

TRƯỜNG ĐẠI HỌC TÀI NGUYÊN VÀ MÔI TRƯỜNG TP.HCM
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO ĐỒ ÁN TỐT NGHIỆP
NGÀNH CÔNG NGHỆ THÔNG TIN

ĐỀ TÀI
XÂY DỰNG ỨNG DỤNG ĐẶT XE VỚI FIREBASE VÀ
FLUTTER

Giảng viên hướng dẫn: **ThS. Đặng Đức Trung**
Sinh viên thực hiện: **Phạm Thị Thảo Dương**
MSSV: **0950080095**
Lớp: **09_DH_CNPM3**
Khóa: **2020-2024**

TP. Hồ Chí Minh, ngày 09 tháng 01 năm 2025

TRƯỜNG ĐẠI HỌC TÀI NGUYÊN VÀ MÔI TRƯỜNG TP.HCM
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO ĐỒ ÁN TỐT NGHIỆP
NGÀNH CÔNG NGHỆ THÔNG TIN

ĐỀ TÀI
XÂY DỰNG ỨNG DỤNG ĐẶT XE VỚI FIREBASE VÀ
FLUTTER

Giảng viên hướng dẫn: **ThS. Đặng Đức Trung**

Sinh viên thực hiện: **Phạm Thị Thảo Dương**

MSSV: **0950080095**

Lớp: **09_DH_CNPM3**

Khóa: **2020-2024**

TP. Hồ Chí Minh, ngày 09 tháng 01 năm 2025

LỜI CẢM ƠN

Đồ án tốt nghiệp là thành quả của quá trình học tập kéo dài suốt bốn năm đại học của em tại trường Đại Học Tài Nguyên Và Môi Trường. Để có thể thực hiện và hoàn thành được đồ án này, em đã nhận được sự giúp đỡ và hướng dẫn tận tình của các thầy cô và bạn bè trong khoa Công Nghệ Thông Tin, Trường Đại học Tài nguyên và Môi trường TPHCM.

Em xin gửi lời cảm ơn tới giảng viên hướng dẫn chính của em, thầy **Đặng Đức Trung**, người đã trực tiếp hướng dẫn và hỗ trợ em trong suốt quá trình thực hiện đề tài. Nhờ sự chỉ dẫn tận tình và nghiêm túc của thầy mà em đã có thể hoàn thành tốt đồ án tốt nghiệp của mình, đồng thời em còn học thêm được rất nhiều kỹ năng mềm từ thầy.

Thời gian 3 tháng không dài cũng không ngắn, mặc dù em đã nỗ lực hết mình để hoàn thiện đồ án này, tuy nhiên do em chưa có nhiều kinh nghiệm nên sẽ khó tránh khỏi những thiếu sót. Vì vậy, em rất mong nhận được góp ý và chỉ bảo của thầy cô để em có thể hoàn thiện đồ án của mình một cách tốt nhất.

Em xin chân thành cảm ơn!

LỜI CAM ĐOAN

Em xin cam đoan đồ án tốt nghiệp “**Xây dựng ứng dụng đặt xe với Firebase và Flutter**” là trung thực và không có bất kỳ sự sao chép của ai, do em tự nghiên cứu, đọc, dịch tài liệu, tổng hợp và thực hiện. Nội dung lý thuyết trong đồ án em có sử dụng một số tài liệu tham khảo như đã trình bày trong phần tài liệu tham khảo.

Em xin hoàn toàn chịu trách nhiệm trước nhà trường nếu phát hiện bất cứ sự sai phạm hay sao chép trong đề tài này!

Sinh viên thực hiện
(ký tên và ghi rõ họ tên)

NHẬN XÉT

(Của giảng viên hướng dẫn)

Kết luận: (Đồng ý hoặc Không đồng ý cho sinh viên nộp báo cáo)

.....

TP. Hồ Chí Minh, ngày... tháng....năm 2025

GIẢNG VIÊN HƯỚNG DẪN

ThS. Đặng Đức Trung

NHẬN XÉT

(Của giảng viên phản biện)

Kết luận: (Đồng ý hoặc Không đồng ý cho sinh viên nộp báo cáo)

.....

TP. Hồ Chí Minh, ngày... tháng....năm 2025

GIẢNG VIÊN PHẢN BIỆN

LỊCH LÀM VIỆC

Tuần	Nội dung	Địa điểm	Nhận xét của GV hướng dẫn	Chữ ký của GV hướng dẫn
1 (25/8- 31/8)	<ul style="list-style-type: none"> - Xác định yêu cầu dự án, phân tích yêu cầu người dùng. - Lập kế hoạch chi tiết cho đề tài. 	<ul style="list-style-type: none"> - Trường ĐH Tài Nguyên và Môi Trường TPHCM 		
2 (1/9- 7/9)	<ul style="list-style-type: none"> - Nghiên cứu và tìm hiểu về Firebase và Google API. - Phân tích và thiết kế cơ sở dữ liệu, xác định các bảng và mối quan hệ. 			
3 (8/9- 14/9)	<ul style="list-style-type: none"> - Thiết kế sơ bộ giao diện người dùng (UI) và trải nghiệm người dùng (UX) của ứng dụng. - Thiết kế kiến trúc hệ thống, xác định các thành phần chính. 			
4 (15/9- 21/9)	<ul style="list-style-type: none"> - Thiết lập môi trường phát triển cài đặt các công cụ. - Bắt đầu phát triển Backend với Firebase 			
5 (22/9- 28/9)	- Tiếp tục phát triển Backend, triển khai các chức năng chính			

	như đăng nhập, đăng ký, lưu trữ dữ liệu người dùng.			
6 (29/9-5/10)	<ul style="list-style-type: none"> - Bắt đầu phát triển frontend, xây dựng các màn hình chính của ứng dụng. - Tích hợp firebase Authentication. 			
7 (6/10-12/10)	<ul style="list-style-type: none"> - Phát triển chức năng đặt xe, tìm kiếm xe, hiển thị danh sách xe - Tích hợp Firebase Realtime Database 			
8 (13/10-19/10)	<ul style="list-style-type: none"> - Tích hợp Google Map API, hiển thị bản đồ, vị trí người dùng. - Phát triển chức năng chỉ đường và tính toán lộ trình. 			
9 (20/10-26/10)	<ul style="list-style-type: none"> - Phát triển chức năng theo dõi chuyến đi trong thời gian thực. - Tích hợp Firebase Cloud Functions. 			
10 (27/10-02/11)	<ul style="list-style-type: none"> - Phát triển chức năng thanh toán trực tuyến. - Kiểm thử và xử lý lỗi các chức năng đã phát triển. 			
11 (03/11-09/11)	<ul style="list-style-type: none"> - Hoàn thiện giao diện người dùng, đảm bảo tính thân thiện và dễ sử dụng. 			

	- Thu thập phản hồi về người dùng (giáo viên và sinh viên).			
12 (10/11- 16/11)	- Phân tích phản hồi từ người dùng. - Tiến hành sửa lỗi theo phân tích.			
13 (17/11- 23/11)	- Cải thiện và tối ưu hóa ứng dụng dựa trên phản hồi. - Tiếp tục kiểm thử và xử lý các lỗi phát sinh.			
14 (24/11- 30/11)	- Chuẩn bị tài liệu hướng dẫn sử dụng và tài liệu kỹ thuật. - Kiểm tra và sửa chữa các lỗi cuối cùng.			
15 (01/12- 07/12)	- Hoàn thiện báo cáo đồ án tốt nghiệp, chuẩn bị slide thuyết trình.			

TÓM TẮT

- Về vấn đề cần giải quyết: Đề tài này được thực hiện nhằm mục đích nâng cao trình độ của bản thân về đề tài, để hiểu rõ hơn về cách thức hoạt động của framework Flutter cũng như hệ thống quản lý của Firebase (đặc biệt là Firebase Realtime Database). Từ đó chúng ta áp dụng vào và xây dựng thành ứng dụng hoàn chỉnh.
- Kỹ thuật sử dụng:
 - Framework: Flutter
 - Ngôn ngữ: Dart
 - Cơ sở dữ liệu: Firebase
- Ứng dụng bao gồm các chức năng chính như:
 - Xác thực và quản lý tài khoản người dùng và tài xế bằng email.
 - Tìm kiếm và đặt xe
 - Tính năng bản đồ: Sử dụng Goong Maps để hiển thị vị trí hiện tại của người dùng, vị trí tài xế, điểm đón và điểm đến.
 - Thanh toán và quản lý lịch sử chuyến đi
 - Quản lý tài xế

MỤC LỤC

CHƯƠNG I: TỔNG QUAN TÀI LIỆU	1
1.1 Cơ sở lý thuyết.....	1
1.1.1 Phạm vi đề tài	1
1.1.2 Thiết bị phần mềm và cách kỹ thuật sử dụng:.....	2
1.1.3 Yêu cầu ứng dụng.....	2
1.2 Kỹ thuật	3
1.2.1 Nền tảng Flutter	3
1.2.2 Ngôn ngữ Dart	4
1.2.3 Cơ sở dữ liệu Firebase	5
1.2.4 Android Studio	6
CHƯƠNG II: PHƯƠNG PHÁP THỰC HIỆN	8
2.1 Các yêu cầu kỹ thuật.....	8
2.2 Quy trình tìm kiếm tài xế.....	8
2.3 Phân tích và thiết kế hệ thống.....	10
2.3.1 Biểu đồ Use Case.....	10
2.3.2 Biểu đồ hoạt động.....	17
2.3.3 Biểu đồ tuần tự	24
2.3.4 Sơ đồ ERD.....	37
2.3.5 Class Diagram	38
2.3.6 Deployment diagram (Sơ đồ triển khai hệ thống)	39
2.3.7 Thiết kế cơ sở dữ liệu	40
2.3.7.1 Mô tả dữ liệu.....	40
2.3.7.2 Cơ sở dữ liệu thử nghiệm	42
CHƯƠNG III: CÀI ĐẶT THỬ NGHIỆM, KẾT QUẢ VÀ ĐÁNH GIÁ	46

3.1 Mô tả kết quả	46
3.1.1 Ứng dụng của người dùng	46
3.1.2 Ứng dụng tài xế	60
3.1.3 Web Admin Panel.....	72
3.2 Test case	75
3.2.1 Chức năng Google Maps và Goong Maps	75
3.2.2 Chức năng đặt xe	75
3.2.3 Hiệu suất (Performance).....	76
3.2.4 UI/UX	77
KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	78
1 Đánh giá kết quả đạt được	78
2 Kiến thức lý thuyết được cung cấp.....	78
3 Thảo luận kết quả	78
4 Hướng phát triển.....	79
TÀI LIỆU THAM KHẢO	80

DANH MỤC CÁC BẢNG

Bảng 2.1 Quản lý Actor	10
Bảng 2.2 Quản lý nghiệp vụ	10
Bảng 2.3 Bảng cơ sở dữ liệu Users	40
Bảng 2.4 Bảng cơ sở dữ liệu Drivers	40
Bảng 2.5 Bảng cơ sở dữ liệu Online Drivers.....	41
Bảng 2.6 Bảng cơ sở dữ liệu Trip Requests	41
Bảng 2.7 Dữ liệu thử nghiệm - Users	42
Bảng 2.8 Dữ liệu thử nghiệm - Drivers	43
Bảng 2.9 Dữ liệu thử nghiệm – Online Drivers	44
Bảng 2.10 Dữ liệu thử nghiệm – Trip Requests	44
Bảng 3.1 Bảng Test Case chức năng Google Maps và Goong Maps.....	75
Bảng 3.2 Bảng Test Case chức năng đặt xe	75
Bảng 3.3 Bảng Test Case về hiệu suất	76
Bảng 3.4 Bảng Test Case về UX/UI.....	77

DANH MỤC CÁC HÌNH

Hình 1.1 Khái niệm Firebase	5
Hình 1.2 Android Studio	6
Hình 2.1 Biểu đồ Use Case tổng quát.....	11
Hình 2.2 Biểu đồ Use Case phân rã.....	13
Hình 2.3 Biểu đồ Use Case quản lý người dùng	14
Hình 2.4 Biểu đồ Use Case quản lý tài xế.....	15
Hình 2.5 Biểu đồ Use Case quản lý chuyến đi	16
Hình 2.6 Biểu đồ hoạt động – Đăng ký	17
Hình 2.7 Biểu đồ hoạt động – Đón khách	18
Hình 2.8 Biểu đồ hoạt động – Chặn (admin)	19
Hình 2.9 Biểu đồ hoạt động – Cập nhật thông tin cá nhân.....	20
Hình 2.10 Biểu đồ hoạt động – Đặt xe	21
Hình 2.11 Biểu đồ hoạt động – Thanh toán	22
Hình 2.12 Biểu đồ hoạt động – Tự thanh toán	23
Hình 2.13 Biểu đồ tuần tự đăng ký.....	24
Hình 2.14 Biểu đồ tuần tự đăng nhập.....	26
Hình 2.15 Biểu đồ tuần tự thanh toán	28
Hình 2.16 Biểu đồ tuần tự chẵn (admin)	29
Hình 2.17 Biểu đồ tuần tự - Cập nhật thông tin cá nhân	31
Hình 2.18 Biểu đồ tuần tự - Đặt xe	33
Hình 2.19 Biểu đồ tuần tự - Đón khách.....	35
Hình 2.20 Sơ đồ ERD	37
Hình 2.21 Class Diagram	38
Hình 2.22 Deployment diagram	39

Hình 3.1 Giao diện đăng ký.....	47
Hình 3.2 Giao diện đăng nhập.....	48
Hình 3.3 Giao diện thông báo lấy vị trí.....	49
Hình 3.4 Giao diện giới thiệu ứng dụng	50
Hình 3.5 Giao diện trang chủ	51
Hình 3.6 Giao diện thông tin cá nhân.....	52
Hình 3.7 Giao diện chỉnh sửa thông tin cá nhân	53
Hình 3.8 Giao diện tìm kiếm địa điểm	54
Hình 3.9 Giao diện đặt xe.....	55
Hình 3.10 Giao diện thông báo không có tài xế.....	56
Hình 3.11 Giao diện tìm kiếm tài xế	57
Hình 3.12 Giao diện khi đặt xe xong.....	58
Hình 3.13 Giao diện lịch sử chuyến đi	59
Hình 3.14 Giao diện thanh toán.....	60
Hình 3.15 Giao diện đăng ký tài xế.....	62
Hình 3.16 Giao diện đăng nhập tài xế	63
Hình 3.17 Giao diện trang chủ tài xế.....	64
Hình 3.18 Giao diện trạng thái tài xế	65
Hình 3.19 Giao diện trang chuyển đi.....	66
Hình 3.20 Giao diện trang hồ sơ tài xế.....	67
Hình 3.21 Giao diện yêu cầu chuyến đi	68
Hình 3.22 Giao diện trang thu nhập	69
Hình 3.23 Giao diện trong chuyến đi	70
Hình 3.24 Giao diện thanh toán.....	71
Hình 3.25 Giao diện trang admin	72
Hình 3.27 Giao diện quản lý người dùng	73

Hình 3.28 Giao diện quản lý chuyến đi.....73

Hình 3.29 Xem chi tiết chuyến đi.....74

KÝ HIỆU CÁC CỤM TỪ VIẾT TẮT

STT	Từ viết tắt	Mô tả
1	UX	User Experience: trải nghiệm người dùng
2	UI	User Interface: giao diện người dùng
3	Admin	Người quản trị
4	IDE	Integrated Development Environment
5	push notifications	Thông báo đầy
6	SDK	Software Development Kit
7	Back-end	Cơ sở dữ liệu hạ tầng
8	Comments	Bình luận
9	User	Người dùng
10	API	Application Programming Interface là một giao diện cho phép các ứng dụng hoặc hệ thống khác nhau giao tiếp với nhau.

MỞ ĐẦU

1 Cơ sở khoa học và tính thực tiễn của đề tài

1.1 Tính thực tiễn của đề tài

Việc di chuyển trong các thành phố lớn ngày càng trở nên phức tạp do tắc nghẽn giao thông và nhu cầu đi lại cao. Đề tài “**Xây dựng ứng dụng đặt xe với Firebase và Flutter**” giúp giải quyết vấn đề này bằng cách cung cấp một phương tiện di chuyển tiện lợi, nhanh chóng và an toàn. Ứng dụng này mang lại tiện ích đáng kể cho cả người dùng và tài xế. Người dùng có thể dễ dàng đặt xe, theo dõi hành trình và thanh toán, trong khi tài xế có thể quản lý các chuyến đi, theo dõi thu nhập và tối ưu hóa lộ trình, cải thiện hiệu suất và trải nghiệm cho cả hai bên. Ngoài ra, việc phát triển ứng dụng đặt xe không chỉ tạo ra lợi ích kinh tế cho các nhà phát triển mà còn thúc đẩy sự cạnh tranh trong ngành vận tải, từ đó cải thiện chất lượng dịch vụ và giá cả hợp lý hơn cho người tiêu dùng. Các công nghệ và giải pháp được phát triển trong đề tài này cũng có thể áp dụng vào nhiều lĩnh vực khác như giao hàng, vận chuyển hàng hóa và các dịch vụ khác liên quan đến di chuyển và logistics.

1.2 Cơ sở khoa học

Đề tài dựa trên các lý thuyết và phương pháp thiết kế hệ thống thông tin, bao gồm phân tích yêu cầu, thiết kế cơ sở dữ liệu và phát triển phần mềm. Firebase cung cấp nền tảng phát triển backend mạnh mẽ với các dịch vụ như Authentication (xác thực người dùng), Firestore (cơ sở dữ liệu NoSQL), Cloud Functions (xử lý sự kiện phía server) và Cloud Messaging (gửi thông báo đẩy). Sử dụng Firebase giúp chúng ta rút ngắn thời gian phát triển đồng thời còn giúp tăng khả năng bảo mật và độ tin cậy của hệ thống. Google Maps API cung cấp các dịch vụ liên quan đến bản đồ và định vị như hiển thị bản đồ, tính toán lộ trình, tìm kiếm địa điểm và chuyển đổi địa chỉ, giúp tạo ra trải nghiệm người dùng mượt mà và trực quan. Việc tích hợp các API hiện đại để phát triển các tính năng phức tạp như định vị thời gian thực, quản lý dữ liệu đồng bộ và tương tác người dùng hiệu quả, tạo ra một sản phẩm hoàn chỉnh, đáp ứng nhu cầu thực tế của người dùng và tài xế.

Nhờ vào cơ sở khoa học vững chắc và tính thực tiễn cao, đề tài “**Xây dựng ứng dụng đặt xe với Firebase và Flutter**” không chỉ mang lại giá trị học thuật mà còn có tiềm năng ứng dụng rộng rãi trong ngành vận tải, góp phần vào sự phát triển của nền kinh tế số và nâng cao chất lượng cuộc sống.

2 Mục tiêu nghiên cứu

Mục đích nghiên cứu của đề tài là tìm hiểu và áp dụng các ngôn ngữ lập trình, công cụ phát triển phần mềm, và các nghiệp vụ liên quan để xây dựng một ứng dụng đặt xe hoàn chỉnh, hiệu quả và thân thiện với người dùng. Cụ thể, ngôn ngữ Dart sử dụng Flutter cho việc phát triển ứng dụng di động đa nền tảng. Các công cụ phát triển phần mềm sẽ bao gồm Firebase để xây dựng hệ thống backend mạnh mẽ và linh hoạt với các dịch vụ như Authentication, Realtime Database, Cloud Functions và Cloud Messaging, và Goong Maps API để tích hợp các dịch vụ bản đồ và định vị như hiển thị bản đồ, tính toán lộ trình, tìm kiếm địa điểm và chuyển đổi địa chỉ. Ngoài ra, cũng sẽ sử dụng các công cụ phát triển là Android Studio và Visual Studio Code. Đề tài cũng sẽ nghiên cứu các nghiệp vụ liên quan như quản lý người dùng và tài xế, quy trình đặt xe, theo dõi và định vị, thanh toán trực tuyến, và hệ thống đánh giá và phản hồi. Mục tiêu cuối cùng là tạo ra một ứng dụng đáp ứng nhu cầu thực tế của người dùng và tài xế, cải thiện trải nghiệm di chuyển và mang lại giá trị kinh tế.

3 Đối tượng và phạm vi nghiên cứu

Đề tài “**Xây dựng ứng dụng đặt xe với Firebase và Flutter**” tập trung vào nghiên cứu và triển khai một hệ thống đặt xe trực tuyến trên nền tảng di động. Đối tượng nghiên cứu bao gồm các công nghệ chính như Firebase, một hệ thống cơ sở dữ liệu thời gian thực và các dịch vụ liên quan do Google phát triển, và Goong Maps API, bộ API cung cấp chức năng bản đồ và định vị.

Phạm vi nghiên cứu của đề tài bao gồm việc phân tích yêu cầu hệ thống để xác định các chức năng cần thiết, thiết kế hệ thống với kế hoạch kiến trúc, cơ sở dữ liệu và giao diện người dùng, phát triển ứng dụng sử dụng Firebase cho backend và Goong Maps API cho chức năng bản đồ và định vị. Quá trình phát triển cũng bao gồm kiểm thử ứng dụng để đảm bảo tính ổn định và bảo mật, triển khai trên các nền tảng di động, thu thập và xử lý phản hồi người dùng để cải tiến ứng dụng. Thông qua đề tài này, chúng ta sẽ

nắm vững kiến thức về phát triển ứng dụng di động, sử dụng các công nghệ hiện đại, và quy trình phát triển phần mềm từ phân tích yêu cầu đến triển khai và bảo trì hệ thống.

4 Phương pháp nghiên cứu

Phương pháp phân tích tổng hợp lý thuyết:

- Nghiên cứu tài liệu: Tìm hiểu và phân tích các tài liệu liên quan đến các công nghệ sẽ sử dụng trong đề tài như Firebase, Google Maps API, Flutter. Điều này bao gồm đọc sách, bài báo khoa học, tài liệu hướng dẫn chính thức, và các tài nguyên trực tuyến khác để hiểu rõ về cách thức hoạt động và ứng dụng của các công nghệ này.
- Phân tích và thiết kế hệ thống: Áp dụng các mô hình lý thuyết như các sơ đồ luồng dữ liệu (Data Flow Diagrams), và các sơ đồ quan hệ thực thể (Entity-Relationship Diagrams) để thiết kế cấu trúc hệ thống. Xác định các yêu cầu chức năng và phi chức năng của hệ thống thông qua các phương pháp phỏng vấn, khảo sát, và quan sát người dùng thực tế.
- Phân tích ứng dụng hiện có: Nghiên cứu và phân tích các ứng dụng đặt xe hiện có trên thị trường như Uber, Grab để hiểu rõ về các tính năng, giao diện người dùng, và trải nghiệm người dùng. Điều này giúp xác định các yêu cầu cần thiết và các yếu tố có thể cải tiến trong ứng dụng mới.

Phương pháp thực nghiệm khoa học:

- Thu thập yêu cầu:
 - Phỏng vấn và khảo sát người dùng: Tiến hành phỏng vấn và khảo sát người dùng tiềm năng để hiểu rõ nhu cầu và mong muốn của họ đối với ứng dụng đặt xe.
 - Phân tích tài liệu: Xem xét các tài liệu và nghiên cứu liên quan để xây dựng cơ sở lý thuyết cho ứng dụng.
- Phân Tích Dữ Liệu:
 - Thu thập và phân tích dữ liệu từ các cuộc khảo sát và phỏng vấn để xác định các yêu cầu chức năng và phi chức năng của ứng dụng.
 - Phân tích dữ liệu về tương tác và hiệu quả học tập của sinh viên với các ứng dụng hiện có để rút ra những điểm mạnh và điểm yếu cần cải thiện.

- Thủ Nghiệm và Đánh Giá:

- Triển khai thử nghiệm ứng dụng với một nhóm sinh viên và giảng viên trong một khoảng thời gian nhất định để thu thập phản hồi.
- Dựa trên phản hồi từ người dùng, tiến hành kiểm thử và chỉnh sửa ứng dụng để cải thiện hiệu suất và trải nghiệm người dùng, đảm bảo ứng dụng đáp ứng đúng nhu cầu và mong đợi của người dùng.

5 Bố cục đề tài

Đề tài này là tài nghiên cứu cá nhân của em dưới sự hướng dẫn của giảng viên nhằm mục đích nâng cao trình độ, nhận thức của bản thân và cách thức hoạt động của nền tảng công nghệ Flutter và hệ thống quản lý của Firebase.

Bố cục đề tài gồm 4 phần:

- **Chương 1: Tổng quan tài liệu**
- **Chương 2: Phương pháp thực hiện**
- **Chương 3: Cài đặt thử nghiệm**
- **Chương 4: Kết luận và hướng phát triển**

CHƯƠNG I: TỔNG QUAN TÀI LIỆU

1.1 Cơ sở lý thuyết

1.1.1 Phạm vi đề tài

Đề tài tập trung xây dựng một hệ thống đặt xe đơn giản, bao gồm ba thành phần chính: ứng dụng người dùng (user app), ứng dụng tài xế (driver app) và bảng điều khiển quản trị viên (admin web dashboard). Hệ thống được xây dựng và thiết kế với các chức năng cơ bản nhằm đáp ứng nhu cầu quản lý và sử dụng dịch vụ đặt xe trong một phạm vi nhỏ, không yêu cầu tích hợp đầy đủ các chức năng phức tạp như hệ thống đặt xe thực tế. Cụ thể:

- **Ứng dụng người dùng (User App):**

- Cho phép người dùng đăng ký tài khoản và đăng nhập vào ứng dụng.
- Người dùng có thể cập nhật thông tin cá nhân.
- Xem danh sách lịch sử đặt xe.
- Chức năng đặt chuyến đi và theo dõi tài xế theo thời gian thực.

- **Ứng dụng tài xế (Driver App):**

- Tài xế có thể đăng ký và đăng nhập vào ứng dụng.
- Cập nhật trạng thái hoạt động (online/offline).
- Xem và chỉnh sửa thông tin tài khoản.
- Nhận yêu cầu chuyến đi từ người dùng và thực hiện chuyến đi.

- **Bảng điều khiển quản trị viên (Admin Web Dashboard):**

- Xem thông tin người dùng và tài xế.
- Xem thông tin cơ bản về các chuyến đi.

❖ **Hạn chế của hệ thống:**

- Không hỗ trợ thanh toán trực tuyến.
- Không xây dựng các thuật toán phức tạp như gán tài xế tự động hoặc tối ưu hóa tuyến đường.

Mục tiêu: Đề tài được xây dựng với mục đích xây dựng một ứng dụng đặt xe đơn giản, dễ phát triển với các chức năng cơ bản cho người dùng và cả tài xế. Từ đó có thể phát triển thêm các tính năng phức tạp hơn để đề tài được hoàn thiện tốt hơn trong tương lai.

1.1.2 Thiết bị phần mềm và cách kỹ thuật sử dụng:

Ngôn ngữ lập trình: Dart.

Framework: Flutter

Cơ sở dữ liệu: Firebase Realtime Database

Backend: Firebase

Công cụ hỗ trợ: Android Studio.

1.1.3 Yêu cầu ứng dụng

❖ Ứng dụng người dùng (User App)

- Chức năng chính
 - Tìm kiếm xem tài xế nào gần vị trí người dùng nhất.
 - Hiển thị bản đồ và định vị vị trí của người dùng.
 - Theo dõi chuyến đi.
 - Thanh toán sau khi kết thúc chuyến đi.
- Giao tiếp (input-output)
 - **Input:** Nhận thông tin tài xế từ Firebase Realtime Database.
 - **Output:** Gửi vị trí và yêu cầu đặt xe đến Firebase.

❖ Ứng dụng Tài xế

- Chức năng chính
 - Hiển thị yêu cầu đặt xe từ người dùng.
 - Theo dõi vị trí của khách hàng.
 - Cập nhật trạng thái chuyến đi (đã nhận, đang di chuyển, hoàn thành).
- Giao tiếp (in-out)
 - **In:** Nhận yêu cầu từ Firebase (địa chỉ, thông tin khách).
 - **Out:** Gửi trạng thái chuyến đi và vị trí của tài xế lên Firebase.

❖ Firebase

- Thành phần

- **Realtime Database:** Lưu trữ và đồng bộ dữ liệu thời gian thực (vị trí tài xế, yêu cầu của chuyến đi).
- **Authentication:** Xác thực tài khoản (email, số điện thoại).
- **Cloud Messaging:** Gửi thông báo đến thiết bị.
- **Storage:** Lưu trữ hình ảnh

- Vai trò: Trung gian lưu trữ và truyền tải dữ liệu giữa các ứng dụng.

❖ Goong Maps API

- Vai trò: Cung cấp bản đồ, định vị và chỉ đường.

- Giao tiếp (in-out)

- **In:** Nhận vị trí từ người dùng hoặc tài xế.
- **Out:** Trả về thông tin bản đồ, chỉ đường, và tính toán khoảng cách và thời gian.

1.2 Kỹ thuật

1.2.1 Nền tảng Flutter

Flutter là một framework mã nguồn mở được phát triển bởi Google, nhằm mục đích giúp các lập trình viên xây dựng các ứng dụng đa nền tảng từ một bộ mã nguồn duy nhất. Với Flutter, bạn có thể tạo ứng dụng cho iOS, Android, web, máy tính để bàn (Windows, macOS, Linux), và thậm chí các thiết bị nhúng như đồng hồ thông minh hay hệ thống xe hơi.

❖ Các đặc điểm nổi bật của Flutter

- **Có thể chạy trên nhiều nền tảng:** cho phép chạy trên nhiều nền tảng như iOS, Android và cả web.
- **Hiệu năng cao:** Sử dụng công nghệ đồ họa tăng tốc phần cứng và engine Flutter riêng biệt (được viết bằng C++), các ứng dụng Flutter có hiệu năng tốt trên mọi thiết bị.

- **Xây dựng giao diện mạnh mẽ và đẹp mắt:** Flutter có thư viện widget phong phú với mục đích hỗ trợ thiết kế giao diện trực quan hơn và đáp ứng trên mọi màn hình và tương thích đến từng pixel.
- **Tính năng tải lại:** Khi lập trình viên thay đổi gì trong ứng dụng thì tính năng này xem thay đổi trong mã ngay lập tức mà không cần khởi động lại ứng dụng, tăng hiệu suất làm việc.

Flutter được viết bằng ngôn ngữ Dart, cung cấp môi trường lập trình hiện đại với cú pháp dễ học và hiệu quả trong việc phát triển giao diện người dùng.

1.2.2 Ngôn ngữ Dart

Dart là một ngôn ngữ lập trình tối ưu hóa cho máy khách, giúp phát triển các ứng dụng nhanh chóng trên nhiều nền tảng. Mục tiêu của nó là cung cấp một ngôn ngữ hiệu quả cho việc phát triển ứng dụng đa nền tảng, đồng thời kết hợp với một nền tảng runtime linh hoạt để hỗ trợ các framework ứng dụng.

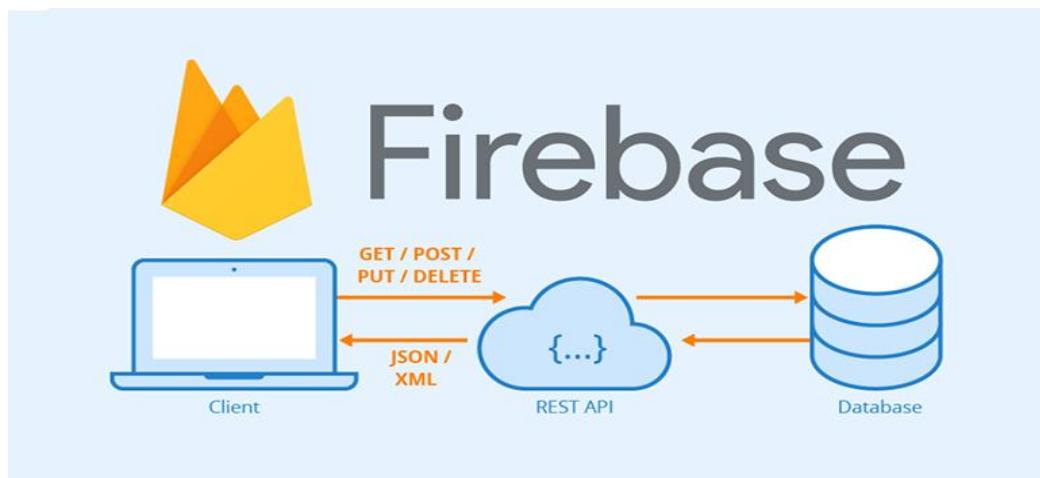
Mỗi ngôn ngữ lập trình đều có phạm vi kỹ thuật riêng – những quyết định trong quá trình phát triển sẽ định hình khả năng và ưu điểm của ngôn ngữ đó. Dart được thiết kế đặc biệt để phù hợp với việc phát triển ứng dụng máy khách, ưu tiên tối đa cả quá trình phát triển và trải nghiệm sản phẩm chất lượng cao trên nhiều nền tảng biên dịch, bao gồm web, di động và máy tính để bàn.

Dart không chỉ là nền tảng của Flutter, mà còn cung cấp ngôn ngữ và runtime hỗ trợ cho các ứng dụng Flutter. Bên cạnh đó, Dart còn hỗ trợ nhiều công cụ cốt lõi cho lập trình viên, như định dạng mã, phân tích và kiểm tra mã nguồn [10].

❖ Ứng dụng của Dart

Dart không chỉ giúp phát triển ứng dụng trên một nền tảng, mà còn mở ra cơ hội sáng tạo trên cả Android và iOS. Trong hành trình đó, Flutter, một framework mạnh mẽ được Google phát triển, nổi lên như công cụ vượt trội. Flutter cho phép xây dựng các ứng dụng đẹp mắt trên cả hai nền tảng này, tất cả chỉ với một mã nguồn duy nhất [6].

1.2.3 Cơ sở dữ liệu Firebase



Hình 1.1 Khái niệm Firebase

Firebase ra đời vào năm 2011 và được Google mua lại vào năm 2014, trở thành một phần quan trọng trong hệ sinh thái phát triển ứng dụng của Google. Nó giúp các nhà phát triển tích hợp các tính năng mạnh mẽ mà không phải quản lý hạ tầng phức tạp. Vì vậy, Firebase còn được gọi là Google Firebase [4].

Firebase cung cấp các SDK cho phía client, cho phép ứng dụng tương tác trực tiếp với các dịch vụ backend mà không cần thiết lập middleware giữa ứng dụng và dịch vụ [4].

❖ Firebase Realtime Database

Firebase Realtime Database là cơ sở dữ liệu đám mây lưu trữ dữ liệu dưới dạng JSON. Dữ liệu được đồng bộ hóa theo thời gian thực trên tất cả các ứng dụng được kết nối. Khi phát triển ứng dụng đa nền tảng (Apple, Android, JavaScript), tất cả ứng dụng khách dùng chung một phiên bản cơ sở dữ liệu và tự động cập nhật khi có dữ liệu mới [5].

Đặc điểm nổi bật

- Đồng bộ hóa thời gian thực: Với tính năng này thay vì dùng http để yêu cầu thì nó dùng dữ liệu đồng bộ hóa. Tức là khi dữ liệu thay đổi thì nó sẽ cập nhật liên tục và chỉ trong vòng vài mili giây.

- Lưu trữ ngoại tuyến: Vì dữ liệu được lưu trữ ở ổ đĩa nên khi không có mạng các ứng dụng Firebase vẫn có thể phản hồi được. Sau đó khi có kết nối mạng trở lại thì dữ liệu sẽ được đồng bộ hóa lại với máy chủ.
- Khả năng mở rộng: Realtime Database có thể mở rộng để phục vụ hàng triệu người dùng đồng thời.
- Quyền truy cập và bảo mật
 - Firebase sử dụng **Firebase Authentication** để xác thực người dùng.
 - Kiểm soát quyền truy cập vào dữ liệu trong mỗi cơ sở dữ liệu bằng Firebase Realtime Database Security Rules tùy chỉnh cho mỗi phiên bản cơ sở dữ liệu [5].
- Dễ dàng tích hợp: Cung cấp SDK cho nhiều nền tảng: Android, iOS, Web, và các ứng dụng server.

1.2.4 Android Studio



Hình 1.2 Android Studio

Android Studio là môi trường phát triển ứng dụng Android, được xây dựng trên nền tảng IntelliJ IDEA, một công cụ mạnh mẽ dành cho nhà phát triển và trình soạn thảo mã. Nhờ đó, Android Studio cung cấp nhiều tính năng giúp tăng năng suất khi phát triển ứng dụng Android, bao gồm [7]:

- Hệ thống xây dựng linh hoạt dựa trên Gradle.
- Trình mô phỏng nhanh với nhiều tính năng.
- Môi trường phát triển tích hợp hỗ trợ tất cả thiết bị Android.
- Tính năng **Live Edit** cho phép cập nhật trực tiếp các thành phần trong trình mô phỏng và trên thiết bị thực.

- Mã mẫu và tích hợp GitHub để giúp xây dựng tính năng phổ biến và nhập mã mẫu.
- Các khung thử nghiệm đa dạng và công cụ kiểm tra ứng dụng.
- Công cụ tìm lỗi mã nguồn (lint) giúp phát hiện vấn đề về hiệu suất, khả năng sử dụng, tương thích phiên bản và các lỗi khác.
- Hỗ trợ phát triển với C++ và NDK.
- Tích hợp sẵn hỗ trợ Google Cloud Platform, dễ dàng tích hợp Google Cloud Messaging và App Engine.

CHƯƠNG II: PHƯƠNG PHÁP THỰC HIỆN

2.1 Các yêu cầu kỹ thuật

❖ Yêu cầu phần mềm

- Ngôn ngữ lập trình: Dart.
- Framework phát triển ứng dụng: Flutter.
- Hệ quản trị cơ sở dữ liệu: Firebase Realtime Database

❖ Yêu cầu FrontEnd

- Flutter Framework: Hỗ trợ phát triển ứng dụng đa nền tảng (Android).
- Giao diện người dùng (UI/UX)
 - Hiển thị giao diện bản đồ (Google Maps).
 - Hiển thị thông tin chuyến đi, tài xế, trạng thái chuyến đi.
- Tích hợp bản đồ
- Theo dõi vị trí tài xế theo thời gian thực: Tích hợp plugin **geolocator**.

❖ Yêu cầu BackEnd

- Firebase Realtime Database
- Firebase Authentication
- Firebase Cloud Messaging (FCM)

❖ Yêu cầu tích hợp API

- Goong Maps API
 - Directions API: Tìm đường đi giữa điểm đón và trả khách.
 - Geocoding API: Chuyển đổi địa chỉ thành tọa độ GPS.
 - Places API: Gợi ý địa điểm cho khách hàng.

2.2 Quy trình tìm kiếm tài xế

- Quét các tài xế đang online trong bán kính 22km => lưu vào danh sách ‘availableNearbyOnlineDriversList’
- Kiểm tra danh sách tài xế khả dụng
 - Nếu danh sách ‘availableNearbyOnlineDriversList’ rỗng
 - Thông báo: “Không có tài xế”

- Hủy yêu cầu chuyến đi và dừng tìm kiếm
- Ngược lại và tiếp tục
- Chọn tài xế đầu tiên trong danh sách
 - Lấy tài xế hiện tại từ 'availableNearbyOnlineDriversList[0]'.
 - Gửi yêu cầu tới tài xế đó bằng cách
 - Ghi `tripID` vào trường `newTripStatus` của tài xế trên Firebase.
 - Trích xuất `deviceToken` của tài xế từ Firebase.
 - Gửi thông báo đẩy (Push Notification) đến tài xế với `tripRequestRef`.
- Chờ phản hồi từ tài xế (dùng bộ đếm thời gian)
 - Thiết lập bộ đếm thời gian (timeout = 20 giây)
 - Nếu `newTripStatus` của tài xế là "accepted":
 - ✓ Tài xế đã nhận chuyến.
 - ✓ Kết thúc bộ đếm và dừng tìm kiếm.
 - Nếu tài xế không phản hồi sau 20 giây:
 - ✓ Đặt trạng thái `newTripStatus` của tài xế là "timeout".
 - ✓ Loại tài xế khỏi danh sách `availableNearbyOnlineDriversList`.
 - ✓ Chọn tài xế tiếp theo trong danh sách.
- Chọn tài xế tiếp theo trong danh sách:
 - Nếu danh sách vẫn còn tài xế khả dụng: thì chọn tài xế đầu tiên trong danh sách.
 - Ngược lại, nếu danh sách trống:
 - Thông báo: "Không có tài xế".
 - Hủy yêu cầu chuyến đi và dừng tìm kiếm.

2.3 Phân tích và thiết kế hệ thống

2.3.1 Biểu đồ Use Case

❖ Các Actor

Bảng 2.1 Quản lý Actor

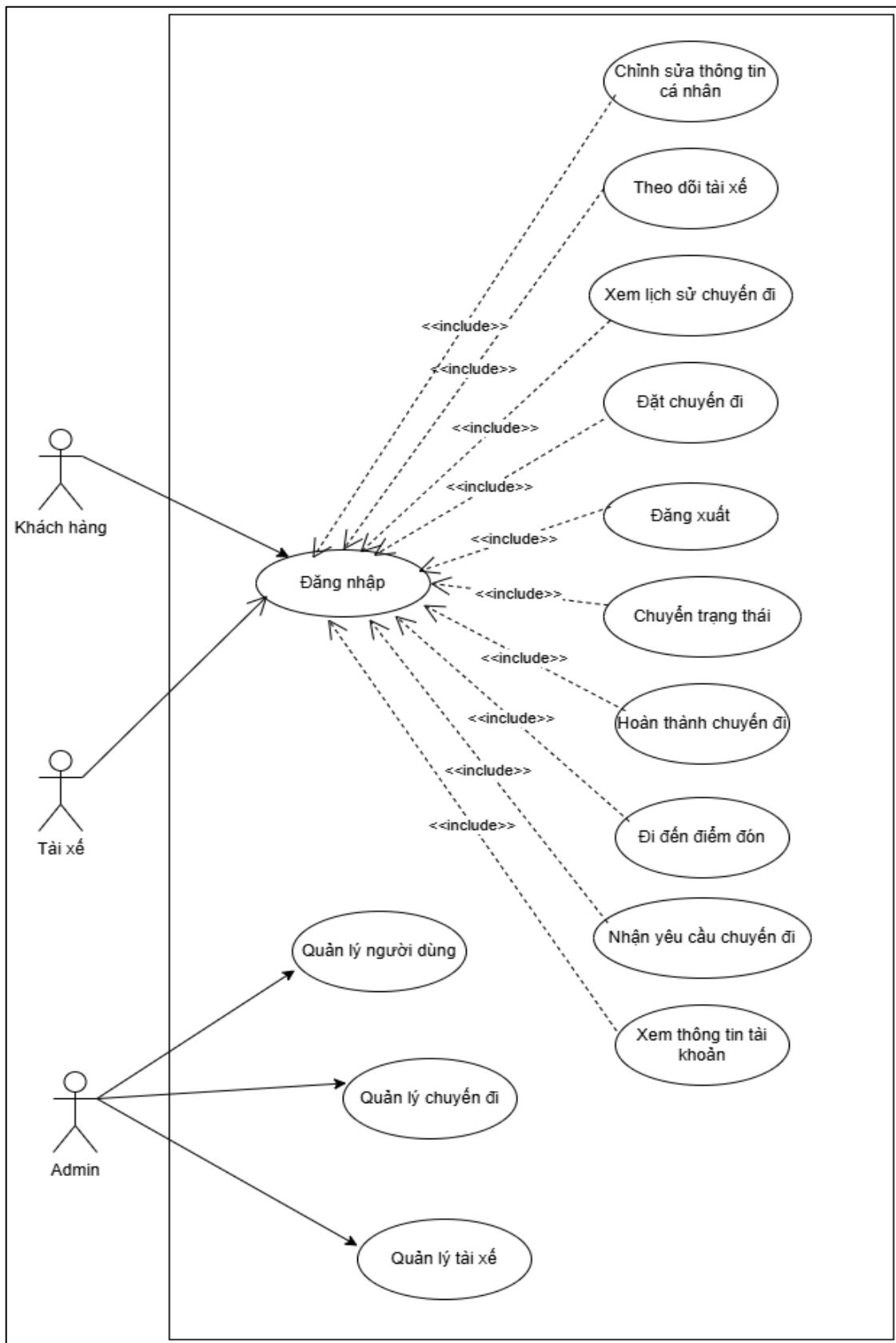
Quản Lý Actor		
STT	Tên	Mô Tả
1	Admin	Quản lý tài khoản tài xế, khách hàng
2	Người dùng	Đặt chuyến đi, theo dõi hành trình, đánh giá dịch vụ, và thanh toán cho các chuyến đi.
3	Tài xế	Nhận chuyến đi từ khách hàng, điều hướng đến điểm đón, và vận chuyển khách đến điểm đến.

❖ Quản lý nghiệp vụ

Bảng 2.2 Quản lý nghiệp vụ

Quản Lý Nghiệp Vụ		
STT	Tên Use case	Mô tả
1	Quản lý tài khoản	Người dùng (người dùng và tài xế) đăng ký, đăng nhập, chỉnh sửa thông tin cá nhân.
2	Đặt chuyến đi	Khách hàng đặt chuyến xe bằng cách nhập điểm đón, điểm đến và tiến hành đặt xe.
3	Theo dõi chuyến đi	Hiển thị vị trí xe theo thời gian thực trên bản đồ khi tài xế đang di chuyển đến điểm đón.
4	Tính cước và thanh toán	Tự động tính giá cước dựa trên khoảng cách, thời gian, phương tiện; hỗ trợ thanh toán tiền mặt.
5	Lịch sử chuyến đi	Khách hàng và tài xế xem lại các chuyến đi đã thực hiện, bao gồm thông tin chi tiết và chi phí.
6	Tính doanh thu	Tính doanh thu từ các chuyến đi của tài xế

❖ Biểu đồ Use Case tổng quát



Hình 2.1 Biểu đồ Use Case tổng quát

Tên use case: Biểu đồ Use Case tổng quát đặt xe.

Tác nhân: Người dùng, Tài xế, Quản trị viên.

Chức năng use case: Biểu đồ này mô tả quy trình người dùng đặt xe, tài xế nhận yêu cầu và hoàn thành chuyến đi.

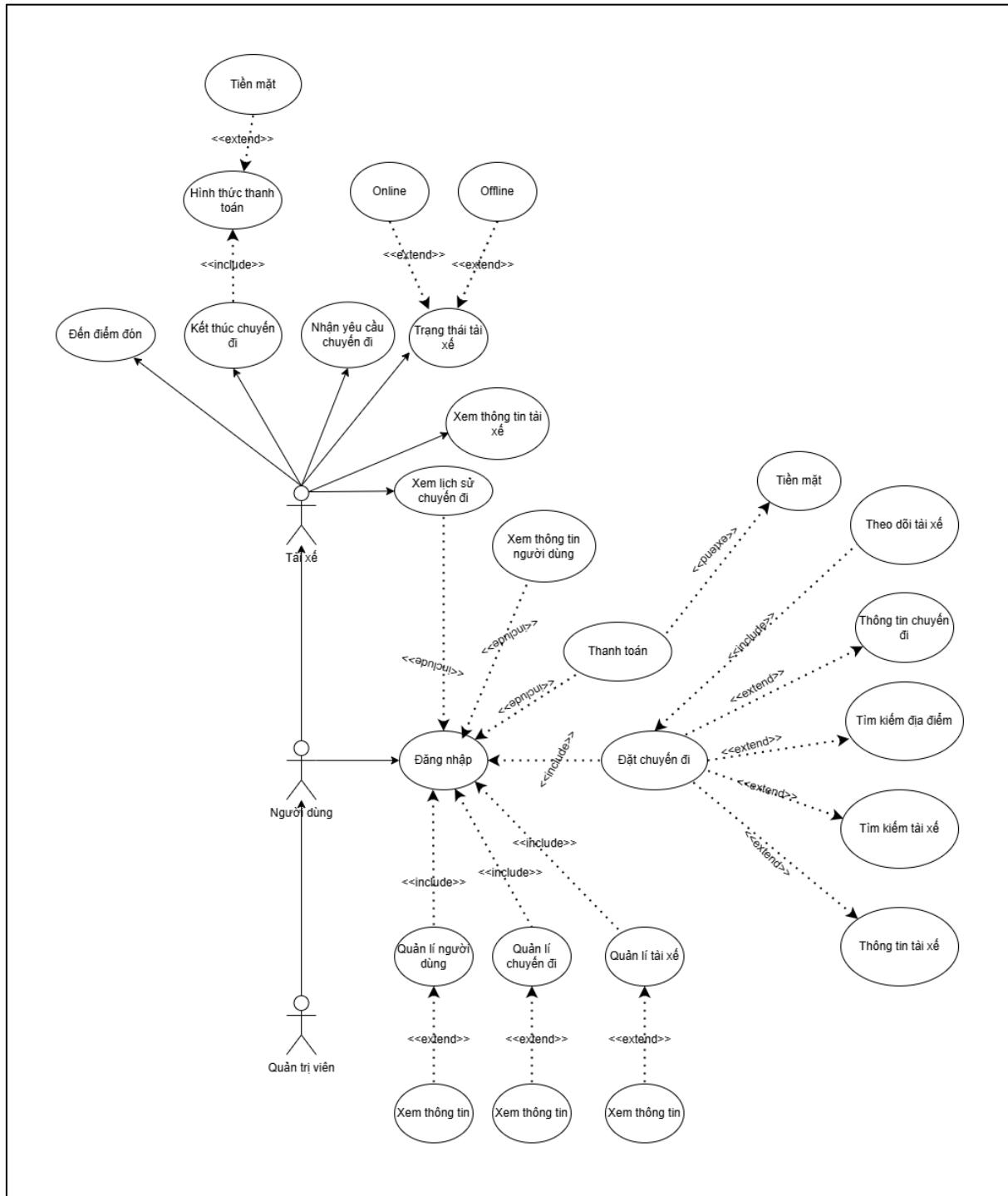
Dòng sự kiện chính:

- Người dùng đăng nhập vào ứng dụng và thực hiện đặt xe, cung cấp thông tin điểm đón và điểm đến.
- Hệ thống tìm kiếm tài xế phù hợp và gửi yêu cầu đến tài xế.
- Tài xế nhận yêu cầu, di chuyển đến điểm đón và thực hiện chuyến đi.
- Sau khi hoàn thành chuyến đi, hệ thống lưu thông tin chuyến đi và hiển thị hóa đơn cho người dùng.

Dòng sự kiện phụ:

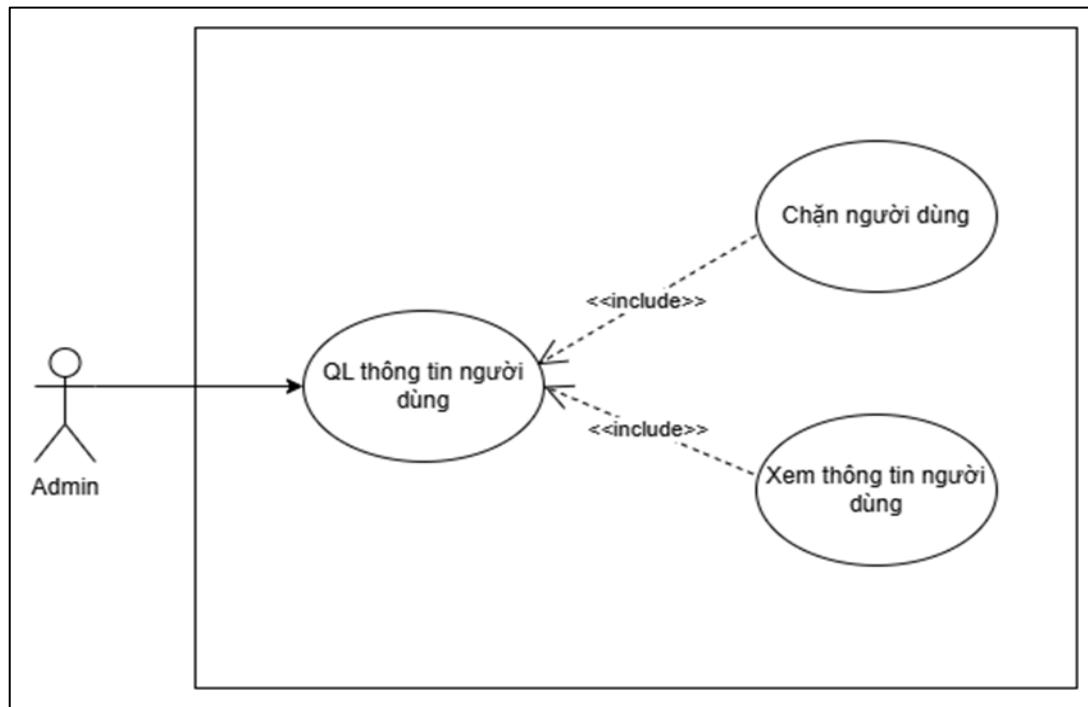
- Nếu không có tài xế khả dụng, hệ thống thông báo cho người dùng và yêu cầu thử lại sau.

❖ Biểu đồ Use Case phân rã



Hình 2.2 Biểu đồ Use Case phân rã

❖ Biểu đồ Use Case quản lý người dùng



Hình 2.3 Biểu đồ Use Case quản lý người dùng

Tên Use Case: Quản lý thông tin người dùng.

Tác nhân: Admin (Quản trị viên).

Chức năng Use Case: Biểu đồ này mô tả chức năng của quản trị viên trong việc quản lý thông tin người dùng, bao gồm xem thông tin và chặn người dùng nếu cần thiết.

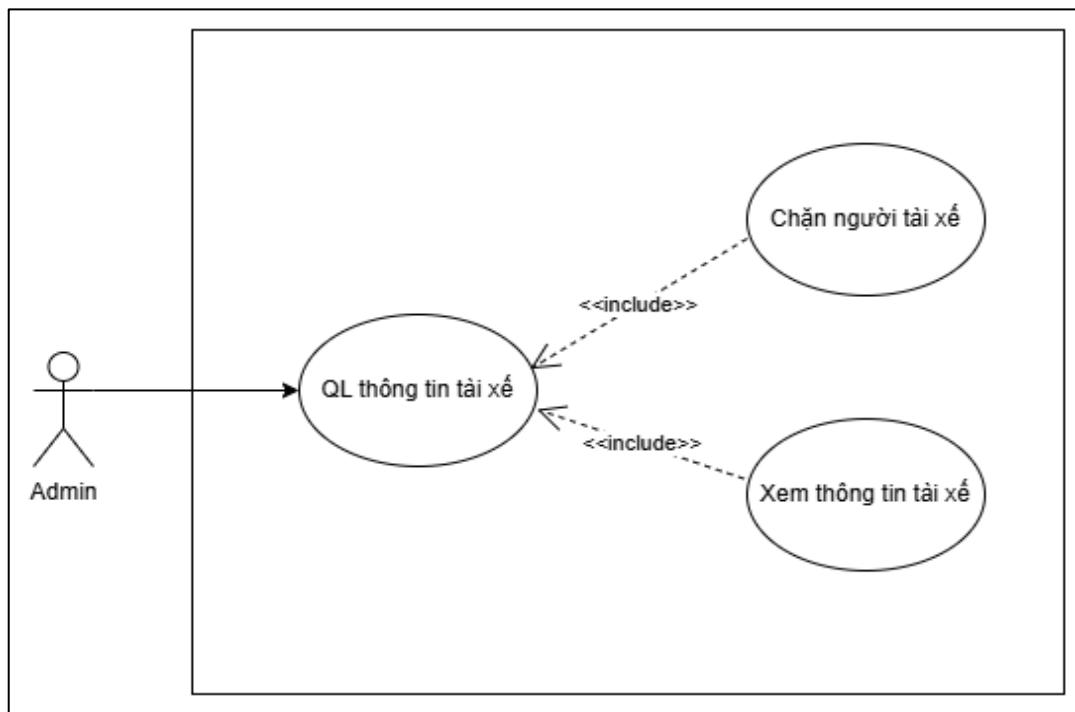
Dòng sự kiện chính:

- Quản trị viên vào hệ thống quản lý.
- Quản trị viên chọn chức năng "Quản lý thông tin người dùng".
- Hệ thống hiển thị danh sách người dùng.
- Quản trị viên có thể:
 - Xem thông tin chi tiết của từng người dùng.
 - Chặn người dùng khi phát hiện vi phạm.

Dòng sự kiện phụ:

- Nếu thông tin người dùng không tồn tại, hệ thống thông báo lỗi.

❖ Biểu đồ Use Case quản lý tài xế



Hình 2.4 Biểu đồ Use Case quản lý tài xế

Tên Use Case: Quản lý thông tin tài xế.

Tác nhân: Admin (Quản trị viên).

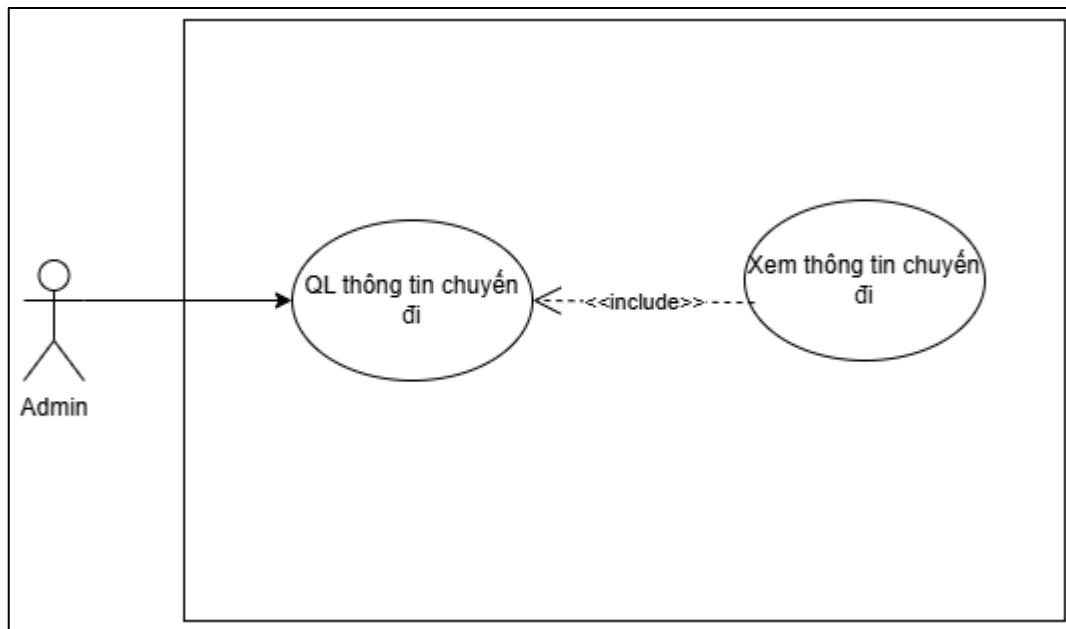
Chức năng Use Case: Biểu đồ mô tả chức năng của quản trị viên trong việc quản lý thông tin tài xế, bao gồm xem thông tin và chặn tài xế nếu cần thiết.

Dòng sự kiện chính:

- Quản trị viên đăng nhập vào hệ thống quản lý.
- Quản trị viên chọn chức năng "Quản lý thông tin tài xế".
- Hệ thống hiển thị danh sách tài xế.
- Quản trị viên có thể:
 - Xem thông tin chi tiết của từng tài xế.
 - Chặn tài xế khi phát hiện vi phạm hoặc hành vi không phù hợp.

Dòng sự kiện phụ:

- Nếu thông tin tài xế không tồn tại, hệ thống thông báo lỗi cho quản trị viên.
- ❖ Biểu đồ Use Case quản lý chuyến đi



Hình 2.5 Biểu đồ Use Case quản lý chuyến đi

Tên Use Case: Quản lý thông tin chuyến đi.

Tác nhân: Admin (Quản trị viên).

Chức năng Use Case: Biểu đồ mô tả chức năng của quản trị viên trong việc quản lý thông tin các chuyến đi, bao gồm kiểm tra và xem thông tin chuyến đi.

Dòng sự kiện chính:

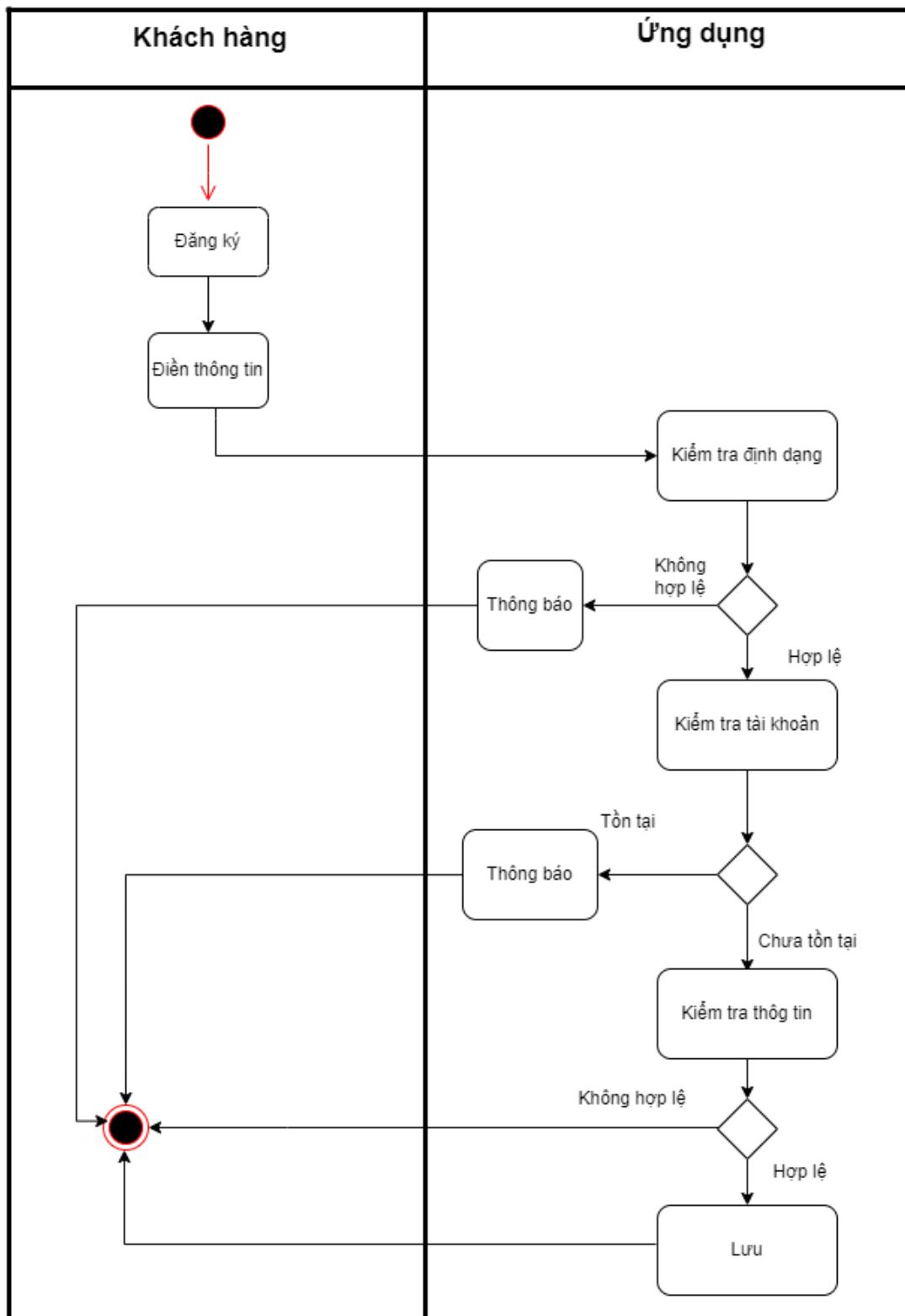
- Quản trị viên đăng nhập vào hệ thống quản lý.
- Quản trị viên chọn chức năng "Quản lý thông tin chuyến đi".
- Hệ thống hiển thị danh sách các chuyến đi.
- Quản trị viên có thể: Xem chi tiết thông tin của từng chuyến đi.

Dòng sự kiện phụ:

- Nếu không tìm thấy thông tin chuyến đi, hệ thống thông báo lỗi cho quản trị viên.

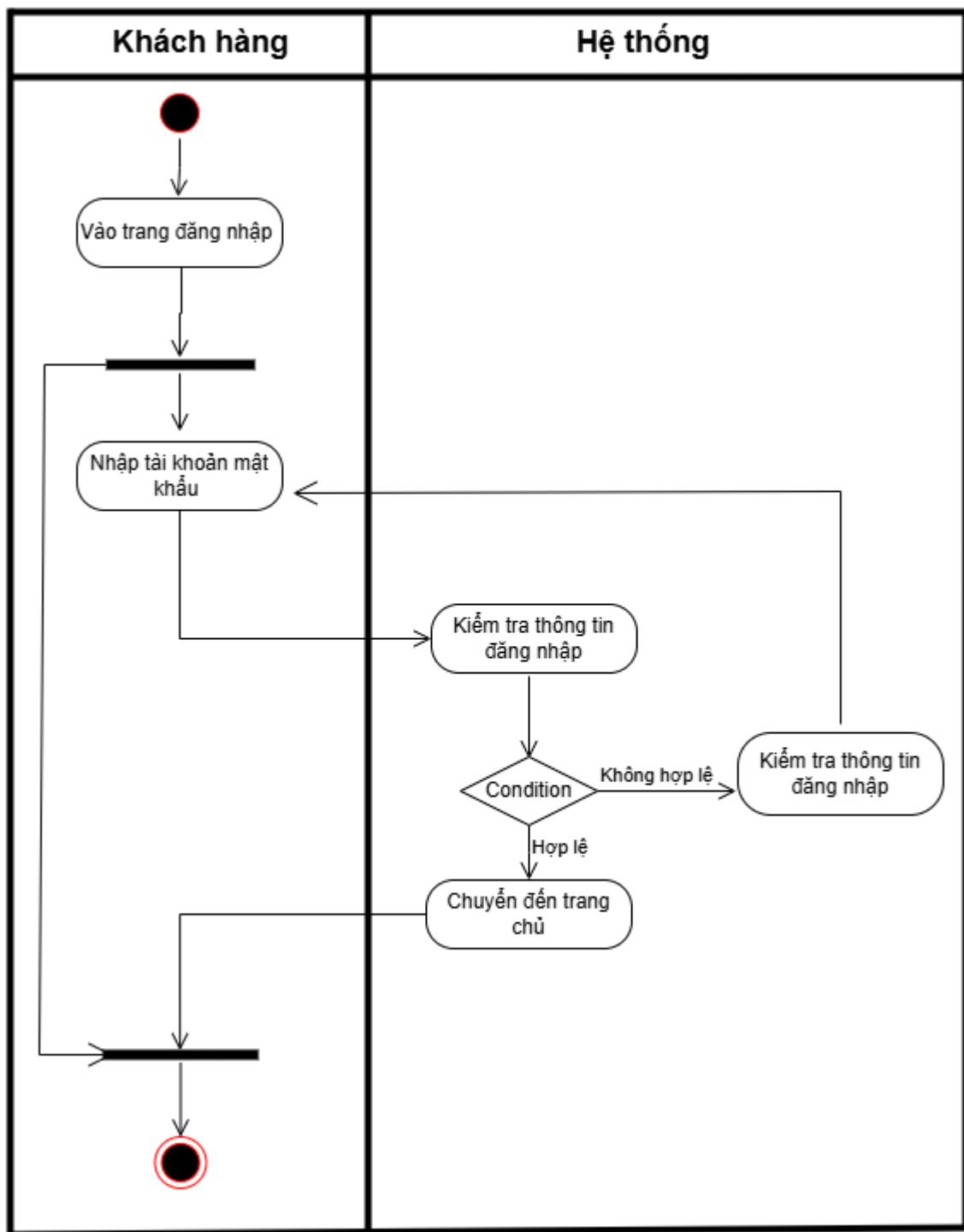
2.3.2 Biểu đồ hoạt động

❖ Biểu đồ hoạt động – Đăng ký



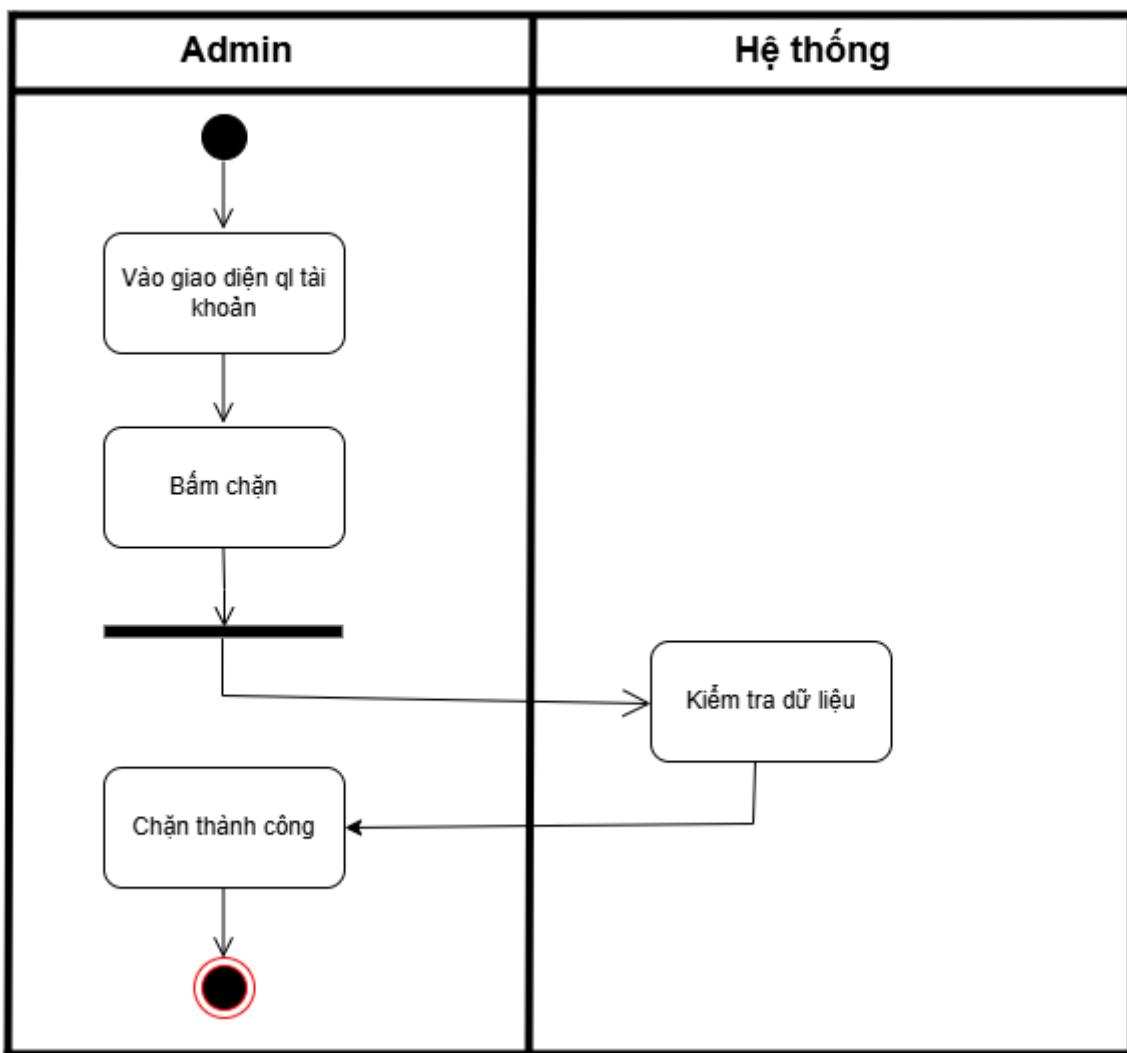
Hình 2.6 Biểu đồ hoạt động – Đăng ký

❖ Biểu đồ hoạt động – Đăng nhập



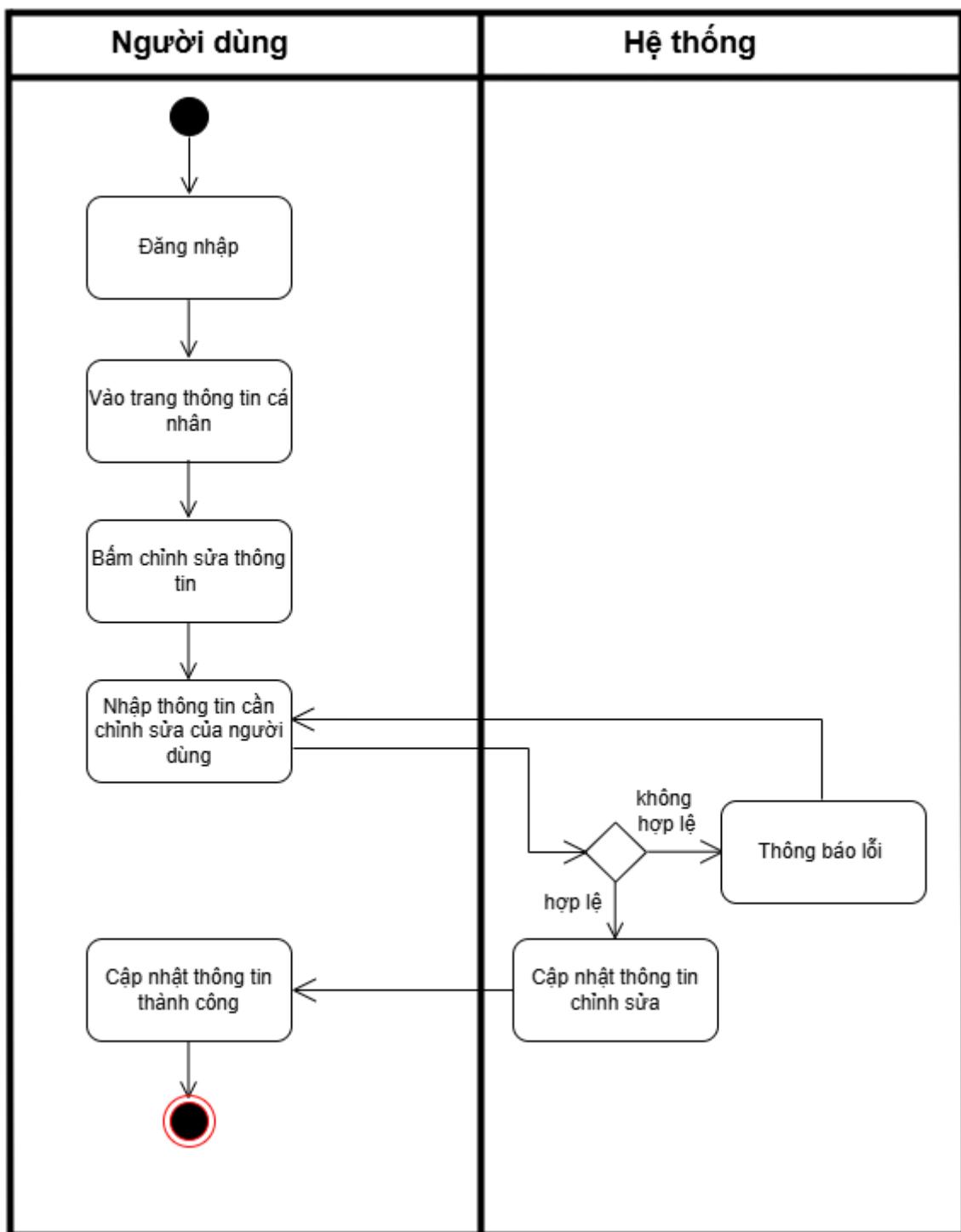
Hình 2.7 Biểu đồ hoạt động – Đăng ký

❖ Biểu đồ hoạt động – Chặn (admin)



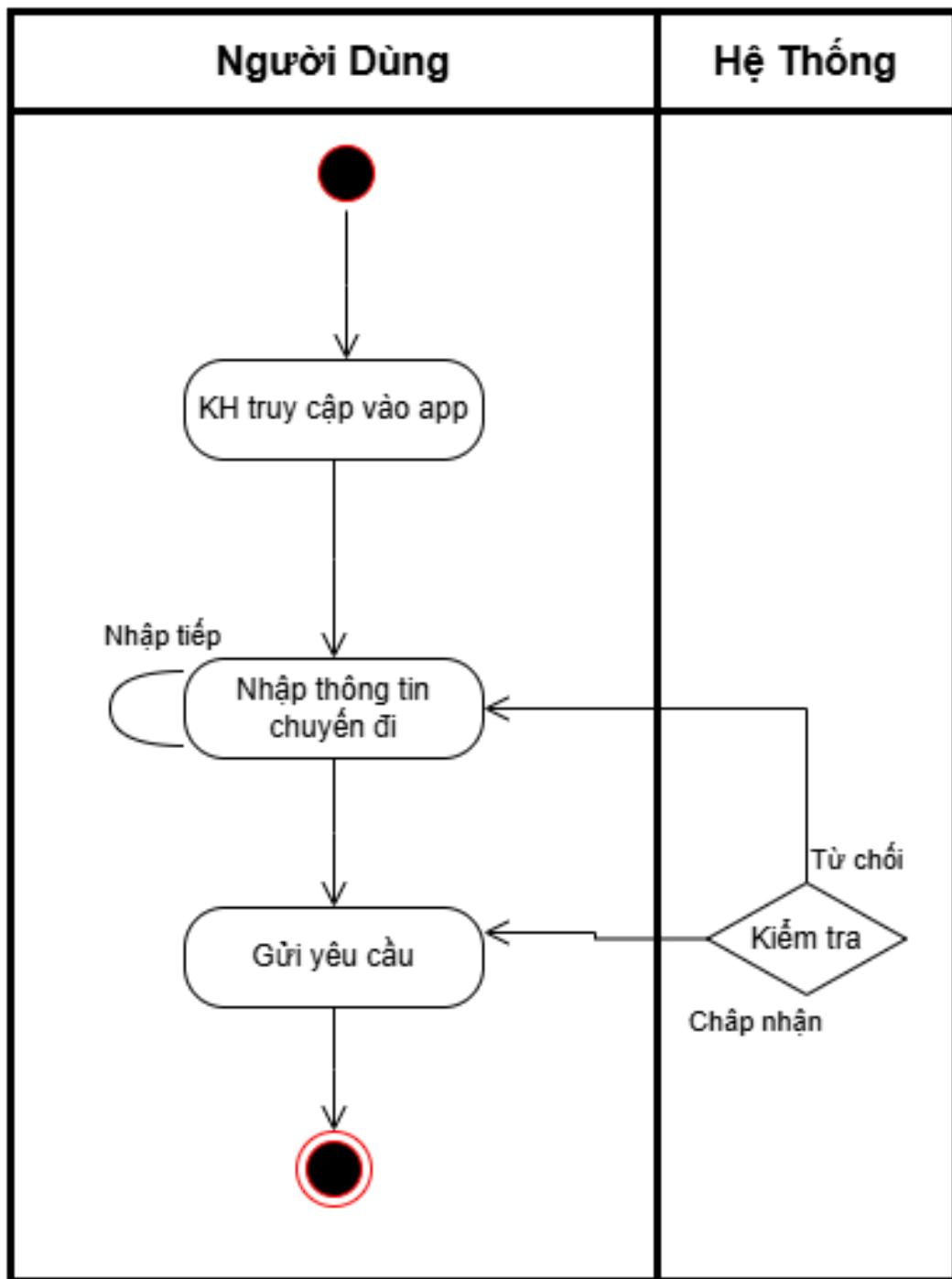
Hình 2.8 Biểu đồ hoạt động – Chặn (admin)

❖ Biểu đồ hoạt động – Cập nhật thông tin cá nhân



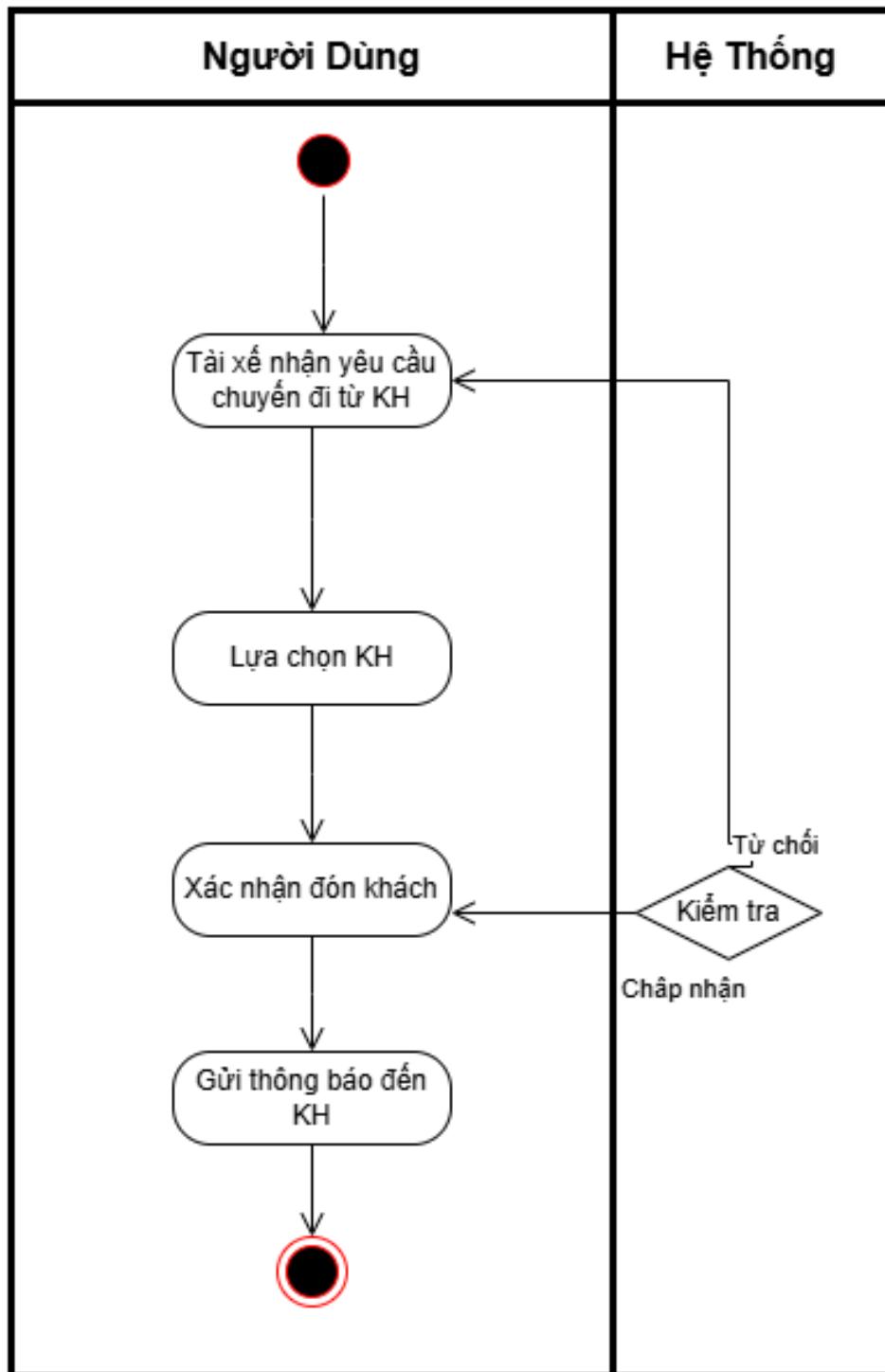
Hình 2.9 Biểu đồ hoạt động – Cập nhật thông tin cá nhân

❖ Biểu đồ hoạt động – Đặt xe



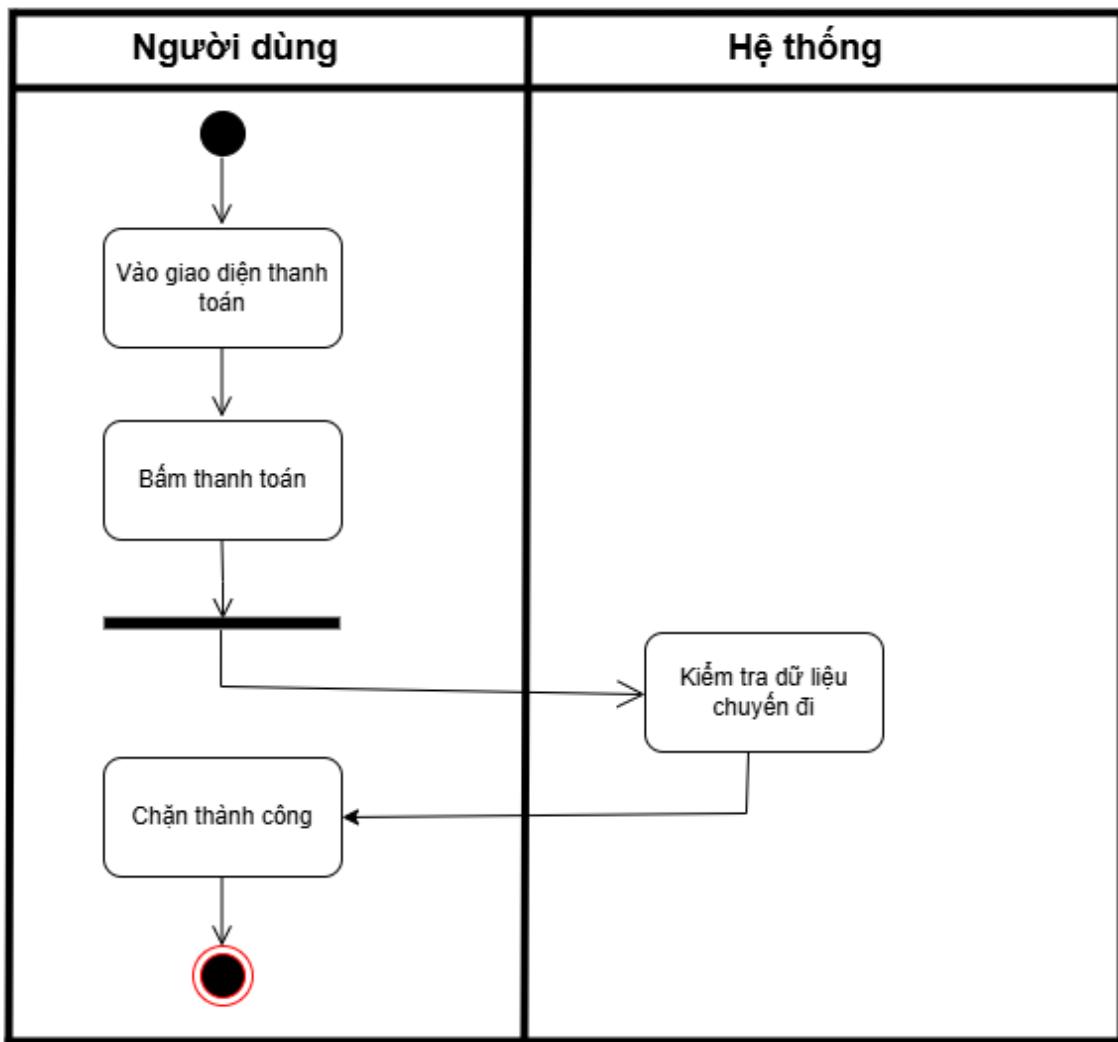
Hình 2.10 Biểu đồ hoạt động – Đặt xe

❖ Biểu đồ hoạt động – Đón khách



Hình 2.11 Biểu đồ hoạt động – Đón khách

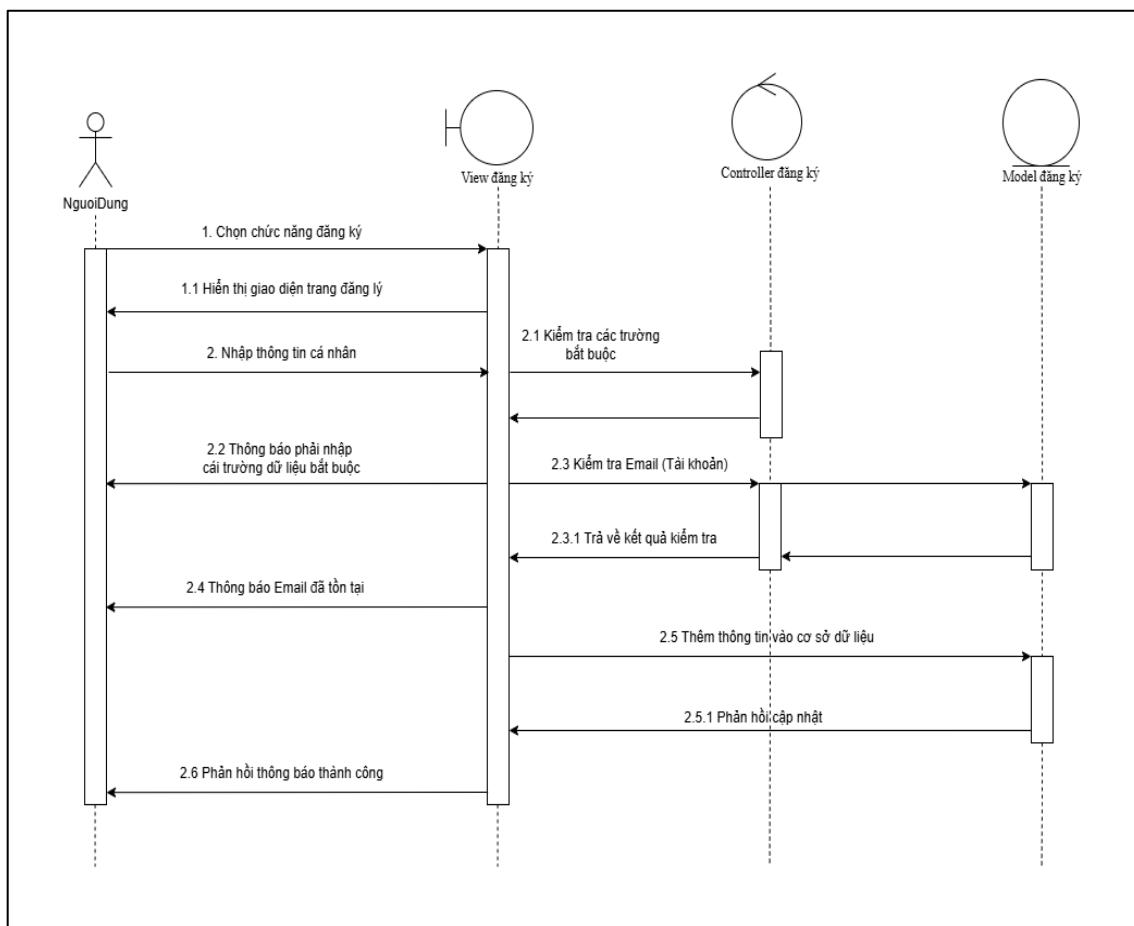
❖ Biểu đồ hoạt động – Thanh toán



Hình 2.12 Biểu đồ hoạt động – Thanh toán

2.3.3 Biểu đồ tuần tự đăng ký

❖ Biểu đồ tuần tự đăng ký



Hình 2.13 Biểu đồ tuần tự đăng ký

Tên Biểu đồ: Quy trình đăng ký tài khoản.

Các Thành Phần Chính:

- Người dùng (NguoiDung): Tác nhân thực hiện đăng ký tài khoản.
- View đăng ký: Thành phần giao diện để người dùng nhập thông tin.
- Controller đăng ký: Thành phần điều khiển, xử lý logic từ giao diện và kết nối với cơ sở dữ liệu.
- Model đăng ký: Thành phần chịu trách nhiệm thực hiện các tác vụ liên quan đến cơ sở dữ liệu.

Luồng Chính:

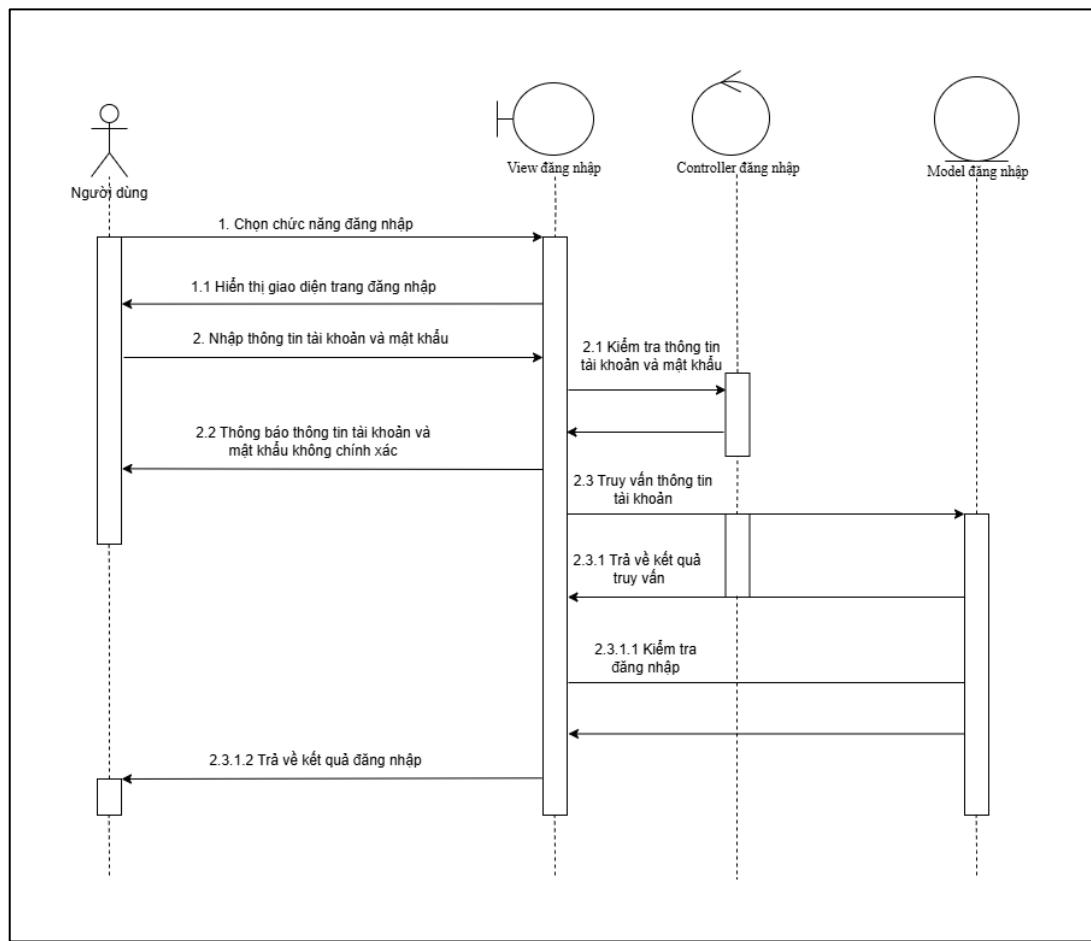
- Người dùng chọn chức năng đăng ký.

- View hiển thị giao diện đăng ký.
- Người dùng nhập thông tin cá nhân.
- Hệ thống kiểm tra các trường bắt buộc:
 - Nếu thiếu thông tin, thông báo cho người dùng.
- Hệ thống kiểm tra xem email đã tồn tại trong cơ sở dữ liệu hay chưa:
 - Nếu email đã tồn tại, thông báo lỗi cho người dùng.
- Nếu thông tin hợp lệ, hệ thống thêm thông tin đăng ký vào cơ sở dữ liệu.
- Gửi phản hồi thành công đến người dùng.

Luồng Phụ:

- Nếu trong quá trình thêm dữ liệu xảy ra lỗi (ví dụ: kết nối cơ sở dữ liệu), hệ thống sẽ thông báo lỗi và yêu cầu thử lại.

❖ Biểu đồ tuần tự đăng nhập



Hình 2.14 Biểu đồ tuần tự đăng nhập

Tên Biểu Đồ: Quy trình đăng nhập hệ thống.

Các Thành Phần Chính:

- Người dùng: Tác nhân thực hiện đăng nhập vào hệ thống.
- View đăng nhập: Giao diện để nhập thông tin tài khoản và mật khẩu.
- Controller đăng nhập: Xử lý logic đăng nhập và kiểm tra thông tin tài khoản, mật khẩu.
- Model đăng nhập: Thực hiện truy vấn cơ sở dữ liệu để kiểm tra thông tin tài khoản.

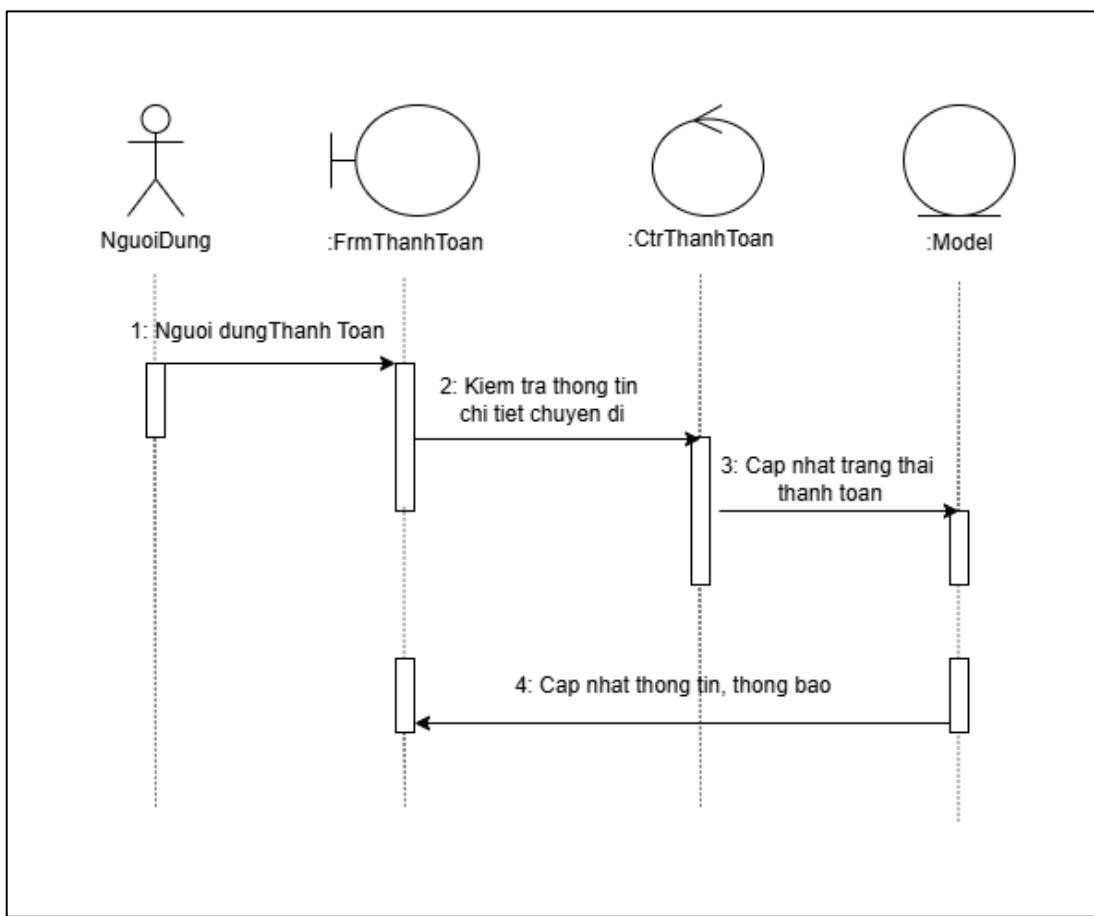
Luồng Chính:

- Người dùng chọn chức năng đăng nhập.
 - View hiển thị giao diện đăng nhập.
- Người dùng nhập thông tin tài khoản và mật khẩu.
- Controller đăng nhập:
 - Kiểm tra thông tin tài khoản và mật khẩu.
- Model đăng nhập:
 - Truy vấn cơ sở dữ liệu để kiểm tra tính hợp lệ của tài khoản và mật khẩu.
 - Trả kết quả truy vấn cho Controller đăng nhập.
- Controller đăng nhập kiểm tra kết quả truy vấn:
 - Nếu thông tin chính xác, trả kết quả đăng nhập thành công.
 - Nếu thông tin không chính xác, thông báo lỗi "Thông tin tài khoản hoặc mật khẩu không chính xác" cho người dùng.

Luồng Phụ:

- Trong trường hợp tài khoản bị khóa hoặc không tồn tại, hệ thống hiển thị thông báo tương ứng cho người dùng.

❖ Biểu đồ tuần tự thanh toán



Hình 2.15 biểu đồ tuần tự thanh toán

Tên Biểu Đồ: Quy trình thanh toán hệ thống

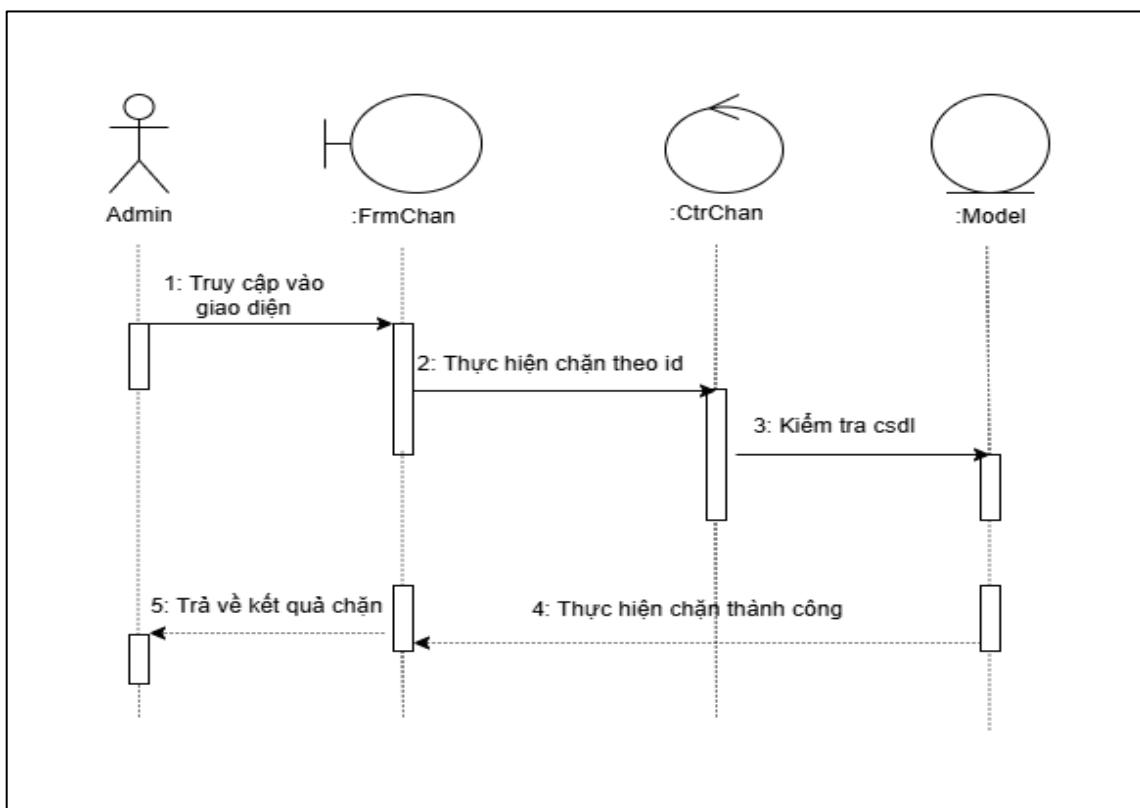
Các Thành Phần Chính:

- Người dùng: Tác nhân thực hiện yêu cầu thanh toán.
- FrmThanhToan (Form thanh toán): Giao diện nhập thông tin thanh toán và xác nhận.
- CtrThanhToan (Controller thanh toán): Bộ điều khiển xử lý logic của quy trình thanh toán.
- Model (Cơ sở dữ liệu): Nơi lưu trữ và truy xuất thông tin giao dịch.

Luồng Chính:

- Người dùng chọn chức năng thanh toán: Gửi yêu cầu tới FrmThanhToan để hiển thị giao diện nhập thông tin thanh toán.
- Người dùng nhập thông tin thanh toán: Thông tin được gửi từ FrmThanhToan tới CtrThanhToan.
- CtrThanhToan kiểm tra thông tin thanh toán: Gửi yêu cầu kiểm tra thông tin đến Model để xác minh chi tiết giao dịch.
- Model xử lý và trả kết quả:
 - Cập nhật trạng thái thanh toán (thành công hoặc thất bại) trong cơ sở dữ liệu.
 - Trả kết quả về cho CtrThanhToan.
- CtrThanhToan xử lý kết quả: Thanh toán thành công: Cập nhật thông tin và thông báo xác nhận cho FrmThanhToan.

❖ Biểu đồ tuần tự chặn (admin)



Hình 2.16 Biểu đồ tuần tự chặn (admin)

Tên Biểu Đồ: Quy trình hệ thống chặn

Các Thành Phần Chính

- Người dùng (Admin): Tác nhân thực hiện thao tác chặn trên hệ thống.
- Giao diện chặn (FrmChan): Giao diện để admin thao tác chặn.
- Controller chặn (CtrChan): Xử lý logic chặn và gửi yêu cầu tới cơ sở dữ liệu.
- Model chặn: Thực hiện truy vấn cơ sở dữ liệu để thực hiện hoặc kiểm tra thông tin chặn.

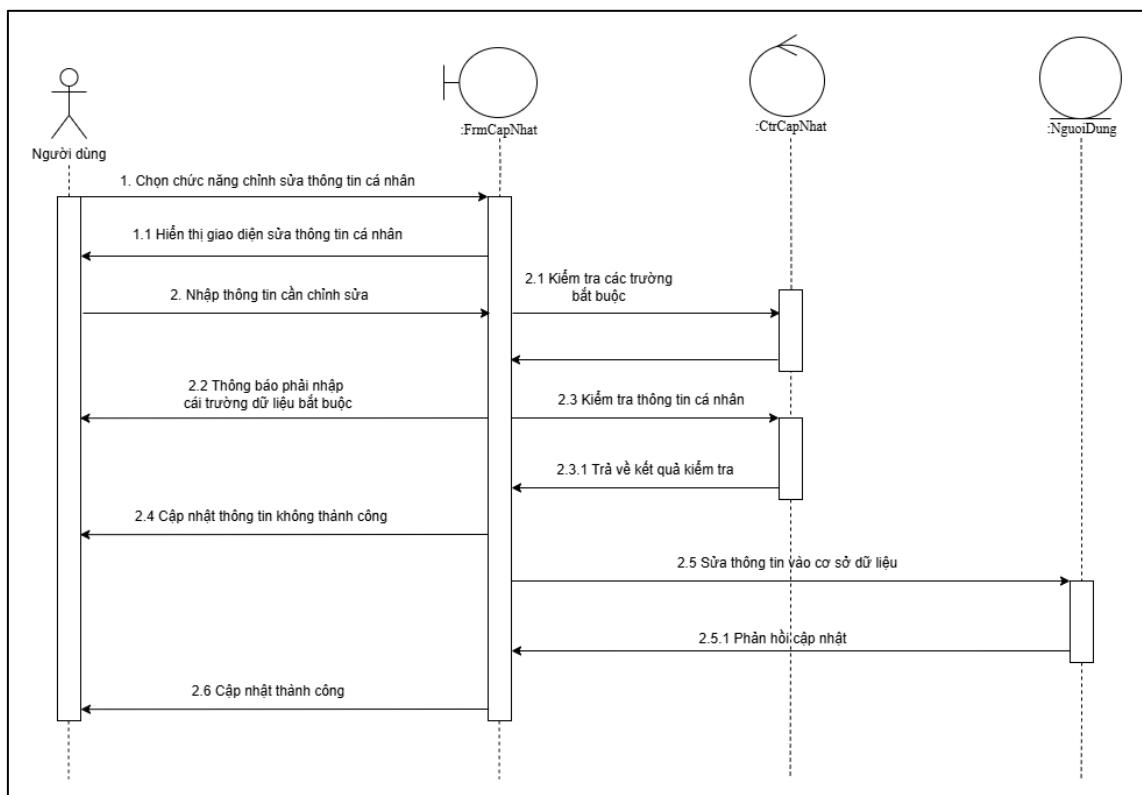
Luồng Chính

- Người dùng (Admin) truy cập giao diện chặn.
 - Giao diện chặn (FrmChan) hiển thị các tùy chọn và yêu cầu thông tin cần thiết (ví dụ: ID người dùng cần chặn).
- Người dùng thực hiện thao tác chặn.
 - Giao diện gửi thông tin ID người dùng cần chặn đến Controller chặn.
- Controller chặn thực hiện logic chặn:
 - Nhận thông tin từ giao diện.
 - Gửi yêu cầu kiểm tra hoặc chặn đến Model chặn.
- Model chặn:
 - Truy vấn cơ sở dữ liệu:
 - Kiểm tra xem ID có tồn tại không.
 - Nếu tồn tại, cập nhật trạng thái chặn trong cơ sở dữ liệu.
 - Trả kết quả (chặn thành công hoặc thất bại) cho Controller chặn.
- Controller chặn xử lý kết quả:

- Nếu chặn thành công:
 - Trả kết quả "Chặn thành công" cho giao diện chặn.
- Nếu chặn thất bại (ID không tồn tại hoặc lỗi khác):
 - Trả thông báo lỗi "Không thể thực hiện chặn" cho giao diện.
- Giao diện hiển thị kết quả chặn cho người dùng (Admin).

Luồng Phụ

- Trường hợp ID không tồn tại:
 - Hệ thống hiển thị thông báo lỗi "ID không tồn tại."
 - Trường hợp lỗi hệ thống hoặc không thể kết nối cơ sở dữ liệu:
 - Hiển thị thông báo lỗi "Lỗi hệ thống, vui lòng thử lại sau."
- ❖ Biểu đồ tuần tự - Cập nhật thông tin cá nhân



Hình 2.17 Biểu đồ tuần tự - Cập nhật thông tin cá nhân

Quy trình: Cập nhật thông tin cá nhân

Các Thành Phần Chính

- Người dùng: Người thực hiện yêu cầu chỉnh sửa thông tin cá nhân.
- FrmCapNhat (Giao diện cập nhật): Nơi người dùng nhập và gửi thông tin cần chỉnh sửa.
- CtrCapNhat (Controller cập nhật): Xử lý logic, kiểm tra thông tin và gửi yêu cầu đến cơ sở dữ liệu.
- NguoiDung (Model): Tương tác với cơ sở dữ liệu để lưu hoặc kiểm tra thông tin cá nhân.

Luồng Chính

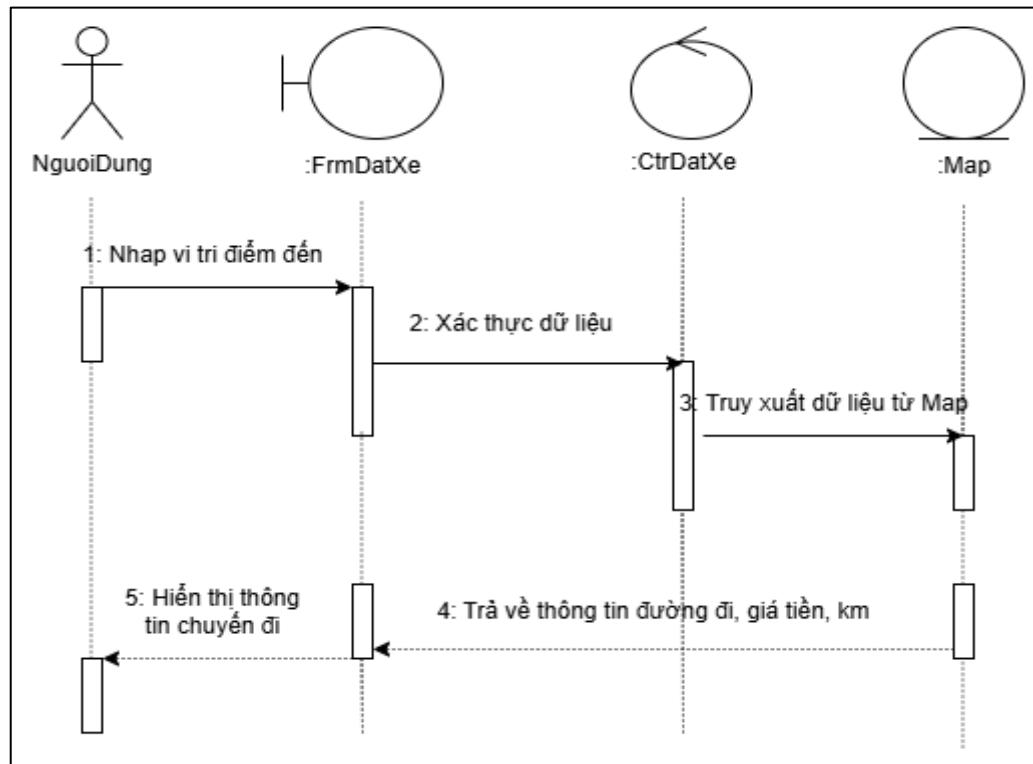
- Người dùng chọn chức năng chỉnh sửa thông tin cá nhân.
 - FrmCapNhat: Hiển thị giao diện chỉnh sửa thông tin cá nhân.
- Người dùng nhập thông tin cần chỉnh sửa.
- CtrCapNhat thực hiện kiểm tra:
 - Kiểm tra các trường dữ liệu bắt buộc:
 - Nếu thiếu, trả thông báo lỗi cho FrmCapNhat (2.2: "Thông báo phải nhập các trường dữ liệu bắt buộc").
 - Quay lại bước nhập thông tin (2.4: "Cập nhật thông tin không thành công").
 - Nếu đầy đủ, thực hiện kiểm tra thông tin cá nhân.
- NguoiDung thực hiện kiểm tra thông tin trong cơ sở dữ liệu:
 - Truy vấn cơ sở dữ liệu để xác minh thông tin.
 - Trả kết quả kiểm tra cho CtrCapNhat (2.3.1).
- CtrCapNhat xử lý kết quả:

- Nếu thông tin hợp lệ:
 - Gửi yêu cầu cập nhật thông tin vào cơ sở dữ liệu (2.5).
 - Nhận phản hồi từ cơ sở dữ liệu (2.5.1).
 - Trả kết quả cập nhật thành công cho FrmCapNhat (2.6).
 - Nếu không hợp lệ, thông báo lỗi.
- FrmCapNhat hiển thị kết quả cập nhật (thành công hoặc không thành công) cho người dùng.

Luồng Phụ

- Trường hợp lỗi hệ thống hoặc không thể kết nối cơ sở dữ liệu: Trả thông báo lỗi "Hệ thống bận, vui lòng thử lại sau."
- Trường hợp thông tin cá nhân không hợp lệ: Thông báo lỗi chi tiết: "Thông tin không hợp lệ."

❖ Biểu đồ tuần tự - Đặt xe



Hình 2.18 Biểu đồ tuần tự - Đặt xe

Quy trình: Đặt xe

Các Thành Phần Chính

- NguoiDung (Người dùng): Người sử dụng hệ thống để đặt xe.
- FrmDatXe (Giao diện đặt xe): Giao diện nhập vị trí và thông tin đặt xe.
- CtrDatXe (Controller đặt xe): Xử lý logic đặt xe và liên kết với bản đồ.
- Map (Dữ liệu bản đồ): Cung cấp thông tin bản đồ, khoảng cách, thời gian và chi phí.

Luồng Chính

- Người dùng nhập vị trí điểm đến vào giao diện đặt xe (FrmDatXe).
- FrmDatXe gửi dữ liệu đến CtrDatXe để xác thực thông tin:
 - Controller kiểm tra dữ liệu đầu vào (ví dụ: kiểm tra định dạng địa chỉ hoặc vị trí hợp lệ).
 - Nếu dữ liệu không hợp lệ, trả thông báo lỗi.
- CtrDatXe gửi yêu cầu đến Map để truy xuất dữ liệu:
 - Map xác định thông tin liên quan như:
 - Tuyến đường phù hợp.
 - Khoảng cách di chuyển (km).
 - Dự tính chi phí chuyến đi.
- Map trả kết quả cho CtrDatXe:
 - Bao gồm:
 - Thông tin tuyến đường.
 - Thời gian dự kiến di chuyển.
 - Chi phí dự kiến.

- CtrDatXe gửi thông tin về FrmDatXe để hiển thị cho người dùng.

Luồng Phụ

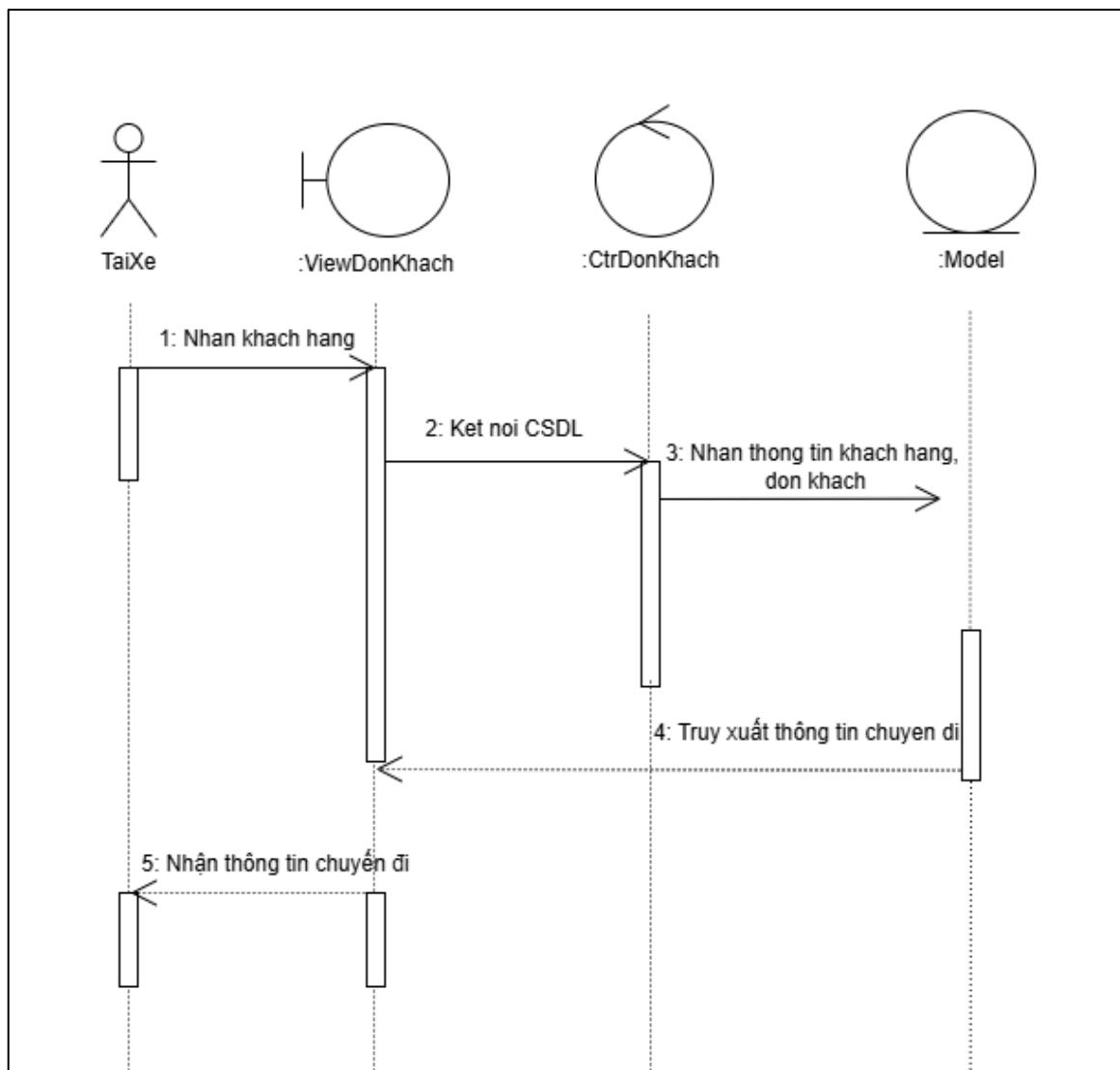
- Trường hợp nhập sai dữ liệu:

- Hệ thống thông báo lỗi: "Vui lòng nhập địa chỉ hợp lệ."

- Trường hợp lỗi kết nối hoặc không lấy được dữ liệu bản đồ:

- Hệ thống thông báo: "Không thể lấy thông tin bản đồ, vui lòng thử lại sau."

❖ Biểu đồ tuần tự - Đón khách



Hình 2.19 Biểu đồ tuần tự - Đón khách

Quy trình: Đón khách

Các thành phần chính

- Tài xế (TaiXe): Người nhận yêu cầu đón khách qua ứng dụng.
- Giao diện đặt chuyến (ViewDonKhach): Thành phần giao diện xử lý và hiển thị thông tin yêu cầu đón khách.
- Bộ điều khiển đơn khách (CtrDonKhach): Xử lý logic ứng dụng và kết nối với cơ sở dữ liệu.
- Mô hình dữ liệu (Model): Quản lý và lưu trữ thông tin khách hàng, chuyến đi trong cơ sở dữ liệu.

Luồng chính

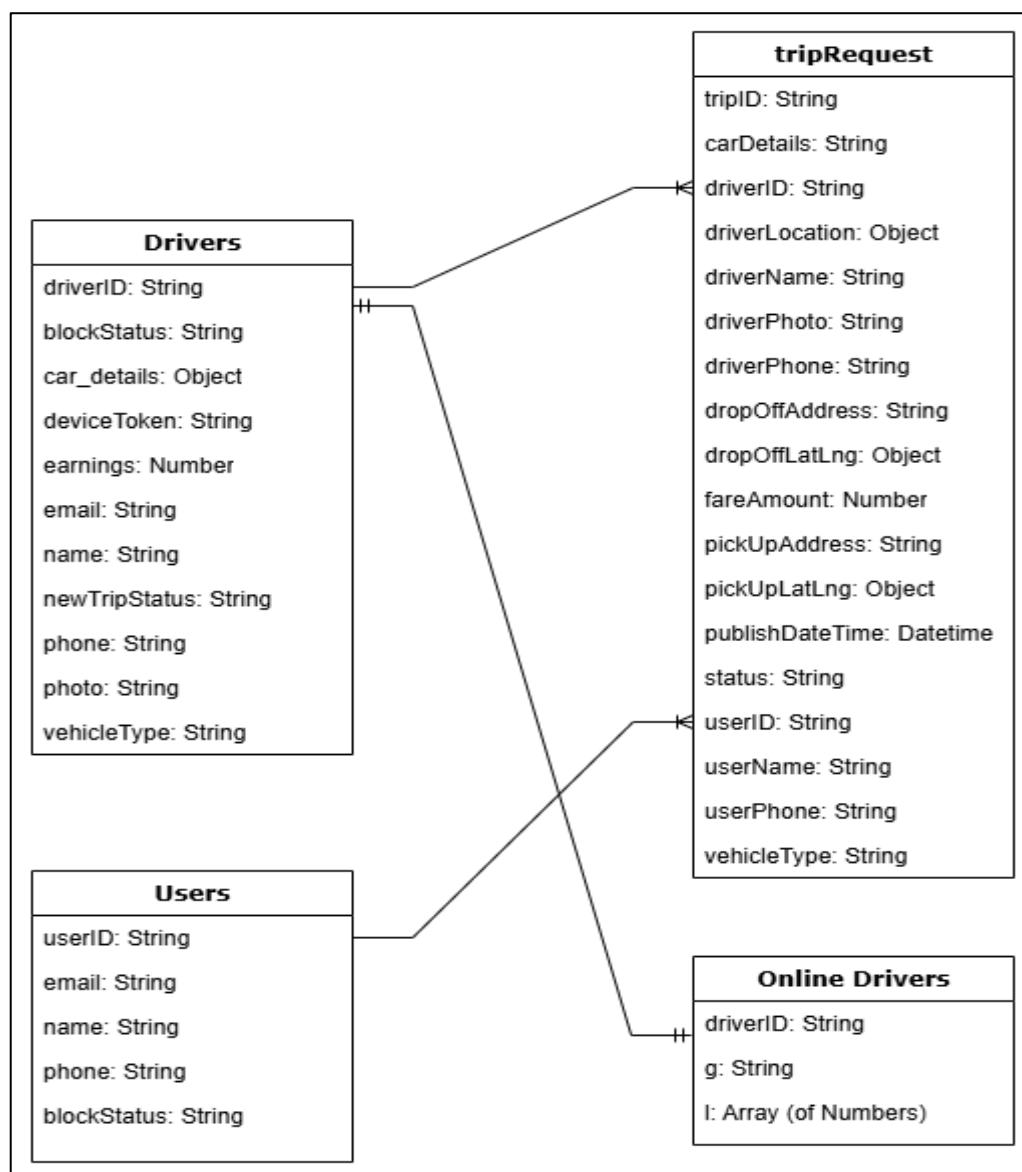
- Nhận yêu cầu đặt chuyến: Tài xế nhận thông báo yêu cầu đặt chuyến thông qua ứng dụng.
- Gửi yêu cầu kết nối CSDL: Giao diện đặt chuyến (ViewDonKhach) chuyển yêu cầu sang bộ điều khiển đơn khách (CtrDonKhach) để kết nối cơ sở dữ liệu.
- Xử lý thông tin khách hàng và chuyến đi: Bộ điều khiển (CtrDonKhach) truy vấn cơ sở dữ liệu để lấy thông tin khách hàng và chi tiết chuyến đi.
- Phản hồi thông tin chuyến đi: Cơ sở dữ liệu trả lại dữ liệu về khách hàng và chuyến đi cho bộ điều khiển.
- Cập nhật thông tin cho tài xế: Bộ điều khiển gửi thông tin đã xử lý đến giao diện ứng dụng để hiển thị cho tài xế.

Luồng phụ

- Xử lý lỗi kết nối:
 - Nếu không kết nối được với cơ sở dữ liệu:
 - Bộ điều khiển thông báo lỗi cho giao diện ứng dụng.

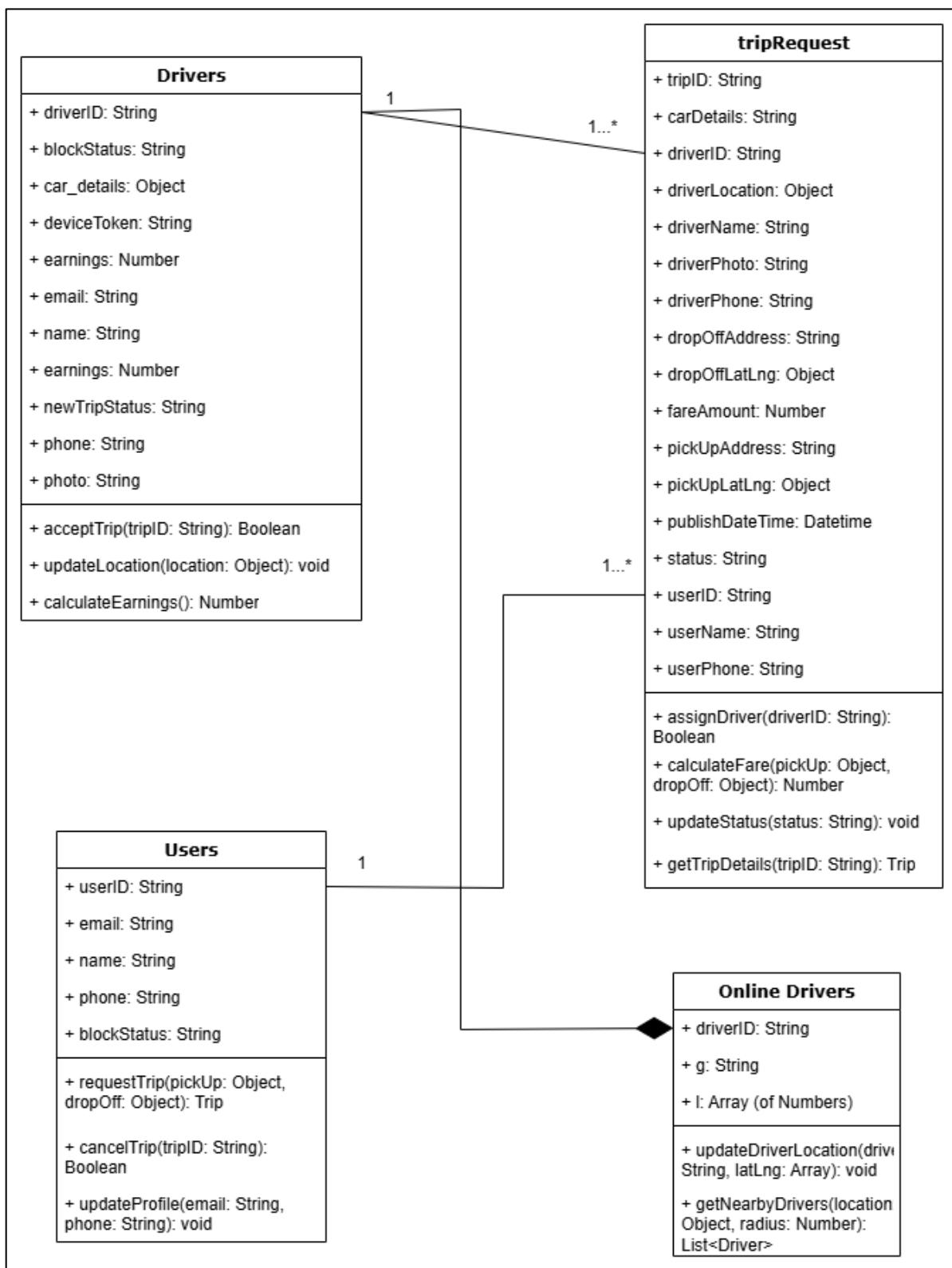
- Giao diện hiển thị thông báo lỗi cho tài xế.
- Khách hàng hủy yêu cầu:
 - Nếu khách hàng hủy chuyến trong khi đang xử lý:
 - Bộ điều khiển nhận thông báo từ hệ thống và hủy truy vấn đến cơ sở dữ liệu.
 - Thông tin hủy chuyến được gửi lại cho giao diện ứng dụng để cập nhật cho tài xế.

2.3.4 Sơ đồ ERD



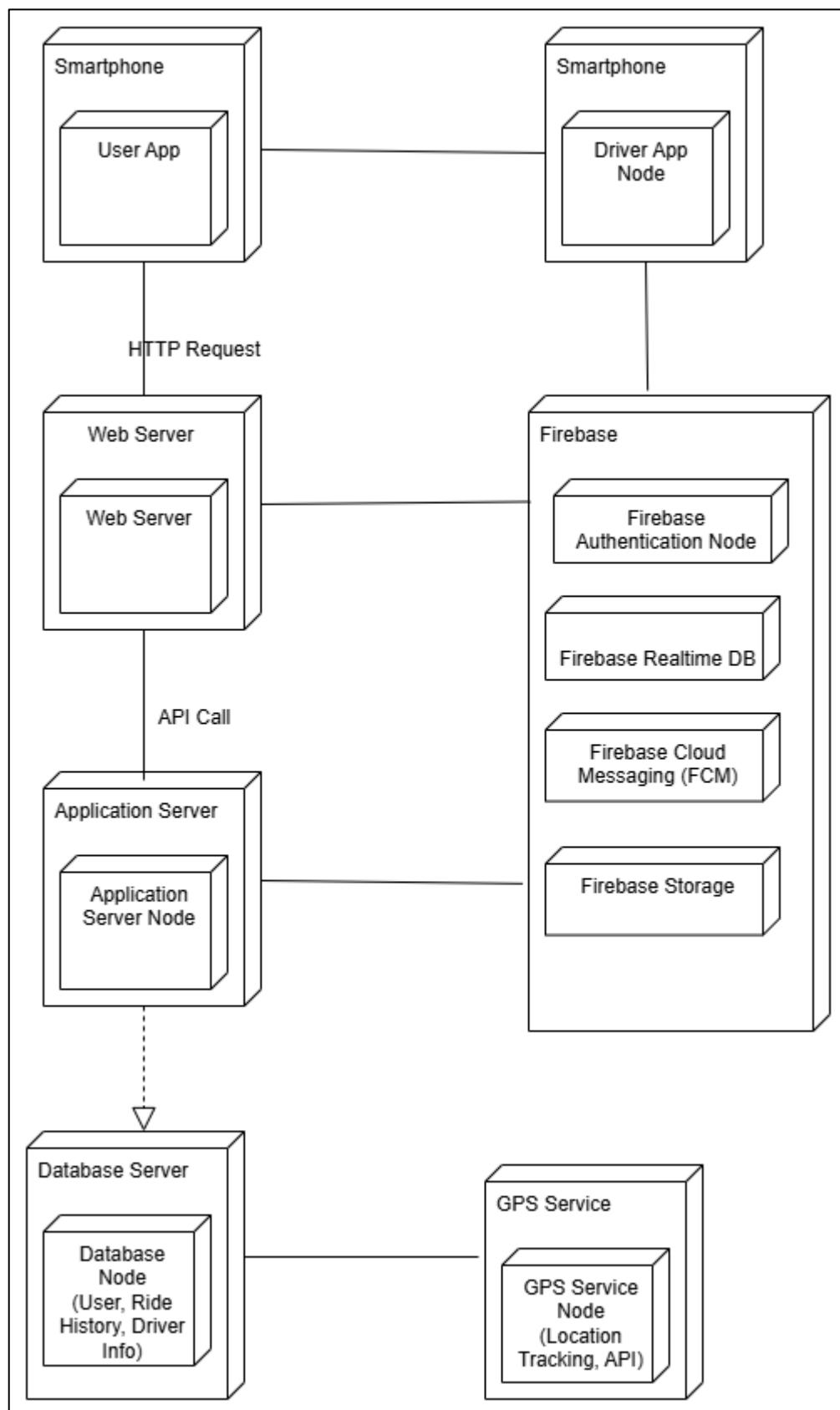
Hình 2.20 Sơ đồ ERD

2.3.5 Class Diagram



Hình 2.21 Class Diagram

2.3.6 Deployment diagram (Sơ đồ triển khai hệ thống)



Hình 2.22 Deployment diagram

2.3.7 Thiết kế cơ sở dữ liệu

2.3.7.1 Mô tả dữ liệu

❖ Bảng Users

Bảng 2.3 Bảng cơ sở dữ liệu Users

STT	Thuộc Tính	Kiểu Dữ Liệu	Mô tả
1	userID	String	ID của người dùng.
2	name	String	Tên người dùng.
3	email	String	Email của người dùng.
4	phone	String	Số điện thoại người dùng.
5	blockStatus	String	Trạng thái tài khoản người dùng ("no""yes").

❖ Bảng Drivers

Bảng 2.4 Bảng cơ sở dữ liệu Drivers

STT	Thuộc Tính	Kiểu Dữ Liệu	Mô tả
1	driverID	String	ID của tài xế thường là chuỗi duy nhất.
2	car_details	Object	Đối tượng chứa thông tin xe của tài xế
		String	Màu xe
		String	Tên hoặc kiểu xe
		String	Biển số xe
3	deviceToken	String	Token thông báo đầy đủ dành cho thiết bị của tài xế.
4	earnings	Number	Thu nhập của tài xế
5	email	String	Email của tài xế.
6	name	String	Tên tài xế
7	newTripStatus	String	Trạng thái chuyến đi mới ("timeout", "accepted")

8	phone	String	Số điện thoại tài xế
9	photo	String	URL của ảnh đại diện tài xế
10	blockStatus	String	Trạng thái tài khoản tài xế ("no","yes").
11	vehicleType	String	Loại xe

❖ Bảng Online Drivers

Bảng 2.5 Bảng cơ sở dữ liệu Online Drivers

STT	Thuộc Tính	Kiểu Dữ Liệu	Mô tả
1	driverID	String	ID của tài xế thường là chuỗi duy nhất.
2	g	String	Giá trị mã hóa cho vị trí địa lý (geohash).
3	l	Array (of Numbers)	Mảng chứa kinh độ (longitude) và vĩ độ (latitude).

❖ Bảng Trip Requests

Bảng 2.6 Bảng cơ sở dữ liệu Trip Requests

STT	Thuộc Tính	Kiểu Dữ Liệu	Mô tả
1	carDetails	String	Mô tả thông tin xe của tài xế
2	driverID	String	ID của tài xế thực hiện chuyến đi.
3	driverLocation	Object	Đối tượng chứa vị trí của tài xế (kinh độ và vĩ độ).
	- latitude	Number	Vĩ độ của tài xế
	- longitude	Number	Kinh độ của tài xế
4	driverName	String	Tên tài xế
5	driverPhone	String	Số điện thoại của tài xế.
6	driverPhoto	String	URL ảnh đại diện tài xế.
7	dropOffAddress	String	Địa chỉ nơi đến
8	dropOffLatLng	Object	Vị trí địa lý nơi đến.

	- latitude	String	Vĩ độ nơi đến.
	- longitude	String	Kinh độ nơi đến.
9	fareAmount	Number	Giá tiền cho chuyến đi
10	pickUpAddress	String	Địa chỉ nơi đón khách.
11	pickUpLatLng	Object	Vị trí địa lý nơi đón khách.
	- latitude	String	Vĩ độ nơi đón khách.
	- longitude	String	Kinh độ nơi đón khách.
12	publishDateTime	Datetime	Ngày giờ chuyến đi được tạo.
13	status	String	Trạng thái chuyến đi ("ended""ongoing""pending")
14	tripID	String	ID chuyến đi.
15	userID	String	ID của người dùng.
16	userName	String	Tên người dùng.
17	userPhone	String	Số điện thoại người dùng.
18	vehicleType	String	Loại xe

2.3.7.2 Cơ sở dữ liệu thử nghiệm

❖ Bảng Users

Bảng 2.7 Dữ liệu thử nghiệm - Users

userID	name	email	phone	password	blockStatus
Gf1mXrbc5wZuT XjupEFtl64JSQ02	duong2	duong2@g mail.com	0916216723	duong123 4	no
QKM7mSO4pFhZ 3WUZrjAnJomWn Xx2	Duong1	duong1@g mail.com	0878061498	duong123 4	no
sEfzN8zWQUX8Q 8RXW9cEpPgST1 I2	thaoduo ng	duong@g mail.com	0916216723	duong123 4	no

❖ Bảng Drivers

Bảng 2.8 Dữ liệu thử nghiệm - Drivers

driverID	dgmIaSLiHeTJw vuisv95rw5ZJ6w2	7vcBJ3GbeBOwE DKgtBfyH1Rzbw82	DPOcwkHYVIWM R3VJp13BAMqnJg63
car_details	Xanh- Toyota- 86C12334	Đỏ-kia- 86C78384	do - kia - 86b5783
deviceToken	c7BHffRORhSh GEp-EMxsP- APA91b G2Q3Axsq-mA MZYSntq2WR 6zEFUWDeqn lV2Ytq1FJvt- F86I10viLZtZaO 89yvnMxFos QvELViv08U SoxNEtze_4E E2kiscrfd7pln BfnB4fjjDDW8	c7BHffRORhSh GEp-EMxsP-A PA91bG2Q3 Axsq-mAMZY Sntq2WR6zE FUWDeqnLV2Y tq1FJvt-F86I10vi LZtZa O89yvnMxFos QvELViv08U SoxNEtze_4EE 2kiscrfd7plnBf nB4fjjDDW8	fis4cO4- TLefCTJPcvT LW6:APA91 bEHo0o_m8m PPrNJl0X0mq 54WBkO16V6 m0v6zw_R70 qY3oBPUZRpm UPLBIB9vfHS GsGScZwNPR S6o3a3Qsdzjk iDAiXEFun8i 1DHjavTswci9aUxj3I
earnings	316.5	0	0
email	dung1@gmail.com	loi@gmail.com	duong4@gmail.com
name	Dunh	Nguyễn Văn Lợi	Duong4
newTripStatus	accepted		
phone	012345678	0984926306	0872568278
photo	Link hình ảnh	Link hình ảnh	Link hình ảnh
vehicleType	Taxi	Truck	Taxi

❖ Bảng Online Drivers

Bảng 2.9 Dữ liệu thử nghiệm – Online Drivers

driverID	g	l
DPOcwkHYVIWMR3VJ p13BAMqnJg63	w3gv77r93r	10.7847114 106.6766148
7vcBJ3GBeBOwEDKgtB fyH1Rzbw82	w3gv776cj	10.7846754 106.6702842

❖ Bảng Trip Requests

Bảng 2.10 Dữ liệu thử nghiệm – Trip Requests

carDetails	do - kia - 86b5783	hồng - toyota - 87C54467	hồng - toyota - 87C54467
driverID	DPOcwkHYVIW MR3VJp13BAM qnJg63	QrckNHpiiZULx udEXoynzVlfYz1 2	QrckNHpiiZULx udEXoynzVlfYz1 2
driverLocation	10.7955117 106.6670417	10.7847208 106.6703033	10.7847208 106.6703033
driverName	Duong4	Nhung	Nhung
driverPhone	0872568244	08764467874	08764467874
dropOffAddress	37 Bờ Bao Tân Thắng	37 Đường Bờ Bao 1	156 Tô Hiến Thành
dropOffLatLng	10.806140528999 106.62182667199	10.8070931 106.6191491	10.780488257999 106.66892052499
fareAmount	129.9	48.9	48.7
pickUpAddress	444/2/1A Đường Cách Mạng Tháng 8, phường 11, Quận 3, Hồ Chí Minh	số 7G, 156, Tô Hiến Thành, Cư xá Bắc Hải, Quận 10, Hồ Chí Minh	số 7G, 156, Tô Hiến Thành, Cư xá Bắc Hải, Quận 10, Hồ Chí Minh
pickUpLatLng	10.7846655	10.78056	10.78056

	106.6702352	106.669005	106.669005
publishDateTime	2024-11-26 11:55:41.163375	2024-12-06 00:28:03.898121	2024-12-05 15:08:04.891099
status	ended	ended	ended
tripID	- OCaoo4XGt94X ATynzch	-ODMrK- _LJ_Vilj25ddm	- ODKr97LRTcjx5 279nBk
userID	QKM7mSO4pFh Z3WUZrjAnJom WnXx2	QKM7mSO4pFh Z3WUZrjAnJom WnXx2	QKM7mSO4pFh Z3WUZrjAnJom WnXx2
userName	Duong1	Duong1	Duong1
userPhone	0878061432	0878061498	0878061498
vehicleType	Taxi	Truck	Taxi

CHƯƠNG III: CÀI ĐẶT THỬ NGHIỆM, KẾT QUẢ VÀ ĐÁNH GIÁ

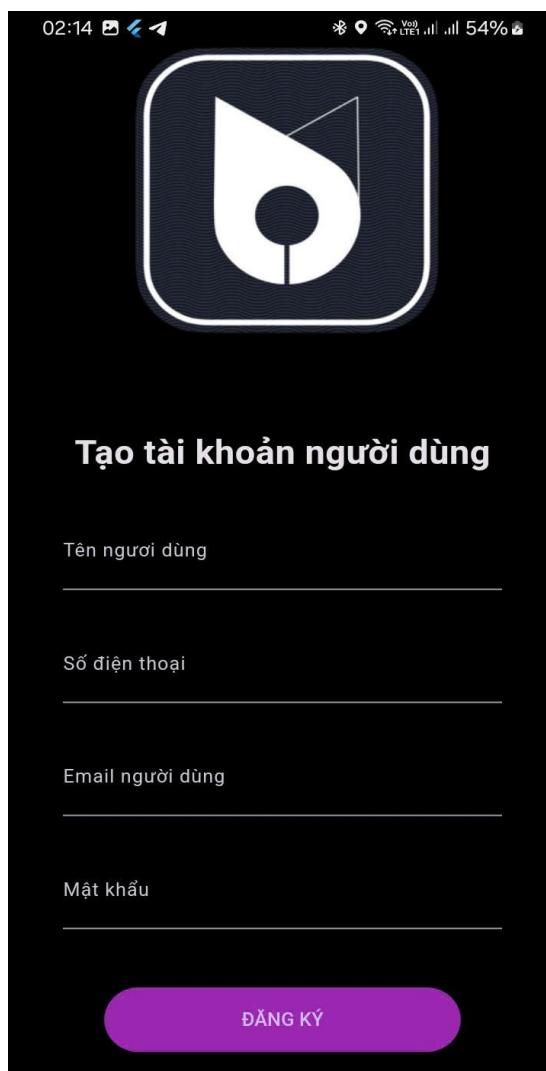
3.1 Mô tả kết quả

3.1.1 Ứng dụng của người dùng

❖ Các bước đặt xe từ người dùng

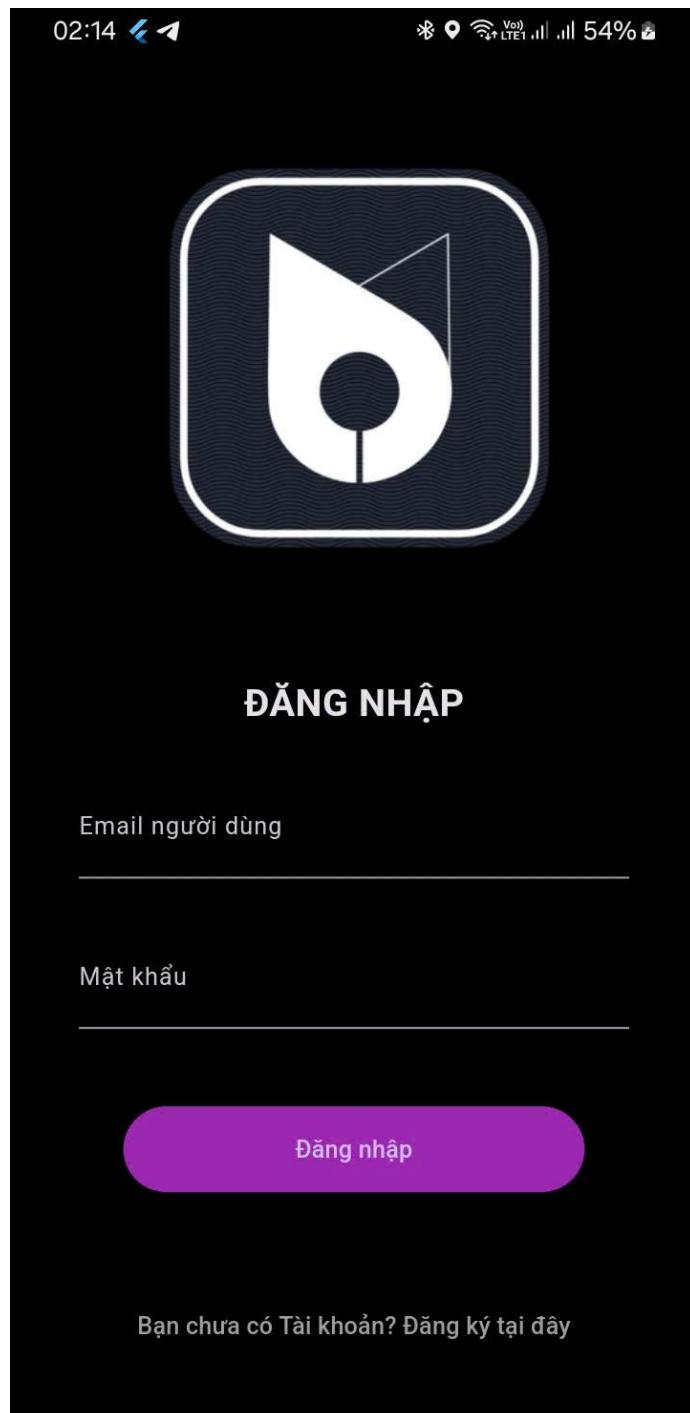
- Mở ứng dụng và đăng nhập
 - Người dùng mở ứng dụng và đăng nhập bằng email và mật khẩu.
 - Sau khi đăng nhập thành công, người dùng được chuyển đến màn hình chính, nơi định vị thời gian thực sẽ tự động lấy vị trí hiện tại của họ.
- Tại màn hình chính, người dùng chọn tùy chọn **icon tìm kiếm** để bắt đầu quy trình.
- Xác định điểm đón: Ứng dụng tự động lấy vị trí hiện tại của người dùng (dựa trên GPS).
- Nhập điểm đến:
 - Người dùng nhập địa chỉ điểm đến hoặc chọn từ gợi ý trên bản đồ.
 - Hệ thống hiển thị tuyến đường dự kiến và khoảng cách giữa điểm đón và điểm đến.
- Xác nhận thông tin đặt xe:
 - Hệ thống hiển thị tóm tắt thông tin đặt xe, bao gồm:
 - Điểm đón và điểm đến.
 - Tuyến đường dự kiến.
 - Thời gian di chuyển ước tính.
 - Giá cước
 - Người dùng nhấn nút **đặt xe** để gửi yêu cầu.
- Kết nối với tài xế:
 - Hệ thống tìm kiếm và gán tài xế gần nhất dựa trên vị trí thời gian thực.
 - Khi tài xế nhận yêu cầu:
 - Người dùng nhận được thông báo và thông tin tài xế (tên, biển số xe, thời gian dự kiến đến).

- Vị trí của tài xế được hiển thị trên bản đồ.
- Theo dõi chuyến đi:
 - Người dùng theo dõi tài xế di chuyển đến điểm đón thông qua bản đồ thời gian thực.
 - Khi tài xế đến, trạng thái cập nhật thành "**Tài xế đã đến**".
- Hoàn thành chuyến đi:
 - Trong suốt chuyến đi, người dùng có thể theo dõi lộ trình trên bản đồ.
 - Khi đến điểm đến, tài xế xác nhận hoàn thành chuyến đi và thu tiền khách hàng.
 - Hệ thống cập nhật trạng thái chuyến đi.
- ❖ Giao diện đăng ký



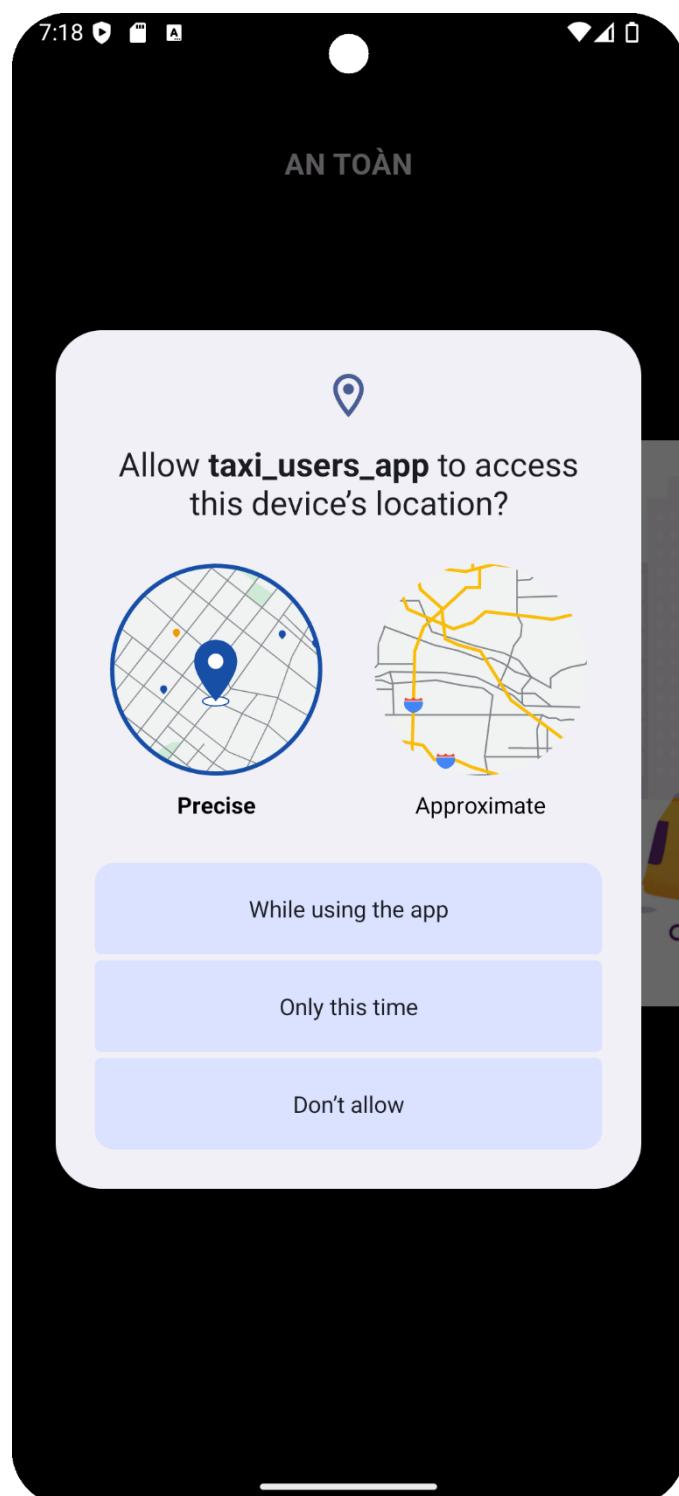
Hình 3.1 Giao diện đăng ký

❖ Giao diện đăng nhập



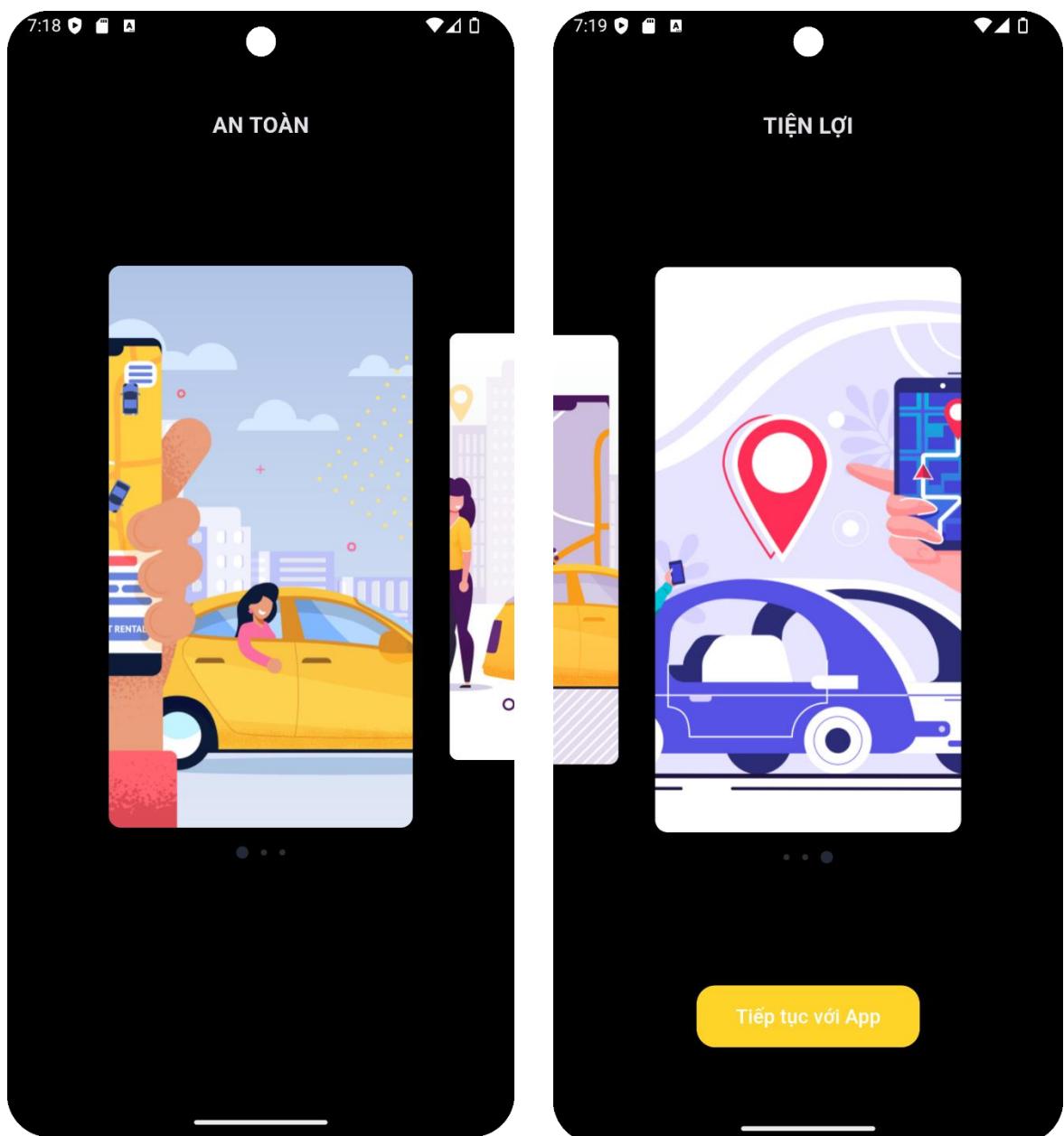
Hình 3.2 Giao diện đăng nhập

- ❖ Giao diện thông báo cho lấy vị trí



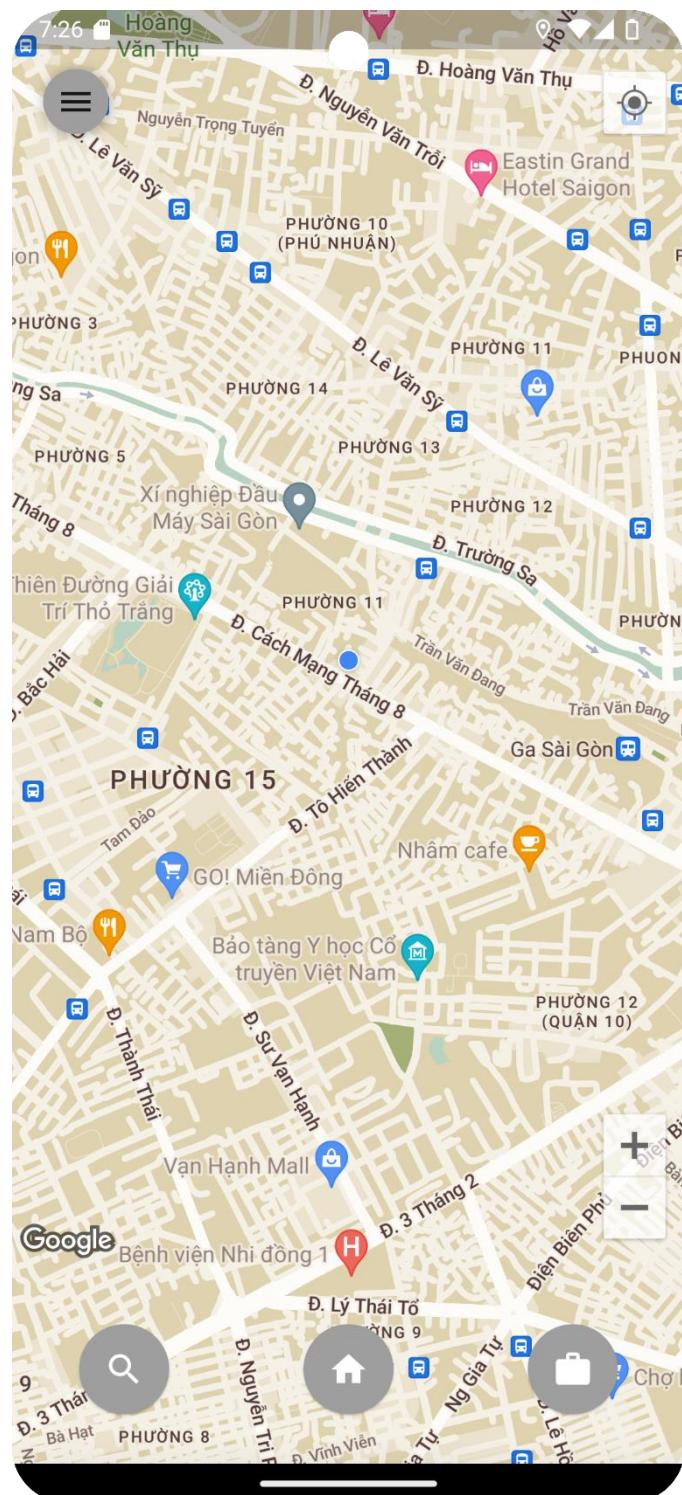
Hình 3.3 Giao diện thông báo lấy vị trí

- ❖ Giao diện giới thiệu về ứng dụng



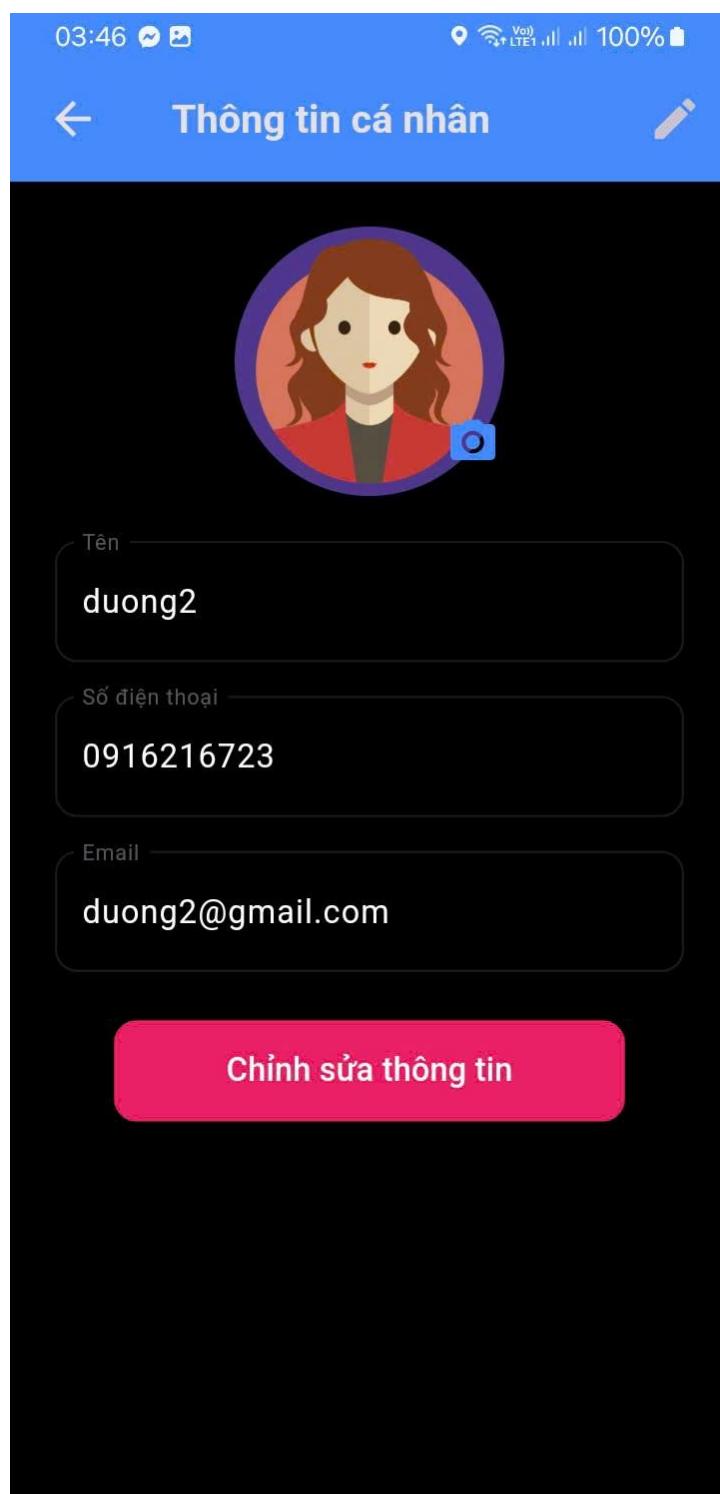
Hình 3.4 Giao diện giới thiệu ứng dụng

❖ Giao diện chính của trang người dùng



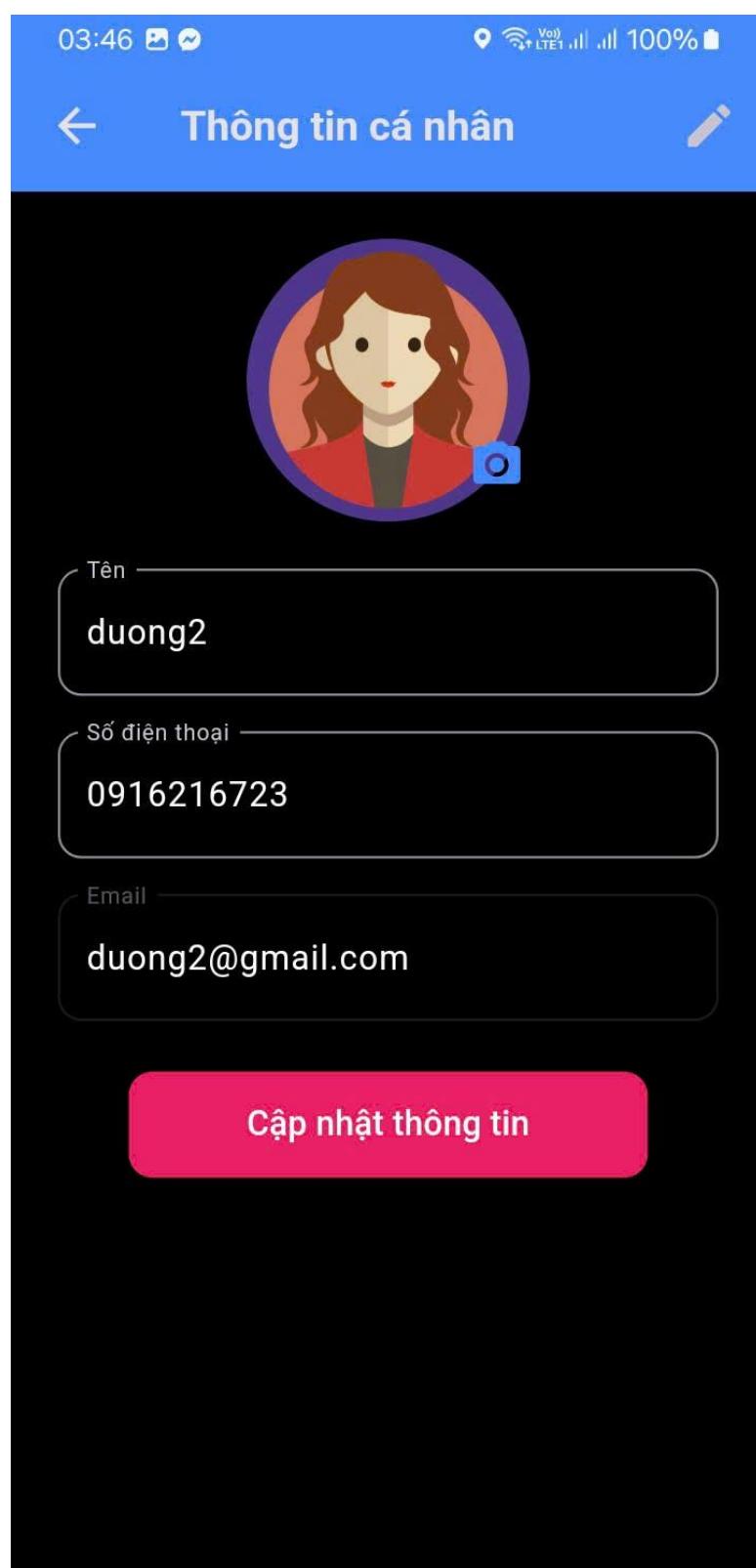
Hình 3.5 Giao diện trang chủ

- ❖ Giao diện thông tin cá nhân



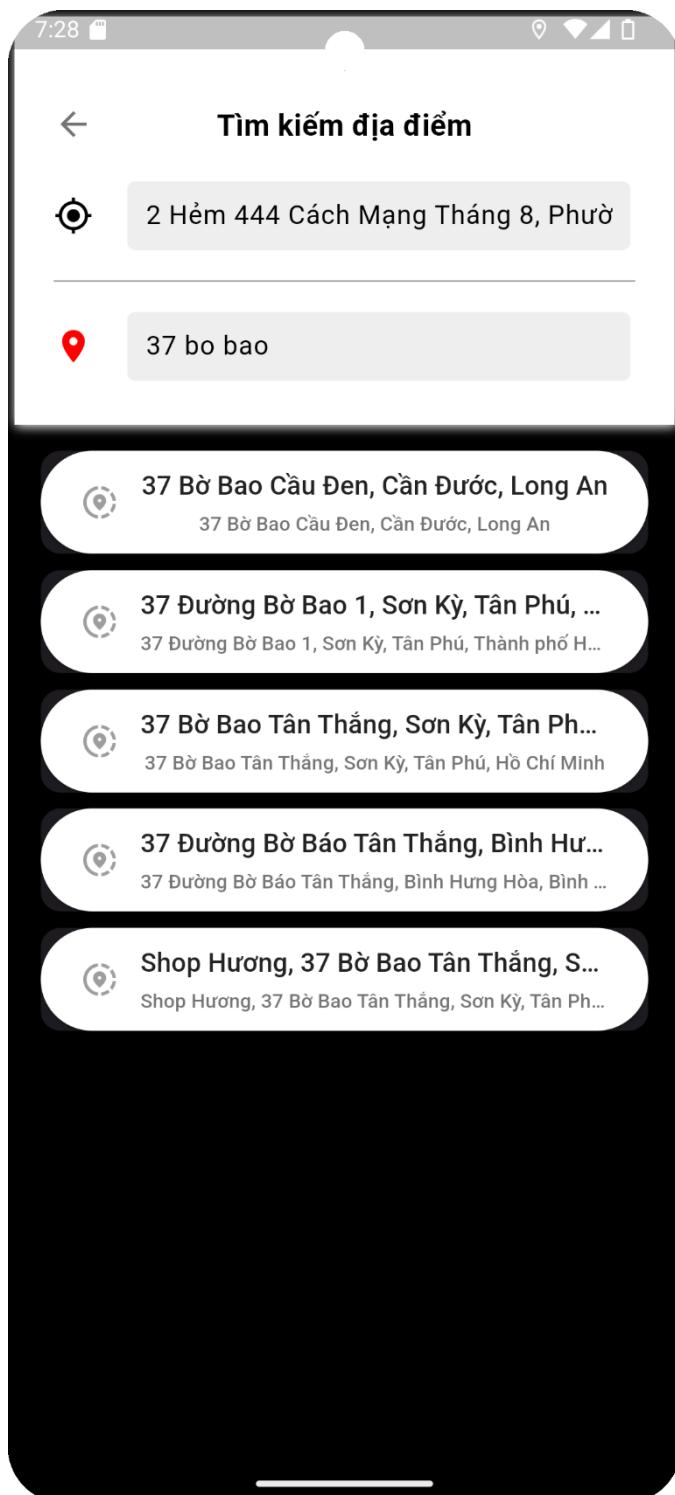
Hình 3.6 Giao diện thông tin cá nhân

- ❖ Giao diện chỉnh sửa thông tin cá nhân



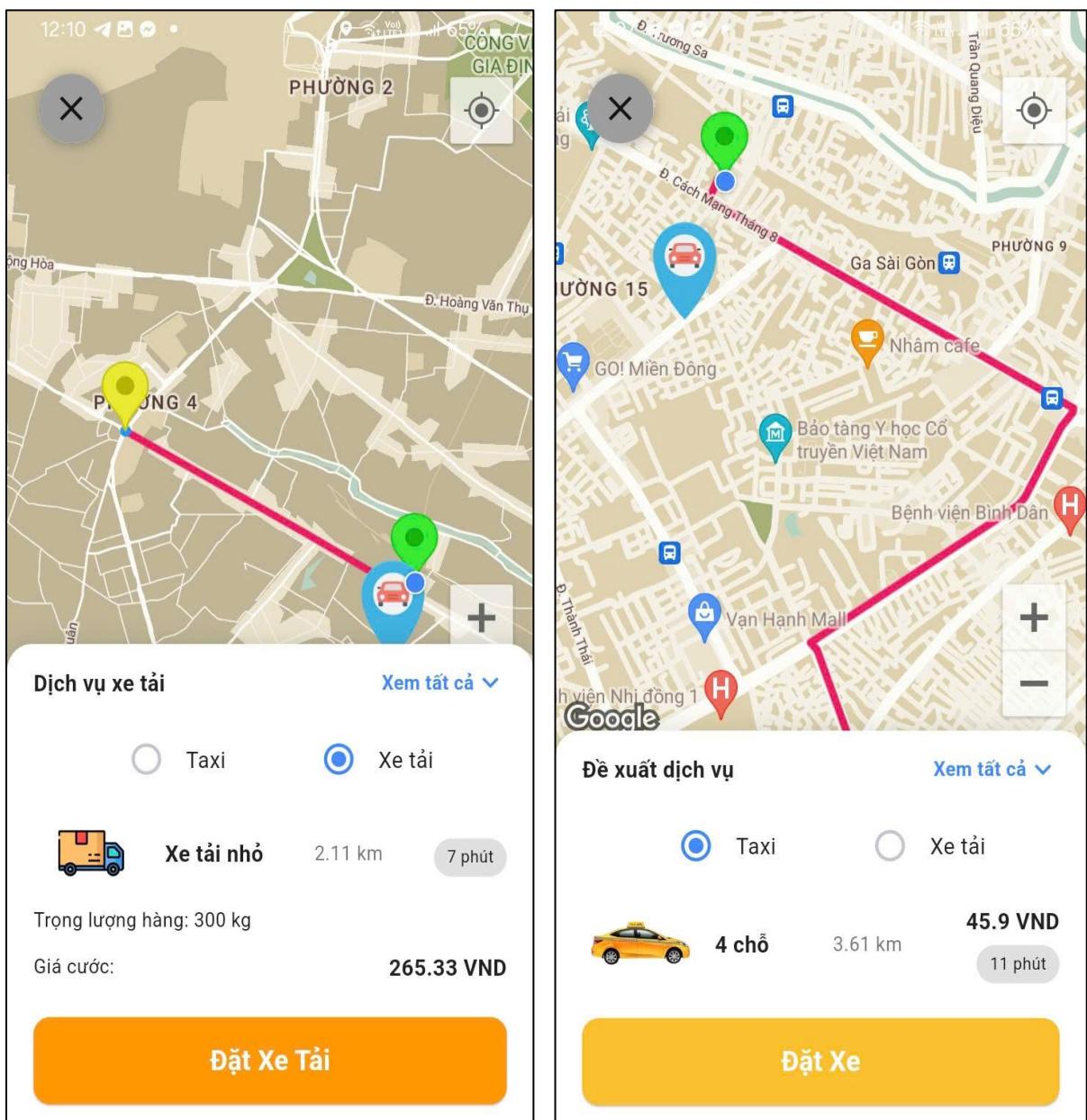
Hình 3.7 Giao diện chỉnh sửa thông tin cá nhân

❖ Giao diện tìm kiếm địa điểm



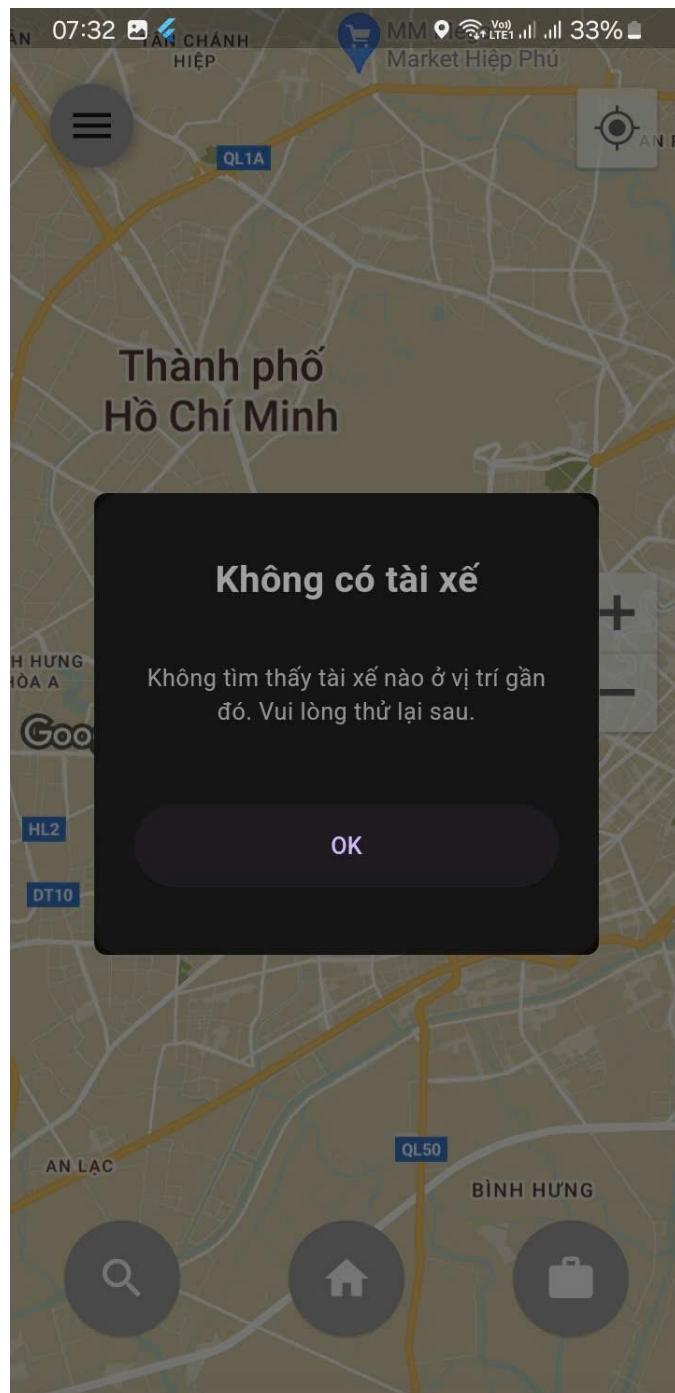
Hình 3.8 Giao diện tìm kiếm địa điểm

❖ Giao diện đặt xe



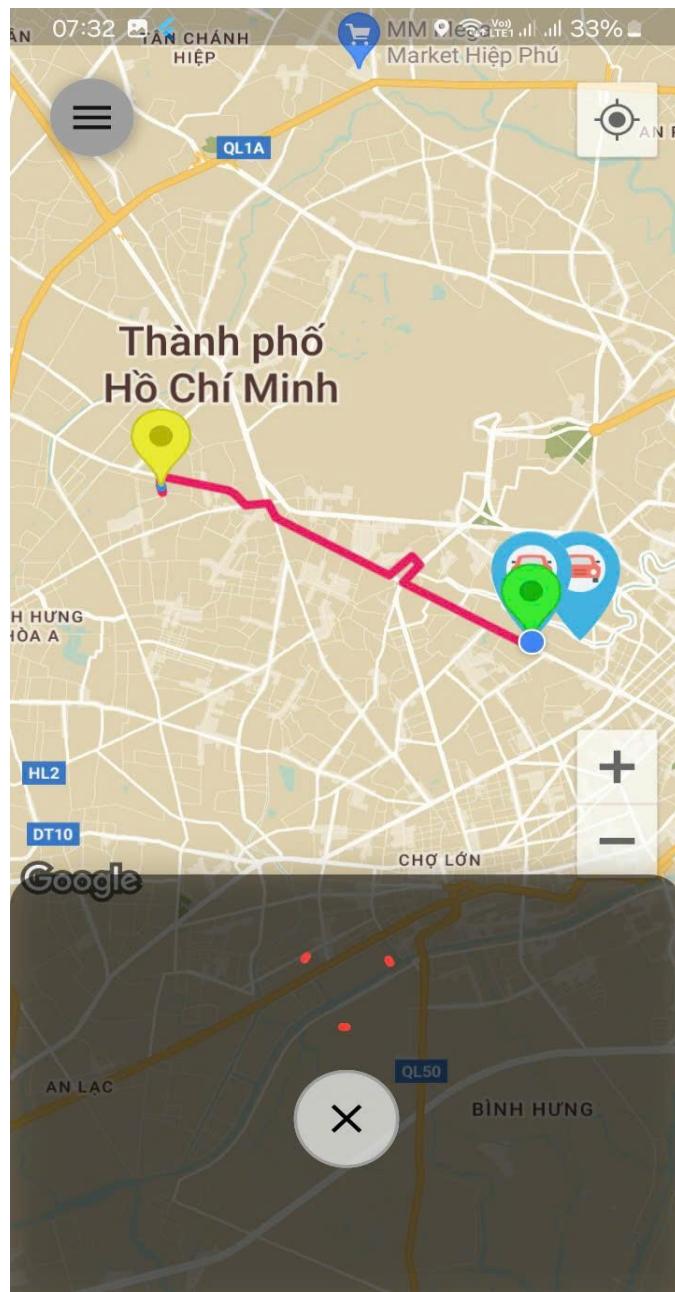
Hình 3.9 Giao diện đặt xe

- ❖ Giao diện thông báo khi không tìm thấy tài xế



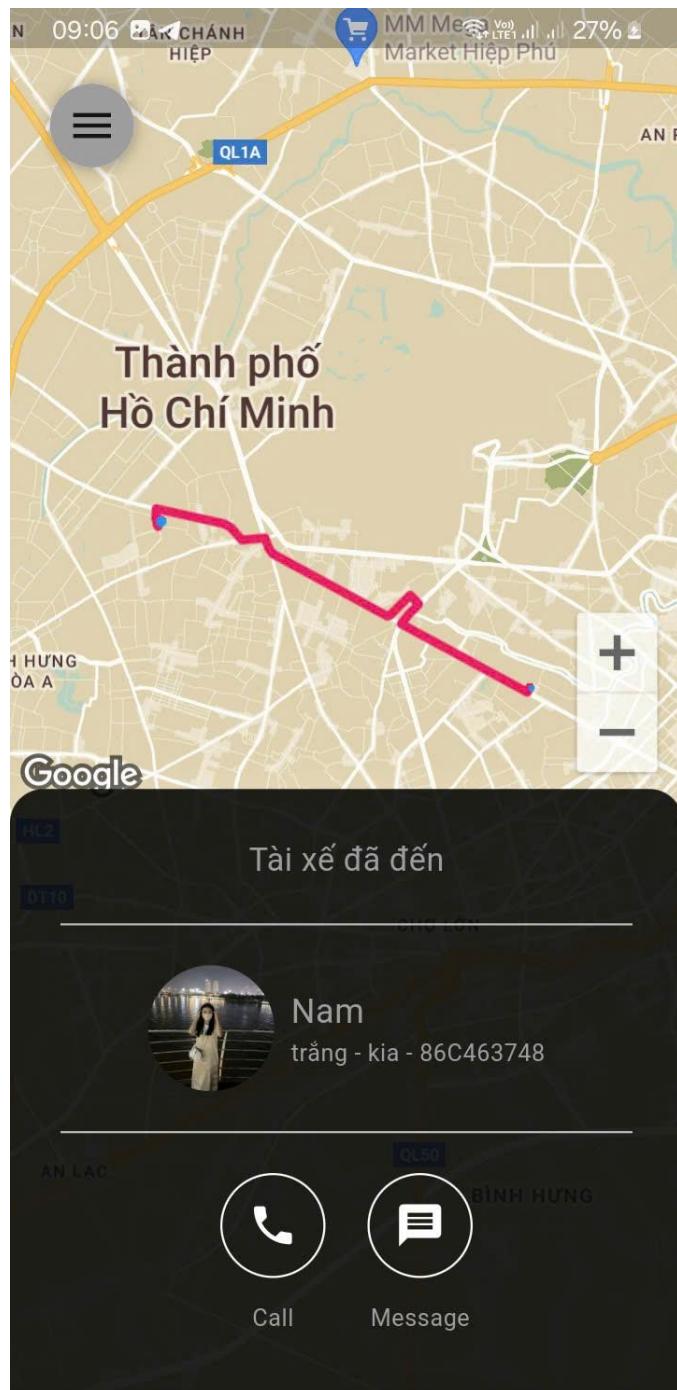
Hình 3.10 Giao diện thông báo không có tài xế

- ❖ Giao diện đang tìm kiếm tài xế



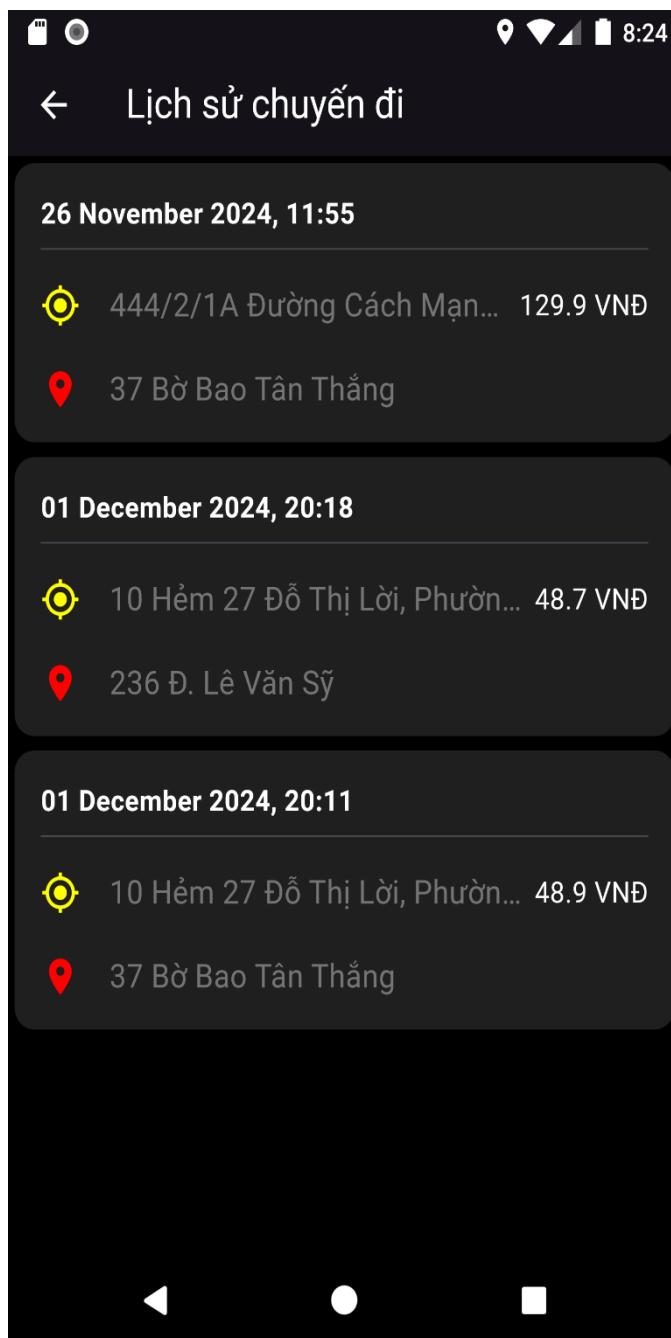
Hình 3.11 Giao diện tìm kiếm tài xế

- ❖ Giao diện sau khi đã đặt xe



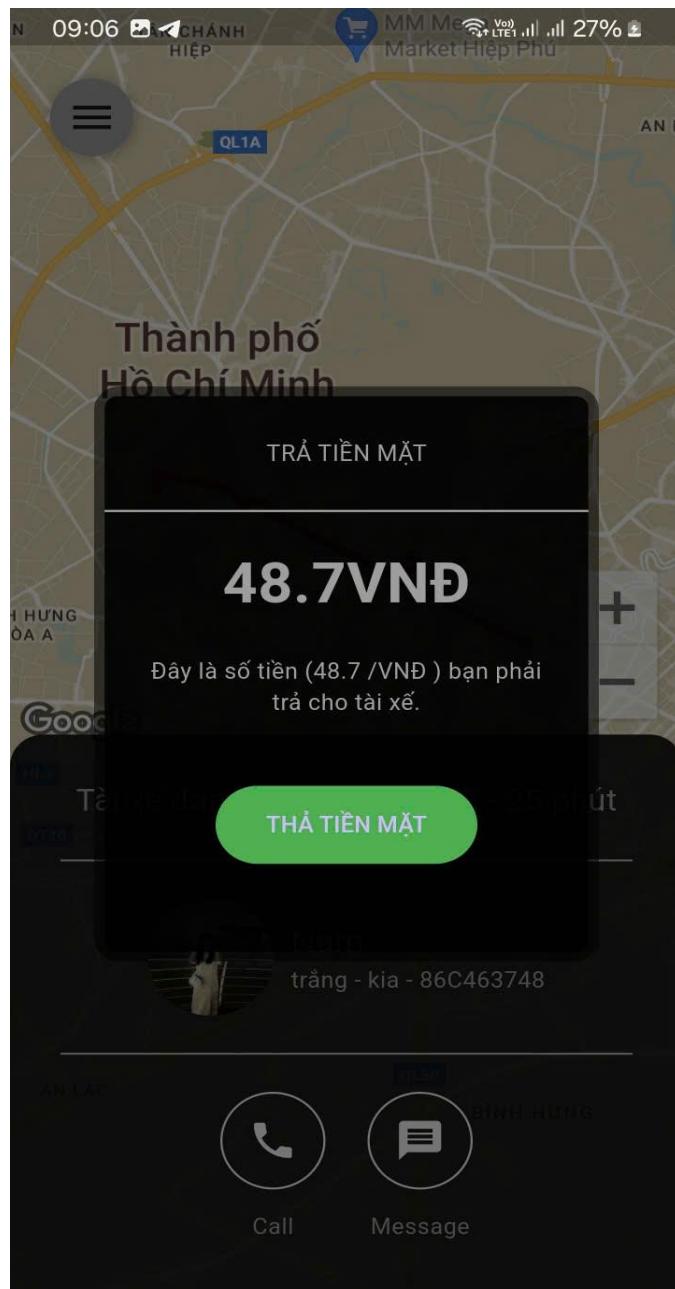
Hình 3.12 Giao diện khi đặt xe xong

- ❖ Giao diện lịch sử chuyến đi



Hình 3.13 Giao diện lịch sử chuyến đi

❖ Giao diện thanh toán



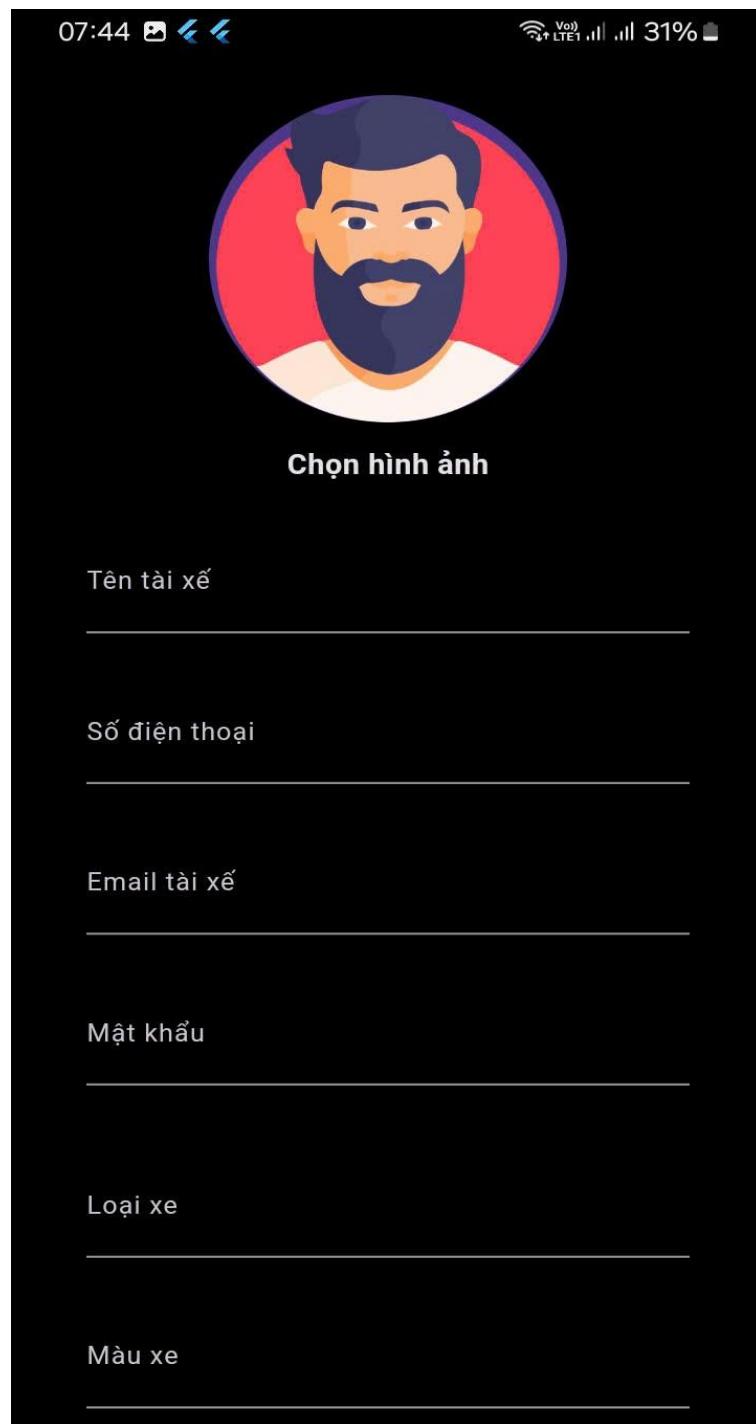
Hình 3.14 Giao diện thanh toán

3.1.2 Ứng dụng tài xế

- ❖ Các bước nhận chuyến đi từ ứng dụng tài xế
- Đăng nhập vào ứng dụng tài xế:
 - Tài xế mở ứng dụng và đăng nhập bằng tài khoản của mình (email và mật khẩu).

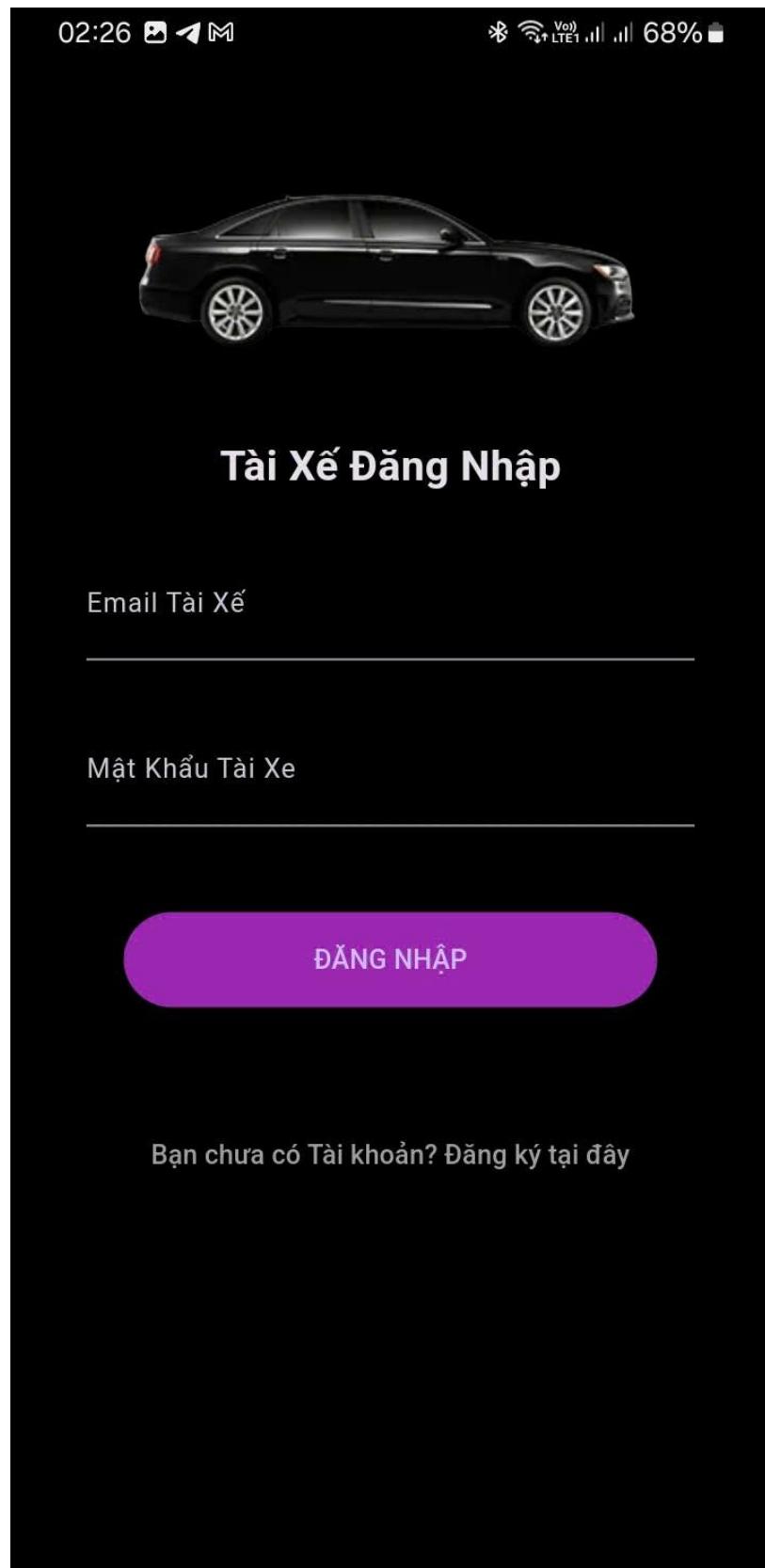
- Sau khi đăng nhập thành công, tài xế được chuyển đến màn hình chính, nơi hiển thị trạng thái hiện tại và thông tin cơ bản.
- Bật trạng thái "Online"
 - Tài xế bật chế độ "**Online**" để thông báo rằng họ đang chấp nhận chuyến đi.
 - Ứng dụng sử dụng định vị GPS để cập nhật vị trí thời gian thực của tài xế và gửi thông tin này đến hệ thống và hiển thị trên bản đồ.
- Nhận yêu cầu đặt xe:
 - Khi người dùng đặt chuyến, hệ thống tự động tìm tài xế phù hợp (dựa trên khoảng cách).
 - Ứng dụng hiển thị yêu cầu chuyến đi với các thông tin
 - Điểm đón và điểm đến.
 - Khoảng cách từ vị trí tài xế đến điểm đón.
 - Giá cước dự kiến
 - Tài xế có giới hạn thời gian 20 giây để chấp nhận hoặc từ chối yêu cầu.
- Chấp nhận yêu cầu:
 - Nếu tài xế đồng ý, họ nhấn nút "**Chấp nhận**".
 - Trạng thái của tài xế được chuyển thành "**Đến**"
 - Vị trí tài xế được cập nhật thời gian thực trên ứng dụng của người dùng.
- Đi đến điểm đón
 - Tài xế sử dụng bản đồ tích hợp trong ứng dụng để dẫn đường đến điểm đón.
- Đón khách
 - Khi đến điểm đón, tài xế nhấn nút "**Đã đến điểm đón**" trên ứng dụng.
 - Hệ thống thông báo cho người dùng rằng tài xế đã đến
 - Sau khi xác nhận hành khách đã lên xe, tài xế nhấn "**Bắt đầu chuyến đi**".
- Thực hiện chuyến đi:
 - Tài xế theo dõi tuyến đường trên bản đồ để di chuyển đến điểm đến.
 - Vị trí của xe được cập nhật thời gian thực cho người dùng.
- Hoàn thành chuyến đi
 - Khi đến điểm đến, tài xế nhấn "**Hoàn thành chuyến đi**".
 - Hệ thống cập nhật trạng thái chuyến đi.

- Thông tin chi tiết về chuyến đi (thời gian, quãng đường, chi phí) được lưu lại.
- ❖ Giao diện đăng ký



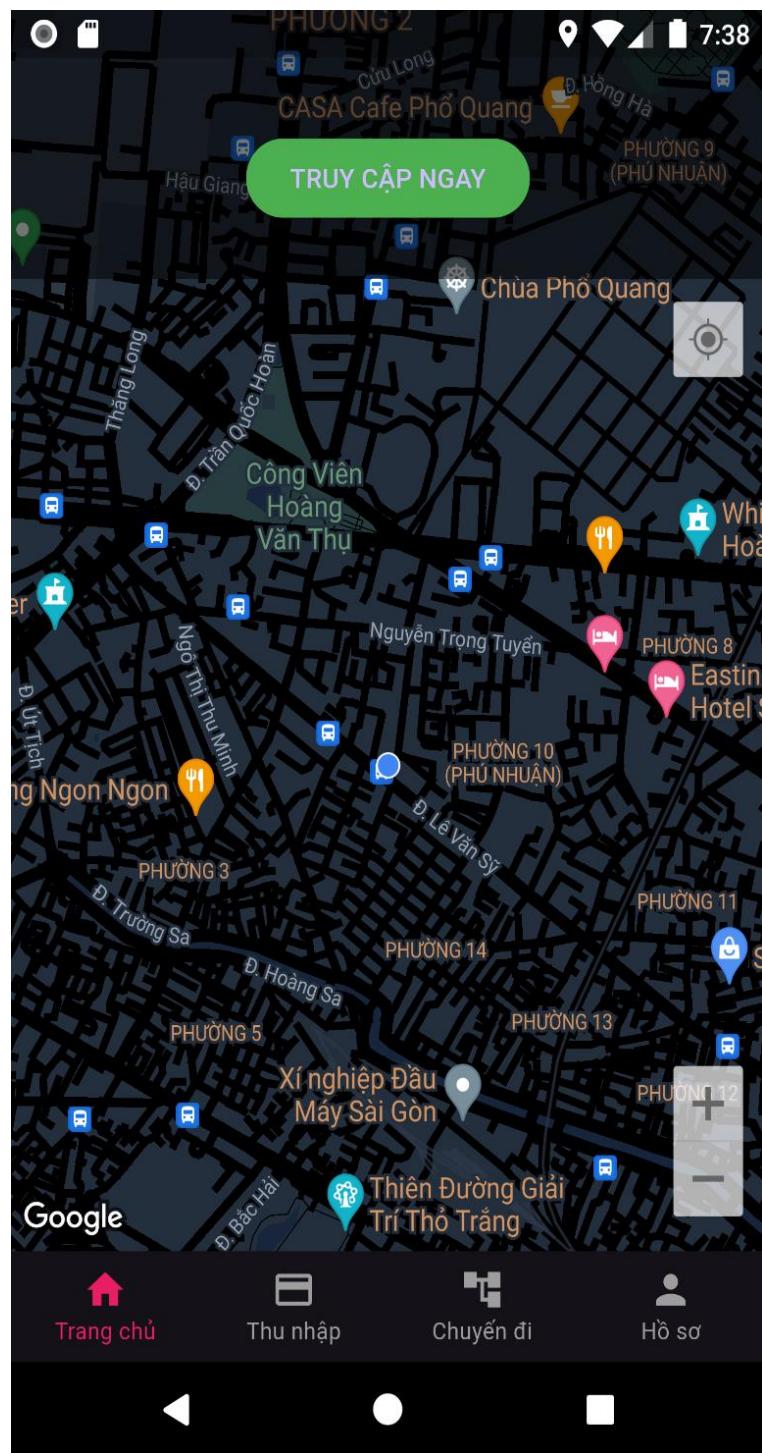
Hình 3.15 Giao diện đăng ký tài xế

- ❖ Giao diện đăng nhập



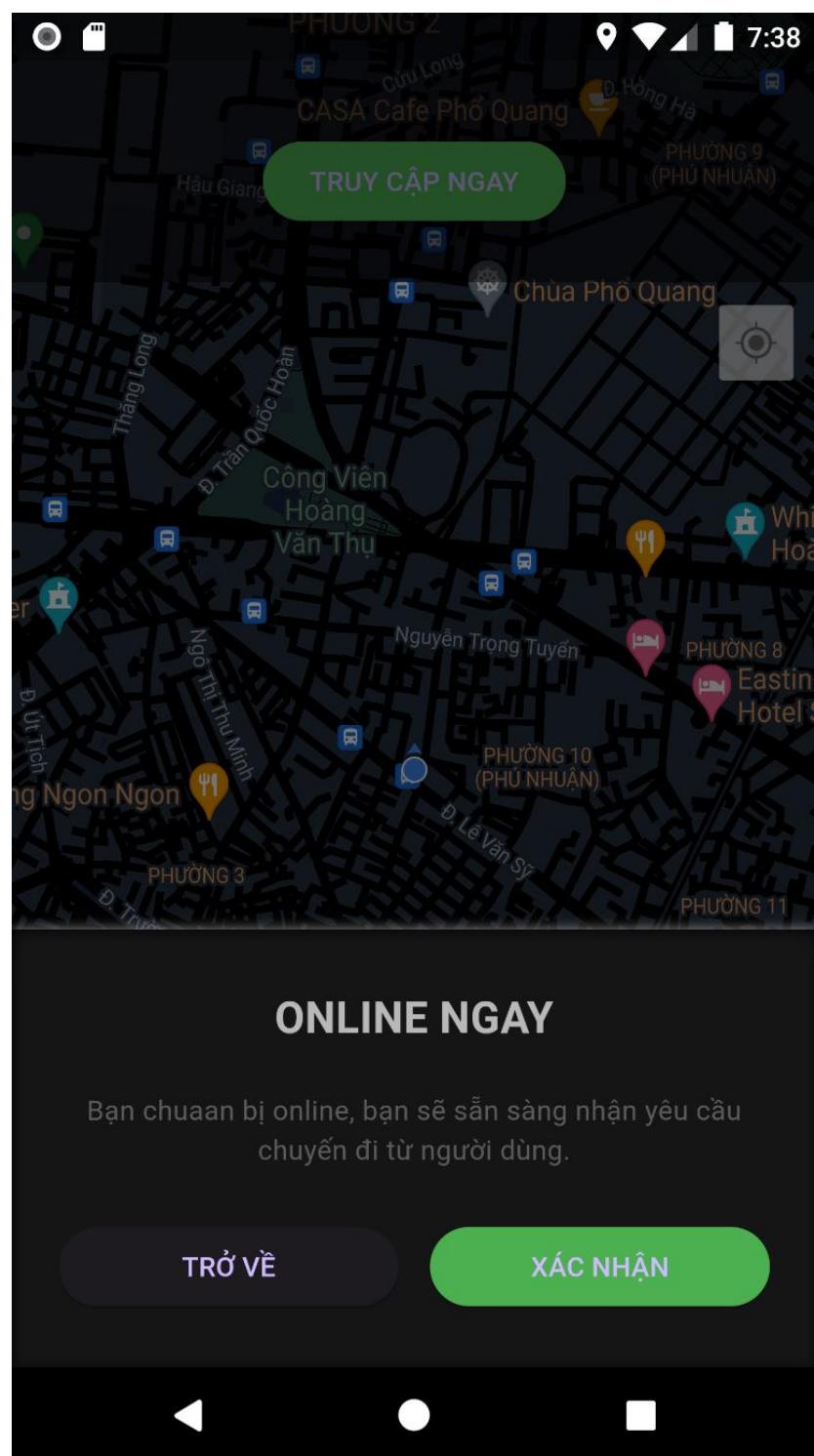
Hình 3.16 Giao diện đăng nhập tài xế

❖ Giao diện trang chủ tài xế



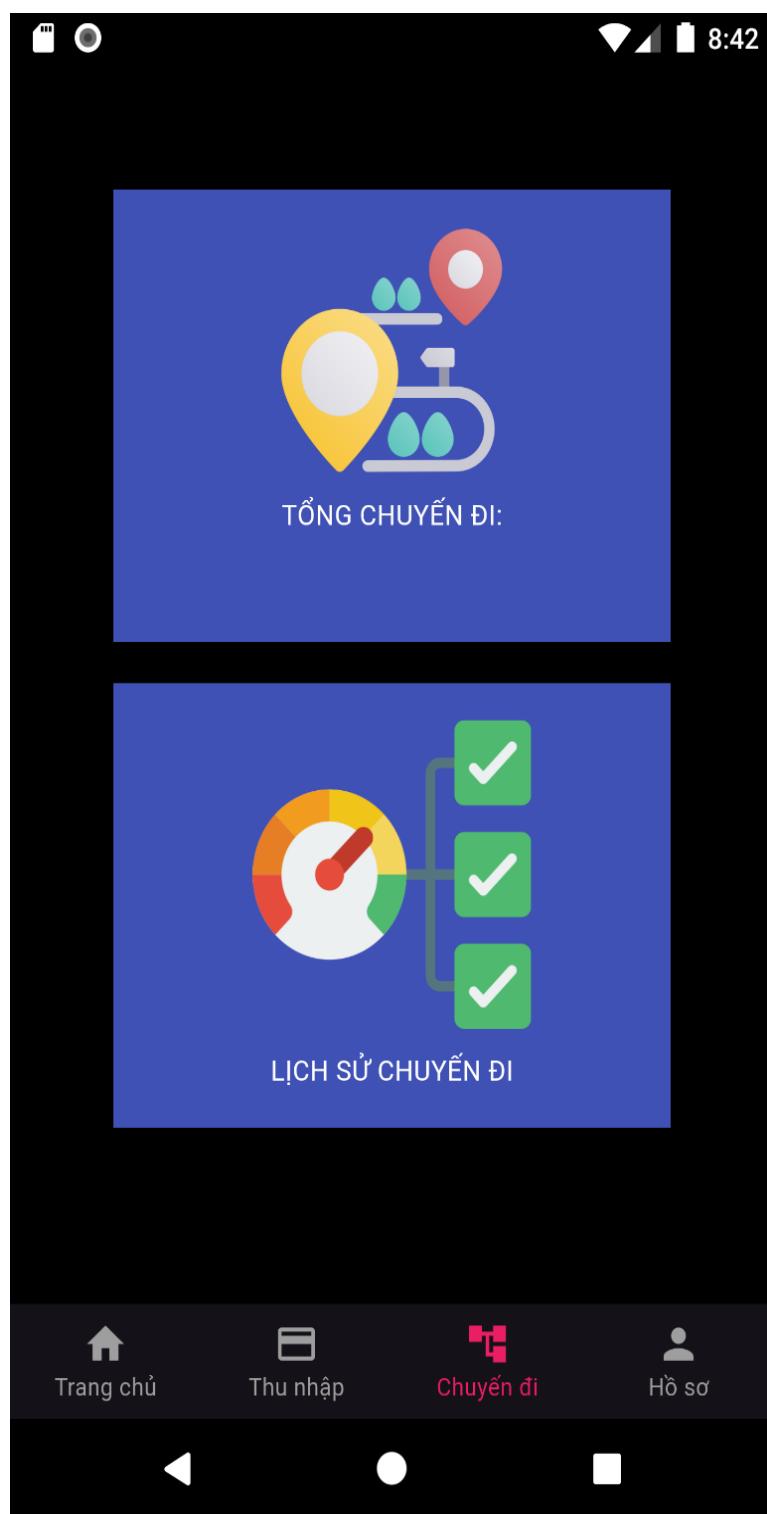
Hình 3.17 Giao diện trang chủ tài xế

- ❖ Giao diện trạng thái tài xế



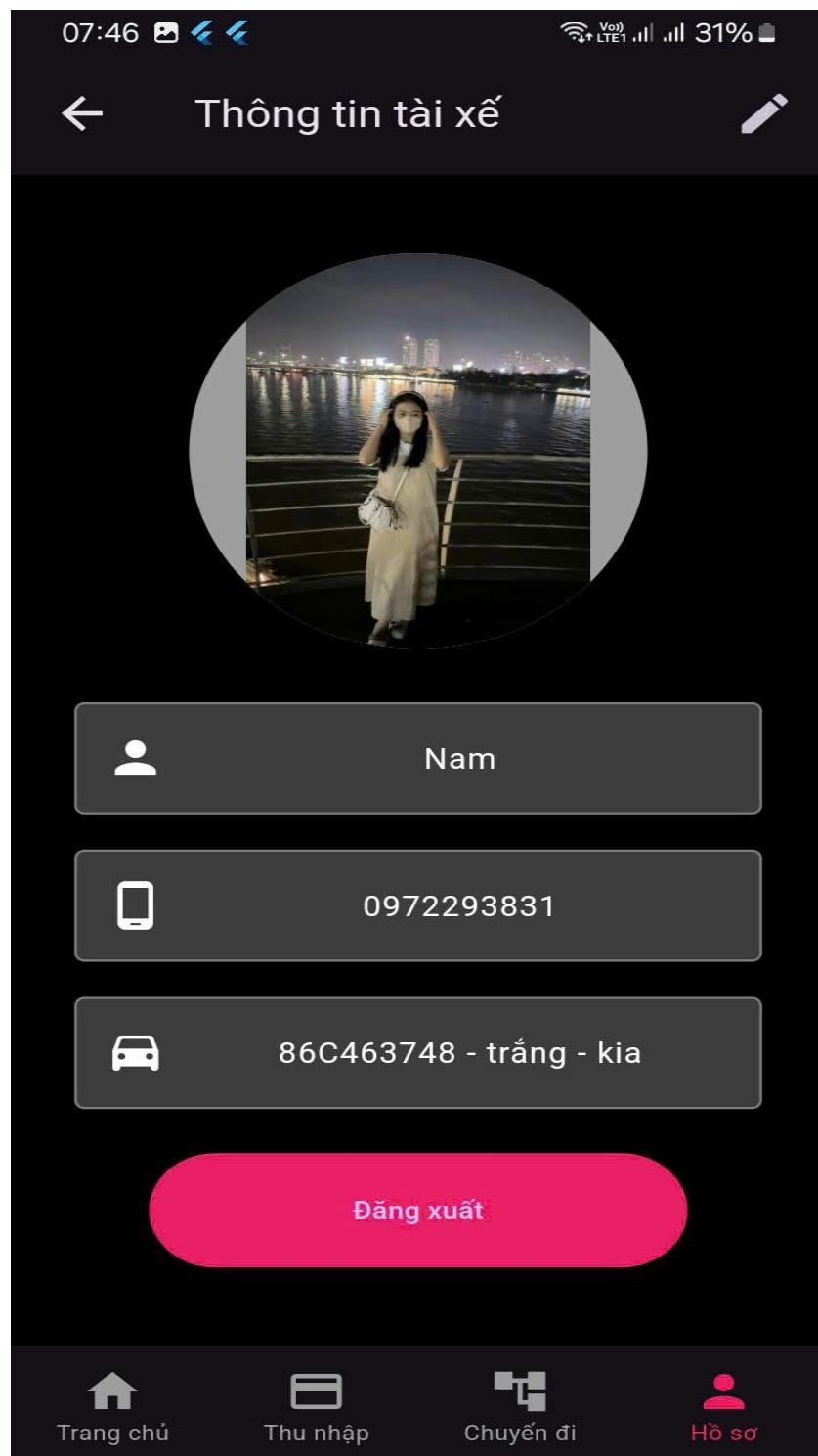
Hình 3.18 Giao diện trạng thái tài xế

- ❖ Giao diện chuyến đi tài xế



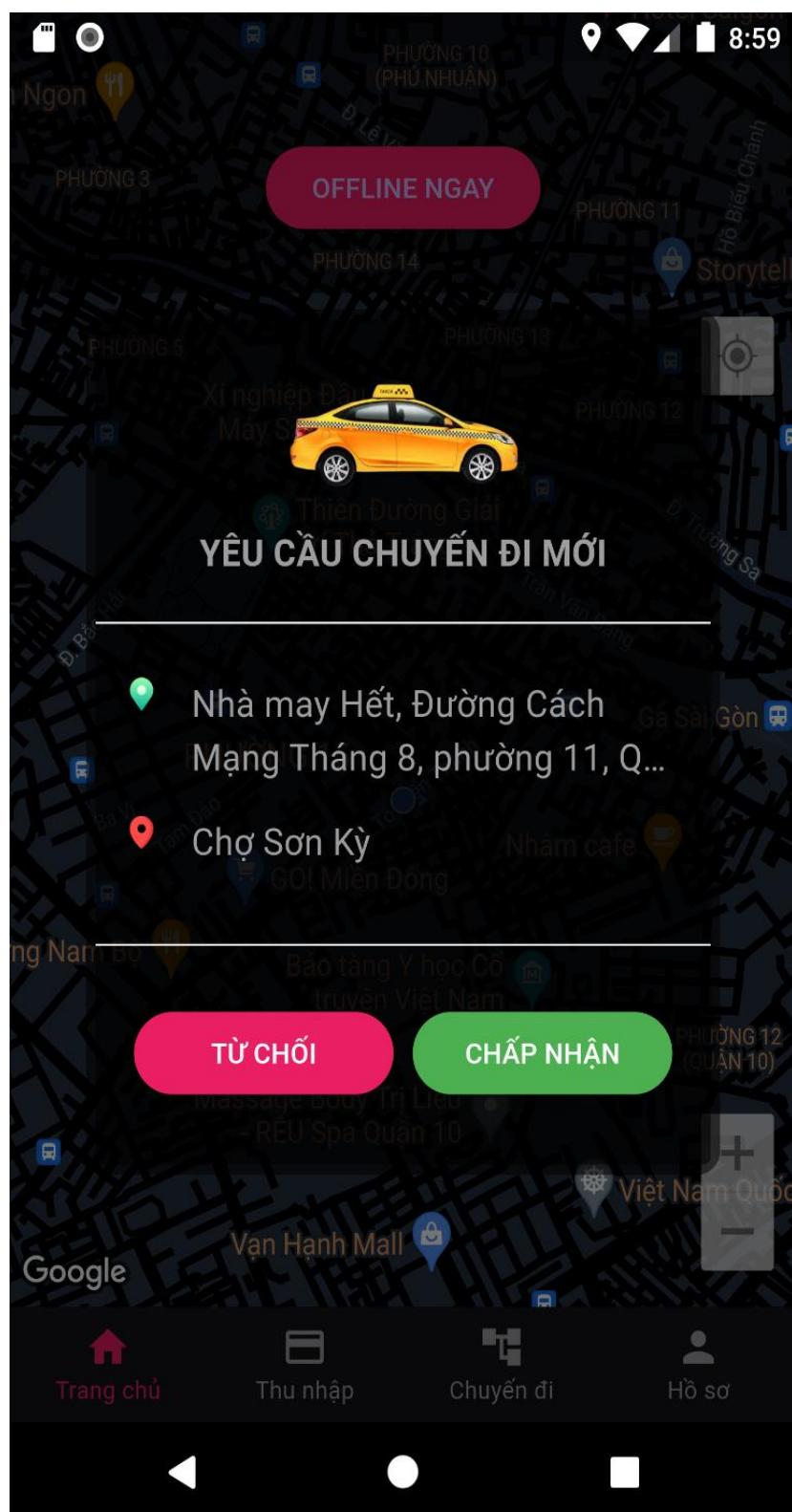
Hình 3.19 Giao diện trang chuyến đi

- ❖ Giao diện hồ sơ tài xế



Hình 3.20 Giao diện trang hồ sơ tài xế

- ❖ Giao diện yêu cầu chuyến đi



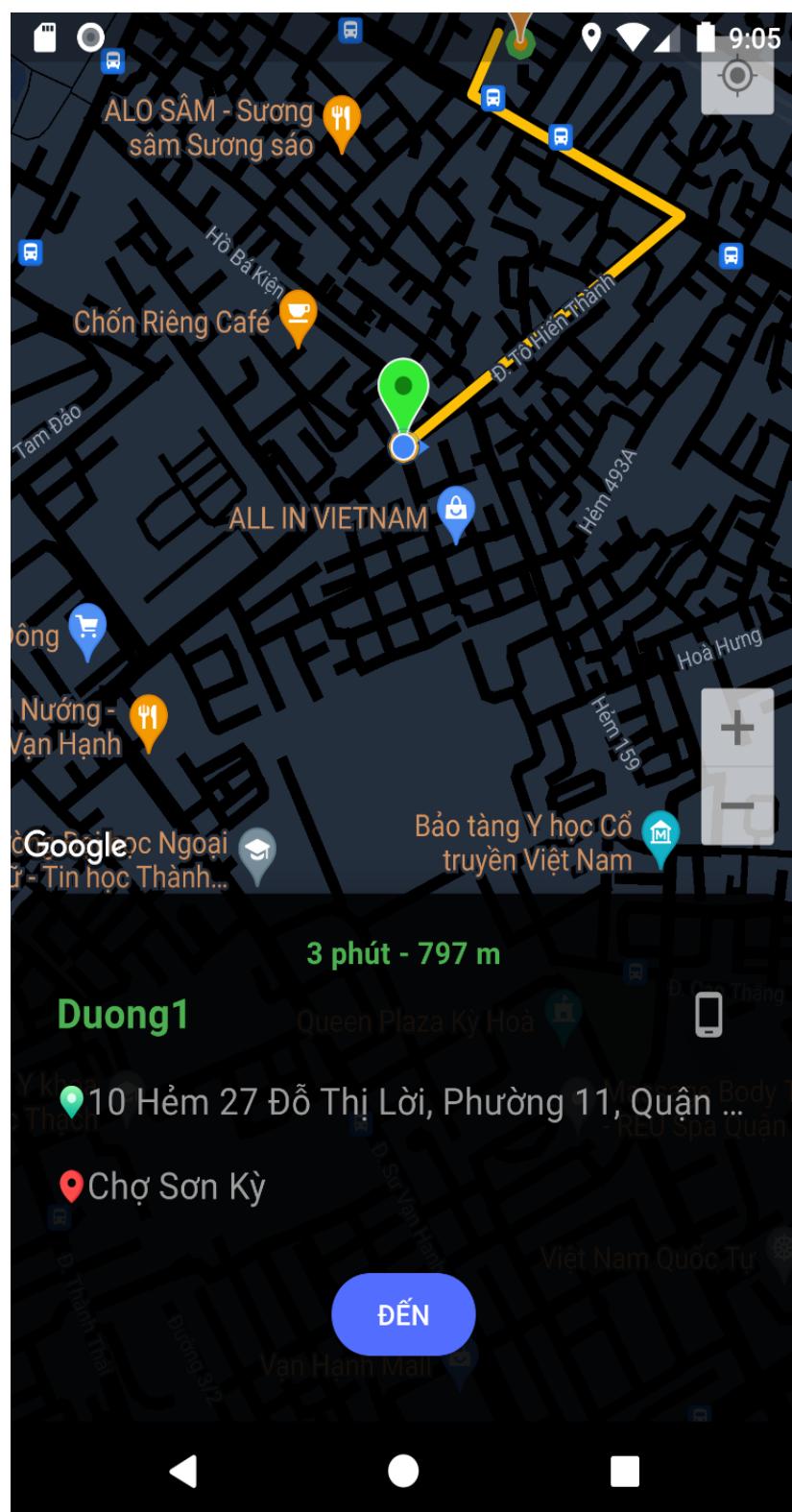
Hình 3.21 Giao diện yêu cầu chuyến đi

- ❖ Giao diện trang thu nhập



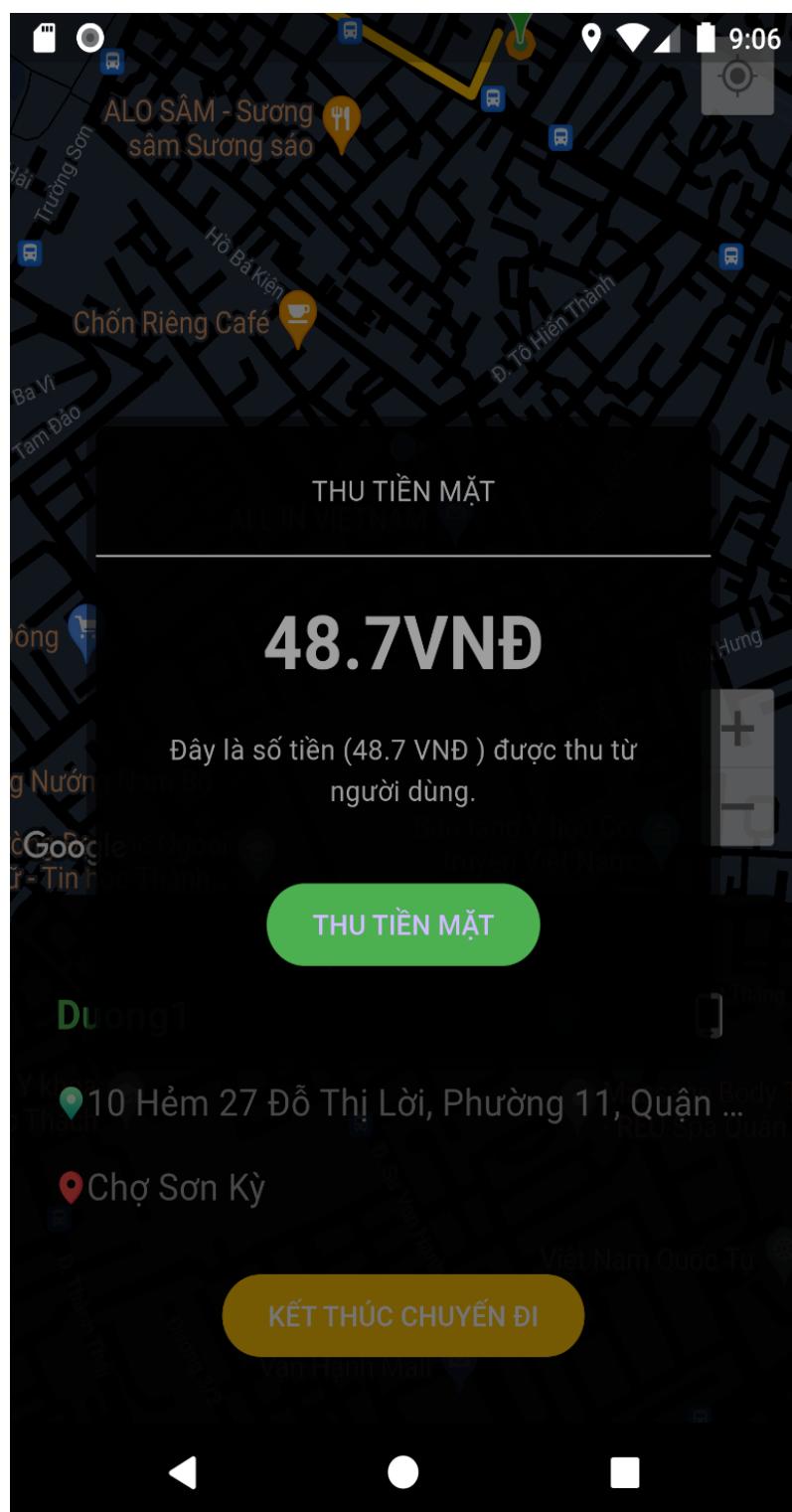
Hình 3.22 Giao diện trang thu nhập

- ❖ Giao diện trong chuyến đi



Hình 3.23 Giao diện trong chuyến đi

- ❖ Giao diện thanh toán



Hình 3.24 Giao diện thanh toán

3.1.3 Web Admin Panel

❖ Giao diện trang chủ admin



Hình 3.25 Giao diện trang admin

❖ Giao diện quản lý tài xế

The screenshot shows the Admin Web driver management interface with a sidebar and a table. The table columns include Driver ID, Hình ảnh, Tên, Chi tiết xe, Điện thoại, Loại xe, Thu nhập, and Trạng thái. Each row contains a 'Chặn Tài Xe' button. The table data is as follows:

Driver ID	Hình ảnh	Tên	Chi tiết xe	Điện thoại	Loại xe	Thu nhập	Trạng thái
80c07bxwg9OqVz5Sqmkwk7YiBvi2		Nguyễn Nhuận Phát	Xanh_Hyundai _ 30G-35307	0984926306	taxi	0 VNĐ	<button>Chặn Tài Xe</button>
f6FLfcBXDKNOEJrGqyqDG0BP5Zp1		Đặng Văn Trường	Trắng_Toyota _ 86C-45367	0944926306	taxi	0 VNĐ	<button>Chặn Tài Xe</button>
hQvglFIW64f2ExKpljopgcP4O3b2		Đoàn Công Hậu	Xám_Hyundai _ 29A-17938	0916216723	truck	0 VNĐ	<button>Chặn Tài Xe</button>
omKWZumudYb8Osdr74Fr4dXyqTaR2		Lê Minh Huy	trắng_toyota _ 59C-24532	0377867956	taxi	237.1 VNĐ	<button>Chặn Tài Xe</button>
qhOitPoH57bOgA0Ur54XfUa65RQ2		Nguyễn Minh Nghĩa	Đỏ_Ford _ 75A-14519	0972293831	truck	1205.2 VNĐ	<button>Chặn Tài Xe</button>

Hình 3.26 Giao diện quản lý tài xế

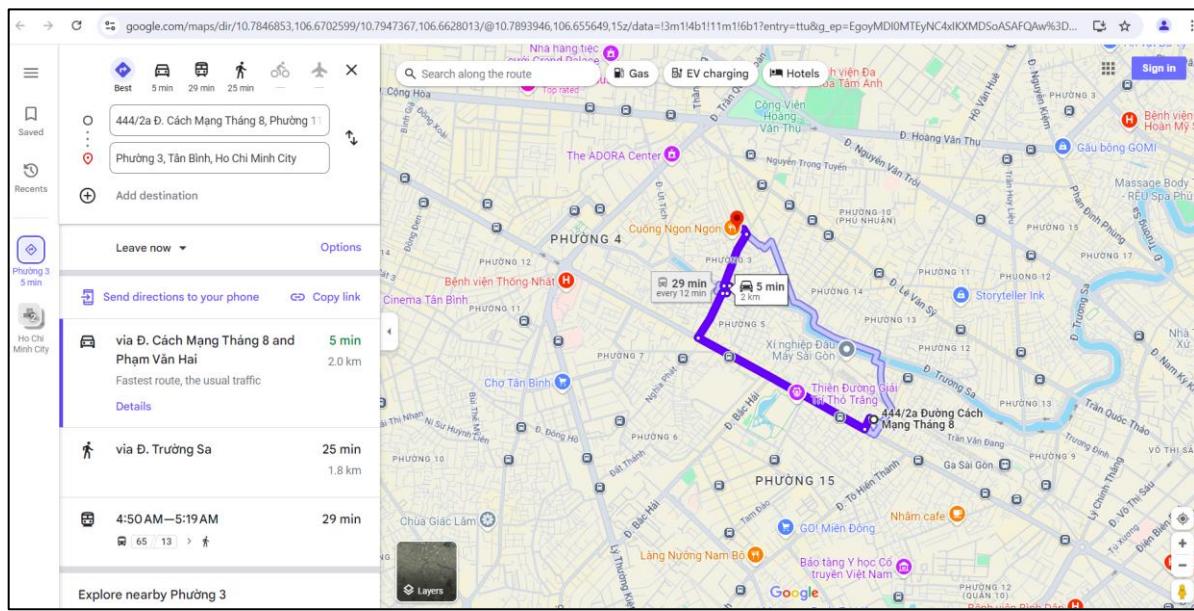
❖ Giao diện quản lý người dùng

Hình 3.27 Giao diện quản lý người dùng

❖ Giao diện quản lý chuyển đổi

Hình 3.28 Giao diện quản lý chuyển đổi

❖ Chi tiết chuyến đi



Hình 3.29 Xem chi tiết chuyến đi

3.2 Test case

3.2.1 Chức năng Google Maps và Goong Maps

Bảng 3.1 Bảng Test Case chức năng Google Maps và Goong Maps

ID	Tên Test Case	Mô tả	Điều kiện đầu vào	Các bước thực hiện	Kết quả kỳ vọng	Trạng thái
TC-01	Định vị vị trí hiện tại	Kiểm tra việc xác định vị trí hiện tại của người dùng.	Ứng dụng đã được cấp quyền truy cập GPS.	1. Mở ứng dụng. 2. Quan sát vị trí hiển thị trên bản đồ.	Vị trí hiện tại của người dùng được hiển thị chính xác trên bản đồ.	Fail
TC-02	Cập nhật vị trí liên tục	Kiểm tra việc bản đồ cập nhật vị trí của tài xế theo thời gian thực.	Tài xế đang di chuyển.	1. Đặt một chuyến xe. 2. Quan sát biểu tượng tài xế trên bản đồ.	Biểu tượng tài xế di chuyển mượt mà và khớp với thực tế.	Pass

3.2.2 Chức năng đặt xe

Bảng 3.2 Bảng Test Case chức năng đặt xe

ID	Tên Test Case	Mô tả	Điều kiện đầu vào	Các bước thực hiện	Kết quả kỳ vọng	Trạng thái
TC-03	Nhập điểm đón/điểm đến	Kiểm tra việc nhập điểm đón và điểm đến của người dùng.	Ứng dụng đã mở bản đồ và giao diện nhập điểm đón/điểm đến.	1. Nhập điểm đón. 2. Nhập điểm đến.	Thông tin điểm đón và điểm đến được nhập chính xác và hiển	Pass

			đến hiển thị.		thị trên bản đồ.	
TC-04	Đặt chuyến xe	Kiểm tra việc gửi yêu cầu đặt xe thành công.	Điểm đón và điểm đến đã được nhập.	1. Nhấn nút "Đặt xe". 2. Quan sát phản hồi từ ứng dụng.	Yêu cầu đặt xe được gửi thành công, tài xế được chỉ định.	Pass

3.2.3 Hiệu suất (Performance)

Bảng 3.3 Bảng Test Case về hiệu suất

ID	Tên Test Case	Mô tả	Điều kiện đầu vào	Các bước thực hiện	Kết quả kỳ vọng	Trạng thái
TC-05	Tốc độ tải bản đồ	Kiểm tra thời gian hiển thị bản đồ sau khi mở ứng dụng.	Kết nối mạng ổn định.	1. Mở ứng dụng. 2. Đo thời gian tải bản đồ.	Bản đồ được hiển thị đầy đủ trong vòng 3 giây.	Fail
TC-06	Độ trễ cập nhật vị trí	Kiểm tra độ trễ giữa vị trí thực tế của tài xế và vị trí hiển thị trên bản đồ.	Tài xế đang di chuyển.	1. Quan sát biểu tượng tài xế trên bản đồ. 2. So sánh với vị trí thực tế.	Độ trễ không vượt quá 2 giây.	Fail

3.2.4 UI/UX

Bảng 3.4 Bảng Test Case về UX/UI

ID	Tên Test Case	Mô tả	Điều kiện đầu vào	Các bước thực hiện	Kết quả kỳ vọng	Trạng thái
TC-07	Kiểm tra hiển thị bản đồ	Kiểm tra kích thước bản đồ hiển thị trên các kích thước màn hình khác nhau.	Ứng dụng đã mở.	1. Kiểm tra giao diện bản đồ trên các thiết bị khác nhau (điện thoại, tablet).	Bản đồ hiển thị đầy đủ, không bị cắt hoặc tràn.	Pass
TC-08	Kiểm tra biểu tượng tài xế	Kiểm tra sự rõ ràng của biểu tượng tài xế trên bản đồ.	Tài xế đang di chuyển.	1. Quan sát biểu tượng tài xế trên bản đồ. 2. So sánh với vị trí thực tế.	Biểu tượng tài xế rõ ràng, không bị mờ hoặc biến dạng.	Pass

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

1 Đánh giá kết quả đạt được

Đề tài "**Xây dựng ứng dụng đặt xe với Firebase và Flutter**" đã đạt được một số chức năng chính bao gồm:

- Xây dựng được một ứng dụng đơn giản cho người dùng đặt xe, theo dõi tài xế theo thời gian thực.
- Xây dựng được một ứng dụng cho tài xế với các chức năng cơ bản và đơn giản.
- Dữ liệu giữa tài xế và người dùng được đồng bộ hóa nhờ Firebase Realtime Database.
- Sử dụng Goong Maps API để tối ưu lộ trình và sử dụng Google Map API hiển thị bản đồ trực quan.

Tuy nhiên, một số khía cạnh chưa được giải quyết triệt để, như tích hợp hệ thống thanh toán trực tuyến hoặc tối ưu thuật toán tìm kiếm tài xế, để đảm bảo trải nghiệm tốt hơn.

2 Kiến thức lý thuyết được củng cố

- Kiến thức về lập trình đa nền tảng với **Flutter**.
- Hiểu biết sâu hơn về các giao thức thời gian thực, đặc biệt trong việc sử dụng Firebase Realtime Database để quản lý dữ liệu động.
- Hiểu rõ hơn về cách hoạt động của API (Goong Maps API, Firebase Authentication, Firebase Messaging).

3 Thảo luận kết quả

Kết quả đạt được đã đáp ứng các mục tiêu cơ bản, tuy nhiên cần bổ sung và cải thiện một số vấn đề như:

- Tích hợp thêm các tính năng nâng cao (thanh toán online, đánh giá tài xế).

- Cải thiện khả năng mở rộng của ứng dụng, nhất là khi số lượng người dùng tăng cao.
- Tối ưu hóa hiệu năng của ứng dụng khi làm việc với dữ liệu lớn.
- Cải thiện lại giao diện người dùng cho cả hệ thống bao gồm ứng dụng người dùng, ứng dụng tài xế và web admin.

4 Hướng phát triển

Dựa trên các kết quả và vấn đề còn tồn tại, ứng dụng có thể được phát triển thêm theo các hướng sau:

- Tích hợp thanh toán trực tuyến: Kết nối với các cổng thanh toán như MoMo, ZaloPay để hỗ trợ thanh toán tiện lợi cho người dùng.
- Hệ thống đánh giá và phản hồi: Xây dựng thêm chức năng đánh giá để khách hàng có thể đánh giá tài xế sau mỗi chuyến đi, góp phần nâng cao chất lượng dịch vụ.
- Phân tích dữ liệu: Tích hợp công cụ phân tích dữ liệu Firebase Analytics để theo dõi hiệu suất hoạt động và hành vi người dùng.
- Nâng cấp thuật toán tìm kiếm tài xế: Sử dụng các thuật toán tối ưu để giảm thời gian tìm kiếm tài xế gần nhất.
- Quản lý lộ trình thông minh: Xây dựng tính năng để xuất lộ trình dựa trên tình trạng giao thông thực tế.
- Cải thiện lại giao diện sao cho đẹp mắt.

TÀI LIỆU THAM KHẢO**Tiếng Việt:**

- [1] N. N. Thắng, Lập trình Flutter: Xây dựng ứng dụng di động từ cơ bản đến nâng cao. Hà Nội, Việt Nam: NXB Hồng Đức, 2022.
- [2] T. D. Thanh, Hướng dẫn lập trình Firebase: Xây dựng hệ thống ứng dụng hoàn chỉnh. TP. Hồ Chí Minh, Việt Nam: NXB Lao động, 2023.
- [3] Tạp chí Công nghệ Thông tin Việt Nam, “Ứng dụng Flutter trong phát triển hệ thống đặt xe trực tuyến,” Tạp chí Công nghệ Thông tin Việt Nam, số 3, trang 15-22, 2023.
- [4] TopDev. (n.d.). *Firebase là gì?*. Retrieved December 8, 2024, from <https://topdev.vn/blog/firebase-la-gi/>.
- [5] Firebase. (n.d.). *Realtime Database*. Retrieved December 8, 2024, from <https://firebase.google.com/docs/database?hl=vi>.
- [6] 200Lab. (n.d.). *Tự học ngôn Dart: Những điều cần biết trước khi bắt đầu*. Retrieved December 8, 2024, from <https://200lab.io/blog/tu-hoc-ngon-dart-nhung-dieu-can-biet-truoc-khi-bat-dau/>.
- [7] Android Developers. (n.d.). *Giới thiệu về Android Studio*. Retrieved December 8, 2024, from <https://developer.android.com/studio/intro?hl=vi>.

Tiếng Anh:

- [8] D. Bayliss, *Flutter Projects: A Practical Guide to Building Cross-Platform Mobile Applications*. Apress, 2023.
- [9] Google Developers, *Flutter Documentation*. [Online]. Available: <https://flutter.dev/docs>. [Accessed: Dec. 1, 2024].
- [10] Dart Team. (n.d.). Overview. Retrieved December 8, 2024, from <https://dart.dev/overview#learning-dart>.