán; kết nối qua API (REST/GraphQL).
- Database: Lưu trữ dữ liệu người dùng, xe, giao dịch (MySQL, MongoDB, AWS S3 cho tệp).
- Dịch vụ bên thứ ba: Thanh toán và thông báo.
- Bảo mật: Mã hóa SSL/TLS, xác thực OAuth/JWT, chống tấn công.

Luồng cơ bản: Client gửi yêu cầu → Server xử lý, truy vấn DB → Trả kết quả; admin giám sát qua bảng điều khiển.

V. TRIỂN KHAI

1. Công cụ, công nghệ đã sử dụng

Dự án T9_Car_Rental được phát triển trên nền tảng Android, sử dụng các công cụ và công nghệ sau:

- Ngôn ngữ lập trình: Java, Kotlin.
- Nền tảng phát triển: Android, hỗ trợ từ API 26 (Android 8.0) đến API 35 (Android 15).
- Môi trường phát triển:
 - IDE: Android Studio phiên bản 2023.1.1 (Ladybug).

- Hệ quản trị cơ sở dữ liệu: Firebase Firestore.
- Thư viện và công cụ hỗ trợ:
 - Firebase: Xác thực người dùng, quản lý dữ liệu thời gian thực, lưu trữ tệp, phân tích và thông báo đầy.
 - Cloudinary: Tải và lưu trữ hình ảnh.
 - ZaloPay SDK: Tích hợp cổng thanh toán ZaloPay.
 - Google Play Services: Đăng nhập Google, bản đồ và định vị.
 - Hình ảnh và giao diện: Glide, Picasso, Material Design Components, ConstraintLayout, RecyclerView.
 - Mạng và thanh toán: OkHttp, Commons Codec.
 - Xác thực và kiểm tra: Android Saripaar, JUnit, Espresso.
 - Quản lý vòng đời: LiveData, ViewModel.
 - Khác: MultiDex, SDP, SSP.
- Quản lý mã nguồn: Git.
- Quản lý phụ thuộc: Gradle.

2. Từng chức năng đã lập trình

2.1. Chức năng dành cho khách hàng (Customer)

Thuê xe:

- Giao diện: CustomerHomeFragment.java, VehicleDetailActivity.java, vehicle_card.xml, activity_detail_car.xml.
- Mô tả: Người dùng xem danh sách xe, chọn xe để xem chi tiết, và đặt xe sau khi xác minh CCCD/bằng lái. Quy trình bao gồm chọn thời gian thuê và thanh toán.

Xem danh sách xe:

- Giao diện: CustomerHomeFragment.java, vehicle_card.xml.
- Mô tả: Hiển thị danh sách xe dưới dạng card với tên xe, giá thuê, đánh giá, và hình ảnh. Dữ liệu lấy từ Firestore qua VehicleAdapter.java.

- Tích hợp: Bộ lọc xe (CustomerVehicleFilterFragment.java) hỗ trợ lọc theo thương hiệu, giá, loại nhiên liệu, và trạng thái sẵn có.

Chọn thời gian thuê xe (ScheduleSelect):

- Giao diện: ScheduleSelect.java, activity_schedule_select.xml.
- Mô tả: Người dùng chọn ngày và giờ nhận/trả xe, kiểm tra lịch trống, và xác nhận đặt xe.

Thanh toán đơn đặt xe:

- Giao diện: CustomerBookingDetailActivity.java, customer_booking_detail.xml.
- Mô tả: Hiển thị chi tiết đơn đặt xe (mã đơn, tên xe, thông tin chủ xe, thời gian thuê, tổng tiền). Tích hợp ZaloPay để thanh toán khi đơn được xác nhận.

Quản lý lịch sử hoạt động:

- Giao diện: CustomerActivityFragment.java, item_activity.xml.
- Mô tả: Hiển thị danh sách đơn đặt xe với trạng thái (chưa xác nhận, đã thanh toán, hoàn thành, hủy) qua BookingAdapter.java.

Quản lý thông báo:

- Giao diện: CustomerNotificationFragment.java, item_notification.xml.
- Mô tả: Hiển thị thông báo về trạng thái tài khoản, xe, và đơn đặt xe. Hỗ trợ đánh dấu đã đọc qua NotificationAdapter.java.

Quản lý hồ sơ cá nhân:

- Giao diện: CustomerSettingFragment.java, activity_profile_management.xml.
- Mô tả: Cập nhật thông tin cá nhân, tải ảnh CCCD và bằng lái. Hỗ trợ thay đổi mật khẩu (UpdatePassword.java).

2.2. Chức năng dành cho chủ xe (Owner)

Thêm và cập nhật xe (AddVehicle/UpdateVehicle):

- Giao diện: AddVehicleActivity.java, UpdateVehicleActivity.java, activity_add_vehicle.xml, activity_update_vehicle.xml.

- Mô tả: Chủ xe thêm xe mới hoặc cập nhật thông tin xe (tên, giá, biển số, hình ảnh, giấy tờ). Dữ liệu được gửi lên Firestore và chờ admin duyệt.

Chap nhận yêu cầu đặt xe từ người dùng:

- Giao diện: OwnerActivityFragment.java, OwnerBookingDetailActivity.java, owner_booking_detail.xml.
- Mô tả: Hiển thị danh sách đơn đặt xe từ khách hàng. Chủ xe xác nhận hoặc từ chối đơn qua OwnerActivityAdapter.java.

Quản lý xe:

- Giao diện: OwnerVehicleFragment.java, owner_vehicle_card.xml.
- Mô tả: Hiển thị danh sách xe với tên, giá, trạng thái xác minh. Hỗ trợ nhấn để cập nhật thông tin xe.

Quản lý thông báo:

- Giao diện: OwnerNotificationsFragment.java, item_notification.xml.
- Mô tả: Hiển thị thông báo về xe (duyệt, từ chối, yêu cầu bổ sung) và đơn đặt xe qua OwnerNotificationAdapter.java.

Quản lý hồ sơ cá nhân:

- Giao diện: OwnerSettingFragment.java.
- Mô tả: Cập nhật thông tin cá nhân và tài liệu xác minh, tương tự khách hàng.

2.3. Chức năng dành cho quản trị viên (Admin)

Quản lý xe:

- Giao diện: AdminVehicleListFragment.java, AdminVehicleDetailActivity.java, admin_vehicle_card.xml, fragment_admin_vehicle_list.xml.
- Mô tả: Hiển thị danh sách xe với trạng thái xác minh. Admin duyệt, từ chối, ẩn xe, hoặc gửi cảnh báo vi phạm (RequestSupplementDialogFragment.java).

Quản lý người dùng:

- Giao diện: AdminUserListFragment.java, item_user.xml, fragment_admin_user_list.xml
- Mô tả: Hiển thị danh sách người dùng với tên, vai trò, trạng thái xác minh. Admin có thể xóa hoặc yêu cầu bổ sung thông tin qua UserAdapter.java.

Thống kê:

- Giao diện: AdminStatisticsFragment.java, SimpleBarChart.java.
- Mô tả: Hiển thị thống kê số lượng xe, đơn đặt xe, và doanh thu theo thời gian dưới dạng biểu đồ.

2.4. Chức năng chung

Đăng nhập/Đăng ký:

- Giao diện: LoginActivity.java, RegisterActivity.java, activity_sign_in.xml, activity_register.xml.
- Mô tả: Đăng nhập qua email, Google, và xác minh số điện thoại (ValidatePhoneActivity.java). Đăng ký yêu cầu thông tin cơ bản và xác minh.

Giao diện giới thiệu:

- Giao diện: StartAppActivity.java, sflash_1.xml, sflash_2.xml, sflash_3.xml, sflash_4.xml.
- Mô tả: Hiển thị 4 màn hình giới thiệu với hiệu ứng trượt, chuyển đến màn hình đăng nhập.

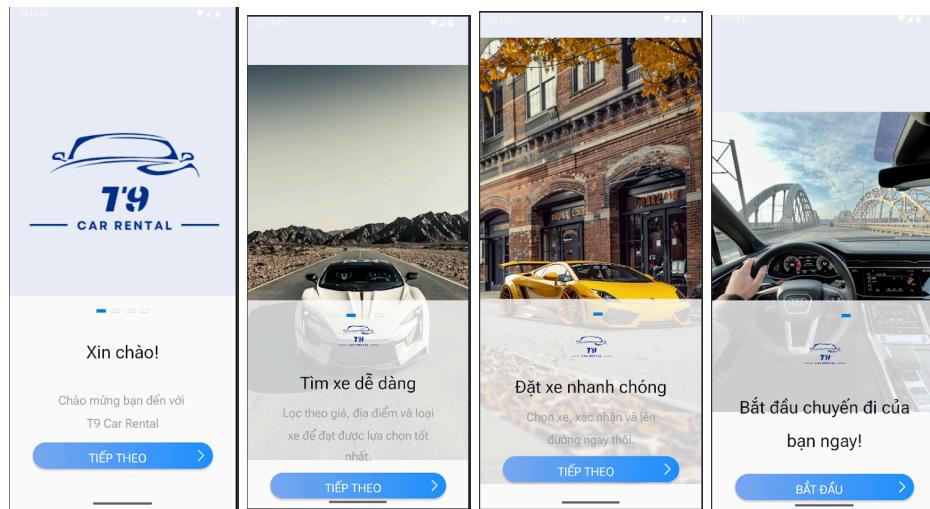
Chat:

- Giao diện: ChatFragment.java, fragment_chat.xml.
- Mô tả: Nhắn tin giữa khách hàng và chủ xe với giao diện bong bóng tin nhắn với AI (MessageAdapter.java).

3. Giao diện thực tế

3.1. Màn hình giới thiệu (Splash Screen):

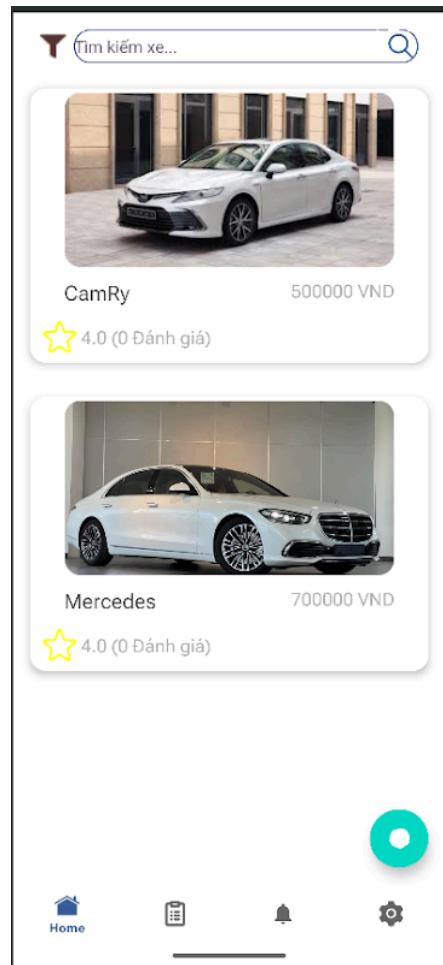
Mô tả: Gồm 4 trang trượt, mỗi trang hiển thị logo, hình ảnh minh họa, tiêu đề, mô tả ngắn, và nút điều hướng. Thanh trạng thái hiển thị thời gian. Màu nền sáng, nút điều hướng nổi bật.



Hình 5.3.1: Màn hình giới thiệu ứng dụng.

3.2. Danh sách xe (Customer Home):

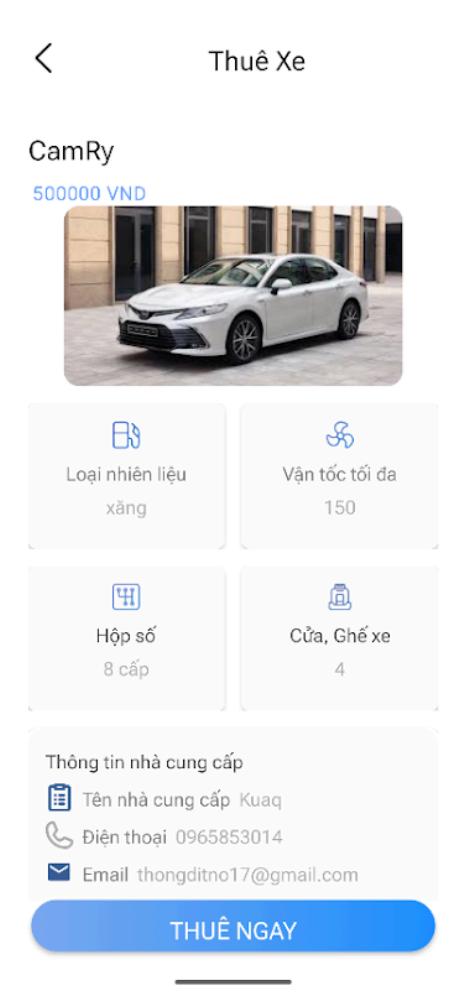
Mô tả: Hiển thị danh sách xe dưới dạng card, bao gồm ảnh xe, tên, giá thuê, và đánh giá. Thanh điều hướng dưới cùng với các mục Home, Activity, Notifications, Tài khoản. Màu nền trắng.



Hình 5.3.2: Danh sách xe trên giao diện khách hàng.

3.3. Chi tiết xe (Vehicle Detail):

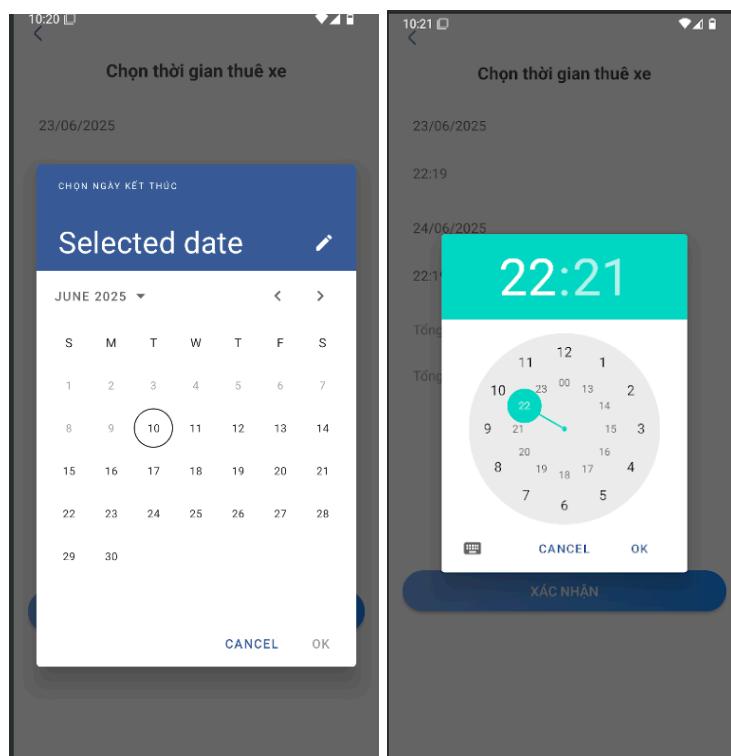
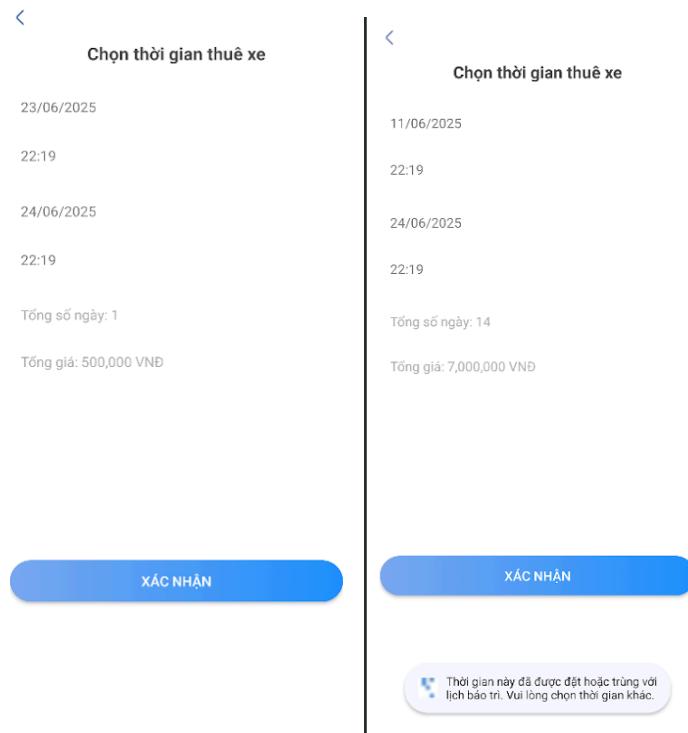
Mô tả: Hiển thị ảnh xe lớn, thông tin chi tiết (tên, giá, nhiên liệu, tốc độ tối đa, hộp số, số ghế/cửa, biển số) và thông tin chủ xe (tên, số điện thoại, email, địa chỉ). Nút đặt xe ở dưới cùng. Màu nền trắng.



Hình 5.3.3: Giao diện chi tiết xe.

3.4. Chọn thời gian thuê xe (Schedule Select):

Mô tả: Giao diện chọn ngày và giờ nhận/trả xe, kèm lịch và bộ chọn giờ. Hiển thị giá ước tính và nút xác nhận. Có thông báo lỗi nếu thời gian không hợp lệ. Màu nền trắng.

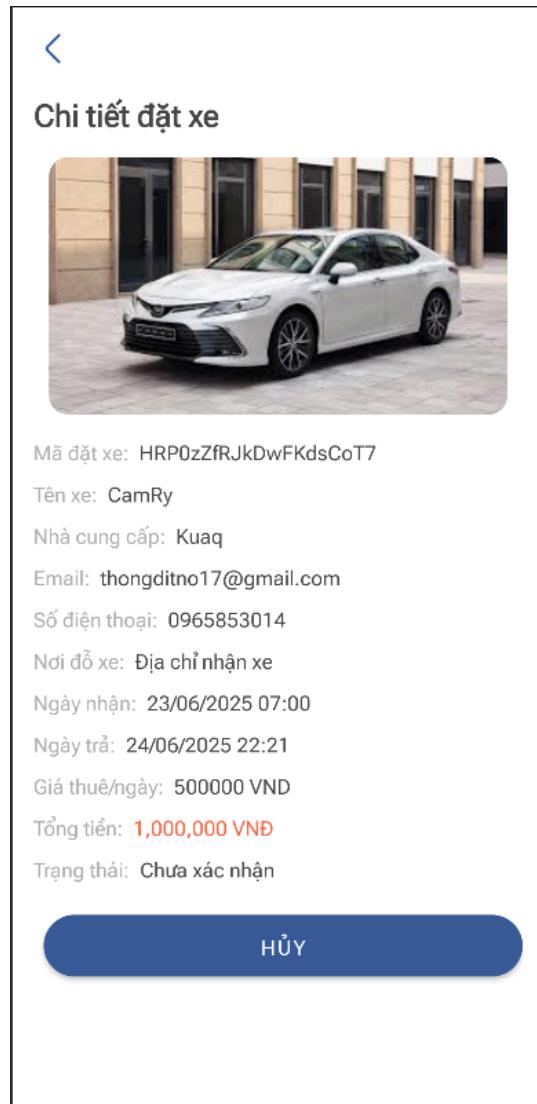




Hình 5.3.4: Giao diện chọn thời gian thuê xe.

3.5. Chi tiết đơn đặt xe và thanh toán (Booking Detail):

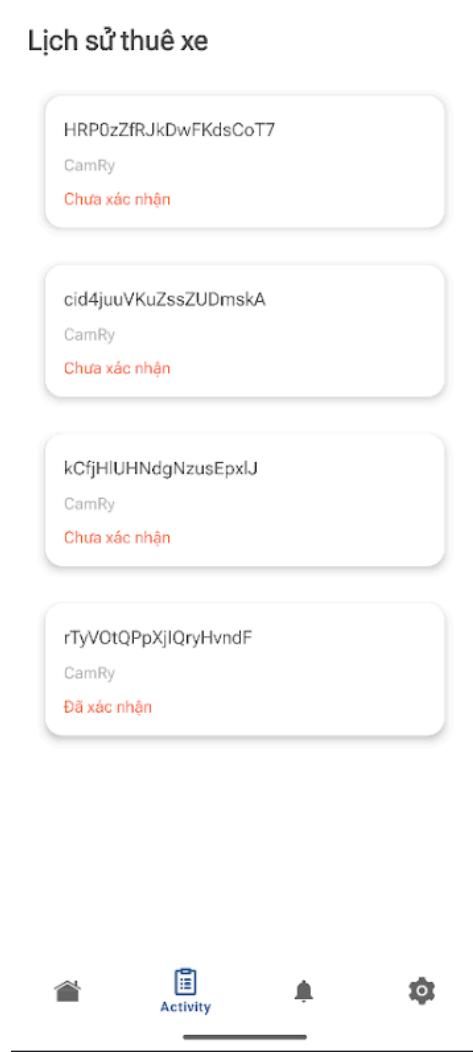
Mô tả: Hiển thị thông tin đơn đặt xe: mã đơn, tên xe, ảnh xe, thông tin chủ xe, địa điểm giao xe, thời gian nhận/trả, giá thuê, tổng tiền, và trạng thái. Có nút thanh toán và hủy. Màn hình trắng.



Hình 5.3.5: Chi tiết đơn đặt xe và thanh toán.

3.6. Lịch sử hoạt động (Customer Activity):

Mô tả: Hiển thị danh sách đơn đặt xe dưới dạng danh mục với tên xe, trạng thái, và mã đơn. Nhấn vào để xem chi tiết. Thanh điều hướng dưới cùng. Màu nền trắng.



Hình 5.3.6: Giao diện lịch sử hoạt động khách hàng.

3.7. Quản lý thông báo (Customer Notifications):

Mô tả: Hiển thị danh sách thông báo với tiêu đề, nội dung, thời gian, và trạng thái (chưa đọc/đã đọc). Nhấn để xem chi tiết hoặc đánh dấu đã đọc. Màu nền trắng.

Thông báo

Tài khoản đã được duyệt Thông báo
10/06/2025 15:50
Tài khoản của bạn đã được duyệt thành công.



Hình 5.3.7: Giao diện quản lý thông báo khách hàng.

3.8. Quản lý hồ sơ cá nhân (Customer Profile):

Mô tả: Hiển thị thông tin cá nhân, ảnh avatar, và nút cập nhật thông tin, tải CCCD/bằng lái. Có tùy chọn đổi mật khẩu và đăng xuất. Màn hình trống.



Hồ sơ cá nhân



Họ và tên
Clone2

Email
vietkuac@gmail.com

Số điện thoại
0942424242

Địa chỉ
Hồ Chí Minh

Thành phố
Hồ Chí Minh

Ngày sinh
10/06/2025

CCCD mặt trước

CHỈNH SỬA HỒ SƠ

Cập nhật thông tin cá nhân



Họ và tên
Clone2

Email
vietkuac@gmail.com

Số điện thoại
0942424242

Ngày sinh
10/06/2025

Địa chỉ
Hồ Chí Minh

Thành phố
Hồ Chí Minh

LƯU THÔNG TIN

Cập nhật căn cước công dân

Hồ sơ cá nhân

CCCD mặt trước



CCCD mặt sau



Bằng lái xe



CHỈNH SỬA HỒ SƠ



Hình 5.3.8: Giao diện quản lý hồ sơ của khách hàng.

3.9. Thêm xe mới (Add Vehicle):

Mô tả: Giao diện nhập thông tin xe mới, bao gồm tên xe, giá thuê, biển số, thương hiệu, nhiên liệu, tốc độ tối đa, hộp số, số ghế/cửa. Có trường tải ảnh xe và giấy tờ, kèm nút gửi dữ liệu. Màu nền trắng.

Thêm mới xe

Tên xe	aaaa
Số chỗ	2
Giá (VND/ngày)	250000
Biển số xe	1241241
Thương hiệu	Toyota
Loại nhiên liệu	Xang
Tốc độ tối đa (km/h)	200
Hộp số	4
Số cửa và ghế	2

Tốc độ tối đa (km/h)
200

Hộp số
4

Số cửa và ghế
2

Ảnh xe



Giấy tờ xe



THÊM

Hình 5.3.9: Giao diện thêm xe mới.

3.10. Cập nhật xe (Update Vehicle):

Mô tả: Giao diện chỉnh sửa thông tin xe hiện tại, cho phép cập nhật giá, biển số, ảnh, và các thông tin khác. Có nút gửi dữ liệu để chờ duyệt. Màu nền trắng.

<

Cập nhật xe

Tên xe
Mercedes

Số chỗ
2

Giá (VND/ngày)
700000

Biển số xe
60C245678

Thương hiệu
ChinaFake

Loại nhiên liệu
Dầu

Tốc độ tối đa (km/h)
250

Hộp số
8

Số cửa và ghế
2

Hình 5.3.10: Giao diện cập nhật xe.

3.11. Chấp nhận yêu cầu đặt xe (Owner Booking):

Mô tả: Hiển thị chi tiết đơn đặt xe với thông tin khách hàng, xe, thời gian thuê, và tổng tiền. Có nút xác nhận và từ chối yêu cầu. Màu nền trắng.

Chi tiết yêu cầu thuê xe



Mã yêu cầu: kCfjHIUHNdgNzusEpxlJ

Tên xe: CamRy

Khách hàng: Clone2

Email: vietkuaq@gmail.com

Số điện thoại: 0942424242

Địa chỉ: Địa chỉ nhận xe

Thời gian nhận: 23/06/2025 07:00

Thời gian trả: 24/06/2025 22:21

Giá thuê/ngày: 500000 VND

Tổng chi phí: **1,000,000 VND**

Trạng thái: Chưa được xác nhận

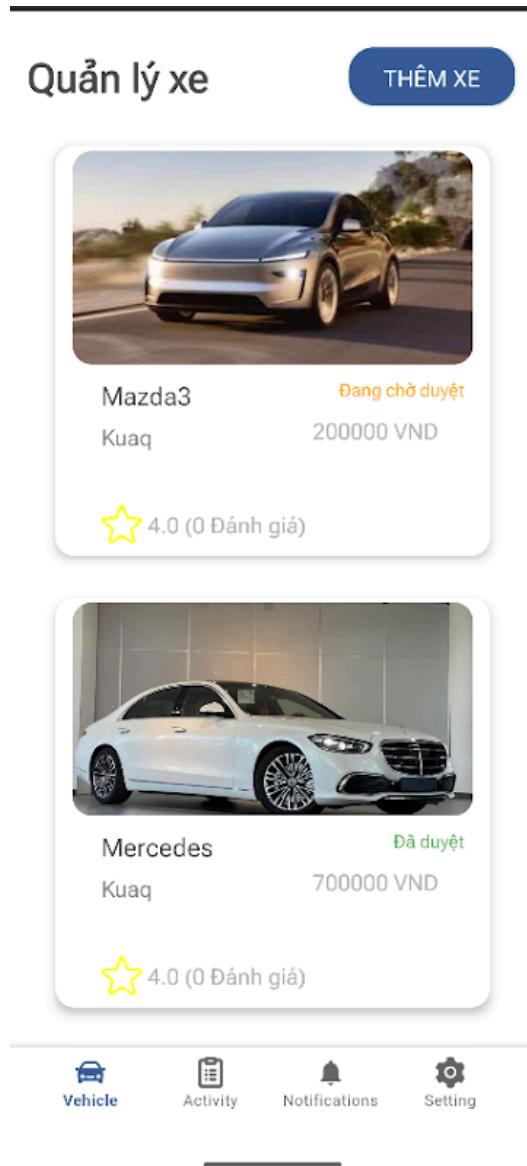
XÁC NHẬN

TỪ CHỐI

Hình 5.3.11: Giao diện xử lý yêu cầu đặt xe của khách.

3.12. Quản lý xe của chủ xe (Owner Vehicles):

Mô tả: Hiển thị danh sách xe với tên, giá, trạng thái xác minh. Nhấn vào để cập nhật thông tin xe. Thanh điều hướng dưới cùng. Màu nền trắng.

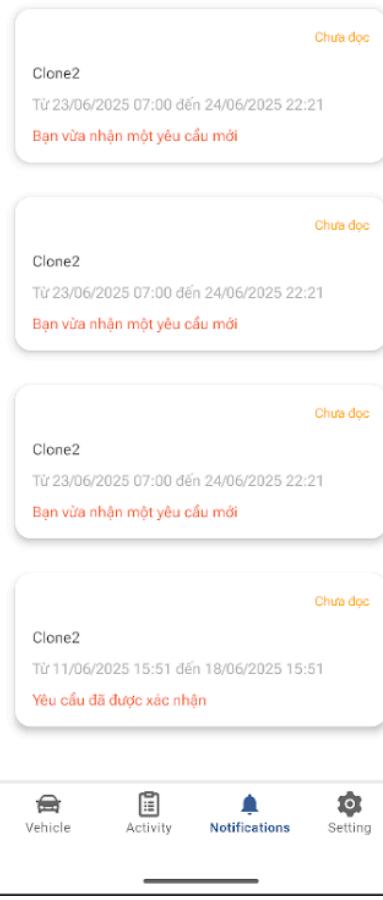


Hình 5.3.12: Giao diện quản lý xe của chủ xe.

3.13. Quản lý thông báo của chủ xe (Owner Notifications):

Mô tả: Hiển thị danh sách thông báo về xe và đơn đặt xe, với tiêu đề, nội dung, thời gian, và trạng thái. Nhấn để xem chi tiết. Màu nền trắng.

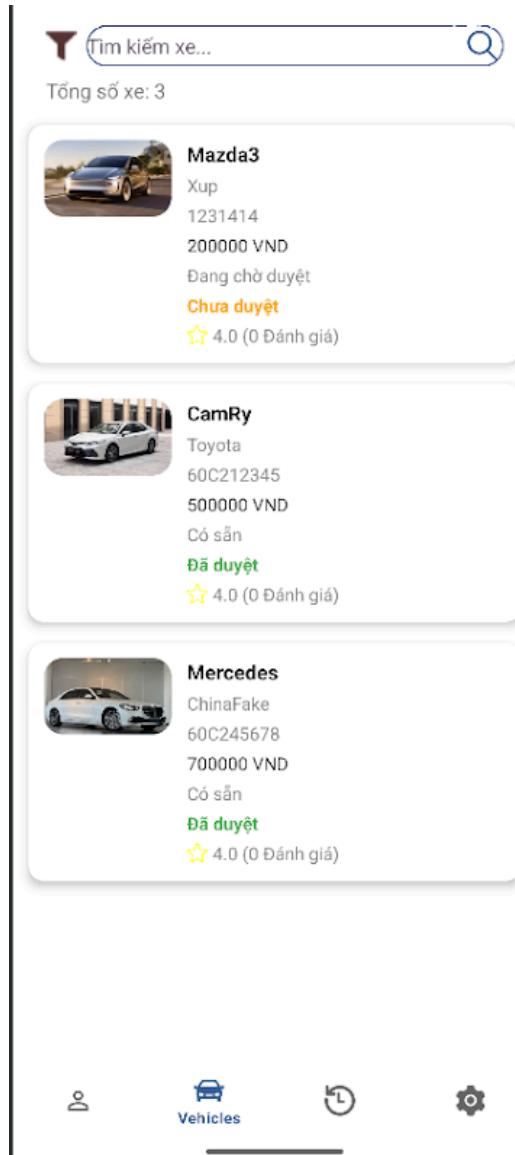
Thông báo



Hình 5.3.13: Giao diện quản lý thông báo chủ xe.

3.14. Quản lý xe của admin (Admin Vehicles):

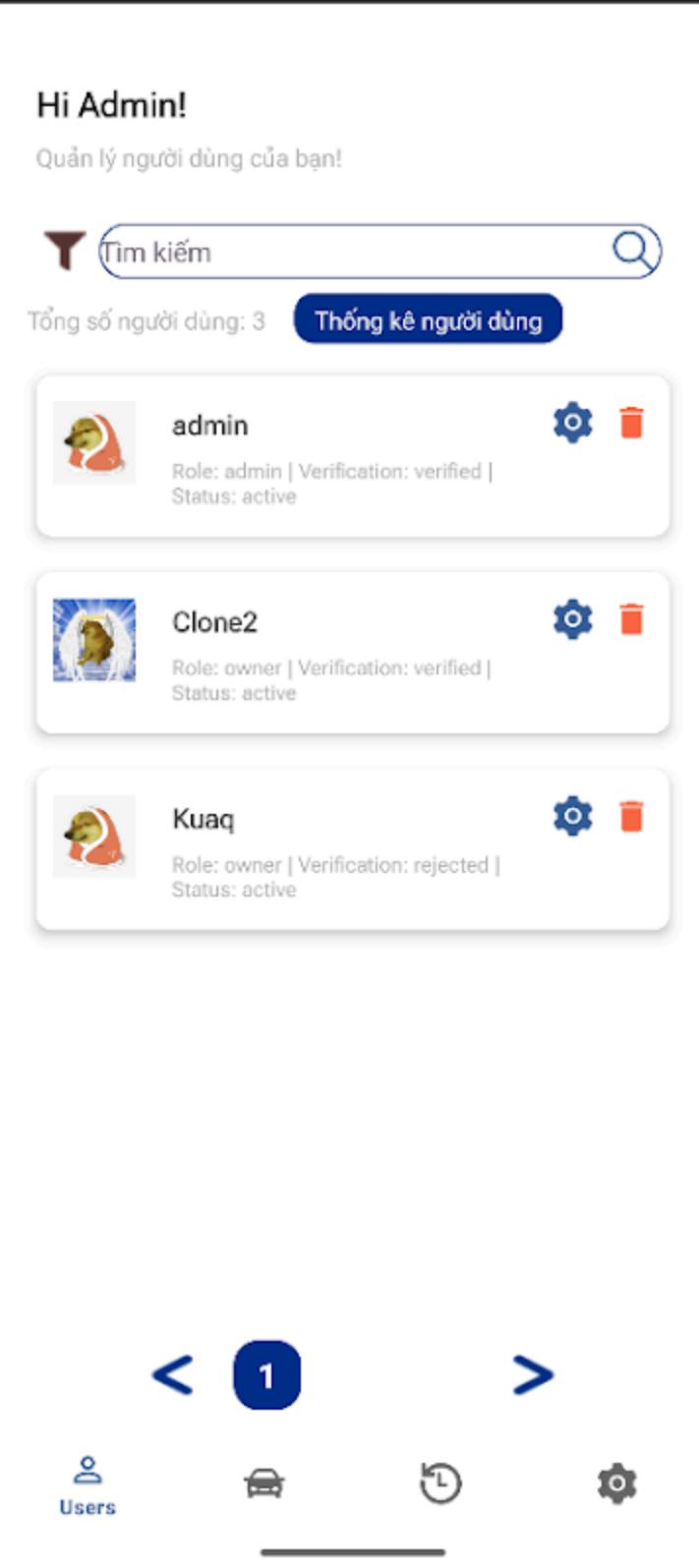
Mô tả: Hiển thị danh sách xe với tên, biển số, trạng thái xác minh. Nhấn vào để xem chi tiết và xử lý (duyệt, từ chối, ẩn). Thanh điều hướng dưới cùng. Màu nền trắng.



Hình 5.3.14: Giao diện quản lý xe của admin.

3.15. Quản lý người dùng của admin (Admin Users):

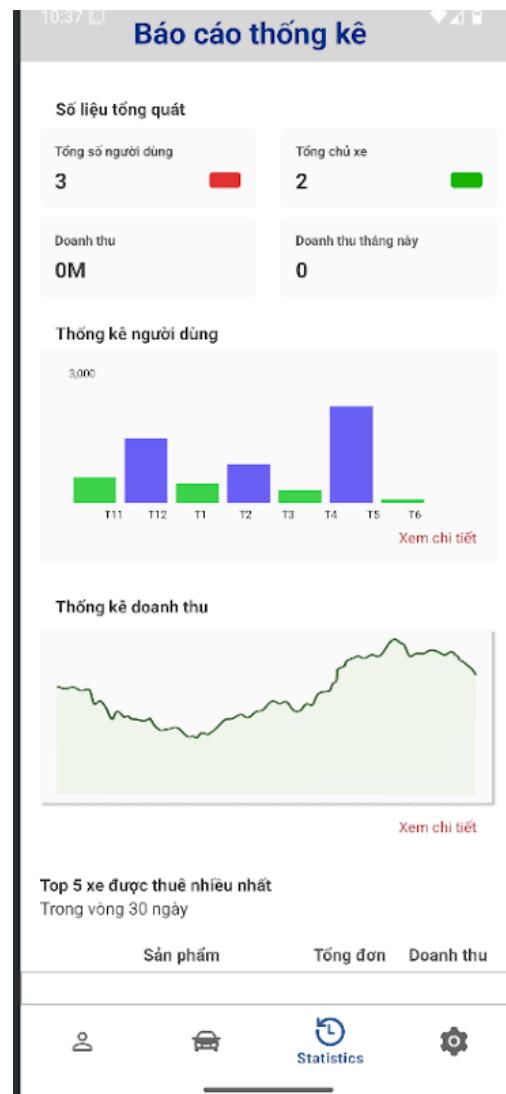
Mô tả: Hiển thị danh sách người dùng với tên, vai trò, trạng thái xác minh, và nút xóa. Nhấn vào để xem chi tiết hoặc yêu cầu bổ sung thông tin. Màu nền trắng.



Hình 5.3.15: Giao diện quản lý người dùng của admin.

3.16. Thống kê của admin (Admin Statistics):

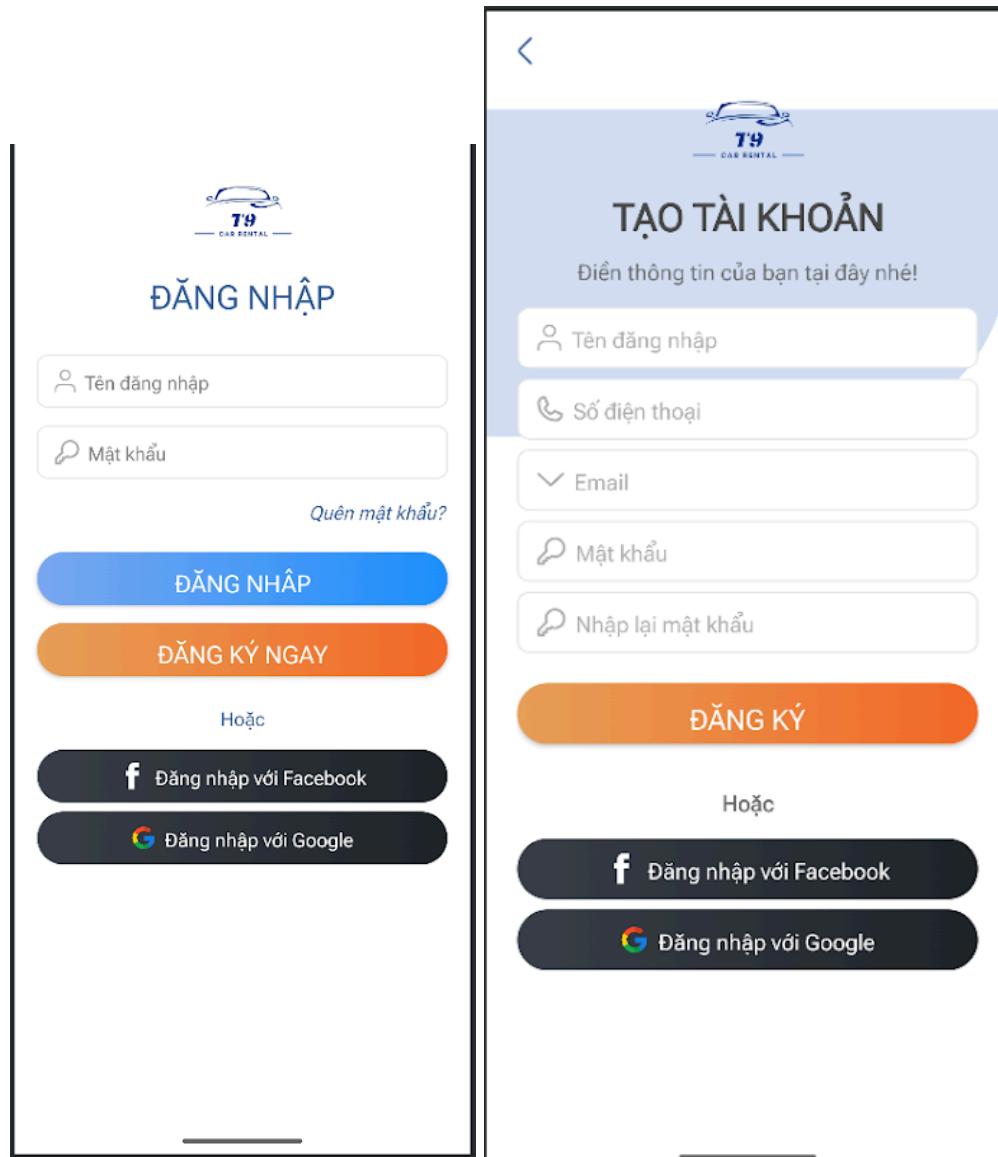
Mô tả: Hiển thị biểu đồ cột với số lượng xe, đơn đặt xe, và doanh thu theo thời gian. Màn hình có màu nền trắng.



Hình 5.3.16: Giao diện thống kê của admin.

3.17. Đăng nhập/Đăng ký:

Mô tả: Giao diện đăng nhập với trường email/mật khẩu, nút Google, và liên kết quên mật khẩu. Giao diện đăng ký yêu cầu tên, email, mật khẩu, số điện thoại. Màn hình có màu nền gradient.



Hình 5.3.17: Giao diện đăng nhập và đăng ký.

3.18. Chat:

Mô tả: Giao diện nhắn tin với bong bóng tin nhắn cho người dùng và đối phương, kèm avatar.



Hình 5.3.18: Giao diện chat.

4. Đoạn code mẫu minh họa các chức năng chính

4.1. Thuê xe (Customer):

Mô tả: Hàm checkCccdAndBook trong VehicleDetailActivity.java kiểm tra trạng thái xác minh CCCD/bằng lái trước khi chuyển đến giao diện chọn thời gian thuê xe. Nếu chưa xác minh, hiển thị dialog yêu cầu cập nhật thông tin.

```
1 usage
private void checkCccdAndBook() {
    FirebaseUser firebaseUser = FirebaseAuth.getInstance().getCurrentUser();
    if (firebaseUser == null) {
        Toast.makeText(context: this, text: "Vui lòng đăng nhập để đặt xe", Toast.LENGTH_SHORT).show();
        return;
    }

    User user = viewModel.getUserLiveData().getValue();
    if (user == null ||
        user.getCiCardFront() == null || user.getCiCardFront().isEmpty() ||
        user.getCiCardBehind() == null || user.getCiCardBehind().isEmpty() ||
        user.getLicenseUrl() == null || user.getLicenseUrl().isEmpty() ||
        !"verified".equals(user.getVerificationStatus())) {
        showVerificationRequiredDialog();
    } else {
        Intent intent = new Intent(packageContext: VehicleDetailActivity.this, ScheduleSelect.class);
        intent.putExtra(name: "vehicleId", vehicleId);
        startActivity(intent);
    }
}

1 usage
private void showVerificationRequiredDialog() {
    new AlertDialog.Builder(context: this)
        .setTitle("Yêu cầu xác minh") Builder
        .setMessage("Bạn cần cập nhật căn cước công dân và bằng lái xe, đồng thời được admin duyệt để thuê xe. Bạn c")
        .setPositiveButton(text: "Cập nhật", (dialog, which) -> {
            Intent intent = new Intent(packageContext: VehicleDetailActivity.this, CCCDActivity.class);
            intent.putExtra(name: "fromVehicleDetail", value: true); // Thêm extra để theo dõi
            startActivity(intent);
        })
        .setNegativeButton(text: "Hủy", (dialog, which) -> dialog.dismiss())
        .create() AlertDialog
        .show();
}
```

4.2. Quản lý xe (Admin):

Mô tả: Hàm approveVehicle và showSupplementDialog trong AdminVehicleDetailActivity.java xử lý duyệt/từ chối xe. Hàm approveVehicle cập nhật trạng thái xe thành "verified" và gửi thông báo. Hàm showSupplementDialog mở dialog nhập lý do từ chối và gửi yêu cầu bổ sung.

```

1 usage
private void approveVehicle() {
    progressDialog = ProgressDialog.show( context: this, title: "Đang duyệt xe", message: "Vui lòng chờ...", indeterminate: true);

    Map<String, Object> updates = new HashMap<>();
    updates.put("verificationStatus", "verified");
    updates.put("vehicleAvailability", "available");

    DocumentReference vehicleRef = db.collection( collectionPath: "Vehicles").document(vehicleId);
    vehicleRef.update(updates).addOnSuccessListener(aVoid -> {
        sendNotification(vehicle.getOwnerId(), title: "Xe đã được duyệt",
            message: "Xe " + vehicle.getVehicleName() + " của bạn đã được duyệt và sẵn sàng cho thuê.", type: "vehicle_verification", vehicleId);
        Toast.makeText( context: this, text: "Xe đã được duyệt", Toast.LENGTH_SHORT).show();
        if (progressDialog != null) progressDialog.dismiss();
        finish();
    }).addOnFailureListener(e -> {
        Log.e(TAG, msg: "Lỗi duyệt xe: " + e.getMessage());
        Toast.makeText( context: this, text: "Lỗi duyệt xe: " + e.getMessage(), Toast.LENGTH_SHORT).show();
        if (progressDialog != null) progressDialog.dismiss();
    });
}

```

```

1 usage
private void showSupplementDialog() {
    RequestSupplementDialogFragment dialog = RequestSupplementDialogFragment.newInstance(vehicle.getOwnerId(), vehicleId, type: "supplement_request_vehicle");
    dialog.setOnSubmitListener(message -> {
        Map<String, Object> updates = new HashMap<>();
        updates.put("verificationStatus", "rejected");
        db.collection( collectionPath: "Vehicles").document(vehicleId).update(updates).addOnSuccessListener(aVoid -> {
            Map<String, Object> notification = new HashMap<>();
            notification.put("userId", vehicle.getOwnerId());
            notification.put("title", "Yêu cầu bổ sung thông tin xe");
            notification.put("message", "Xe " + vehicle.getVehicleName() + " cần bổ sung: " + message);
            notification.put("type", "supplement_request_vehicle");
            notification.put("vehicleId", vehicleId);
            notification.put("createdAt", Timestamp.now());
            notification.put("isRead", false);
            notification.put("role", "owner");

            db.collection( collectionPath: "Notifications").add(notification).addOnSuccessListener(doc -> db.collection( collectionPath: "Notifications").document(doc.getId()).update( id: "notificationId", doc.getId()));
            Toast.makeText( context: this, text: "Đã gửi yêu cầu bổ sung", Toast.LENGTH_SHORT).show();
            finish();
        }).addOnFailureListener(e -> {
            Log.e(TAG, msg: "Lỗi cập nhật trạng thái xe: " + e);
            Toast.makeText( context: this, text: "Lỗi gửi yêu cầu: " + e.getMessage(), Toast.LENGTH_SHORT).show();
        });
    });
    dialog.show(getSupportFragmentManager(), tag: "RequestSupplementDialog");
}

```

4.3. Quản lý User (Admin):

Mô tả: Hàm onBindViewHolder trong UserAdapter.java hiển thị thông tin người dùng (tên, vai trò, trạng thái) và xử lý sự kiện nhấn/xóa. Sự kiện xóa hiển thị dialog xác nhận trước khi xóa người dùng khỏi Firestore.

```

@Override
public void onBindViewHolder(@NonNull UserViewHolder holder, int position) {
    User user = userList.get(position);

    // Bind data
    holder.textViewUsername.setText(user.getUsername() != null && !user.getUsername().isEmpty() ? user.getUsername() : "Unknown User");
    String verificationStatus = user.getVerificationStatus() != null ? user.getStatus() : "Not Set";
    String status = user.getStatus() != null ? user.getStatus() : "Not Set";
    holder.textViewDescription.setText(String.format("Role: %s | Verification: %s | Status: %s",
        user.getCurrentRole() != null ? user.getCurrentRole() : "Not Set", verificationStatus, status));

    // Load avatar
    Glide.with(holder.itemView.getContext()).RequestManager
        .load(user.getAvatarUrl() != null && !user.getAvatarUrl().isEmpty() ? user.getAvatarUrl() : R.drawable.ic_person).RequestBuilder<Drawable>
        .placeholder(R.drawable.ic_person)
        .into(holder.imageViewAvatar);

    // Click listener for item
    holder.itemView.setOnClickListener(v -> {
        if (clickListener != null) {
            clickListener.onUserClick(position);
        }
    });

    // Click listener for delete button
    holder.deleteButton.setOnClickListener(v -> {
        if (holder.itemView.getContext() != null) {
            new AlertDialog.Builder(holder.itemView.getContext())
                .setTitle("Xác nhận xóa")
                .setMessage("Bạn có muốn xóa " + (user.getUsername() != null && !user.getUsername().isEmpty() ? user.getUsername() : "người dùng này") + "?")
                .setNegativeButton("Hủy", (dialog, which) -> dialog.dismiss())
                .setPositiveButton("Xác nhận", (dialog, which) -> {
                    if (deleteListener != null) {
                        deleteListener.onDeleteClick(position);
                    }
                })
                .create()
                .show();
        }
    });
}

```

4.4. Thanh toán (Customer):

Mô tả: Hàm createOrderAndPay, checkout, và postPayment trong CustomerBookingDetailActivity.java xử lý thanh toán qua ZaloPay. createOrderAndPay tạo đơn hàng, checkout mở ZaloPay để thanh toán, và postPayment cập nhật trạng thái đơn sau khi thanh toán thành công.

```

    }

    private void createOrderAndPay() {
        long amount = (long) booking.getTotalAmount();
        if (amount <= 0) {
            Toast.makeText(context, text: "Số tiền không hợp lệ!", Toast.LENGTH_SHORT).show();
            return;
        }

        executorService.execute() -> {
            try {
                CreateOrder createOrder = new CreateOrder();
                JSONObject data = createOrder.createOrder(String.valueOf(amount));
                mainHandler.post() -> {
                    if (data == null) {
                        Toast.makeText(context, text: "Không thể tạo đơn hàng!", Toast.LENGTH_SHORT).show();
                        Log.e(TAG, msg: "ZaloPay response: null");
                        return;
                    }
                    try {
                        Log.d(TAG, msg: "ZaloPay response: " + data.toString());
                        String code = data.getString(name: "returncode");
                        if ("1".equals(code)) {
                            String token = data.getString(name: "zptranstoken");
                            checkout(token);
                        } else {
                            Toast.makeText(context, text: "Lỗi tạo đơn hàng: returncode " + code, Toast.LENGTH_SHORT).show();
                            Log.e(TAG, msg: "ZaloPay returncode: " + code);
                        }
                    } catch (Exception e) {
                        Toast.makeText(context, text: "Lỗi xử lý response: " + e.getMessage(), Toast.LENGTH_SHORT).show();
                        Log.e(TAG, msg: "ZaloPay Exception: " + e.getMessage());
                    }
                };
            } catch (Exception e) {
                mainHandler.post() -> {
                    Toast.makeText(context, text: "Lỗi tạo đơn hàng: " + e.getMessage(), Toast.LENGTH_SHORT).show();
                    Log.e(TAG, msg: "ZaloPay Exception: " + e.getMessage());
                };
            }
        };
    }
}

```

```

1 usage
private void checkout(String token) {
    ZaloPaySDK.getInstance().payOrder(activity: this, token, s: "demozpdk://app", new PayOrderListener() {
        no usages
        @Override
        public void onPaymentSucceeded(String transId, String zpTransToken, String appTransId) {
            runOnUiThread() -> {
                postPayment(transId, appTransId);
            };
        }

        no usages
        @Override
        public void onPaymentCanceled(String zpTransToken, String appTransId) {
            runOnUiThread() -> Toast.makeText(context: CustomerBookingDetailActivity.this, text: "Bạn đã hủy thanh toán.", Toast.LENGTH_SHORT).show();
        }

        no usages
        @Override
        public void onPaymentError(ZaloPayError zaloPayError, String zpTransToken, String appTransId) {
            runOnUiThread() -> Toast.makeText(context: CustomerBookingDetailActivity.this, text: "Thanh toán thất bại: " + zaloPayError, Toast.LENGTH_SHORT).show();
        }
    });
}

```

```

1 usage
private void postPayment(String transId, String appTransId) {
    Map<String, Object> updates = new HashMap<>();
    updates.put("paymentStatus", "paid");
    updates.put("paymentMethod", "ZaloPay");
    updates.put("paymentId", transId);
    updates.put("status", "paid");
    updates.put("updatedAt", Timestamp.now());
    updates.put("orderId", orderId);

    db.collection( collectionPath: "Bookings").document(bookId) DocumentReference
        .update(updates) Task<Void>
        .addOnSuccessListener(aVoid -> {
            createPaymentSuccessNotification(booking.getOwnerId());
            Toast.makeText( context, text: "Thanh toán thành công!", Toast.LENGTH_SHORT).show();
            Intent intent = new Intent( packageContext: CustomerBookingDetailActivity.this, CustomerMainActivity.class);
            intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP | Intent.FLAG_ACTIVITY_NEW_TASK);
            startActivity(intent);
            finish();
        })
        .addOnFailureListener(e -> {
            Log.e(TAG, msg: "Lỗi cập nhật booking: " + e.getMessage());
            Toast.makeText( context, text: "Lỗi cập nhật booking: " + e.getMessage(), Toast.LENGTH_SHORT).show();
        });
}

```

4.5. Chọn thời gian thuê xe (ScheduleSelect, Customer):

Mô tả: Hàm checkScheduleConflict trong ScheduleSelectViewModel.java kiểm tra lịch trống của xe bằng cách so sánh thời gian nhận/trả với các đơn đặt xe hiện có. Hàm confirmBooking tạo đơn đặt xe mới trên Firestore nếu lịch hợp lệ.

```

1 usage
public LiveData<Boolean> checkScheduleConflict(String vehicleId, Timestamp startTime, Timestamp endTime) {
    db.collection( collectionPath: "Bookings") CollectionReference
        .whereEqualTo( field: "vehicleId", vehicleId) Query
        .whereIn( field: "status", List.of("pending", "confirmed", "paid"))
        .get() Task<QuerySnapshot>
        .addOnSuccessListener(querySnapshot -> {
            boolean _isAvailable_ = true;
            for (var doc : querySnapshot.getDocuments()) {
                Booking existingBooking = doc.toObject(Booking.class);
                if (existingBooking != null &&
                    existingBooking.getStartTime() != null &&
                    existingBooking.getEndTime() != null) {
                    Date existingStart = existingBooking.getStartTime().toDate();
                    Date existingEnd = existingBooking.getEndTime().toDate();
                    Date newStart = startTime.toDate();
                    Date newEnd = endTime.toDate();
                    if (!(newEnd.before(existingStart) || newStart.after(existingEnd))) {
                        _isAvailable_ = false;
                        break;
                    }
                }
            }
            scheduleAvailability.setValue(_isAvailable_);
        })
        .addOnFailureListener(e -> scheduleAvailability.setValue(null));
    return scheduleAvailability;
}

```

```

1 usage
private void confirmBooking() {
    if (startDateTime == null || endDateTime == null) {
        Toast.makeText(context: this, R.string.no_date_selected, Toast.LENGTH_SHORT).show();
        return;
    }
    if (endDateTime.before(startDateTime)) {
        Toast.makeText(context: this, R.string.invalid_date_range, Toast.LENGTH_SHORT).show();
        return;
    }
    if (tvSelectStartTime.getText().toString().equals(getString(R.string.select_start_time)) ||
        tvSelectEndTime.getText().toString().equals(getString(R.string.select_end_time))) {
        Toast.makeText(context: this, R.string.no_time_selected, Toast.LENGTH_SHORT).show();
        return;
    }
    if (totalDays <= 0 || totalAmount <= 0) {
        Toast.makeText(context: this, R.string.invalid_booking, Toast.LENGTH_SHORT).show();
        return;
    }

    viewModel.checkScheduleConflict(vehicleId, new Timestamp(startDateTime.getTime()), new Timestamp(endDateTime.getTime()))
        .observe(owner: this, isAvailable -> {
            if (isAvailable == null) {
                Toast.makeText(context: this, R.string.booking_failed, Toast.LENGTH_SHORT).show();
                return;
            }
            if (!isAvailable) {
                Toast.makeText(context: this, R.string.time_overlap_error, Toast.LENGTH_SHORT).show();
                return;
            }
        })
}

```

```

Booking booking = new Booking();
booking.setVehicleId(vehicleId);
booking.setCustomerId(firebaseAuth.getInstance().getCurrentUser().getUid());
booking.setOwnerId(vehicle.getOwnerId());
booking.setStartTime(new Timestamp(startDateTime.getTime()));
booking.setEndTime(new Timestamp(endDateTime.getTime()));
booking.setStatus("pending");
booking.setCreatedAt(Timestamp.now());
booking.setUpdatedAt(Timestamp.now());
booking.setTotalAmount(totalAmount);
booking.setPickupLocation("Địa chỉ nhận xe"); // TODO: Lấy từ input
booking.setDropoffLocation("Địa chỉ trả xe"); // TODO: Lấy từ input

viewModel.createBooking(booking).observe(owner: this, bookingId -> {
    if (bookingId != null) {
        Notification notification = new Notification();
        notification.setUserId(vehicle.getOwnerId());
        notification.setBookingId(bookingId);
        notification.setMessage("Có booking mới: #" + bookingId);
        notification.setType("new_booking");
        db.collection(collectionPath: "Notifications").add(notification);

        Intent intent = new Intent(packageContext: ScheduleSelect.this, RequestSuccessActivity.class);
        intent.putExtra(name: "bookingId", bookingId);
        startActivity(intent);
        Toast.makeText(context: this, R.string.booking_success, Toast.LENGTH_SHORT).show();
        finish();
    } else {
        Toast.makeText(context: this, R.string.booking_failed, Toast.LENGTH_SHORT).show();
    }
});
}
}

```

4.6. Thêm xe mới (AddVehicle, Owner):

Mô tả: Hàm addVehicle trong AddVehicleActivity.java kiểm tra dữ liệu nhập (tên, giá, biển số, ảnh), tải ảnh lên Cloudinary, và gửi thông tin xe lên Firestore với trạng thái "pending".

```
29     public class AddVehicleActivity extends AppCompatActivity {
30         private void addVehicle() {
31             ...
32
33             vehicle.setVehicleName(vehicleName.getText().toString().trim());
34             vehicle.setVehicleSeats(vehicleSeats.getText().toString().trim());
35             vehicle.setVehiclePrice(vehiclePrice.getText().toString().trim() + " VND");
36             vehicle.setVehicleNumber(vehicleNumber.getText().toString().trim());
37             vehicle.setVehicleBrand(vehicleBrand.getText().toString().trim());
38             vehicle.setFuelType(fuelType.getText().toString().trim());
39             vehicle.setMaxSpeed(maxSpeed.getText().toString().trim());
40             vehicle.setTransmission(transmission.getText().toString().trim());
41             vehicle.setDoorsAndSeats(doorsAndSeats.getText().toString().trim());
42             vehicle.setVehicleAvailability("pending");
43             vehicle.setVehicleImageUrl(vehicleImageUrl);
44             vehicle.setDocumentImageUrl(documentImageUrl);
45             vehicle.setVerificationStatus("pending");
46             vehicle.setCreatedAt(Timestamp.now());
47             if (firebaseUser != null) {
48                 vehicle.setOwnerId(firebaseUser.getUid());
49             } else {
50                 toast("Không thể thêm xe: Người dùng chưa đăng nhập.");
51                 return;
52             }
53
54             progressDialog = ProgressDialog.show(context, title: "Đang thêm xe", message: "Vui lòng chờ...", indeterminate: true);
55
56             db.collection(collectionPath: "Vehicles")
57                 .add(vehicle).Task<DocumentReference>
58                     .addOnSuccessListener(documentReference -> {
59                         String vehicleId = documentReference.getId();
60                         vehicle.setVehicleId(vehicleId);
61                         updateVehicleId(vehicleId);
62                         createAdminNotification(vehicleId);
63                         if (progressDialog != null) progressDialog.dismiss();
64                         toast("Xe đã được thêm thành công, đang chờ duyệt.");
65                         setResult(resultCode);
66                         finish();
67                     })
68                     .addOnFailureListener(e -> {
69                         if (progressDialog != null) progressDialog.dismiss();
70                         Log.e(TAG, msg: "Thêm xe thất bại: " + e.getMessage());
71                         toast("Thêm xe thất bại: " + e.getMessage());
72                     });
73         }
74     }
```

4.7. Cập nhật xe (UpdateVehicle, Owner):

Mô tả: Hàm update trong UpdateVehicleActivity.java kiểm tra các thay đổi so với dữ liệu xe hiện tại, tải ảnh mới (nếu có) lên Cloudinary, và cập nhật Firestore với trạng thái "pending".

```

    usage
    private void update() {
        if (!validateInputs()) {
            return;
        }

        Map<String, Object> data = new HashMap<>();
        boolean flag = false;
        String plateNumber = editTextVehicleNumber.getText().toString().trim();
        String seats = editTextVehicleSeats.getText().toString().trim();
        String price = editTextVehiclePrice.getText().toString().trim();
        String brand = editTextVehicleBrand.getText().toString().trim();
        String fuelType = editTextFuelType.getText().toString().trim();
        String maxSpeed = editTextMaxSpeed.getText().toString().trim();
        String transmission = editTextTransmission.getText().toString().trim();
        String doorsAndSeats = editTextDoorsAndSeats.getText().toString().trim();

        // Check for changes
        if (!plateNumber.equals(vehicle.getVehicleNumber())) {
            data.put("vehicleNumber", plateNumber);
            flag = true;
        }
        if (!seats.equals(vehicle.getVehicleSeats())) {
            data.put("vehicleSeats", seats);
            flag = true;
        }
        if (!price.equals(vehicle.getVehiclePrice() != null ? vehicle.getVehiclePrice().replace("$", "") : "")) {
            data.put("vehiclePrice", price + " VND");
            flag = true;
        }
        if (!brand.equals(vehicle.getVehicleBrand())) {
            data.put("vehicleBrand", brand);
            flag = true;
        }
        if (!fuelType.equals(vehicle.getFuelType())) {
            data.put("fuelType", fuelType);
            flag = true;
        }

        if (!transmission.equals(vehicle.getTransmission())) {
            data.put("transmission", transmission);
            flag = true;
        }
        if (!doorsAndSeats.equals(vehicle.getDoorsAndSeats())) {
            data.put("doorsAndSeats", doorsAndSeats);
            flag = true;
        }

        if (vehicleImageUri != null || documentImageUri != null) {
            uploadImagesAndUpdate(data, flag);
        } else if (flag) {
            updateFirestore(data);
        } else {
            Toast.makeText(context, "Không có thay đổi để cập nhật", Toast.LENGTH_SHORT).show();
        }
    }
}

```

4.8. Chấp nhận yêu cầu đặt xe (Owner):

Mô tả: Hàm onConfirmBooking trong OwnerBookingDetailActivity.java cập nhật trạng thái đơn đặt xe thành "confirmed" trên Firestore và gửi thông báo đến khách hàng. Hàm onRejectBooking xử lý từ chối đơn.

```
1 usage
private void confirmBooking() {
    Calendar calendar = Calendar.getInstance();
    calendar.setTime(booking.getEndTime().toDate());
    calendar.add(Calendar.DAY_OF_MONTH, amount);
    Timestamp maintenanceEndTime = new Timestamp(calendar.getTime());

    Map<String, Object> updates = new HashMap<>();
    updates.put("status", "confirmed");
    updates.put("updatedAt", Timestamp.now());
    updates.put("maintenanceEndTime", maintenanceEndTime);

    db.collection( collectionPath: "Bookings" ).document(bookingId) DocumentReference
        .update(updates) Task<Void>
        .addOnSuccessListener( aVoid -> {
            createNotification(booking.getCustomerId(), message: "Yêu cầu đặt xe #"+ bookingId + " đã được xác nhận", type: "status_update");
            Toast.makeText( context: this, text: "Đã xác nhận yêu cầu", Toast.LENGTH_SHORT).show();
            createOrder();
            finish();
        })
        .addOnFailureListener(e -> {
            Log.e(TAG, msg: "Lỗi xác nhận yêu cầu: " + e.getMessage());
            Toast.makeText( context: this, text: "Lỗi xác nhận: " + e.getMessage(), Toast.LENGTH_SHORT).show();
        });
}
```

```
1 usage
private void rejectBooking() {
    Map<String, Object> updates = new HashMap<>();
    updates.put("status", "rejected");
    updates.put("updatedAt", Timestamp.now());

    db.collection( collectionPath: "Bookings" ).document(bookingId) DocumentReference
        .update(updates) Task<Void>
        .addOnSuccessListener( aVoid -> {
            createNotification(booking.getCustomerId(), message: "Yêu cầu đặt xe #"+ bookingId + " đã bị từ chối", type: "status_update");
            Toast.makeText( context: this, text: "Đã từ chối yêu cầu", Toast.LENGTH_SHORT).show();
            finish();
        })
        .addOnFailureListener(e -> {
            Log.e(TAG, msg: "Lỗi từ chối yêu cầu: " + e.getMessage());
            Toast.makeText( context: this, text: "Lỗi từ chối: " + e.getMessage(), Toast.LENGTH_SHORT).show();
        });
}
```

VI. KIỂM THỬ

1. Cách kiểm thử từng tính năng

Dựa trên các chức năng đã triển khai trong phần Triển khai, dưới đây là cách kiểm thử từng tính năng chính của ứng dụng T9_Car_Rental. Kiểm thử được thực hiện trên thiết bị Android (API 26 trở lên) sử dụng Android Studio 2023.1.1 (Ladybug). Các tính năng được kiểm thử thủ công (manual testing) và tự động (unit test, instrumentation test) khi áp dụng.

1.1. Chức năng dành cho khách hàng (Customer)

Thuê xe:

- Mô tả kiểm thử: Người dùng thực hiện quy trình thuê xe từ xem danh sách xe, xem chi tiết xe, chọn thời gian thuê, đến thanh toán đơn đặt xe.

- Bước kiểm thử:
 1. Mở ứng dụng, đăng nhập bằng tài khoản khách hàng đã xác minh.
 2. Tại màn hình chính (CustomerHomeFragment), kiểm tra danh sách xe hiển thị đầy đủ thông tin (tên, giá, đánh giá, ảnh).
 3. Nhấn vào một xe để mở chi tiết xe (VehicleDetailActivity), xác minh thông tin xe và chủ xe hiển thị đúng.
 4. Nhấn nút "Đặt xe", kiểm tra yêu cầu xác minh CCCD/bằng lái. Nếu chưa xác minh, dialog yêu cầu cập nhật xuất hiện.
 5. Tại màn hình chọn thời gian (ScheduleSelect), chọn ngày/giờ nhận/trả xe, xác nhận đơn. Kiểm tra thông báo lỗi nếu thời gian không hợp lệ.
 6. Tại màn hình chi tiết đơn (CustomerBookingDetailActivity), kiểm tra thông tin đơn và nút "Thanh toán" xuất hiện khi đơn được xác nhận.
 7. Nhấn "Thanh toán", thực hiện giao dịch qua ZaloPay, xác minh trạng thái đơn cập nhật thành "paid" sau khi thanh toán thành công.
- Điều kiện kiểm thử:
 1. Tài khoản khách hàng đã xác minh CCCD/bằng lái.
 2. Có xe khả dụng trên Firestore.
 3. ZaloPay SDK được cấu hình đúng (môi trường sandbox).
- Kết quả mong đợi:
 - Danh sách xe hiển thị đầy đủ, chi tiết xe chính xác.

- Đơn đặt xe được tạo và thanh toán thành công, thông báo gửi đến chủ xe.
- Công cụ kiểm thử:
 - Manual testing trên thiết bị thực.
 - Logcat để kiểm tra log Firestore và ZaloPay API.
 - Unit test: ExampleUnitTest.java (kiểm tra logic cơ bản).
 - Instrumentation test: ExampleInstrumentedTest.java (kiểm tra package name).

Xem danh sách xe:

- Mô tả kiểm thử: Kiểm tra hiển thị danh sách xe và chức năng lọc.
- Bước kiểm thử:
 1. Mở màn hình chính (CustomerHomeFragment).
 2. Kiểm tra danh sách xe hiển thị với các card chứa tên, giá, đánh giá, và ảnh.
 3. Nhấn nút lọc (CustomerVehicleFilterFragment), chọn các tiêu chí (thương hiệu, giá, nhiên liệu, trạng thái), áp dụng bộ lọc.
 4. Xác minh danh sách xe được cập nhật theo tiêu chí lọc.
- Điều kiện kiểm thử: Có dữ liệu xe trên Firestore.
- Kết quả mong đợi: Danh sách xe hiển thị đúng, bộ lọc hoạt động chính xác.
- Công cụ kiểm thử: Manual testing, Logcat.

Chọn thời gian thuê xe (ScheduleSelect):

- Mô tả kiểm thử: Kiểm tra khả năng chọn thời gian thuê xe và xử lý lịch trùng.
- Bước kiểm thử:

1. Từ màn hình chi tiết xe, nhấn "Đặt xe" để mở ScheduleSelect.
2. Chọn ngày/giờ nhận/trả xe, nhấn "Xác nhận".
3. Kiểm tra thông báo lỗi nếu thời gian trùng với đơn khác hoặc không hợp lệ.
4. Nếu hợp lệ, xác minh đơn đặt xe được tạo trên Firestore.
 - Điều kiện kiểm thử: Xe đã chọn có lịch trống.
 - Kết quả mong đợi: Thời gian hợp lệ tạo đơn thành công, thời gian trùng hiển thị lỗi.
 - Công cụ kiểm thử: Manual testing, Logcat, Firestore Console.

Thanh toán đơn đặt xe:

- Mô tả kiểm thử: Kiểm tra quy trình thanh toán qua ZaloPay.
- Bước kiểm thử:
 1. Mở chi tiết đơn đặt xe (CustomerBookingDetailActivity) với trạng thái "confirmed" và "unpaid".
 2. Nhấn nút "Thanh toán", kiểm tra ứng dụng ZaloPay mở ra.
 3. Thực hiện thanh toán thành công/hủy/thất bại, xác minh trạng thái đơn cập nhật tương ứng ("paid"/không đổi).
 4. Kiểm tra thông báo gửi đến chủ xe khi thanh toán thành công.
- Điều kiện kiểm thử: ZaloPay SDK cấu hình đúng, đơn đặt xe hợp lệ.
- Kết quả mong đợi: Thanh toán thành công cập nhật trạng thái, thông báo gửi đúng.

- Công cụ kiểm thử: Manual testing, ZaloPay sandbox, Firestore Console.

Quản lý lịch sử hoạt động:

- Mô tả kiểm thử: Kiểm tra hiển thị danh sách đơn đặt xe.
- Bước kiểm thử:
 1. Mở màn hình lịch sử (CustomerActivityFragment).
 2. Kiểm tra danh sách đơn hiển thị đúng (mã đơn, tên xe, trạng thái).
 3. Nhấn vào một đơn để xem chi tiết (CustomerBookingDetailActivity).
- Điều kiện kiểm thử: Tài khoản có lịch sử đơn đặt xe.
- Kết quả mong đợi: Danh sách đơn hiển thị đầy đủ, chi tiết đơn chính xác.
- Công cụ kiểm thử: Manual testing, Firestore Console.

Quản lý thông báo:

- Mô tả kiểm thử: Kiểm tra hiển thị và đánh dấu thông báo.
- Bước kiểm thử:
 1. Mở màn hình thông báo (CustomerNotificationFragment).
 2. Kiểm tra danh sách thông báo hiển thị (tiêu đề, nội dung, thời gian, trạng thái).
 3. Nhấn vào thông báo chưa đọc, xác minh trạng thái cập nhật thành "đã đọc".
- Điều kiện kiểm thử: Tài khoản có thông báo trên Firestore.
- Kết quả mong đợi: Thông báo hiển thị đúng, trạng thái cập nhật chính xác.
- Công cụ kiểm thử: Manual testing, Firestore Console.

Quản lý hồ sơ cá nhân:

- Mô tả kiểm thử: Kiểm tra cập nhật thông tin cá nhân và tài liệu xác minh.

- Bước kiểm thử:
 1. Mở màn hình hồ sơ (CustomerSettingFragment).
 2. Cập nhật thông tin (tên, email, số điện thoại, địa chỉ), nhấn lưu.
 3. Tải lên ảnh CCCD/bằng lái (CCCDActivity), xác minh ảnh lưu trên Firebase Storage.
 4. Nhấn đổi mật khẩu (UpdatePassword), nhập mật khẩu mới, xác nhận.
- Điều kiện kiểm thử: Tài khoản đã đăng nhập.
- Kết quả mong đợi: Thông tin cập nhật đúng, ảnh lưu thành công, mật khẩu đổi được.
- Công cụ kiểm thử: Manual testing, Firebase Console.

1.2. Chức năng dành cho chủ xe (Owner)

Thêm và cập nhật xe (AddVehicle/UpdateVehicle):

- Mô tả kiểm thử: Kiểm tra quy trình thêm và cập nhật xe.
- Bước kiểm thử:
 1. Mở giao diện thêm xe (AddVehicleActivity), nhập đầy đủ thông tin (tên, giá, biển số, thương hiệu, nhiên liệu, tốc độ, hộp số, số ghế/cửa).
 2. Tải lên ảnh xe và giấy tờ, nhấn gửi.
 3. Kiểm tra dữ liệu xe được tạo trên Firestore với trạng thái "pending".
 4. Mở giao diện cập nhật xe (UpdateVehicleActivity), chỉnh sửa thông tin, tải lại ảnh, nhấn gửi.
 5. Xác minh dữ liệu xe được cập nhật trên Firestore, trạng thái vẫn "pending".
- Điều kiện kiểm thử: Tài khoản chủ xe đã xác minh.
- Kết quả mong đợi: Xe được thêm/cập nhật thành công, trạng thái chờ duyệt.

- Công cụ kiểm thử: Manual testing, Firebase Console, Cloudinary Dashboard.

Chấp nhận yêu cầu đặt xe từ người dùng:

- Mô tả kiểm thử: Kiểm tra xử lý đơn đặt xe từ khách hàng.
- Bước kiểm thử:
 1. Mở màn hình danh sách đơn đặt xe (OwnerActivityFragment).
 2. Nhấn vào một đơn để xem chi tiết (OwnerBookingDetailActivity).
 3. Nhấn nút "Xác nhận" hoặc "Từ chối", kiểm tra trạng thái đơn cập nhật trên Firestore ("confirmed"/"rejected").
 4. Xác minh thông báo được gửi đến khách hàng.
- Điều kiện kiểm thử: Có đơn đặt xe từ khách hàng.
- Kết quả mong đợi: Trạng thái đơn cập nhật đúng, thông báo gửi thành công.
- Công cụ kiểm thử: Manual testing, Firestore Console.

Quản lý xe:

- Mô tả kiểm thử: Kiểm tra hiển thị danh sách xe của chủ xe.
- Bước kiểm thử:
 1. Mở màn hình quản lý xe (OwnerVehicleFragment).
 2. Kiểm tra danh sách xe hiển thị (tên, giá, trạng thái xác minh).
 3. Nhấn vào một xe để mở giao diện cập nhật (UpdateVehicleActivity).
 4. Điều kiện kiểm thử: Chủ xe có xe trong Firestore.
- Kết quả mong đợi: Danh sách xe hiển thị đúng, chuyển đến cập nhật thành công.
- Công cụ kiểm thử: Manual testing, Firestore Console.

Quản lý thông báo:

- Mô tả kiểm thử: Tương tự khách hàng, kiểm tra hiển thị và đánh dấu thông báo về xe và đơn đặt xe.
- Bước kiểm thử: Tương tự đối với người dùng
- Điều kiện kiểm thử: Tài khoản có thông báo.
- Kết quả mong đợi: Thông báo hiển thị đúng, trạng thái cập nhật được.
- Công cụ kiểm thử: Manual testing, Firestore Console.

Quản lý hồ sơ cá nhân:

- Mô tả kiểm thử: Tương tự khách hàng, kiểm tra cập nhật thông tin cá nhân và tài liệu xác minh.
- Bước kiểm thử: Tương tự đối với người dùng
- Điều kiện kiểm thử: Tài khoản đã đăng nhập.
- Kết quả mong đợi: Thông tin cập nhật đúng, tài liệu lưu thành công.
- Công cụ kiểm thử: Manual testing, Firebase Console.

1.3 Chức năng dành cho quản trị viên (Admin)

Quản lý xe:

- Mô tả kiểm thử: Kiểm tra quy trình duyệt/tù chối xe.
- Bước kiểm thử:
 1. Đăng nhập bằng tài khoản admin duy nhất.
 2. Mở màn hình quản lý xe (AdminVehicleListFragment).
 3. Kiểm tra danh sách xe hiển thị (tên, biển số, trạng thái).
 4. Nhấn vào một xe để xem chi tiết (AdminVehicleDetailActivity).
 5. Nhấn "Duyệt" để cập nhật trạng thái xe thành "verified", hoặc "Tù chối" để mở dialog nhập lý do tù chối (RequestSupplementDialogFragment).
 6. Xác minh trạng thái xe và thông báo gửi đến chủ xe trên Firestore.

- Điều kiện kiểm thử: Có xe trong trạng thái "pending".
- Kết quả mong đợi: Trạng thái xe cập nhật đúng, thông báo gửi thành công.
- Công cụ kiểm thử: Manual testing, Firestore Console.

Quản lý người dùng:

- Mô tả kiểm thử: Kiểm tra hiển thị và quản lý danh sách người dùng.
- Bước kiểm thử:
 1. Mở màn hình quản lý người dùng (AdminUserListFragment).
 2. Kiểm tra danh sách người dùng hiển thị (tên, vai trò, trạng thái xác minh).
 3. Nhấn nút xóa, xác minh dialog xác nhận xuất hiện, và người dùng bị xóa khỏi Firestore.
 4. Nhấn vào một người dùng để yêu cầu bổ sung thông tin, kiểm tra thông báo gửi đến người dùng.
- Điều kiện kiểm thử: Có dữ liệu người dùng trong Firestore.
- Kết quả mong đợi: Danh sách hiển thị đúng, xóa và gửi thông báo thành công.
- Công cụ kiểm thử: Manual testing, Firestore Console.

Thống kê:

- Mô tả kiểm thử: Kiểm tra giao diện thống kê (chỉ là mác, chưa hoàn thiện).
- Bước kiểm thử:
 1. Mở màn hình thống kê (AdminStatisticsFragment).
 2. Kiểm tra biểu đồ cột hiển thị (dữ liệu giả lập).
- Điều kiện kiểm thử: Không yêu cầu dữ liệu thực.
- Kết quả mong đợi: Biểu đồ hiển thị đúng giao diện, không có dữ liệu thực.
- Công cụ kiểm thử: Manual testing.

Cài đặt:

- Mô tả kiểm thử: Kiểm tra giao diện cài đặt (chỉ là mác, chưa hoàn thiện).
- Bước kiểm thử:
 1. Mở màn hình cài đặt (AdminSettingsFragment).
 2. Kiểm tra các mục cài đặt hiển thị (dữ liệu giả lập).
 3. Điều kiện kiểm thử: Không yêu cầu chức năng thực.
- Kết quả mong đợi: Giao diện hiển thị đúng, không có chức năng thực.
- Công cụ kiểm thử: Manual testing.

1.4 Chức năng chung

Đăng nhập/Đăng ký:

- Mô tả kiểm thử: Kiểm tra quy trình tạo tài khoản, xác minh, và đăng nhập.
 - Bước kiểm thử:
 1. Mở giao diện đăng ký (RegisterActivity), nhập thông tin (tên, email, mật khẩu, số điện thoại), nhấn đăng ký.
 2. Mở giao diện xác minh số điện thoại (ValidatePhoneActivity), nhập mã OTP.
 3. Đăng nhập bằng tài khoản vừa tạo (LoginActivity), kiểm tra chuyển đến màn hình chính.
 4. Kiểm tra lỗi khi nhập sai email/mật khẩu hoặc số điện thoại không hợp lệ.
 - Điều kiện kiểm thử: Kết nối Firebase Authentication hoạt động.
 - Kết quả mong đợi: Tài khoản được tạo, xác minh, và đăng nhập thành công. Lỗi hiển thị đúng.
 - Công cụ kiểm thử: Manual testing, Firebase Console.

Giao diện giới thiệu:

- Mô tả kiểm thử: Kiểm tra hiệu ứng trượt và điều hướng.
- Bước kiểm thử:
 1. Mở ứng dụng, kiểm tra 4 màn hình giới thiệu (StartAppActivity).
 2. Vuốt trái/phải để chuyển trang, nhấn "Tiếp theo"/"Bắt đầu" để đến màn hình đăng nhập.
- Điều kiện kiểm thử: Không yêu cầu.
- Kết quả mong đợi: Hiệu ứng trượt mượt, điều hướng đúng.
- Công cụ kiểm thử: Manual testing.

Chat:

- Mô tả kiểm thử: Kiểm tra gửi và nhận tin nhắn.
- Bước kiểm thử:
 1. Mở màn hình chat (ChatFragment) từ giao diện khách hàng hoặc chủ xe.
 2. Gửi tin nhắn, kiểm tra hiển thị bong bóng tin nhắn đúng vai trò (người dùng/đối phương).
 3. Kiểm tra tin nhắn được lưu và hiển thị trên Firestore.
- Điều kiện kiểm thử: Có kết nối Firestore, hai tài khoản đang hoạt động.
- Kết quả mong đợi: Tin nhắn gửi và hiển thị đúng.
- Công cụ kiểm thử: Manual testing, Firestore Console.

1.5. Flow kiểm thử cụ thể

Flow: Tạo tài khoản - Xác nhận - Đăng nhập - Upload thông tin

- Cập nhật CCCD - Admin duyệt - Thông báo về khách hàng:

- Bước kiểm thử:
 1. Mở RegisterActivity, nhập thông tin (tên, email, mật khẩu, số điện thoại), nhấn đăng ký.
 2. Tại ValidatePhoneActivity, nhập mã OTP để xác minh.
 3. Đăng nhập bằng tài khoản vừa tạo tại LoginActivity.

4. Mở CustomerSettingFragment, cập nhật thông tin cá nhân (tên, địa chỉ).
 5. Mở CCCDActivity, tải lên ảnh CCCD/bằng lái, nhấn gửi.
 6. Chuyển sang tài khoản admin, mở AdminUserListFragment, chọn người dùng vừa tạo, nhấn duyệt (UserAdapter.java).
 7. Quay lại tài khoản khách hàng, mở CustomerNotificationFragment, kiểm tra thông báo "Tài khoản đã được duyệt".
- Điều kiện kiểm thử:
 - Firebase Authentication và Storage hoạt động.
 - Tài khoản admin duy nhất đã đăng nhập.
 - Kết quả mong đợi:
 - Tài khoản được tạo, xác minh, và duyệt thành công.
 - Thông báo hiển thị đúng trên giao diện khách hàng.
 - Công cụ kiểm thử: Manual testing, Firebase Console, Firestore Console.

Flow: Tạo tài khoản để thêm xe cho thuê - Upload thông tin - Thêm xe - Upload đủ thông tin - Chờ admin duyệt - Duyệt xong cho thuê:

- Bước kiểm thử:
 1. Mở RegisterActivity, tạo tài khoản chủ xe, nhập thông tin, nhấn đăng ký.
 2. Xác minh email bằng link được gửi về
 3. Đăng nhập tài khoản chủ xe tại LoginActivity.
 4. Mở OwnerSettingFragment, cập nhật thông tin cá nhân và tài liệu xác minh.

5. Mở AddVehicleActivity, nhập thông tin xe (tên, giá, biển số, thương hiệu, nhiên liệu, tốc độ, hộp số, số ghế/cửa), tải ảnh xe/giấy tờ, nhấn gửi.
 6. Chuyển sang tài khoản admin, mở AdminVehicleListFragment, chọn xe vừa thêm, nhấn duyệt (AdminVehicleDetailActivity).
 7. Quay lại tài khoản chủ xe, mở OwnerNotificationsFragment, kiểm tra thông báo "Xe đã được duyệt".
 8. Mở OwnerVehicleFragment, xác minh xe hiển thị trạng thái "verified" và sẵn sàng cho thuê.
- Điều kiện kiểm thử:
 - Firebase Authentication, Storage, và Firestore hoạt động.
 - Tài khoản admin duy nhất đã đăng nhập.
 - Kết quả mong đợi:
 - Tài khoản chủ xe được tạo, xe được thêm và duyệt thành công.
 - Thông báo hiển thị đúng, xe sẵn sàng cho thuê.
 - Công cụ kiểm thử: Manual testing, Firebase Console, Firestore Console, Cloudinary Dashboard.

2. Ghi nhận lỗi và cách xử lý

2.1. Lỗi thực tế: Truy vấn ràng buộc Firestore phức tạp gây không hiển thị dữ liệu:

Mô tả: Các truy vấn Firestore với nhiều điều kiện (ví dụ: whereEqualTo và whereIn trong ScheduleSelectViewModel.java) dẫn đến dữ liệu không được tải hoặc tải chậm, đặc biệt khi danh sách xe hoặc đơn đặt xe lớn.

Nguyên nhân: Ràng buộc truy vấn phức tạp làm tăng độ trễ hoặc vượt giới hạn Firestore (ví dụ: số lượng điều kiện trong whereIn).

Cách xử lý:

- Chuyển một số logic lọc dữ liệu sang client-side. Ví dụ: Trong ScheduleSelectViewModel.java, thay vì dùng whereIn với danh sách trạng thái, tải tất cả đơn đặt xe và lọc trạng thái tại client.
- Tối ưu hóa cấu trúc dữ liệu Firestore, sử dụng chỉ mục (index) cho các truy vấn phổ biến (tạo index trong Firebase Console).
- Thêm bộ nhớ đệm (cache) cục bộ để lưu trữ dữ liệu xe/đơn đặt xe, giảm số lượng truy vấn Firestore.

Kết quả sau xử lý: Dữ liệu hiển thị nhanh hơn, giảm lỗi không tải được.

2.2. Lỗi giả định: Tải ảnh xe thất bại do URL không hợp lệ trong Cloudinary:

Mô tả lỗi: Trong VehicleAdapter.java hoặc OwnerVehicleAdapter.java, hàm onBindViewHolder sử dụng Glide để tải ảnh xe từ Cloudinary, nhưng URL trả về null hoặc không hợp lệ, dẫn đến hiển thị placeholder (ic_car) thay vì ảnh thật.

Nguyên nhân: Lỗi trong quá trình tải ảnh lên Cloudinary (CloudinaryApi.java) hoặc dữ liệu Firestore lưu URL sai.

Cách xử lý:

- Thêm kiểm tra URL trước khi tải trong VehicleAdapter.java: Nếu URL null hoặc rỗng, bỏ qua Glide và đặt ảnh mặc định.
- Kiểm tra logic tải ảnh trong AddVehicleActivity.java và UpdateVehicleActivity.java, đảm bảo URL được lưu đúng vào Firestore sau khi tải lên Cloudinary.
- Ghi log chi tiết trong CloudinaryApi.java để xác định lỗi tải ảnh.

Kết quả sau xử lý: Ảnh xe hiển thị đúng hoặc fallback an toàn.

2.3. Lỗi giả định: Thanh toán ZaloPay thất bại do token hết hạn:

Mô tả lỗi: Trong CustomerBookingDetailActivity.java, hàm createOrderAndPay tạo đơn hàng thành công, nhưng khi gọi checkout, ZaloPay trả về lỗi token không hợp lệ, khiến thanh toán thất bại.

Nguyên nhân: Token ZaloPay (zptranstoken) có thời hạn ngắn, nhưng ứng dụng không xử lý trường hợp token hết hạn trước khi thanh toán.

Cách xử lý:

- Thêm logic retry trong createOrderAndPay: Nếu checkout thất bại với lỗi token, gọi lại createOrder để lấy token mới.
- Hiển thị thông báo người dùng rõ ràng (ví dụ: "Token hết hạn, vui lòng thử lại") thay vì lỗi chung.
- Kiểm tra thời gian tạo token trong CreateOrder.java, đảm bảo giao dịch hoàn thành trong thời hạn token.

Kết quả sau xử lý: Thanh toán ổn định hơn, người dùng được thông báo rõ ràng.

2.4. Lỗi giả định: Thông báo không gửi được đến khách hàng/chủ xe:

Mô tả lỗi: Trong AdminVehicleDetailActivity.java (hàm sendNotification) hoặc CustomerBookingDetailActivity.java (hàm createPaymentSuccessNotification), thông báo không được lưu vào Firestore hoặc không hiển thị trên giao diện người nhận.

Nguyên nhân: Lỗi kết nối Firestore, dữ liệu thông báo không được gán notificationId đúng, hoặc truy vấn Firestore trong NotificationAdapter.java không lấy được thông báo mới.

Cách xử lý:

- Thêm kiểm tra kết nối mạng trước khi gửi thông báo trong sendNotification và createPaymentSuccessNotification.

- Đảm bảo notificationId được cập nhật ngay sau khi thêm vào Firestore (đã có trong mã nguồn, nhưng cần kiểm tra lại).
- Tối ưu hóa truy vấn trong NotificationAdapter.java, sử dụng snapshot listener để tự động cập nhật thông báo mới.

Kết quả sau xử lý: Thông báo gửi và hiển thị đúng.

3. Đánh giá mức độ hoàn thiện

Dựa trên các chức năng đã triển khai và kết quả kiểm thử, mức độ hoàn thiện của ứng dụng T9_Car_Rental được đánh giá như sau:

Mức độ hoàn thiện tổng thể: Khoảng 70%.

Lý do:

- Hai chức năng chính (Quản lý xe và Quản lý người dùng của admin) đã hoàn thiện tốt, hoạt động ổn định với các quy trình duyệt/tù chói xe và người dùng, gửi thông báo chính xác.
- Chức năng thuê xe của khách hàng (từ xem danh sách xe, chọn thời gian, đến thanh toán) được triển khai đầy đủ, tích hợp ZaloPay thành công.
- Chức năng thêm/cập nhật xe và chấp nhận đơn đặt xe của chủ xe hoạt động tốt, hỗ trợ quy trình cho thuê xe.
- Tuy nhiên, một số hạn chế:
 - Chức năng Thống kê và Cài đặt của Admin chưa có logic thực tế, giảm tính hoàn thiện.
 - Chưa triển khai đăng nhập qua Facebook và Google, giới hạn tùy chọn xác thực.
 - Lỗi truy vấn Firestore phức tạp ảnh hưởng đến hiệu suất hiển thị dữ liệu, dù đã xử lý phần nào ở client-side.
 - Chức năng chat (ChatFragment.java) chỉ hỗ trợ cơ bản, chưa tối ưu cho trải nghiệm người dùng (ví dụ: không có thông báo thời gian thực).

Điểm mạnh:

- Quy trình thuê xe và quản lý xe/người dùng được tích hợp chặt chẽ với Firestore, đảm bảo tính nhất quán dữ liệu.
- Tích hợp ZaloPay ổn định, hỗ trợ thanh toán trực tuyến an toàn.
- Giao diện người dùng trực quan, sử dụng Material Design và font Roboto Flex, phù hợp với tiêu chuẩn Android.
- Hệ thống thông báo hoạt động tốt, hỗ trợ tương tác giữa khách hàng, chủ xe, và admin.

Điểm yếu:

- Hiệu suất truy vấn Firestore chưa tối ưu, cần cải thiện cấu trúc dữ liệu và thêm chỉ mục.
- Thiếu chức năng đăng nhập qua Facebook/Google, hạn chế trải nghiệm người dùng.
- Chức năng thống kê và cài đặt chưa hoàn thiện, không đáp ứng nhu cầu thực tế.
- Thiếu kiểm thử tự động (unit test, UI test) cho các chức năng chính, chủ yếu dựa vào kiểm thử thủ công.

VII. KẾT QUẢ VÀ ĐÁNH GIÁ

1. Tính khả thi của ứng dụng

Việc phát triển một ứng dụng thuê xe ô tô trong bối cảnh thị trường hiện nay không chỉ khả thi mà còn có tiềm năng lớn. Với sự gia tăng của nhu cầu di chuyển cá nhân mà không cần sở hữu xe, mô hình thuê xe đang trở thành một lựa chọn phổ biến, đặc biệt tại các đô thị lớn. Ứng dụng này có thể giải quyết được vấn đề tiếp cận xe dễ dàng, giúp người dùng thuê phương tiện phù hợp theo nhu cầu, từ xe du lịch đến xe chuyên dụng.

Tuy nhiên, để đảm bảo tính khả thi, ứng dụng phải tích hợp các cơ chế kiểm soát chặt chẽ về danh tính người thuê và chủ xe, thông qua các quy trình KYC như xác minh CCCD, giấy phép lái xe và trạng thái xe. Một thách thức lớn trong triển khai là đảm bảo tính minh bạch giữa chủ xe và khách thuê, cũng như có cơ chế xử lý tranh chấp hiệu quả. Ngoài ra,

yêu cầu tuân thủ các quy định pháp lý liên quan đến hợp đồng cho thuê xe cũng là một yếu tố cần xem xét.

2. Mức độ hoàn thành các tính năng so với kế hoạch

Khi đối chiếu với kế hoạch ban đầu, các tính năng cốt lõi như đăng ký tài khoản, xác minh danh tính, tìm kiếm xe theo nhu cầu, đặt thuê và thanh toán đều đã được xây dựng và hoạt động tương đối ổn định. Tuy nhiên, một số tính năng mở rộng như đánh giá chủ xe, hệ thống bảo hiểm tích hợp, kiểm duyệt giấy tờ xe tự động vẫn chưa hoàn thiện như mong muốn.

Hệ thống KYC đã được triển khai hiệu quả, giúp xác minh danh tính chủ xe và khách thuê theo các tiêu chí đã đặt ra, như khớp thông tin CCCD, email và số điện thoại duy nhất trong hệ thống. Đây là một bước tiến quan trọng nhằm đảm bảo an toàn cho giao dịch. Tuy nhiên, tính năng kiểm tra trạng thái xe cần được tối ưu để hạn chế việc thuê nhầm xe không sẵn sàng hoặc chưa đăng kiểm đầy đủ.

3. Khả năng triển khai thực tế (thị trường, đối tượng mục tiêu)

Về thị trường, ứng dụng này có tiềm năng cao khi nhắm đến khách hàng cá nhân có nhu cầu thuê xe theo thời gian ngắn hoặc dài hạn, bao gồm du khách, người đi công tác và cả những người cần phương tiện trong trường hợp khẩn cấp. Ngoài ra, mô hình này cũng phù hợp với các chủ xe muốn tăng thu nhập bằng cách cho thuê phương tiện nhàn rỗi.

Một thách thức quan trọng trong triển khai thực tế là xây dựng hệ sinh thái đủ mạnh để thu hút số lượng lớn chủ xe tham gia nền tảng. Việc thiết kế cơ chế định giá hợp lý, bảo hiểm giao dịch và xây dựng niềm tin giữa các bên sẽ quyết định mức độ thành công của ứng dụng. Trong giai đoạn đầu, có thể cần hợp tác với các doanh nghiệp cho thuê xe truyền thống để tạo độ phủ thị trường trước khi mở rộng mô hình chia sẻ phương tiện giữa cá nhân.

VIII. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

1. Tổng kết những gì đạt được

Cho đến thời điểm hiện tại, ứng dụng thuê xe ô tô đã hoàn thiện gần như toàn bộ các chức năng mà nhóm đưa ra, đồng thời khắc phục được đa số các lỗi về tính năng và logic trong quy trình vận hành của hệ thống và luồng hoạt động của ứng dụng. Hệ thống xác minh danh tính (KYC) được triển khai hiệu quả, đảm bảo tính hợp pháp của giao dịch, đồng thời giúp giảm thiểu rủi ro liên quan đến việc thuê xe từ các nguồn không đáng tin cậy. Tính năng tìm kiếm và đặt xe hoạt động tốt, giúp người dùng dễ dàng tiếp cận các phương tiện phù hợp với nhu cầu di chuyển của mình.

Bên cạnh đó, cơ chế kiểm soát dữ liệu đầu vào như xác minh CCCD, email, số điện thoại và giấy phép lái xe đã giúp tăng cường sự minh bạch giữa các bên tham gia hệ thống. Việc thiết lập hệ thống quy tắc nghiệp vụ chặt chẽ giúp tối ưu trải nghiệm người dùng và giảm thiểu các trường hợp vi phạm quy định.

2. Hướng cải tiến về mặt tính năng/ giao diện/ công nghệ

Việc cải tiến ứng dụng cần tập trung vào tối ưu trải nghiệm người dùng và nâng cao hiệu quả vận hành. Để giúp khách thuê xe tìm kiếm phương tiện một cách nhanh chóng và thuận tiện hơn, hệ thống lọc tìm kiếm có thể được cải tiến theo hướng hiển thị xe theo vị trí gần nhất hoặc phân loại theo các nhóm nhu cầu khác nhau như xe gia đình, xe du lịch, xe địa hình. Một yếu tố quan trọng trong nâng cao độ tin cậy của nền tảng là tích hợp hệ thống đánh giá và phản hồi giữa chủ xe và khách thuê, từ đó giúp người dùng có cái nhìn khách quan hơn trước khi đưa ra quyết định.

Bên cạnh đó, xây dựng cơ chế bảo hiểm cho giao dịch sẽ giúp đảm bảo quyền lợi cho cả chủ xe và khách thuê, giảm thiểu rủi ro khi xảy ra sự cố ngoài ý muốn. Trong lĩnh vực công nghệ, việc ứng dụng blockchain để lưu trữ lịch sử giao dịch và kiểm duyệt thông tin xe là một hướng đi tiềm năng, giúp nâng cao tính minh bạch và đảm bảo các thông tin liên quan không thể bị chỉnh sửa tùy tiện. Ngoài ra, cải thiện giao diện ứng dụng

theo hướng đơn giản, trực quan nhưng vẫn đảm bảo đầy đủ thông tin cần thiết sẽ giúp người dùng thao tác dễ dàng hơn, hạn chế sai sót trong quá trình thuê xe.

Ứng dụng có thể mở rộng theo mô hình đăng ký thành viên, giúp khách hàng quen thuộc có chính sách ưu đãi riêng. Nếu phát triển theo hướng cá nhân hóa trải nghiệm, hệ thống có thể gợi ý các phương tiện phù hợp dựa trên lịch sử thuê xe và hành vi sử dụng của khách. Tất cả những cải tiến này không chỉ giúp ứng dụng vận hành hiệu quả hơn mà còn tạo ra sự khác biệt trên thị trường, tăng khả năng cạnh tranh với các dịch vụ thuê xe truyền thống.

3. Tiềm năng thương mại hóa

Xét về khía cạnh kinh doanh, mô hình thuê xe trực tuyến có tiềm năng thương mại hóa cao, đặc biệt nếu có chiến lược tiếp cận thị trường phù hợp. Các nguồn doanh thu có thể bao gồm phí giao dịch từ người thuê xe, phí hoa hồng từ chủ xe, hoặc các dịch vụ liên quan như bảo hiểm xe, hỗ trợ tài chính khi thuê xe dài hạn.

Việc hợp tác với các công ty bảo hiểm, ngân hàng để cung cấp dịch vụ thanh toán trả góp có thể mở ra một hướng đi mới, giúp tăng khả năng tiếp cận khách hàng và mở rộng thị phần. Ngoài ra, mô hình thuê xe theo nhóm hoặc theo đăng ký thành viên có thể giúp tối ưu việc sử dụng xe, đồng thời tạo giá trị gia tăng cho nền tảng.

Về dài hạn, ứng dụng có thể mở rộng sang các thị trường khác ngoài Việt Nam, đặc biệt là những khu vực có nhu cầu cao về dịch vụ thuê xe nhưng chưa có nền tảng phù hợp. Sự kết hợp giữa công nghệ và tính thực tiễn sẽ quyết định sự thành công của ứng dụng, biến nó thành một giải pháp hữu ích trong ngành vận tải và chia sẻ phương tiện trong tương lai.

IX. TÀI LIỆU THAM KHẢO

[1] Khánh, L. Đ. D. (n.d.). *Slide chủ đề 4 – Thiết kế giao diện* [Slide bài giảng].

Trường Đại học Công nghệ Thông tin – ĐHQG TP.HCM.

- [2] Khánh, L. Đ. D. (n.d.). *Slide chủ đề – Các hướng phát triển trong TMĐT* [Slide bài giảng]. Trường Đại học Công nghệ Thông tin – ĐHQG TP.HCM.
- [3] Android Developers. (n.d.). *Guide to app architecture*. Android Developers. <https://developer.android.com/topic/architecture>
- [4] Firebase. (n.d.). *Firebase documentation*. <https://firebase.google.com/docs>
- [5] Deitel, P., & Deitel, H. (2017). *Android how to program* (3rd ed.). Pearson Education.
- [6] Meier, R. (2012). *Professional Android 4 Application Development*. Wiley.
- [7] Nguyễn Văn Hiệp. (2021). *Lập trình Android cơ bản với Java*. NXB Hồng Đức.
- [8] Trần Hữu Dũng. (2022). *Firebase – Giải pháp backend cho lập trình viên mobile*. Đại học Bách Khoa TP.HCM.

X. PHỤ LỤC

1. Business Rule Catalogue

STT	Tên Business Rule	ID	Mô tả Business Rule	Phân loại	Nguồn luật	Thực thể áp dụng	Hệ thống kiểm soát	Ghi chú
1	Họ tên khớp với CCCD	KYC -BRL 01	Họ tên người dùng phải trùng khớp với thông tin từ CCCD hoặc NCDC	Dữ liệu định danh	Nghị định 137/2015/NĐ-CP	Người dùng	So sánh dữ liệu CCCD và tên nhập	Áp dụng với tài khoản chủ xe

			nếu có tích hợp.					
2	Định dạng CCCD hợp lệ	KYC -BRL 02	Số CCCD phải là chuỗi 12 số hợp lệ, không có ký tự đặc biệt.	Dữ liệu phải là chuỗi 12 số hợp lệ, không có ký tự đặc biệt.	Cục Cảnh sát QLHC & NCDC	Người dùng	Regex kiểm định định dạng	Áp dụng với mọi người dùng
3	CCCD không trùng lặp	KYC -BRL 03	Số CCCD phải duy nhất trong toàn hệ thống.	Dữ liệu đầu vào	Chính sách nội bộ	Người dùng	Kiểm tra trùng lặp trong DB	Tránh tạo tài khoản ảo
4	Email định dạng hợp lệ	KYC -BRL 04	Email phải đúng định dạng và chứa ký tự “@”, tên miền hợp lệ.	Dữ liệu đầu vào	Chuẩn hệ thống	Người dùng	Regex + validat e	Áp dụng khi đăng ký
5	Email duy nhất	KYC -BRL 05	Email không được trùng với	Dữ liệu đầu vào	Chính sách hệ thống	Người dùng	Kiểm tra trùng	Tránh tạo nhiều tài khoản

			email đã đăng ký khác.				lắp DB	cùng email
6	SĐT hợp lệ	KYC -BRL 06	Số điện thoại phải bắt đầu bằng 0 và đủ 10 chữ số.	Dữ liệu đầu vào	Quy định hệ thống	Người dùng	Regex kiểm định	Có thẻ tích hợp xác minh SMS OTP
7	SĐT duy nhất	KYC -BRL 07	SĐT không được trùng với người dùng khác trong hệ thống.	Dữ liệu đầu vào	Chính sách hệ thống	Người dùng	Truy vấn DB khi tạo mới/sử a đổi	Cảnh báo trùng số
8	Giấy phép lái xe còn hạn	KYC -BRL 08	GPLX phải còn hiệu lực tại thời diễn thuê và rõ ràng khi upload.	Giấy tờ pháp lý	Bộ GTVT	Người dùng	Check ngày hết hạn, hình ảnh rõ	Có thẻ yêu cầu kiểm duyệt thủ công

Bảng 1: Tiêu chí liên quan đến người dùng

STT	Tên Busin ess Rule	ID	Mô tả Busines s Rule	Phân loại	Nguồn luật	Thực thể áp dụng	Hệ thống kiểm soát	Ghi chú
1	Biển số xe đúng định dạng	VEH-BRL01	Biển số xe phải theo định dạng được pháp luật Việt Nam công nhận.	Dữ liệu xe	Luật Giao thông đường bộ	Xe	Rege x định dạng và khôn g trùng DB	Áp dụng khi chủ xe đăng ký
2	Biển số duy nhất	VEH-BRL02	Mỗi xe chỉ có một biển số duy nhất trong hệ thống.	Ràng buộc logic	Quy định hệ thống	Xe	Kiểm tra trùng lặp DB	Tránh đăng trùng nhiều lần
3	Giấy tờ xe hợp lệ	VEH-BRL03	Xe phải có đăng ký, đăng kiểm, bảo hiểm rõ ràng,	Giấy tờ pháp lý	Bộ GTVT, nội bộ	Xe	Yêu cầu upload file, có thể kiểm duyệt	Có thể yêu cầu ảnh rõ nét, PDF

			còn hiệu lực.					
4	Loại xe thuộc danh sách cho phép	VEH-BRL0 4	Xe phải thuộc danh mục hạch thống hỗ trợ: Sedan, SUV, MPV, Xe máy, Xe bán tải,...	Danh mục hệ thống	Chính sách nội bộ	Xe	Kiểm tra lựa chọn trong danh sách hệ thống	Không cho nhập tự do
5	Số chỗ ngồi hợp lệ	VEH-BRL0 5	Số chỗ ngồi phải phù hợp với loại xe đăng ký.	Dữ liệu xác thực	Bộ GTVT	Xe	Ràng buộc kết hợp loại xe – số chỗ	Áp dụng với xe 4–7 chỗ trở lên
6	Trạng thái xe là “Sẵn sàng”	VEH-BRL0 6	Xe chỉ được hiển thị để thuê nếu có trạng thái “Sẵn	Trạng thái vận hành	Luồng nghiệp vụ	Xe	Kiểm tra trạng thái xe trong DB	Tránh thuê xe đang sửa hoặc đã đặt

			sàng” (`available').					
7	Giá thuê xe hợp lệ	VEH- BRL0 7	Giá thuê phải > 0 và nằm trong mức hợp lý (ví dụ: < 5 triệu/ngà y).	Chính sách giá	Chính sách hệ thống	Xe	Kiểm tra số và cảnh báo nếu vượt ngưỡng	Có thể có giá min/max

Bảng 2: Tiêu chí liên quan đến xe

STT	Tên Business Rule	ID	Mô tả Business Rule	Phân loại	Nguồn luật	Thực thể áp dụng	Hệ thống kiểm soát	Ghi chú
1	Xác minh số CCC D	CAR- BRL0 1	Số CCCD phải là chuỗi 12 chữ số, định dạng hợp lệ, không trùng lặp và trùng với dữ liệu	Dữ liệu đầu vào	Luật Căn cước công dân, NCDC	Người dùng	Rege x kiểm định + xác minh NCD C (nếu tích hợp)	Phát hiện sai định dạng hoặc trùng lặp

			NCDC nêu có.					
2	Email duy nhất	CAR-BRL0 2	Email phải duy nhất trong toàn hệ thống	Dữ liệu đầu vào	Quy định hệ thống	Người dùng	Rege x + kiểm tra duy nhất trong DB	Áp dụng khi đăng ký tài khoản
3	SĐT duy nhất	CAR-BRL0 3	Số điện thoại phải là 10 chữ số, bắt đầu bằng 0 và duy nhất trong hệ thống	Dữ liệu đầu vào	Quy định hệ thống	Người dùng	Rege x + kiểm tra duy nhất trong DB	Có thể tích hợp với SMS OTP để xác thực
4	Kiểm tra GPL X	CAR-BRL0 4	Giấy phép lái xe của khách thuê phải hợp lệ và còn thời hạn	Quy trình xác thực	Nghị định 123/2021/NĐ-C P	Người dùng	Kiểm tra file + xác nhận thời gian hiệu lực	Dùng khi người dùng là khách thuê

5	Trạng thái xe	CAR-BRL0 5	Xe chỉ được thuê khi trạng thái là 'Sẵn sàng'	Luôn ghiệp vụ	Quy định hệ thống	Xe	Kiểm tra trạng thái trong DB trước khi hiển thị	Trạng thái do chủ xe hoặc hệ thống cập nhật
6	Định dạng biển số xe	CAR-BRL0 6	Biển số xe phải theo định dạng chuẩn	Dữ liệu đầu vào	Luật Giao thông đường bộ	Xe	Rege x kiểm định định dạng	Kiểm tra khi chủ xe đăng thông tin xe
7	Giá thuê xe	CAR-BRL0 7	Giá thuê xe phải lớn hơn 0 và được chủ xe xác nhận	Ràng buộc nghiệp vụ	Chính sách giá hệ thống	Xe	Kiểm tra số + cảnh báo ngoài người	Hệ thống có thể thiết lập min/max
8	Tên xe hợp lệ	CAR-BRL0 8	Tên xe không được chứa ký tự đặc biệt hoặc	Dữ liệu đầu vào	Chuẩn nhập liệu	Xe	Rege x kiểm định	Áp dụng với nội dung hiển thị trên ứng dụng

			từ ngữ không phù hợp					
9	Giấy tờ xe hợp lệ	CAR-BRL09	Tất cả xe đăng lên phải có ảnh giấy tờ xe rõ ràng, đúng định dạng và còn hiệu lực	Tài liệu đính kèm	Yêu cầu quản trị hệ thống	Xe	Yêu cầu upload file hợp lệ, hệ thống lưu trữ	Có thể có chức năng duyệt của Admin
10	Chủ xe phải KYC trước khi đăng xe	CAR-BRL10	Tài khoản chủ xe phải được xác minh danh tính thành công trước khi được phép đăng tin xe	Phân quyề n hệ thốn g	Chính sách xác thực người dùng	Người dùng (chủ xe)	Ràng buộc logic + trạng thái tài khoản	Có thể yêu cầu xác thực thủ công hoặc tự động

Bảng 3: Nguyên tắc nghiệp vụ tổng quát

2. Data Dictionary

Tên hiển thị	Kiểu dữ liệu	Yêu cầu dữ liệu	Nguồn dữ liệu	Giá trị mặc định	Giá trị / Ý nghĩa
Họ và tên	Chuỗi ký tự (Text)	Bắt buộc	Người dùng		Họ và tên đầy đủ
Số CCCD	Chuỗi ký tự (Text)	Bắt buộc	Người dùng		12 số định danh
Email	Chuỗi ký tự (Text)	Bắt buộc	Người dùng		Địa chỉ email hợp lệ
Số điện thoại	Chuỗi ký tự (Text)	Bắt buộc	Người dùng		Số di động 10 chữ số

Bảng 4: Trường dữ liệu: Người dùng

Tên hiển thị	Kiểu dữ liệu	Yêu cầu dữ liệu	Nguồn dữ liệu	Giá trị mặc định	Giá trị / Ý nghĩa
Biển số xe	Chuỗi ký tự (Text)	Bắt buộc	Người dùng		Biển số đăng ký xe
Loại xe	Danh sách lựa chọn	Bắt buộc	Người dùng		Sedan, SUV, MPV...
Hình ảnh	Chuỗi ký tự (Text)	Không bắt buộc	Người dùng		URL ảnh minh họa xe
Trạng thái xe	Giá trị nhị phân	Bắt buộc	Người dùng	Sẵn sàng	Sẵn sàng / Không sẵn sàng

Bảng 5: Trường dữ liệu: Xe

Tên hiển thị	Kiểu dữ liệu	Yêu cầu dữ liệu	Nguồn dữ liệu	Giá trị mặc định	Giá trị / Ý nghĩa
Ngày bắt đầu thuê	Ngày tháng	Bắt buộc	Người dùng		VD: 01/01/2025
Ngày kết thúc thuê	Ngày tháng	Bắt buộc	Người dùng		Sau ngày bắt đầu

Giá thuê mỗi ngày	Số thập phân	Bắt buộc	Người dùng	0.00	Đơn vị: VND
Trạng thái giao dịch	Danh sách lựa chọn	Bắt buộc	Hệ thống	Đang chờ	Đang chờ, Đã xác nhận, Đã hoàn thành, Đã hủy

Bảng 6: Trường dữ liệu: Giao dịch thuê xe

Tên hiển thị	Kiểu dữ liệu	Yêu cầu dữ liệu	Nguồn dữ liệu	Giá trị mặc định	Giá trị / Ý nghĩa
Số sao	Số nguyên	Bắt buộc	Người dùng		Từ 1 đến 5
Nội dung đánh giá	Chuỗi ký tự (Text)	Không bắt buộc	Người dùng		Đánh giá của người thuê

Bảng 7: Trường dữ liệu: Đánh giá & Góp ý

Tên hiển thị	Kiểu dữ liệu	Yêu cầu dữ liệu	Nguồn dữ liệu	Giá trị mặc định	Giá trị / Ý nghĩa
Tài khoản quản trị	Chuỗi ký tự (Text)	Bắt buộc	Người dùng		Tên đăng nhập
Mật khẩu	Chuỗi ký tự (Text)	Bắt buộc	Người dùng		Được mã hóa

Bảng 8: Trường dữ liệu: Quản trị viên

3. Glossary

STT	Tiêu chí	Định nghĩa	Mô tả
1	Chủ xe	Người sở hữu xe và đăng ký xe để cho thuê trên nền tảng	Chủ xe cần đăng ký hồ gồm: họ tên, CCCD, số điện thoại, tài khoản ngân hàng để nhận thanh toán. Có thể quản lý nhiều xe.
2	Khách thuê	Người sử dụng dịch vụ để	Phải có tài khoản đã xác

		thuê xe trên nền tảng	minh CCCD và bằng lái hợp lệ. Có thể xem xe, đặt xe, thanh toán và đánh giá sau thuê.
3	Xe	Phương tiện được đưa lên nền tảng để cho thuê	Có biển số xe, mô tả xe, ảnh xe, giấy tờ xe đầy đủ
4	Biển số xe	Mã định danh hợp pháp của xe	Định dạng: XXX-XXX.XX ví dụ: 51A-123.45. Là duy nhất trong hệ thống.
5	Giá Thuê	Số tiền khách phải trả để thuê xe trong một khoảng thời gian nhất định	Có thể tính theo giờ hoặc theo ngày. Được hiển thị trong giao diện đặt xe. Đơn vị: VNĐ.
6	Giao dịch thuê xe	Hợp đồng thuê xe giữa chủ xe và khách thuê	Bao gồm thời gian thuê, giá, thông tin xe và người dùng
7	Trạng thái xe	Tình trạng hiện tại của xe trên nền tảng	Gồm: “Sẵn sàng”, “Đang thuê”, “Bảo trì”, “Không khả dụng”
8	OTP	Mã xác thực dùng một lần	Được gửi qua SMS hoặc email để xác minh danh tính người dùng trong quá trình đăng ký/đăng nhập
9	CCCD	Căn cước công dân	Bắt buộc phải cung cấp ảnh CCCD để xác minh khi đăng ký tài khoản

10	Giấy phép lái xe	Bằng lái xe của khách thuê	Phải được xác minh trước khi đặt xe
11	Đặt xe	Quy trình khách thuê chọn xe, đặt lịch và xác nhận thanh toán	Bao gồm chọn loại xe, thời gian thuê, xác minh giấy tờ và đặt cọc nếu cần
12	Xác minh tài khoản	Quy trình kiểm tra danh tính người dùng	Qua OTP, CCCD, bằng lái hoặc email
13	Ví điện tử	Phương thức thanh toán điện tử	Bao gồm: Momo, ZaloPay, VNPay...
14	Hủy giao dịch	Quy trình khách thuê hoặc chủ xe huỷ đặt xe trước giờ thuê	Có thể kèm theo phí hủy theo chính sách
15	Phí phạt	Khoản tiền cộng thêm khi người thuê vi phạm điều khoản thuê	Ví dụ: trả xe trễ, gây hư hỏng, không đủ đầy xăng khi trả xe
16	Feedback	Phản hồi đánh giá từ khách thuê hoặc chủ xe	Bao gồm đánh giá sao và bình luận chi tiết
17	Bảo hiểm xe	Bảo hiểm đi kèm với xe cho thuê	Có thể tính vào giá thuê hoặc mua riêng. Bao gồm bảo hiểm cơ bản hoặc mở rộng
18	Khuyến mãi	Chương trình giảm giá được áp dụng cho người thuê xe	Ví dụ: Giảm 10% khi thuê lần đầu, mã giảm giá theo chiến dịch
19	Mã giới thiệu	Mã mà người dùng chia sẻ để giới thiệu người	Cả người giới thiệu và người được giới thiệu có thể nhận

		dùng mới	ưu đãi
20	Lịch sử thuê xe	Danh sách tất cả các giao dịch thuê đã thực hiện	Bao gồm thời gian, loại xe, đánh giá, phí phạt nếu có
21	Thời gian	Khoảng thời gian người dùng thuê xe	Bao gồm ngày/giờ bắt đầu và kết thúc
22	Kỹ thuật viên	Nhân sự nội bộ phụ trách bảo trì xe	Truy cập hệ thống nội bộ để cập nhật tình trạng xe
23	QR Check-in/out	Quét mã QR để xác nhận nhận/trả xe	Thay cho ký giấy hoặc xác nhận thủ công
24	API	Giao diện lập trình ứng dụng giữa hệ thống và dịch vụ bên ngoài	Sử dụng để kết nối với hệ thống thanh toán, OTP, bản đồ...
25	Hệ thống	Thành phần phần mềm xử lý logic, lưu trữ, đồng bộ dữ liệu	Bao gồm server backend, cơ sở dữ liệu, dashboard quản trị

Bảng 9: Glossary (Bản Tiếng Việt)

No.	Term	Definition	Description
1	Vehicle Owner	A person who owns and lists a vehicle for rent on the platform	Must register with full name, ID card, phone number, and bank account. Can manage multiple vehicles.
2	Renter	A user who rents a vehicle	Requires a verified account

		on the platform	with ID and valid driver's license. Can browse, book, pay, and leave reviews.
3	Vehicle	The car/motorbike available for rental	Must include license plate, description, photos, and legal documents.
4	License Plate	Official vehicle identification code	Format: XXX-XXX.XX (e.g., 51A-123.45). Unique in the system.
5	Rental Price	The amount paid by renter to use a vehicle	Can be hourly or daily. Displayed on booking interface. Currency: VND.
6	Rental Transaction	The contract between vehicle owner and renter	Includes rental time, price, vehicle, owner and renter info. Has a unique transaction ID.
7	Vehicle Status	Current availability or condition of the vehicle	Includes: "Available", "In Use", "Under Maintenance", "Unavailable". Auto-updated via system.
8	OTP	One-Time Password	Sent via SMS/email to verify identity during registration or booking.
9	Citizen ID	Vietnamese national ID card	Required to verify renter/owner identity.

			Uploaded as photo.
10	Driving License	Legal document that permits a person to drive	Must be verified before booking.
11	Booking	Process of selecting and reserving a vehicle	Includes selecting vehicle, rental time, verifying documents, payment, and deposit if needed.
12	Account Verification	Identity validation process	Via OTP, Citizen ID, driver's license, or email.
13	E-wallet	Electronic payment method	Examples: Momo, ZaloPay, VNPay.
14	Cancel Transaction	Process to cancel a vehicle booking before rental start	May involve cancellation fee depending on policy.
15	Penalty Fee	Extra charge due to rental policy violation	E.g.: late return, damage, fuel not refilled.
16	Feedback / Review	Comments and rating from renter or owner	Includes star rating and optional text/photo feedback.
17	Vehicle Insurance	Insurance plan covering the rental vehicle	May be included in rental price or purchased separately.
18	Promotion	Discount offers for renters	E.g.: 10% off for first rental, coupon codes from campaigns.

19	Referral Code	Code shared to invite new users	Both referrer and invitee can receive rewards.
20	Rental History	List of past rental transactions	Shows time, type of vehicle, review, penalties, etc.
21	Rental Duration	Period for which a vehicle is rented	Includes start and end date/time.
22	Technician	Internal staff in charge of vehicle maintenance	Updates vehicle status and performs periodic checks.
23	QR Check-in/out	QR code scanning to confirm vehicle pickup/return	Digital alternative to manual signatures.
24	API	Application Programming Interface	Used to connect with external services such as OTP, payment, maps.
25	System	The backend logic and data processing platform	Includes server, database, admin dashboard, synchronization modules, and APIs.

Bảng 10: Glossary (Bản Tiếng Anh)