

**TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN**



**NGUYỄN KHÁNH MINH - 51900132
TRẦN MINH TÀI - 51900204**

XÂY DỰNG HỆ THỐNG ĐẶT VÉ XE TÍCH HỢP AI

DỰ ÁN CÔNG NGHỆ THÔNG TIN 2

KỸ THUẬT PHẦN MỀM

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023

**TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN**



NGUYỄN KHÁNH MINH - 51900132

TRẦN MINH TÀI - 51900204

XÂY DỰNG HỆ THỐNG ĐẶT VÉ XE TÍCH HỢP AI

DỰ ÁN CÔNG NGHỆ THÔNG TIN 2

KỸ THUẬT PHẦN MỀM

Người hướng dẫn

ThS. Nguyễn Quốc Bình

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023

LỜI CẢM ƠN

Trước tiên, nhóm chúng tôi xin gửi lời cảm ơn chân thành và lòng biết ơn sâu sắc đến ThS. Nguyễn Quốc Bình. Là một người đã luôn hỗ trợ và hướng dẫn tận tình cho nhóm trong suốt quá trình nghiên cứu và hoàn thành dự án với đề tài ***“Xây dựng hệ thống web đặt vé xe có tích hợp AI”***.

Trong thời gian học tập và rèn luyện tại trường Đại học Tôn Đức Thắng, nhóm chúng tôi đã nhận được rất nhiều sự giúp đỡ tận tình của quý thầy cô. Thầy cô là người đã hướng dẫn và truyền cảm hứng trong học tập từ những ngày đầu tiên bước chân vào môi trường đại học. Bên cạnh đó, quý thầy cô còn truyền đạt rất nhiều kiến thức hay và bổ ích, rèn luyện những kỹ năng hữu ích cho công việc trong tương lai. Với lòng biết ơn sâu sắc từ tận đáy lòng, chúng tôi xin chân thành gửi lời cảm ơn đến quý thầy cô, đặc biệt là ThS. Nguyễn Quốc Bình – người đã trực tiếp dẫn dắt nhóm chúng tôi trong môn DACNTT2.

Dự án này là kết quả của sự nỗ lực tìm hiểu và kiến thức mà nhóm chúng tôi đã học và tìm kiếm thông tin nên nó khó tránh khỏi những sai sót, rất mong quý thầy cô có thể đóng góp ý kiến để có được thêm nhiều bài học và rút kinh nghiệm cho lần sau hoàn thành tốt hơn.

Lời cuối cùng nhóm chúng tôi xin chân thành cảm ơn và gửi lời chúc tốt đẹp nhất đến quý thầy cô đã tạo cơ hội cho chúng tôi có thể trau dồi bản thân trong môn học này.

TP. Hồ Chí Minh, ngày 04 tháng 09 năm 2023

Tác giả

(Ký tên và ghi rõ họ tên)

CÔNG TRÌNH ĐƯỢC HOÀN THÀNH TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG

Nhóm xin cam đoan đây là sản phẩm đồ án của riêng của nhóm và được sự hướng dẫn của ThS. Nguyễn Quốc Bình. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, dự án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

Nếu phát hiện có bất kỳ sự gian lận nào chúng tôi xin hoàn toàn chịu trách nhiệm về nội dung đồ án của mình. Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do nhóm gây ra trong quá trình thực hiện (nếu có).

TP. Hồ Chí Minh, ngày 04 tháng 09 năm 2023

Tác giả

(ký tên và ghi rõ họ tên)

XÂY DỰNG TRANG WEB ĐẶT VÉ XE CÓ TÍCH HỢP AI

TÓM TẮT

Dự án này tập trung vào việc phát triển một web đặt vé xe kết hợp AI, nhằm tạo ra một công cụ thuận tiện và nhanh chóng để người dùng có thể tìm kiếm, so sánh và đặt vé xe từ điểm A đến điểm B. Web đặt vé xe được thiết kế để cung cấp cho người dùng các tính năng chính sau:

1. **Tìm kiếm và đặt vé:** Web sẽ cung cấp cho người dùng một công cụ tìm kiếm thông minh, cho phép họ tìm kiếm các chuyến xe đi giữa các địa điểm khác nhau. Người dùng có thể tìm kiếm theo điểm khởi hành, điểm đến, thời gian và loại hình xe như xe buýt, tàu hỏa, xe khách hoặc taxi.
2. **So sánh giá vé:** Web sẽ hiển thị cho người dùng các lựa chọn về giá vé từ các nhà vận chuyển khác nhau. Người dùng có thể so sánh giá vé, thời gian đi, đánh giá của nhà vận chuyển và các tiện ích đi kèm để chọn được lựa chọn phù hợp với nhu cầu và ngân sách của họ.
3. **Đặt vé trực tuyến:** Người dùng có thể thực hiện việc đặt vé trực tuyến ngay trên Web mà không cần phải đến quầy vé hay gọi điện thoại. Họ có thể lựa chọn ghế ngồi, thời gian đi và các thông tin cá nhân cần thiết để hoàn tất quá trình đặt vé.

Bằng cách sử dụng web đặt vé xe có tích hợp AI, người dùng sẽ tiết kiệm được thời gian và công sức trong việc tìm kiếm và đặt vé xe. Họ có thể dễ dàng quản lý hành trình đi xe của mình và đảm bảo rằng việc đi lại của họ diễn ra một cách thuận tiện và suôn sẻ.

Qua dự án này, nhóm chúng tôi hy vọng tạo ra một web đặt vé xe hiệu quả và hữu ích, góp phần nâng cao trải nghiệm của người dùng trong việc đi lại bằng xe và giúp thúc đẩy sự phát triển của ngành vận tải.

BUILDING AN AI-INTEGRATED BUS TICKET BOOKING WEBSITE

ABSTRACT

This report focuses on developing an AI-integrated bus ticket booking, in order to create a convenient and fast tool for users to search, compare and book bus tickets from point A to point B. Web AI-integrated bus ticket booking are designed to provide users with the following key features:

1. Search and book tickets: This Web will provide users with a smart search engine, allowing them to search for rides between different locations. Users can search by departure, destination, time and vehicle type such as bus, train, coach or taxi.
2. Compare fares: The Web will show users options for fares from different carriers. Users can compare fares, travel times, carrier ratings, and add-ons to choose the option that suits their needs and budget.
3. Online ticket booking: Users can make online ticket booking right on the Web without having to go to the ticket counter or make a phone call. They can choose their seats, travel time and personal information needed to complete the booking process.

By using AI-integrated bus ticket booking, users will save time and effort in searching and booking bus tickets. They can easily manage their ride itineraries and ensure that their commute is convenient and smooth.

Through this report, we hope to create an efficient and useful bus ticket booking website, contributing to improving the user experience in car travel and helping to promote the development of the transportation industry.

MỤC LỤC

DANH MỤC HÌNH VẼ	vii
DANH MỤC BẢNG BIỂU	viii
DANH MỤC CÁC CHỮ VIẾT TẮT.....	ix
CHƯƠNG 1. TỔNG QUAN.....	1
1.1 Giới thiệu.....	1
1.2 Lý do chọn đề tài.....	1
1.3 Đối tượng nghiên cứu.....	2
1.4 Phạm vi nghiên cứu.....	2
1.5 Mục tiêu đề tài.....	3
CHƯƠNG 2. PHÂN TÍCH YÊU CẦU	4
2.1 Đặc tả yêu cầu	4
2.2 User story	5
2.3 Công nghệ sử dụng.....	6
2.3.1 <i>MongoDB</i>	6
2.3.2 <i>ExpressJS</i>	7
2.3.3 <i>ReactJS</i>	9
2.3.4 <i>NodeJS</i>	10
2.4 Các tác nhân trong hệ thống.....	12
2.5 Các usecase trong hệ thống	12
CHƯƠNG 3. THIẾT KẾ YÊU CẦU.....	14
3.1 Sơ đồ usecase	14
3.2 Đặc tả usecase	15

3.3 Sơ đồ BPMN	33
3.4 Mô hình ERD	35
3.5 Các thiết kế wireframe	35
CHƯƠNG 4. KẾT LUẬN.....	37
4.1 Kết quả đạt được	37
4.2 Hạn chế.....	39
4.3 Hướng phát triển tương lai	40
TÀI LIỆU THAM KHẢO	41

DANH MỤC HÌNH VẼ

Hình 2.3.2.1 Tạo ExpressJS	8
Hình 2.3.2.2 Xử lý định tuyến và tham số	8
Hình 2.3.2.3 Tạo và sử dụng middleware	9
Hình 2.3.2.4 Tạo và hiển thị HTML	9
Hình 3.1.1 Sơ đồ usecase tổng quát	14
Hình 3.3.1 Quy trình đặt vé.....	33
Hình 3.3.2 Quy trình tra cứu	33
Hình 3.3.3 Quy trình thanh toán.....	33
Hình 3.3.4 Quy trình chatbotAI	34
Hình 3.3.5 Quy trình đánh giá.....	34
Hình 3.4.1 Mô hình thực thể ERD	35
Hình 3.5.1 Wireframe chatbotAI	35
Hình 3.5.2 Wireframe đặt vé.....	36
Hình 3.5.3 Wireframe lịch sử vé.....	36
Hình 3.5.4 Wireframe đánh giá.....	36
Hình 4.1.1 Trang chủ web đặt vé	37
Hình 4.1.2 Trang danh sách vé.....	38
Hình 4.1.3 Trang đặt vé.....	38
Hình 4.1.4 Trang chọn vị trí ngồi và số lượng vé	38
Hình 4.1.5 Trang thông tin chi tiết vé thanh toán	39
Hình 4.1.6 Trang chatbotAI	39

DANH MỤC BẢNG BIỂU

Bảng 2.4.1 Tác nhân hệ thống.....	12
Bảng 2.5.1 Các usecase hệ thống	12
Bảng 3.2.1 Usecase đăng nhập.....	15
Bảng 3.2.2 Usecase đăng xuất.....	16
Bảng 3.2.3 Usecase cập nhật thông tin tài khoản.....	16
Bảng 3.2.4 Usecase quản lý thông tin người dùng	17
Bảng 3.2.5 Usecase Đặt vé (số lượng vé và chỗ ngồi)	20
Bảng 3.2.6 Usecase thanh toán	21
Bảng 3.2.7 Usecase xem lịch sử vé.....	22
Bảng 3.2.8 Usecase xem chi tiết vé	23
Bảng 3.2.9 Usecase hủy vé	25
Bảng 3.2.10 Usecase đánh giá.....	27
Bảng 3.2.11 Usecase quản lý chuyến đi.....	28
Bảng 3.2.12 Usecase xem lộ trình.....	29
Bảng 3.2.13 Usecase xem chi tiết nhà xe.....	30
Bảng 3.2.14 Usecase lọc nâng cao	31
Bảng 3.2.15 Usecase cập nhật nhà xe	32

DANH MỤC CÁC CHỮ VIẾT TẮT

AI	Application Programming Interface
API	Artificial Intelligence
ATM	Automated Teller Machine
BPMN	Business Process Modeling Notation
DOM	Document Object Model
ERD	Entity Relationship Diagram
HTTP	Hypertext Transfer Protocol
I/O	Input/Output
IOT	Internet Of Things
JS	Javascript
JSX	Javascript XML
KH	Khách hàng
ML	Machine Learning
MVC	Model View Controller
OTP	One Time Password
URL	Uniform Resource Locator

CHƯƠNG 1. TỔNG QUAN

1.1 Giới thiệu

Web đặt vé xe là một web giúp cho người dùng có thể đặt vé xe trực tuyến một cách nhanh chóng và thuận tiện. Người dùng có thể sử dụng Web để đặt vé xe bus, xe khách hoặc xe du lịch, đồng thời có thể thanh toán vé bằng các phương thức thanh toán trực tuyến như thẻ tín dụng, thẻ ATM hoặc ví điện tử.

Web đặt vé xe thường cung cấp cho người dùng thông tin về các chuyến xe, lịch trình, giá cả và các điểm đến khác nhau. Người dùng có thể tìm kiếm các chuyến xe theo tên địa điểm hoặc theo ngày đi, thời gian khởi hành và đến nơi. Ngoài ra, Web cũng cho phép người dùng đánh giá và đưa ra nhận xét về các chuyến xe, giúp cho những người sử dụng sau có thể dễ dàng chọn lựa.

Web đặt vé xe thường được sử dụng rộng rãi tại các thành phố lớn và các khu vực du lịch. Nó giúp cho người dùng tiết kiệm thời gian và công sức trong việc đặt vé, đồng thời cung cấp cho họ nhiều lựa chọn về các chuyến xe và điểm đến.

1.2 Lý do chọn đề tài

Sự phát triển của ngành vận tải: Ngành vận tải đang phát triển mạnh mẽ, đặc biệt là trong lĩnh vực vận chuyển hành khách. Việc tạo ra một web đặt vé xe sẽ giúp đáp ứng nhu cầu ngày càng tăng về đi lại và giúp thúc đẩy sự phát triển ngành này.

Sự gia tăng của web và công nghệ: Việc phát triển một Web đặt vé xe sẽ tận dụng lợi thế và cung cấp cho người dùng một công cụ thuận tiện để đặt vé và quản lý hành trình đi xe.

Tiện ích và tiết kiệm thời gian: Việc tìm kiếm, so sánh và đặt vé xe có thể trở nên rất mất thời gian và phiền phức. Một web đặt vé xe sẽ giúp giải quyết vấn đề này bằng cách cung cấp cho người dùng một giao diện thân thiện, nhanh chóng và dễ sử dụng, giúp tiết kiệm thời gian và nỗ lực trong việc đặt vé.

Tiềm năng kinh doanh: Đề tài này có tiềm năng kinh doanh lớn. Việc phát triển một web đặt vé xe có thể mang lại lợi nhuận từ việc tính phí dịch vụ đặt vé, hợp tác với các nhà vận chuyển và quảng cáo. Do đó, nó có thể thu hút sự quan tâm của các doanh nghiệp và nhà đầu tư.

Khả năng ứng dụng vào thực tế: Đề tài này có thể dễ dàng được áp dụng và thực hiện. Công nghệ phát triển web đã trở nên phổ biến và tiếp cận rộng rãi, do đó việc phát triển một web đặt vé xe có khả năng ứng dụng vào thực tế và đáp ứng nhu cầu của người dùng.

1.3 Đối tượng nghiên cứu

Người dùng cá nhân: Bao gồm những người có nhu cầu đi lại và đặt vé xe từ điểm A đến điểm B. Đối tượng này có thể bao gồm học sinh, sinh viên, công nhân, du khách, người đi công việc, và những người có nhu cầu di chuyển.

Doanh nghiệp vận tải: Các doanh nghiệp vận tải, bao gồm các công ty xe buýt, tàu hỏa, xe khách, taxi, và các dịch vụ xe hơi đặc biệt, có thể là đối tượng nghiên cứu để đánh giá khả năng hợp tác và tích hợp Web vào hệ thống quản lý của họ.

Nhà cung cấp dịch vụ vận chuyển: Các nhà cung cấp dịch vụ vận chuyển có thể là đối tượng nghiên cứu để tìm hiểu về quy trình đặt vé, tính năng cần thiết và quản lý thông tin liên quan đến việc kết nối và cung cấp dịch vụ thông qua Web.

Nhà phát triển ứng dụng: Đối tượng nghiên cứu cũng có thể bao gồm những nhà phát triển Web, những người phụ trách việc thiết kế, phát triển và duy trì web đặt vé xe. Việc tìm hiểu về yêu cầu kỹ thuật, quy trình phát triển và thử nghiệm có thể rất hữu ích trong quá trình nghiên cứu.

1.4 Phạm vi nghiên cứu

Thiết kế và phát triển web: Nghiên cứu có thể tập trung vào quy trình thiết kế và phát triển Web đặt vé xe. Điều này bao gồm việc xác định yêu cầu chức năng và giao diện người dùng, quy trình phát triển và triển khai web.

Tìm kiếm và đặt vé: Nghiên cứu có thể nghiên cứu về quy trình tìm kiếm và đặt vé trong web. Điều này bao gồm việc phát triển thuật toán tìm kiếm thông minh, tích hợp hệ thống đặt vé, và quy trình thanh toán an toàn và tiện lợi.

Quản lý hành trình đi xe: Nghiên cứu có thể tập trung vào quy trình quản lý hành trình đi xe, bao gồm cung cấp thông tin về lịch trình, thay đổi lịch trình, thông báo về tình trạng giao thông, và cung cấp tùy chọn điều chỉnh và hủy vé.

Tích hợp với nhà vận chuyển và hệ thống quản lý: Nghiên cứu có thể tập trung vào quá trình tích hợp web với các nhà vận chuyển và hệ thống quản lý hiện có. Điều này bao gồm việc xây dựng giao diện lập trình web (API) để kết nối với dữ liệu và dịch vụ của nhà vận chuyển, và tích hợp thông tin về giá vé, lịch trình và khả năng đặt chỗ.

Đánh giá hiệu suất và trải nghiệm người dùng: Nghiên cứu có thể đánh giá hiệu suất và trải nghiệm người dùng của web đặt vé xe. Điều này bao gồm việc thu thập phản hồi từ người dùng, đo lường thời gian tìm kiếm và đặt vé, và đánh giá sự hài lòng và sự tiện lợi của web.

1.5 Mục tiêu đề tài

Mục tiêu chính của đề tài web đặt vé xe là cung cấp cho khách hàng một cách thức thuận tiện và nhanh chóng để đặt vé xe. Website sẽ cung cấp cho khách hàng thông tin về các chuyến xe, giá vé, thời gian khởi hành và thời gian đến của các chuyến xe. Khách hàng có thể dễ dàng tìm kiếm chuyến xe phù hợp với nhu cầu của mình và đặt vé trực tuyến. Website cũng sẽ cung cấp cho khách hàng các thông tin về các nhà xe, các loại xe và các dịch vụ đi kèm.

Để đạt các mục tiêu trên, hệ thống web đặt vé xe cần có các chức năng sau:

- Tìm kiếm chuyến xe: Khách hàng có thể tìm kiếm chuyến xe theo các tiêu chí như: điểm đi, điểm đến, thời gian khởi hành,...
- Đặt vé: Khách hàng có thể đặt vé xe trực tuyến bằng cách cung cấp thông tin cá nhân, thông tin chuyến xe,...
- Thanh toán: Khách hàng có thể thanh toán vé xe trực tuyến bằng nhiều hình thức khác nhau.
- Quản lý thông tin đặt vé: Khách hàng có thể quản lý các thông tin đặt vé của mình như: lịch sử đặt vé, vé đã đặt,...

CHƯƠNG 2. PHÂN TÍCH YÊU CẦU

2.1 Đặc tả yêu cầu

Hệ thống web đặt vé xe được xây dựng với các yêu cầu sau:

- Xây dựng hệ thống web đặt vé cần có những chức năng:
 - Đăng ký tài khoản: cho phép người dùng đăng ký và tạo tài khoản để sử dụng các dịch vụ của web.
 - Đặt vé: cho phép người dùng chọn tuyến đường, thời gian và số lượng ghế cần đặt để có thể đặt vé.
 - Thanh toán: cho phép người dùng thanh toán trực tuyến bằng các phương thức thanh toán khác nhau như thẻ tín dụng, ví điện tử hay tiền mặt khi đến bến xe.
 - Quản lý thông tin: cho phép người dùng quản lý thông tin cá nhân, lịch sử đặt vé và các thông tin liên quan đến tài khoản của mình.
 - Tìm kiếm: cho phép người dùng tìm kiếm thông tin về các tuyến đường, thời gian khởi hành, giá vé và các thông tin liên quan khác.
 - Thông báo: thông báo cho người dùng về các thông tin liên quan đến việc đặt vé như thông tin chuyến xe, thời gian khởi hành, địa điểm lên xe, trạng thái đặt vé và các thông tin khác liên quan đến chuyến đi.
 - Đánh giá và đề xuất: cho phép người dùng đánh giá và đề xuất về chất lượng dịch vụ, tuyến đường, giá cả để cải thiện chất lượng dịch vụ và đáp ứng nhu cầu của người dùng.
- Các ràng buộc về thực thi và thiết kế của web đặt vé xe:
 - Giao diện chương trình được thiết kế đúng theo yêu cầu, đầy đủ chức năng, dễ nhìn, dễ dùng, dễ nhớ, thân thiện với người dùng.
 - Đảm bảo tốc độ xử lý nhanh, kịp thời.
 - Hệ thống hoạt động ổn định 24/24.
 - Đáp ứng số lượng lớn lượt truy cập cùng lúc.
 - Đảm bảo việc bảo mật các thông tin về người dùng.

2.2 User story

Như một người dùng, KH muốn sử dụng một trang web đặt vé xe trực tuyến tích hợp trí tuệ nhân tạo AI để có trải nghiệm đặt vé nhanh chóng, thuận tiện và thông minh hơn. Người dùng muốn có một trang web đặt vé xe thông minh với các tính năng AI sau:

1. Tìm kiếm và so sánh: KH muốn trang web cho phép tìm kiếm vé xe từ nhiều nhà xe khác nhau và so sánh giá cả, thời gian di chuyển, phương tiện vận chuyển, và các tiện ích khác để KH có thể chọn lựa dễ dàng nhất.
2. Gợi ý thông minh: Khi KH nhập địa điểm xuất phát và đích, KH muốn trang web gợi ý các lộ trình tối ưu dựa trên dữ liệu thời gian thực và thông tin lưu lượng giao thông. Ví dụ: nếu có sự cố giao thông trên một tuyến đường, trang web sẽ đề xuất các tuyến điều hướng khác để giúp tránh kẹt xe.
3. Đề xuất điểm đến: Khi KH không biết nơi tốt để đi trong khu vực, KH muốn trang web đề xuất các điểm đến phổ biến và những nơi thú vị dựa trên sở thích cá nhân hoặc phản hồi từ người dùng khác.
4. Tích hợp thanh toán và xác nhận nhanh chóng: KH muốn trang web cho phép thanh toán trực tuyến an toàn và nhanh chóng thông qua các phương thức thanh toán phổ biến. Sau khi thanh toán, KH muốn nhận được xác nhận đặt vé trong thời gian ngắn và được cung cấp thông tin chi tiết về vé và lịch trình của KH.
5. Hỗ trợ khách hàng 24/7: Nếu KH gặp vấn đề hoặc cần hỗ trợ trong quá trình đặt vé, KH muốn có khả năng liên hệ với đội ngũ hỗ trợ khách hàng thông qua trang web hoặc qua các kênh liên lạc khác.
6. Ghi nhớ thông tin cá nhân: Để tiết kiệm thời gian cho các đặt vé sau này, KH muốn trang web lưu trữ thông tin cá nhân và thông tin thanh toán một cách an toàn và bảo mật.
7. Phản hồi sau chuyến đi: KH muốn có khả năng đánh giá và viết nhận xét về chuyến đi của mình để chia sẻ trải nghiệm với người dùng khác.

2.3 Công nghệ sử dụng

MERN Stack là một stack JavaScript rút gọn từ MongoDB, ExpressJS, ReactJS và NodeJS. Nó được thiết kế để giúp phát triển ứng dụng web toàn ngăn xếp dễ dàng hơn và nhanh hơn.

2.3.1 MongoDB

MongoDB là một loại database hướng tài liệu NoSQL. Việc sử dụng một cấu trúc JSON rất linh hoạt thay vì sử dụng một cấu trúc dựa trên table để thích ứng với tài liệu như JSON. MongoDB lưu trữ dữ liệu dưới dạng Document JSON, điều này có nghĩa là mỗi tập hợp có kích thước và số lượng document khác nhau. Truy vấn sẽ diễn ra nhanh chóng khi dữ liệu được lưu trữ trong một document kiểu JSON.

Một số tính năng nổi bật của MongoDB:

- Cấu trúc dữ liệu linh hoạt: MongoDB cho phép lưu trữ dữ liệu dưới dạng tài liệu JSON, có thể được cấu trúc một cách linh hoạt mà không phụ thuộc vào một quy trình cụ thể. Điều này giúp MongoDB thích ứng dễ dàng với nhiều loại dữ liệu.
- Hiệu suất cao: MongoDB có tốc độ đọc và ghi nhanh. MongoDB có cấu trúc dữ liệu phân tán, cho phép dữ liệu được lưu trữ trên nhiều máy chủ.
- Giao diện lập trình ứng dụng (API) của MongoDB dễ sử dụng và hỗ trợ nhiều ngôn ngữ lập trình.
- Mở rộng dễ dàng: Bằng cách thêm các máy chủ mới vào cụm, MongoDB có thể dễ dàng mở rộng theo chiều ngang. Điều này giúp đáp ứng nhu cầu ngày càng tăng về lưu trữ và truy vấn dữ liệu.

MongoDB được sử dụng trong nhiều ứng dụng khác nhau, bao gồm:

- Website và ứng dụng web: Dữ liệu người dùng, dữ liệu sản phẩm và dữ liệu giao dịch đều được lưu trữ trong MongoDB.
- Ứng dụng di động: Dữ liệu người dùng, dữ liệu lịch sử hoạt động và dữ liệu vị trí được lưu trữ trong MongoDB.
- Ứng dụng dữ liệu lớn: MongoDB được sử dụng để phân tích và lưu trữ lượng dữ liệu lớn.

- Ứng dụng AI và ML: Các mô hình AI và ML sử dụng MongoDB để lưu trữ dữ liệu huấn luyện.

Một số ví dụ cụ thể về cách MongoDB đang được sử dụng:

- Facebook: Dữ liệu người dùng, chẳng hạn như tên, địa chỉ email, ảnh hồ sơ và bạn bè, được lưu trữ bằng cách sử dụng MongoDB.
- Twitter: Dữ liệu của các tweet bao gồm văn bản, người dùng và thời gian được lưu trữ bởi MongoDB.
- Netflix sử dụng MongoDB để lưu trữ dữ liệu về phim và chương trình truyền hình, bao gồm tên, thể loại, diễn viên và đạo diễn.

2.3.2 ExpressJS

ExpressJS là framework hỗ trợ xây dựng ứng dụng web dựa trên Node.js. ExpressJS hỗ trợ việc phát triển các ứng dụng web và API bằng cách cung cấp các công cụ và tích hợp các tính năng cần thiết để xử lý các yêu cầu HTTP, quản lý định tuyến (routing) và thực hiện các nhiệm vụ phức tạp khác trong quá trình phát triển ứng dụng web.

Một số khía cạnh quan trọng của ExpressJS:

- Routing: ExpressJS cho phép xác định các đường dẫn mà ứng dụng sẽ xử lý. Các phương thức HTTP như GET, POST, PUT và DELETE cho phép gắn các xử lý phù hợp cho mỗi tuyến đường.
- Middleware: là các hàm được thực thi trước khi được xử lý route yêu cầu, có thể hoàn thành các nhiệm vụ như xác thực, ghi nhật ký, xử lý dữ liệu đầu vào và nhiều nhiệm vụ khác bằng cách sử dụng điều này trước khi yêu cầu đến tới đường đi chính.
- Xử lý yêu cầu và phản hồi: ExpressJS cung cấp các đối tượng request và response để có thể truy cập thông tin về yêu cầu từ phía client và tạo phản hồi tương ứng. Bạn có thể trả về dữ liệu HTML, JSON, hoặc bất kỳ loại dữ liệu nào khác thông qua các phương thức của đối tượng response.
- View Engine: ExpressJS không cần view engine cụ thể, nhưng nó hỗ trợ tích hợp với nhiều view engine phổ biến như EJS, Pug (trước đây là Jade) và

Handlebars. Bằng cách kết hợp dữ liệu động vào các template, View engine giúp tạo và hiển thị các trang HTML dễ dàng.

- Phân cấp ứng dụng: ExpressJS không ép buộc ứng dụng theo một cách nhất định. Để giữ cho mã nguồn dễ quản lý và mở rộng, có thể tổ chức ứng dụng thành các module, quản lý và middleware.
- Các extension và middleware bên thứ ba: Do một cộng đồng rộng lớn và phong phú, ExpressJS cung cấp nhiều extension và middleware bên thứ ba để có thể thêm các chức năng như xác thực, quản lý phiên và bảo mật vào ứng dụng của mình.
- Hỗ trợ API RESTful: ExpressJS cung cấp các công cụ và thư viện để tạo API RESTful.
- Hỗ trợ mô hình MVC: Với ExpressJS, có thể tạo các ứng dụng web theo mô hình Model-View-Controller (MVC).

Ứng dụng của ExpressJS:

- Có thể tạo một ứng dụng ExpressJS cơ bản như sau:

```
const express = require('express');
const app = express();
const port = 3000;

app.get('/', (req, res) => {
  res.send('Chào mừng đến với ứng dụng Express.js cơ bản!');
});

app.listen(port, () => {
  console.log('Ứng dụng Express.js đang lắng nghe tại http://localhost:${port}');
});
```

Hình 2.3.2.1 Tạo ExpressJS

- Xử lý định tuyến và tham số:

Các biểu thức chính quy và tham số động có thể được sử dụng trong ExpressJS. Điều này hỗ trợ xử lý các yêu cầu có nhiều đường dẫn và trích xuất dữ liệu từ URL. Ví dụ:

```
app.get('/users/:id', (req, res) => {
  const userID = req.params.id;
  res.send('User ID: ${userID}');
});
```

Hình 2.3.2.2 Xử lý định tuyến và tham số

- Middleware:

Các middleware có thể được tạo và sử dụng bởi ExpressJS để thực hiện các tác vụ trung gian trước khi cần xử lý route chính. Xác thực, kiểm tra dữ liệu đầu vào, ghi nhật ký và nhiều nhiệm vụ khác là một ví dụ về điều này. Ví dụ:

```
function myMiddleware(req, res, next) {
  // Thực hiện xử lý trước khi điều khiển được chuyển đến tuyến đường chính
  console.log('Middleware được gọi');
  next(); // Chuyển điều khiển đến middleware hoặc tuyến đường tiếp theo
}
```

Hình 2.3.2.3 Tạo và sử dụng middleware

- Template và View Engine:

Để tạo và hiển thị trang HTML một cách đơn giản, ExpressJS cho phép tích hợp với các view engine. Người dùng có thể sử dụng các template để truyền dữ liệu động từ server vào các trang để hiển thị thông tin tương ứng. Ví dụ về việc sử dụng ExpressJS:

```
app.set('view engine', 'ejs');

app.get('/username/:name', (req, res) => {
  const userName = req.params.name;
  res.render('user', {name: userName});
});
```

Hình 2.3.2.4 Tạo và hiển thị HTML

2.3.3 ReactJS

ReactJS là một thư viện JavaScript mã nguồn mở chủ yếu được sử dụng để tạo giao diện người dùng (UI) cho ứng dụng web. Cộng đồng nguồn mở và Facebook đã hỗ trợ nó phát triển, khiến nó nhanh chóng trở thành một trong những công cụ phổ biến nhất để tạo giao diện người dùng động và tương tác trên web. ReactJS sử dụng các thành phần để quản lý trạng thái và tái sử dụng code của ứng dụng.

Một số khái niệm quan trọng trong ReactJS:

- Components (Thành phần): Các thành phần xây dựng độc lập này cho phép bạn xây dựng giao diện. Có thể sử dụng hàm hoặc lớp để tạo thành phần.

Chúng có thể nhận được dữ liệu đầu vào được gọi là "props" và cũng có thể nhận được trạng thái được gọi là "state".

- Props (thuộc tính): Là các dữ liệu đầu vào khi cung cấp cho một thành phần. Truyền thông tin từ thành phần cha sang thành phần con thông qua các giá trị không thay đổi này.
- State (Trạng thái): Là các dữ liệu có thể thay đổi trong một thành phần. Khi state thay đổi, ReactJS sẽ tự động render lại giao diện để phản ánh trạng thái mới.
- Render: Là quá trình biến đổi dữ liệu trong ReactJS thành giao diện người dùng. Khi dữ liệu hoặc trạng thái thay đổi, ReactJS sẽ tự động render lại các thành phần liên quan mà không cần tương tác trực tiếp.
- Virtual DOM: Là một biểu diễn ảo của DOM thật. React sử dụng Virtual DOM để giảm thiểu số lần truy cập trực tiếp vào DOM, giúp cải thiện hiệu suất và tăng tốc độ render.
- JSX: Là một phần cú pháp trong ReactJS cho phép viết giao diện người dùng bằng cách kết hợp JavaScript và XML. JSX giúp tạo ra mã nguồn dễ đọc và dễ hiểu hơn.

2.3.4 NodeJS

Node.js là một môi trường chạy JavaScript ngoài trình duyệt dành cho việc phát triển các ứng dụng có thể được mở rộng. Mô hình kiến trúc hướng sự kiện của Node.js cho phép nó xử lý nhiều yêu cầu đồng thời một cách hiệu quả.

Lợi ích của việc sử dụng NodeJS:

- Khả năng xử lý bất đồng bộ, hoặc không đồng bộ: Node.js được phát triển để xử lý các tác vụ bất đồng bộ. Điều này cho phép nó xử lý nhiều công việc một lúc mà không cần phải chờ đợi các công việc khác được hoàn thành.
- Sự kiện và I/O không bị chặn: Node.js xử lý các hoạt động I/O không đồng bộ bằng cách sử dụng mô hình sự kiện. Thay vì chờ đợi, Node.js sẽ thông báo qua các sự kiện khi một tác vụ I/O hoàn thành.

- Phát triển các ứng dụng trong thời gian thực: Node.js thường được sử dụng để phát triển các ứng dụng thời gian thực như ứng dụng chat, trò chơi trực tuyến và các ứng dụng theo dõi sự kiện do khả năng xử lý bất đồng bộ và giao tiếp sự kiện.
- Xử lý dữ liệu và tệp: Node.js cho phép thực hiện nhiều tác vụ liên quan đến dữ liệu, chẳng hạn như đọc và ghi tệp và xử lý dữ liệu dạng JSON.
- Mô hình xử lý sự kiện: Node.js sử dụng mô hình xử lý sự kiện để xử lý các yêu cầu và sự kiện, giúp tối ưu hóa việc quản lý luồng và tránh việc tạo ra nhiều luồng.
- Chia sẻ mã giữa máy chủ và khách hàng: Sử dụng JavaScript cả trên phía máy chủ và phía khách hàng (browser) cho phép việc chia sẻ mã giữa các phần của ứng dụng, giúp tiết kiệm thời gian và nguồn lực.
- Phát triển đa nền tảng: Node.js có thể chạy trên nhiều nền tảng, bao gồm Windows, macOS và Linux.
- Dễ dàng triển khai: Môi trường phát triển và triển khai ứng dụng Node.js thường khá dễ dàng thiết lập và quản lý.
- Hiệu suất và Tốc độ: Node.js sử dụng kiến thức JavaScript và sử dụng mã máy được biên dịch từ V8 Engine, làm cho nó nhanh và hiệu quả. Điều này đặc biệt hữu ích cho các ứng dụng thời gian thực hoặc ứng dụng yêu cầu xử lý đồng thời (concurrent processing).
- Hỗ trợ cho WebSockets: Node.js cung cấp hỗ trợ tốt cho WebSockets, cho phép phát triển ứng dụng thời gian thực như chat trực tiếp và trò chơi trực tuyến.
- Khả năng mở rộng tốt: Node.js dễ dàng mở rộng và phù hợp với các ứng dụng cần mở rộng ngang hoặc dọc.
- Sản phẩm MVP (Minimum Viable Product) Nhanh hơn: Node.js cho phép các nhà phát triển nhanh chóng xây dựng các sản phẩm MVP hoặc các ứng dụng thử nghiệm một cách nhanh chóng và dễ dàng.

2.4 Các tác nhân trong hệ thống

Bảng 2.4.1 Tác nhân hệ thống

STT	Tác nhân	Mô tả
1	Admin	Là người toàn quyền quản lý và điều hành web đặt vé xe, chịu trách nhiệm quản lý thông tin của khách hàng, nhà cung cấp dịch vụ, thanh toán, quản lý các chính sách hoàn tiền, phát triển tính năng mới và cung cấp hỗ trợ khách hàng.
2	Khách hàng đặt vé	Là người sử dụng web đặt vé xe để tìm kiếm, đặt và thanh toán vé xe.
3	Nhà xe	Là những công ty vận tải cung cấp dịch vụ đưa đón khách hàng từ điểm A đến điểm B.

2.5 Các usecase trong hệ thống

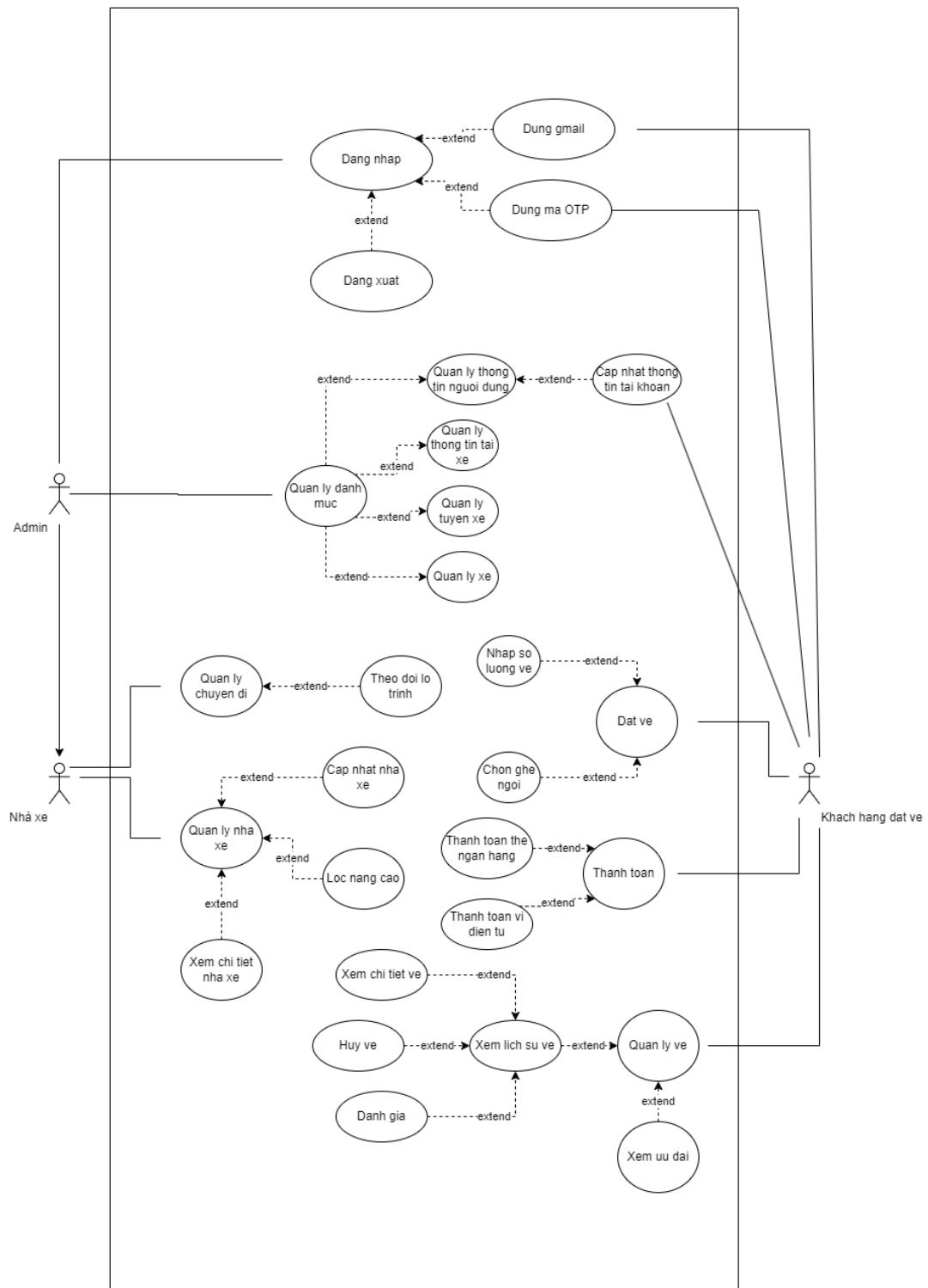
Bảng 2.5.1 Các usecase hệ thống

ID	Tên Use Case	Mô tả
UC01	Đăng nhập	Các tác nhân thực hiện việc đăng nhập để hệ thống cấp quyền hạn của tác nhân đó.
UC02	Đăng xuất	Chỉ các tác nhân đã thực hiện việc đăng nhập mới có thể đăng xuất ra khỏi hệ thống.
UC03	Dùng gmail	Khách hàng sử dụng tài khoản gmail để đăng nhập.
UC04	Dùng mã OTP	Khách hàng sử dụng số điện thoại để đăng nhập.
UC05	Cập nhật thông tin tài khoản	Khách hàng thực hiện chỉnh sửa thông tin cá nhân.
UC06	Quản lý thông tin người dùng	Admin thực hiện việc thêm, sửa và xóa người dùng.
UC07	Quản lý thông tin tài xế	Admin thực hiện việc thêm, sửa và xóa tài xế.
UC08	Quản lý tuyến xe	Admin thực hiện việc thêm, sửa và xóa tuyến xe.

ID	Tên Use Case	Mô tả
UC09	Quản lý xe	Admin thực hiện việc thêm, sửa và xóa xe.
UC10	Chọn số lượng vé	Khách hàng chọn số lượng vé phù hợp số lượng người đi cho chuyến xe của mình.
UC11	Chọn ghế ngồi	Khách hàng chọn các vị trí ghế ngồi.
UC12	Thanh toán	Khách hàng thực hiện việc thanh toán bằng thẻ ngân hàng hoặc ví điện tử.
UC13	Xem lịch sử vé	Khách hàng xem lịch sử vé: vé đã hoàn tất chuyến đi, vé vừa đặt, vé chưa thanh toán, vé hủy...
UC14	Xem chi tiết vé	Khách hàng xem chi tiết thông tin các vé đã đặt.
UC15	Hủy vé	Khách hàng muốn hủy vé đã đặt.
UC16	Đánh giá	Khách hàng đánh giá chuyến xe đã đi.
UC15	Quản lý chuyến đi	Nó cung cấp các tính năng cho nhà xe để tìm kiếm, xem thông tin chi tiết, đặt vé và quản lý các chuyến đi của khách hàng.
UC16	Xem lộ trình	Nhà xe xem vị trí của các chuyến xe để biết lộ trình đi có đúng chưa và thời gian đi có vượt quá thời gian quy định của chuyến đi đó không.
UC17	Xem chi tiết nhà xe	Quản lý các thông tin chi tiết của các nhà xe.
UC18	Lọc nâng cao	Lọc các thông tin cần thiết để dễ dàng tìm kiếm chính xác nhà xe.
UC19	Cập nhật nhà xe	Thêm, sửa, xóa các nhà xe

CHƯƠNG 3. THIẾT KẾ YÊU CẦU

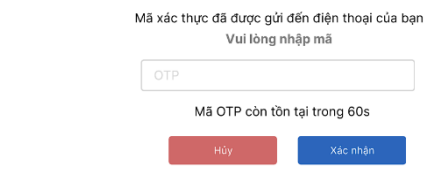
3.1 Sơ đồ usecase




Hình 3.1.1 Sơ đồ usecase tổng quát

3.2 Đặc tả usecase



Bảng 3.2.1 Usecase đăng nhập

Usecase đăng nhập	
Usecase Description	Các tác nhân thực hiện việc đăng nhập để hệ thống cấp quyền hạn của tác nhân đó
Actor chính	User
Trigger	Người dùng truy cập vào hệ thống
Pre – condition	Có số điện thoại để đăng nhập với mã OTP hoặc dùng gmail để đăng nhập
Post – condition	Hệ thống ghi nhận đăng nhập tài khoản thành công
Main flows (Luồng sự kiện chính)	
User	System
1a. Người dùng điền sdt hoặc chọn đăng nhập bằng mail.	
	<p>1b. Hiện thị một popup xác thực khi đăng nhập bằng sdt</p> 
1c. Người dùng nhận mã bằng sms và nhập vào ô, nhấn nút “Xác nhận”	
	1d. Hệ thống ghi nhận và hiển thị trang chủ datvere.
Alternative flow(Luồng sự kiện thay thế): Đăng nhập thông qua mail.	
Exception flow (Luồng sự kiện ngoài lề)	
Người dùng để trống trường dữ liệu bắt buộc, hệ thống sẽ thông báo cho người dùng và bắt buộc người dùng nhập vào mới có thể đăng nhập.	


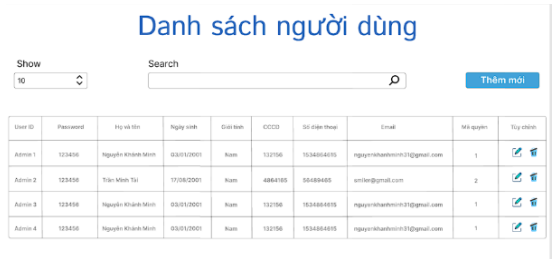
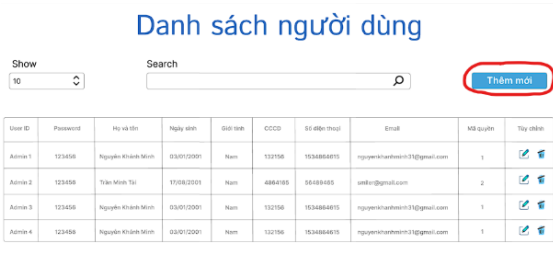
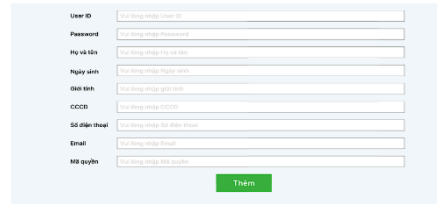
Bảng 3.2.2 Usecase đăng xuất

Usecase đăng xuất	
Usecase Description	Chỉ các tác nhân đã thực hiện việc đăng nhập mới có thể đăng xuất ra khỏi hệ thống
Actor chính	User, admin, nhà xe
Trigger	Người dùng muốn đăng xuất khỏi hệ thống
Pre – condition	Người dùng đang đăng nhập vào hệ thống
Post – condition	Người dùng đăng xuất khỏi hệ thống và chuyển sang vai trò khách.
Main flows (Luồng sự kiện chính)	
User	System
1a. Di chuột vào tên tài khoản tại trang chủ, nhấn vào biểu tượng. 	
1b. Người dùng bấm vào nút đăng xuất	
	1c. Hệ thống đăng xuất tài khoản, chuyển về trang chủ trước khi đăng nhập.
Alternative flow(Luồng sự kiện thay thế): Admin, nhà xe đăng xuất	
Exception flow (Luồng sự kiện ngoài lề)	
Không có.	





Bảng 3.2.3 Usecase cập nhật thông tin tài khoản

Usecase cập nhật thông tin tài khoản	
Usecase Description	Người dùng muốn chỉnh sửa thông tin tài khoản để phục vụ cho việc đặt xe
Actor chính	User
Trigger	User muốn cập nhật thông tin cho tài khoản
Pre – condition	Đã đăng nhập vào hệ thống
Post – condition	Thông tin tài khoản được cập nhật thành công
Main flows (Luồng sự kiện chính)	
User	System
<p>1a. Di chuột vào tên tài khoản tại trang chủ, nhấn vào biểu tượng.</p> 	
<p>1b. Người dùng bấm vào nút “Thông tin tài khoản”</p>	<p>1c. Hệ thống hiển thị trang</p> 
<p>1d. Người dùng chỉnh sửa thông tin và lưu.</p>	<p>1e. Hệ thống lưu lại</p>
Alternative flow(Luồng sự kiện thay thế): Không có	
Exception flow (Luồng sự kiện ngoài lề)	
Không có.	

Bảng 3.2.4 Usecase quản lý thông tin người dùng

Usecase quản lý thông tin người dùng	
Usecase Description	Admin thực hiện thêm, sửa, xóa người dùng.
Actor chính	Admin
Trigger	Khách hàng đã chỉnh sửa lại thông tin cá nhân Khách hàng mới sử dụng web đặt vé Khách hàng không còn sử dụng web đặt vé
Pre – condition	Đã đăng nhập vào hệ thống
Post – condition	Đã cập nhật về người dùng
Main flows (Luồng sự kiện chính)	
User	System
<p>1a. Sau khi đăng nhập chọn mục người dùng</p> 	<p>1b. Hệ thống hiển thị trang danh sách người dùng</p> 
<p>1c. Chọn mục “Thêm mới” để thêm người dùng mới</p> 	<p>1d. Hệ thống hiển thị trang thêm mới người dùng</p> 

1d. Chọn biểu tượng “Chỉnh sửa” để cập nhật lại thông tin cho người dùng

User ID	Password	Họ và tên	Ngày sinh	Giới tính	CCCD	Số điện thoại	Email	Mã quyền	Tùy chỉnh
Admin 1	123456	Nguyễn Khánh Minh	03/01/2001	Nam	132156	1534864615	nguyenkhanhminh31@gmail.com	1	
Admin 2	123456	Trần Minh Tài	17/08/2001	Nam	4864185	56489465	smiler@gmail.com	2	
Admin 3	123456	Nguyễn Khánh Minh	03/01/2001	Nam	132156	1534864615	nguyenkhanhminh31@gmail.com	1	
Admin 4	123456	Nguyễn Khánh Minh	03/01/2001	Nam	132156	1534864615	nguyenkhanhminh31@gmail.com	1	





1e. Hệ thống hiển thị trang chỉnh sửa người dùng

Sửa người dùng

User ID	Admin 1
Password	0231231
Họ và tên	Nguyễn Khánh Minh
Ngày sinh	03/01/2001
Giới tính	Nam
CCCD	15645616
Số điện thoại	5645456546
Email	nguyenkhanhminh31@gmail.com
Mã quyền	1

Lưu lại
Hủy

1f. Chọn biểu tượng “Xóa” để xóa người dùng

User ID	Password	Họ và tên	Ngày sinh	Giới tính	CCCD	Số điện thoại	Email	Mã quyền	Tùy chỉnh
Admin 1	123456	Nguyễn Khánh Minh	03/01/2001	Nam	132156	1534864615	nguyenkhanhminh31@gmail.com	1	
Admin 2	123456	Trần Minh Tài	17/08/2001	Nam	4864185	56489465	smiler@gmail.com	2	
Admin 3	123456	Nguyễn Khánh Minh	03/01/2001	Nam	132156	1534864615	nguyenkhanhminh31@gmail.com	1	
Admin 4	123456	Nguyễn Khánh Minh	03/01/2001	Nam	132156	1534864615	nguyenkhanhminh31@gmail.com	1	

1g. Hệ thống hiển thị thông báo nhắc

Bạn có chắc chắn xóa?


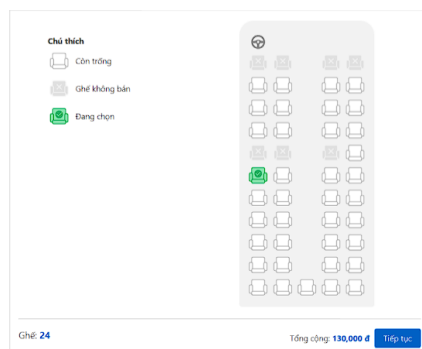
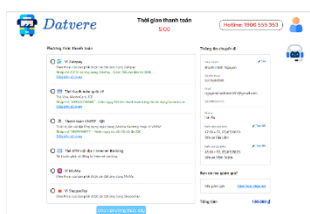
Xóa

Alternative flow(Luồng sự kiện thay thế): Không có

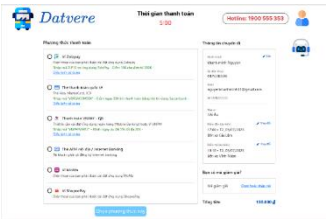
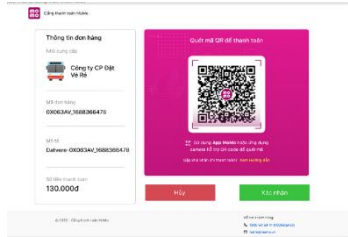
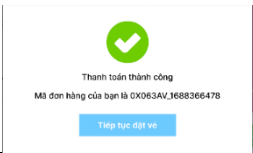
Exception flow (Luồng sự kiện ngoài lề):

Admin nhập sai các trường dữ liệu.


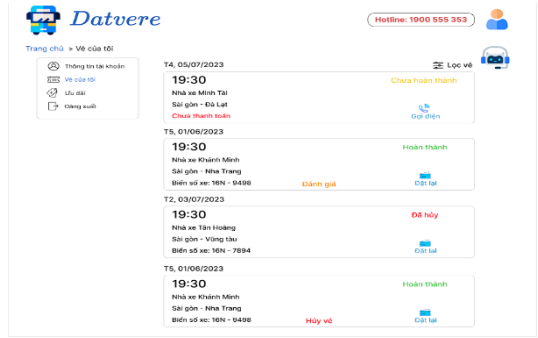
Bảng 3.2.5 Usecase Đặt vé (số lượng vé và chỗ ngồi)

Usecase Đặt vé (số lượng vé và chỗ ngồi)	
Usecase Description	Khách hàng chọn số lượng vé phù hợp với lượng người đi và vị trí ngồi
Actor chính	Khách hàng
Trigger	Khách hàng đặt vé
Pre – condition	Khách hàng chọn chuyến muốn đi
Post – condition	Khách đã chọn số lượng vé và vị trí ngồi
Main flows (Luồng sự kiện chính)	
User	System
<p>1a. Khách hàng chọn mục “Chọn chuyến”</p> 	<p>1b. Hệ thống hiển thị trang chọn vị trí ghế số lượng vé</p> 
<p>1c. Khách hàng nhấn “Tiếp tục”</p>	<p>1d. Hệ thống hiển thị trang thông tin thanh toán</p> 
Alternative flow(Luồng sự kiện thay thế): Không có	
Exception flow (Luồng sự kiện ngoài lề)	
Không có.	


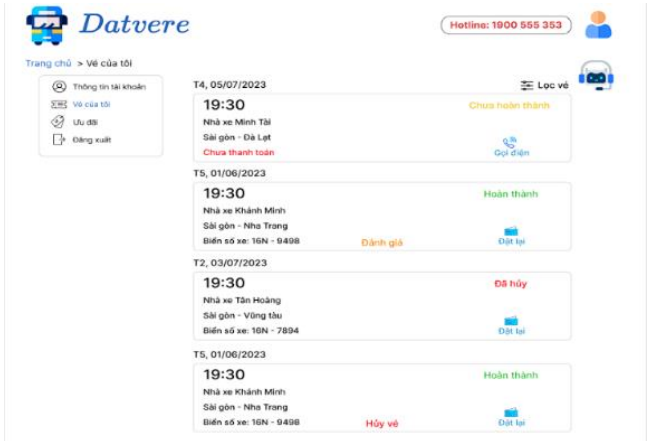
Bảng 3.2.6 Usecase thanh toán

Usecase thanh toán	
Usecase Description	Khách hàng thanh toán cho vé đã đặt
Actor chính	Khách hàng
Trigger	Khách hàng đặt vé
Pre – condition	Khách hàng thanh toán vé
Post – condition	Khách đã hoàn tất thanh toán
Main flows (Luồng sự kiện chính)	
User	System
<p>1a. Khách hàng chọn phương thức để thanh toán và ấn nút “Chọn phương thức này” để tiến hành thanh toán</p> 	<p>1b. Hệ thống hiển thị trang thanh toán</p> 
<p>1c. Sau khi quét mã thanh toán và đã thanh toán xong chọn nút “Xác nhận”</p> 	
Alternative flow(Luồng sự kiện thay thế): Khách hàng chọn nút “Hủy”	
Exception flow (Luồng sự kiện ngoài lề)	
Không có	

Bảng 3.2.7 Usecase xem lịch sử vé

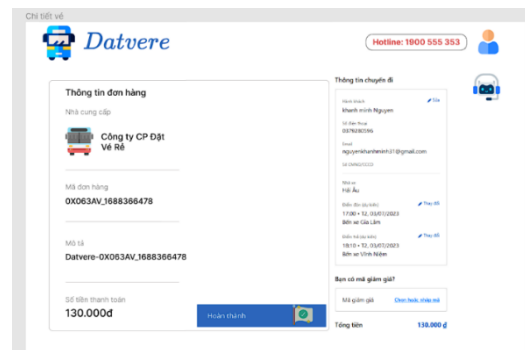
Usecase xem lịch sử vé	
Usecase Description	Khách hàng xem lịch sử vé: vé đã hoàn tất chuyến đi, vé vừa đặt, vé chưa thanh toán, vé hủy,...
Actor chính	Khách hàng
Trigger	Khách hàng vào xem lịch sử vé
Pre – condition	Khách hàng hoàn tất đặt vé, hoàn thành chuyến đi, vé hủy, và vé chưa thanh toán
Post – condition	Khách hàng xem lịch sử vé
Main flows (Luồng sự kiện chính)	
User	System
<p>1a. Khách hàng di chuột vào tài khoản tại trang chủ, nhấn vào biểu tượng</p> 	
<p>1b. Khách hàng nhấp vào mục “Vé của tôi”.</p>	<p>1c. Hệ thống hiển thị trang lịch sử vé</p> 
Alternative flow(Luồng sự kiện thay thế): Không	
Exception flow (Luồng sự kiện ngoài lề)	
Không có	

Bảng 3.2.8 Usecase xem chi tiết vé

Usecase xem chi tiết vé	
Usecase Decription	Khách hàng xem chi tiết thông tin các vé đã đặt.
Actor chính	Khách hàng
Trigger	Khách hàng vào xem lịch sử vé
Pre – condition	Khách hàng hoàn tất đặt vé, hoàn thành chuyến đi, vé hủy, và vé chưa thanh toán
Post – condition	Khách hàng xem chi tiết vé
Main flows (Luồng sự kiện chính)	
User	System
<p>1a. Khách hàng di chuột vào tài khoản tại trang chủ, nhấn vào biểu tượng</p> 	
<p>1b. Khách hàng nhấp vào mục “Vé của tôi”</p>	<p>1c. Hệ thống hiển thị trang lịch sử vé</p> 

1d. Khách hàng chọn 1 vé bất kỳ để xem chi tiết

1e. Hệ thống hiển thị chi tiết vé


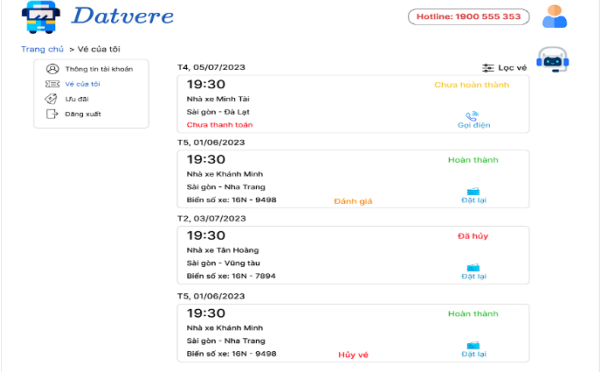


Alternative flow(Luồng sự kiện thay thế): Không

Exception flow (Luồng sự kiện ngoài lề)

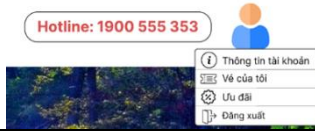
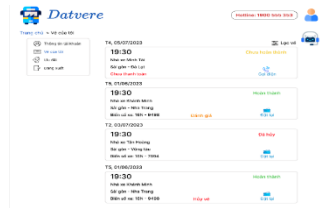
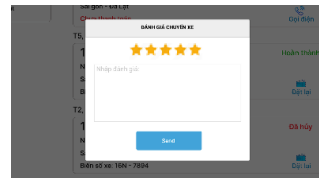
Không có

Bảng 3.2.9 Usecase hủy vé


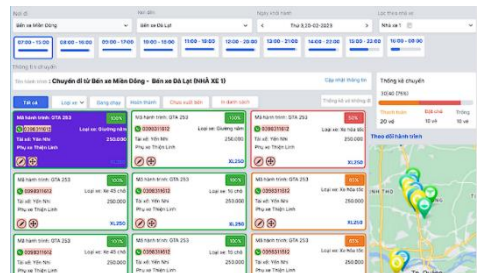
Usecase hủy vé	
Usecase Description	Khách hàng muốn hủy vé chuyển xe đã đặt
Actor chính	Khách hàng
Trigger	Khách hàng vào xem lịch sử vé
Pre – condition	Khách hàng hoàn tất đặt vé, hoàn thành chuyến đi, vé hủy, và vé chưa thanh toán
Post – condition	Khách hàng hủy vé
Main flows (Luồng sự kiện chính)	
User	System
<p>1a. Khách hàng di chuột vào tài khoản tại trang chủ, nhấn vào biểu tượng</p> 	
<p>1b. Khách hàng nhấp vào mục “Vé của tôi”</p>	<p>1c. Hệ thống hiển thị trang lịch sử vé</p> 

1d. Khách hàng chọn mục “Hủy vé”	1e. Hệ thống hiển thị thông báo xác nhận hủy <div data-bbox="987 331 1319 497"> <p>Bạn có chắc chắn muốn hủy vé?</p> <p>Hủy vé</p> </div>
1f. Khách hàng xác nhận hủy vé	1g. Hệ thống hiển thị trang chi tiết hủy vé <div data-bbox="849 622 1422 900"> <div> <p>Nhà cung cấp</p> <p>Công ty CP Đặt Vé Rê</p> <p>Mã đơn hàng OX063AV_1688366478</p> <p>M/ô tả Datvere-OX063AV_1688366478</p> <p>Số tiền thanh toán 130.000đ</p> <p>Hủy vé</p> </div> <div> <p>Thông tin người dùng</p> <p>Có tên Email 0216280596</p> <p>Email nguyenthienhienhien21@gmail.com</p> <p>Số điện thoại 0933600000</p> <p>Địa chỉ Việt Nam</p> <p>Địa chỉ chi nhánh 17200 • TP. Hồ Chí Minh</p> <p>Địa chỉ chi nhánh 1610 • TP. Hồ Chí Minh</p> <p>Địa chỉ chi nhánh 1610 • TP. Hồ Chí Minh</p> <p>Bạn có mã giảm giá?</p> <p>Không giảm giá Chưa hoặc nhập mã</p> <p>Tổng tiền 130.000 đ</p> </div> </div>
Alternative flow(Luồng sự kiện thay thế): Không	
Exception flow (Luồng sự kiện ngoài lề)	
Không có	




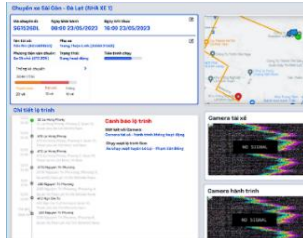
Bảng 3.2.10 Usecase đánh giá

Usecase đánh giá	
Usecase Description	Khách hàng đánh giá chuyến xe
Actor chính	Khách hàng
Trigger	Khách hàng vào xem lịch sử vé
Pre – condition	Khách hàng hoàn tất đặt vé, hoàn thành chuyến đi
Post – condition	Khách đánh giá
Main flows (Luồng sự kiện chính)	
User	System
1a. Khách hàng di chuột vào tài khoản tại trang chủ, nhấn vào biểu tượng 	
1b. Khách hàng nhấp vào mục “Vé của tôi”	1c. Hệ thống hiển thị trang lịch sử vé 
1d. Khách hàng chọn mục “Đánh giá”	1e. Hệ thống hiển thị phần đánh giá 
1f. Khách hàng nhập đánh giá và nhấn “Gửi”	
Alternative flow(Luồng sự kiện thay thế): Không	
Exception flow (Luồng sự kiện ngoài lề)	
Không có	


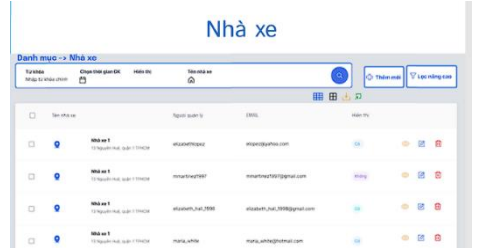


Bảng 3.2.11 Usecase quản lý chuyến đi

Usecase quản lý chuyến đi	
Usecase Decription	Nhà xe quản lý các chuyến đi của KH
Actor chính	Nhà xe
Trigger	Nhà xe tìm kiếm, xem thông tin chi tiết, đặt vé và quản lý các chuyến đi của của khách hàng.
Pre – condition	Đã đăng nhập vào hệ thống
Post – condition	Vào được trang quản lý chuyến đi
Main flows (Luồng sự kiện chính)	
User	System
<p>1a. Sau khi đăng nhập chọn mục “Quản lý chuyến đi”</p> 	<p>1b. Hệ thống hiển thị trang quản lý chuyến xe</p> 
Alternative flow(Luồng sự kiện thay thế): Không có	
Exception flow (Luồng sự kiện ngoài lề)	
Không có	


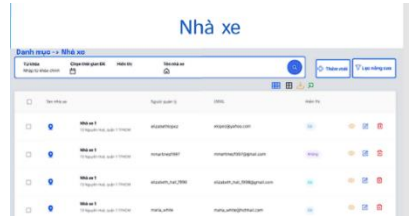
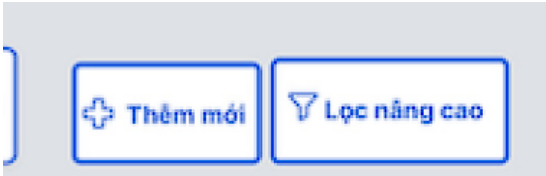
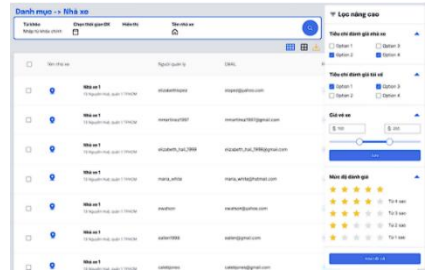
Bảng 3.2.12 Usecase xem lộ trình

Usecase xem lộ trình	
Usecase Description	Nhà xe xem chi tiết lộ trình của một chuyến đi
Actor chính	Nhà xe
Trigger	Nhà xe xem vị trí của các chuyến xe để biết lộ trình đi có đúng chưa và thời gian đi có vượt quá thời gian quy định của chuyến đi đó không.
Pre – condition	Vào trang quản lý chuyến đi
Post – condition	Chọn để xem chi tiết của 1 lộ trình
Main flows (Luồng sự kiện chính)	
User	System
<p>1a. Sau khi đăng nhập chọn mục người dùng</p> 	<p>1b. Hệ thống hiển thị trang quản lý chuyến xe</p> 
<p>1c. Chọn bất kỳ hành trình để xem chi tiết</p> 	<p>1d. Hệ thống hiển thị trang chi tiết lộ trình</p> 
Alternative flow(Luồng sự kiện thay thế): Không có	
Exception flow (Luồng sự kiện ngoài lề)	
Admin nhập sai các trường dữ liệu.	


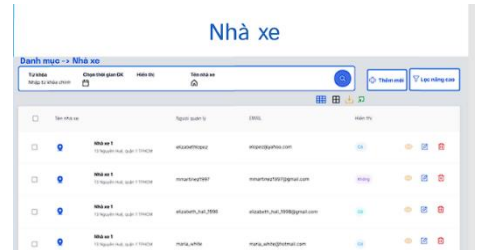


Bảng 3.2.13 Usecase xem chi tiết nhà xe

Usecase xem chi tiết nhà xe	
Usecase Decription	Xem các thông tin chi tiết của nhà xe
Actor chính	Nhà xe
Trigger	Nhà xe xem các thông tin như: người quản lý, sđt, địa chỉ nhà xe,...
Pre – condition	Đăng nhập thành công
Post – condition	Vào trang nhà xe
Main flows (Luồng sự kiện chính)	
User	System
<p>1a. Sau khi đăng nhập chọn mục “Nhà xe”</p> 	<p>1b. Hệ thống hiển thị trang danh sách nhà xe</p> 
<p>1c. Chọn bất kỳ nhà xe để xem thông tin chi tiết</p> 	<p>1d. Hệ thống hiển thị thông tin chi tiết của nhà xe</p> 
Alternative flow(Luồng sự kiện thay thế): Không có	
Exception flow (Luồng sự kiện ngoài lề)	
Không có	

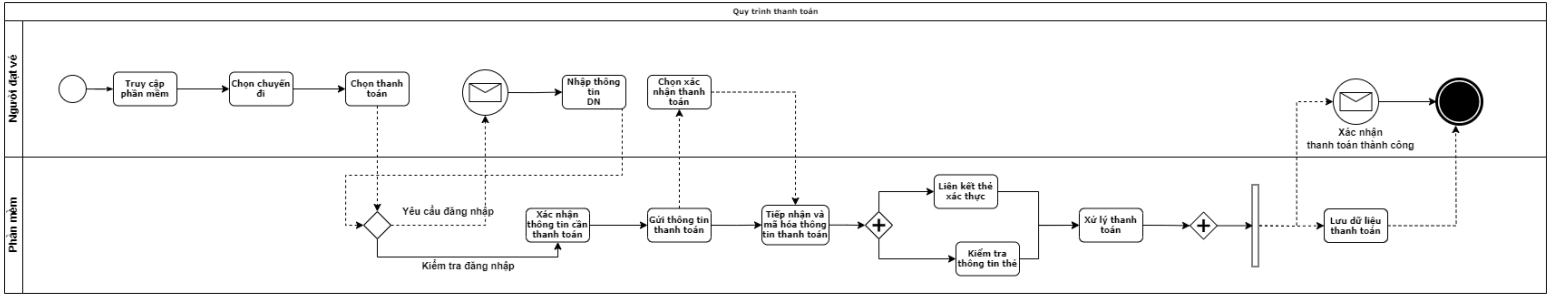
Bảng 3.2.14 Usecase lọc nâng cao

Usecase lọc nâng cao	
Usecase Description	Lọc các thông tin cần thiết để dễ dàng tìm kiếm chính xác nhà xe
Actor chính	Nhà xe
Trigger	Lọc các thông tin theo các tiêu chí như: tài xế, nhà xe, mức độ sao
Pre – condition	Vào trang nhà xe
Post – condition	Vào mục lọc nâng cao
Main flows (Luồng sự kiện chính)	
User	System
<p>1a. Sau khi đăng nhập chọn mục “Nhà xe”</p> 	<p>1b. Hệ thống hiển thị trang danh sách nhà xe</p> 
<p>1c. Chọn mục “Lọc nâng cao”</p> 	<p>1d. Hệ thống hiển thị trang lọc</p> 
Alternative flow(Luồng sự kiện thay thế): Không có	
Exception flow (Luồng sự kiện ngoài lề)	
Không có	

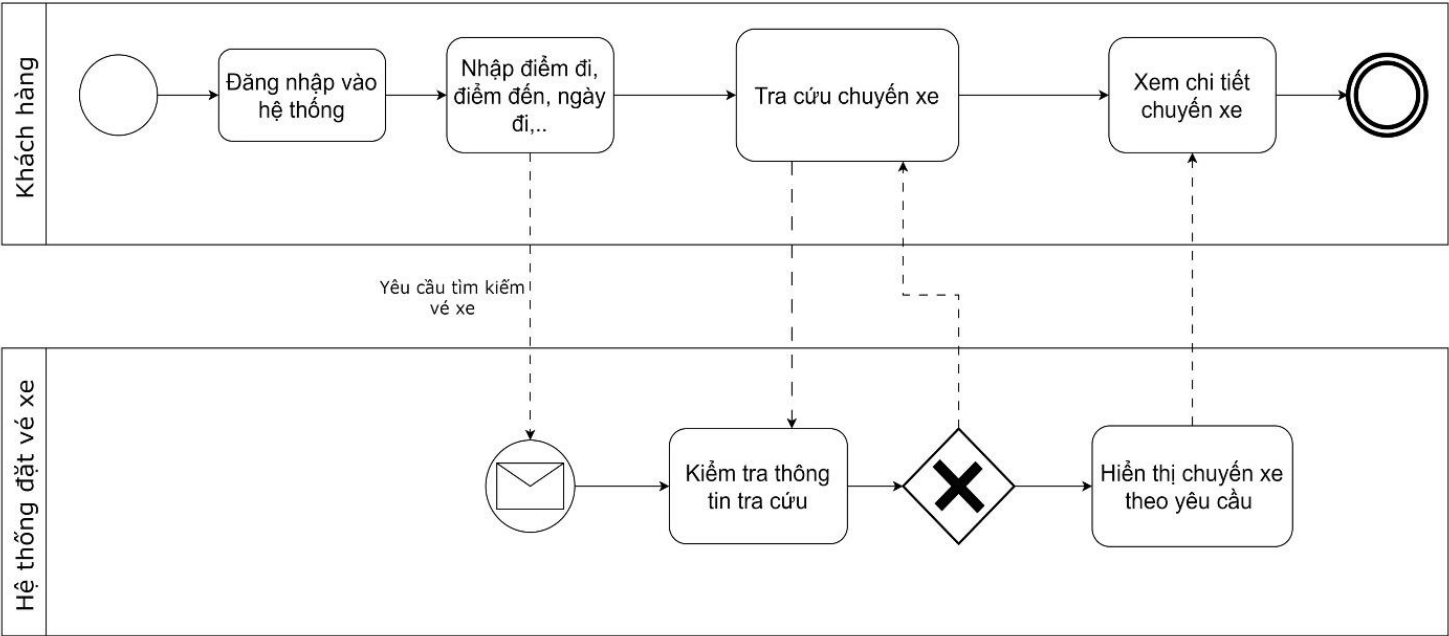
Bảng 3.2.15 Usecase cập nhật nhà xe

Usecase cập nhật nhà xe	
Usecase Description	Thêm, sửa, xóa các nhà xe
Actor chính	Nhà xe
Trigger	Khi có thay đổi về thông tin nhà xe
Pre – condition	Đăng nhập thành công
Post – condition	Vào trang nhà xe
Main flows (Luồng sự kiện chính)	
User	System
<p>1a. Sau khi đăng nhập chọn mục “Nhà xe”</p> 	<p>1b. Hệ thống hiển thị trang danh sách nhà xe</p> 
<p>1c. Chọn mục “Thêm mới”</p> 	<p>1d. Hệ thống hiển thị trang lọc</p> 
Alternative flow(Luồng sự kiện thay thế): Không có	
Exception flow (Luồng sự kiện ngoài lề)	
Không có	

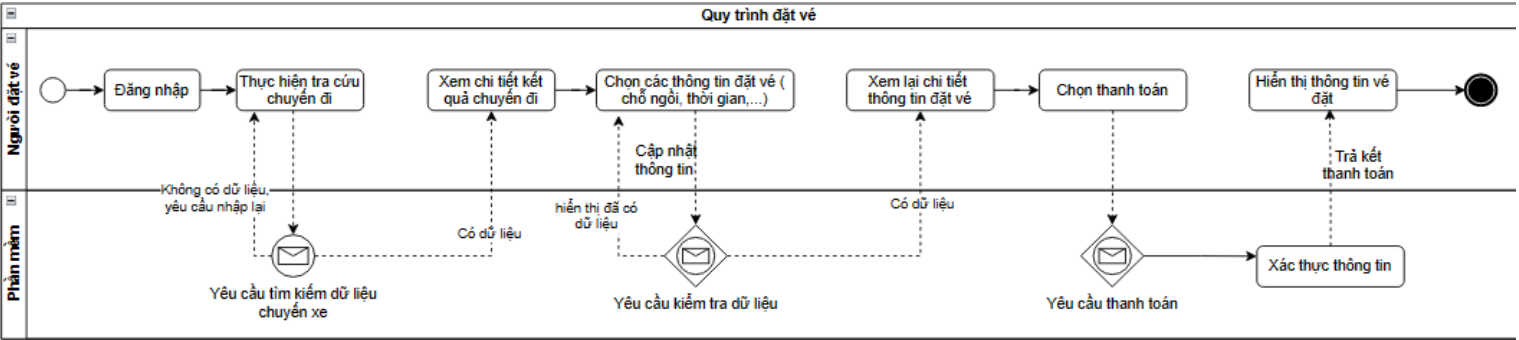
3.3 Sơ đồ BPMN



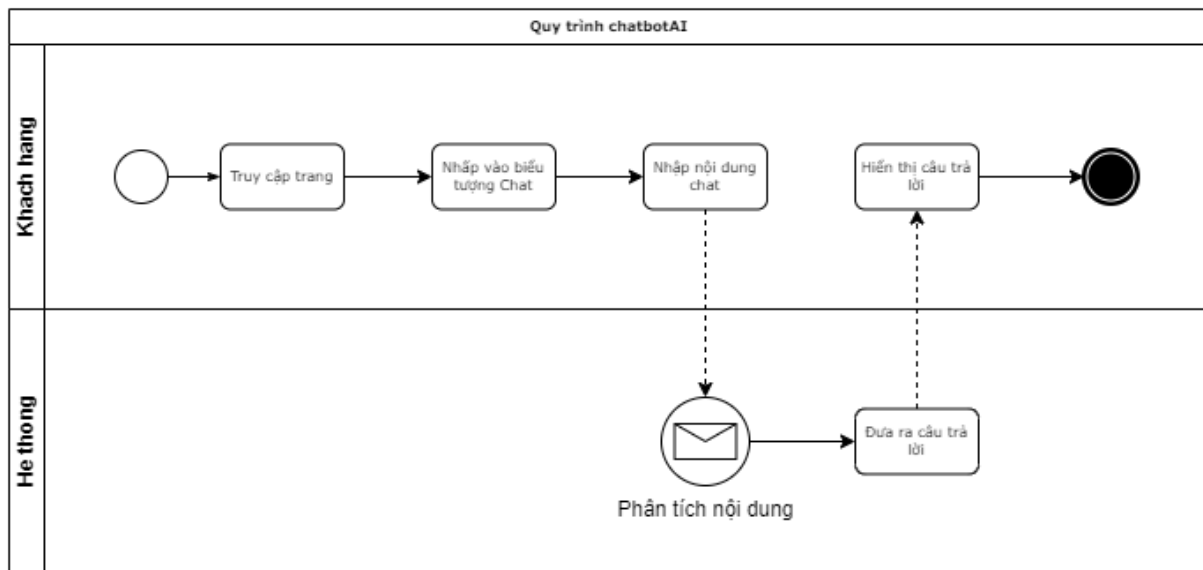
Hình 3.3.3 Quy trình thanh toán



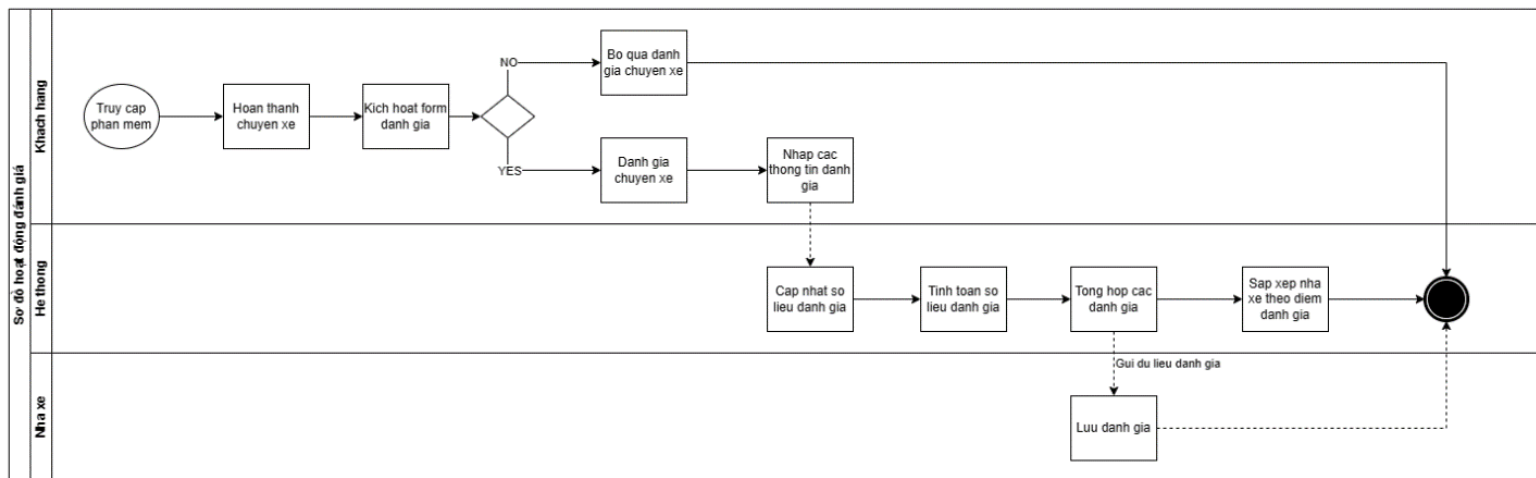
Hình 3.3.2 Quy trình tra cứu



Hình 3.3.1 Quy trình đặt vé

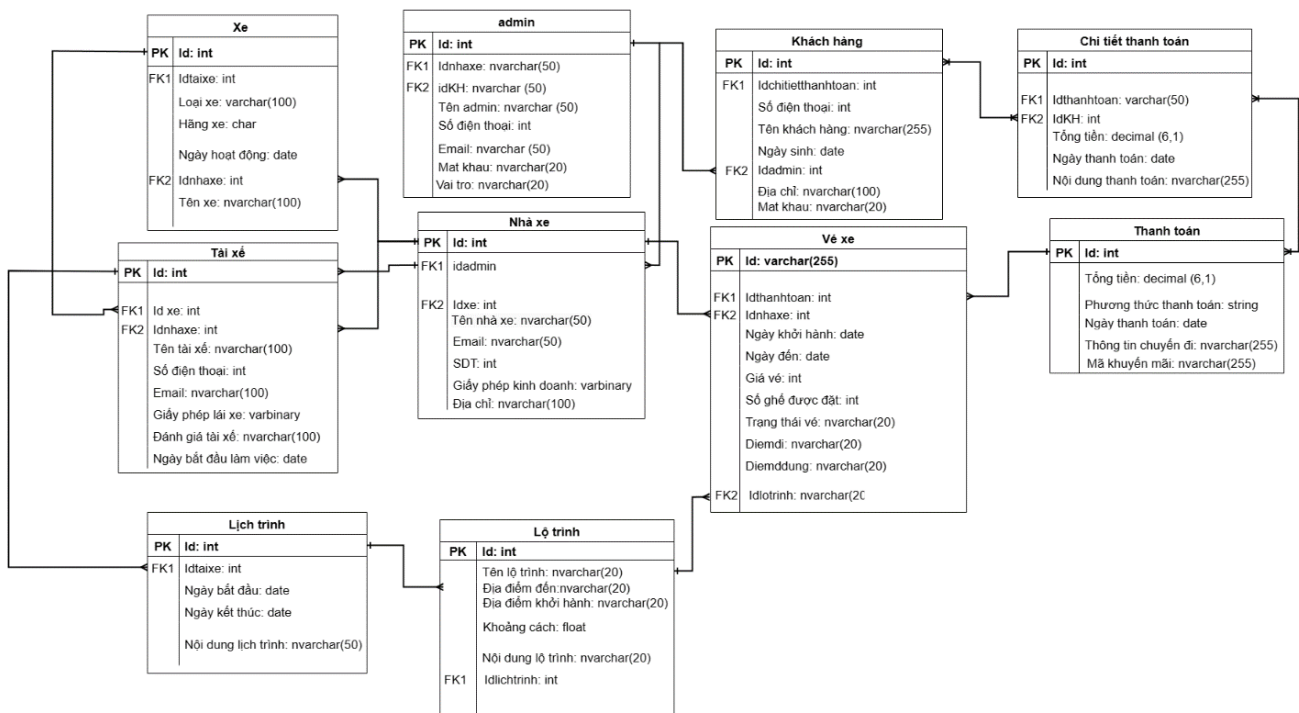


Hình 3.3.4 Quy trình chatbotAI



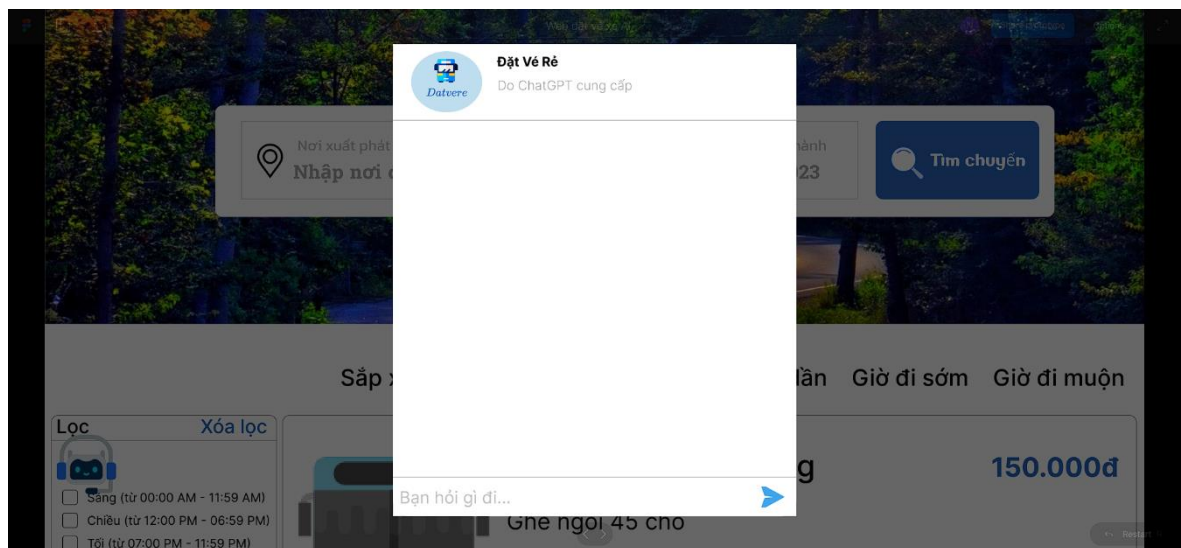
Hình 3.3.5 Quy trình đánh giá

3.4 Mô hình ERD

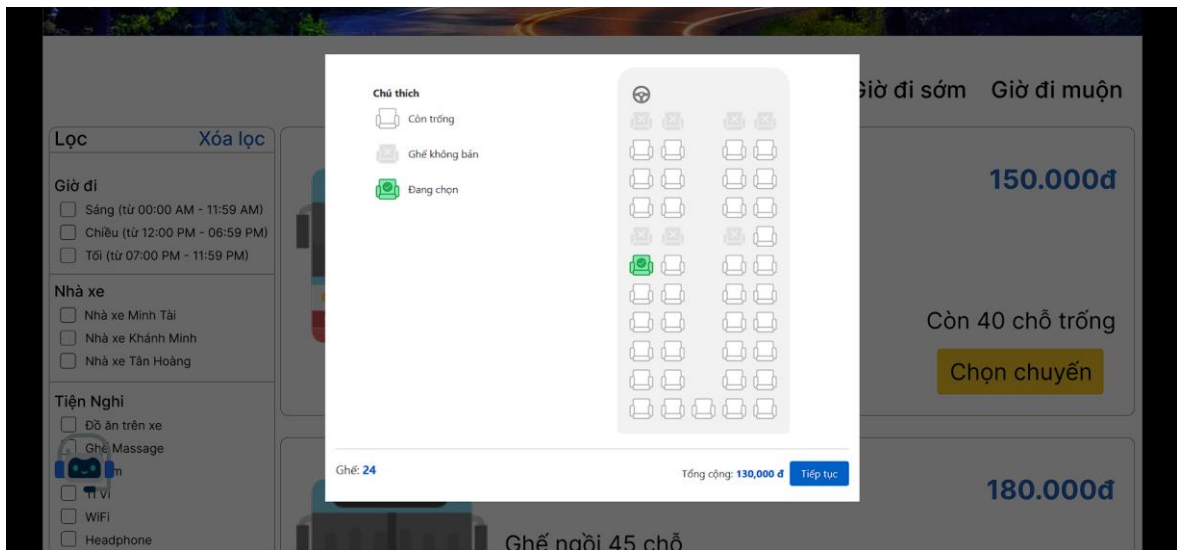


Hình 3.4.1 Mô hình thực thể ERD

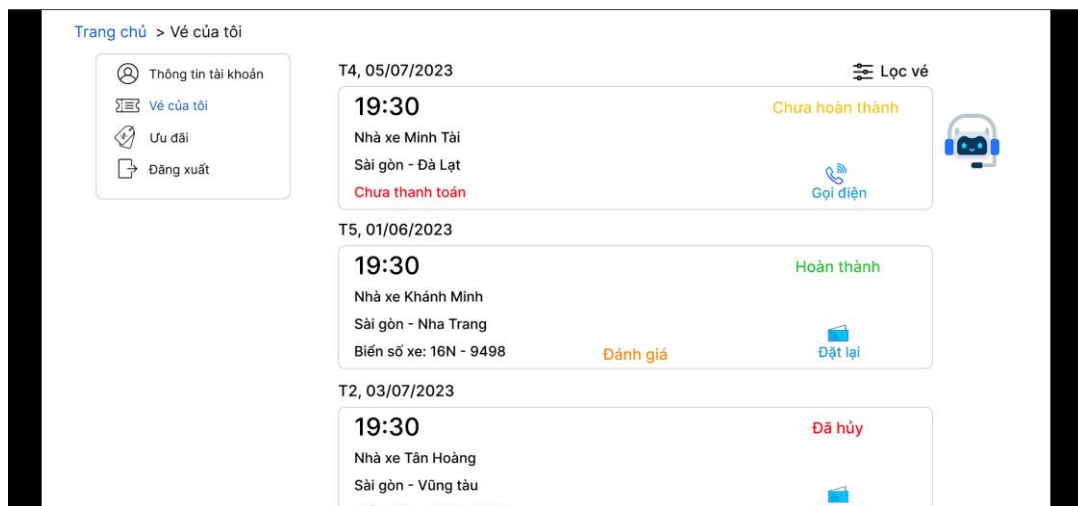
3.5 Các thiết kế wireframe



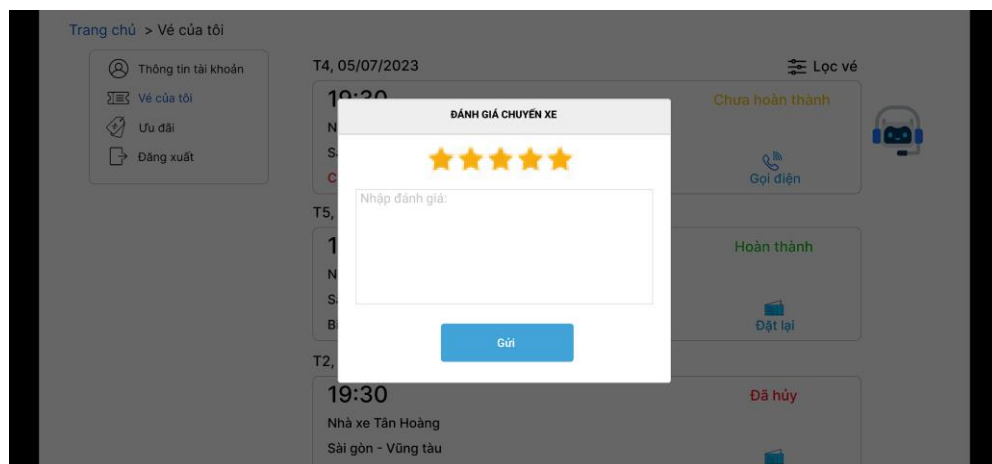
Hình 3.5.1 Wireframe chatbotAI



Hình 3.5.2 Wireframe đặt vé



Hình 3.5.3 Wireframe lịch sử vé



Hình 3.5.4 Wireframe đánh giá

CHƯƠNG 4. KẾT LUẬN

4.1 Kết quả đạt được

Về kiến thức:

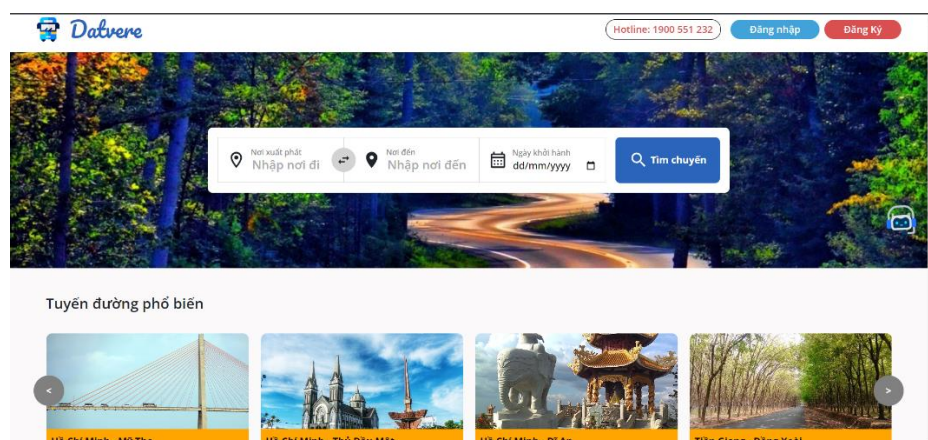
Sau thời gian thực hiện đồ án, nhóm đã tiếp thu được nhiều kiến thức. Có thể nói thông qua đồ án, nhóm đã đạt được:

- Hiểu biết nhiều hơn về các kỹ thuật trong phân tích nghiệp vụ
- Phân tích các thiết kế theo hướng chuyên nghiệp hơn
- Hiểu hơn về các hỗ trợ trong việc đặt vé xe
- Nâng cao tình thần tự học hỏi và tự nghiên cứu

Về Web:

Web đã phần nào xây dựng và đáp ứng hầu hết các chức năng chính:

- Đăng nhập
- Đăng ký
- Đặt vé
- Lọc
- Chatbot AI
- Quản lý thông tin người dùng
- Tìm kiếm
- Đánh giá
- Thanh toán



Hình 4.1.1 Trang chủ web đặt vé

Sắp xếp theo: Sắp xếp theo

Lọc Xóa lọc

Giờ đi


- ☐ Sáng (từ 00:00 - 11:59)
- ☐ Chiều (từ 12:00 - 18:59)
- ☐ Tối (từ 19:00 - 23:59)

Nhà xe

- ☐ Nhà xe Phương Trang
- ☐ Nhà xe Mai Linh
- ☐ Nhà xe Sapa Express

Tiện nghi

- ☐ Đồ ăn trên xe
- ☐ Ghế Massage
- ☐ Ổ cắm
- ☐ TV
- ☐ Wifi
- ☐ Headphone
- ☐ VIP Seater
- ☐ Personal Desk
- ☐ USB Port



Nhà xe Phương Trang 175.000đ

Ghế ngồi - 44 chỗ


🕒 14:50 - Bến xe Phan Thiết

↓ 3 giờ 40 phút

📍 18:30 - Bến xe Thủ Dầu Một

Còn 39 chỗ trống

[Chọn tuyến](#)



Nhà xe Mai Linh 225.000đ

Ghế ngồi - 44 chỗ


🕒 08:50 - Bến xe Phan Thiết

↓ 3 giờ 40 phút

📍 12:30 - Bến xe Thủ Dầu Một

Còn 44 chỗ trống

[Chọn tuyến](#)



Nhà xe Mai Linh 325.000đ

Giường nằm - 44 chỗ

🕒 08:50 - Bến xe Phan Thiết


↓ 3 giờ 40 phút


📍 12:30 - Bến xe Thủ Dầu Một

Còn 44 chỗ trống

[Chọn tuyến](#)

Hình 4.1.2 Trang danh sách vé

 Hotline: 1900 551 232 Trần Minh Tài



Nhà xe Phương Trang

Tuyến: Phan Thiết → Dĩ An

Điểm đón khách
14:50 • Bến xe Phan Thiết
Vị trí bên xe trên bản đồ: [Nhấn vào đây](#)

Điểm trả khách
18:30 • Bến xe Thủ Dầu Một
Vị trí bên xe trên bản đồ: [Nhấn vào đây](#)

3 giờ 40 phút
246.82 km

Giá: 175.000/vé

Còn 39 / 44 chỗ trống


Xem đánh giá: [Nhấn vào đây](#)


Số lượng vé: 0

[Chọn ghế](#)

[Tiếp tục](#)

Hình 4.1.3 Trang đặt vé

 Hotline: 1900 551 232 Trần Minh Tài



Vui lòng chọn số ghế:

Tầng trên		Tầng dưới	
A1	A12	B1	B12
A2	A13	B2	B13
A3	A14	B3	B14
A4	A15	B4	B15
A5	A16	B5	B16
A6	A17	B6	B17
A7	A18	B7	B18
A8	A19	B8	B19
A9	A20	B9	B20
A10	A21	B10	B21
A11	A22	B11	B22

Chú thích

- ☐ Còn trống
- ☒ Không bán
- ☒ Đang chọn

Điểm trả khách
18:30 • Bến xe Thủ Dầu Một
Vị trí bên xe trên bản đồ: [Nhấn vào đây](#)

Ghế: A16 A5

Tổng cộng: 350.000 đ

[Tiếp tục](#)

[Tiếp tục](#)

Hình 4.1.4 Trang chọn vị trí ngồi và số lượng vé

Thông tin thanh toán

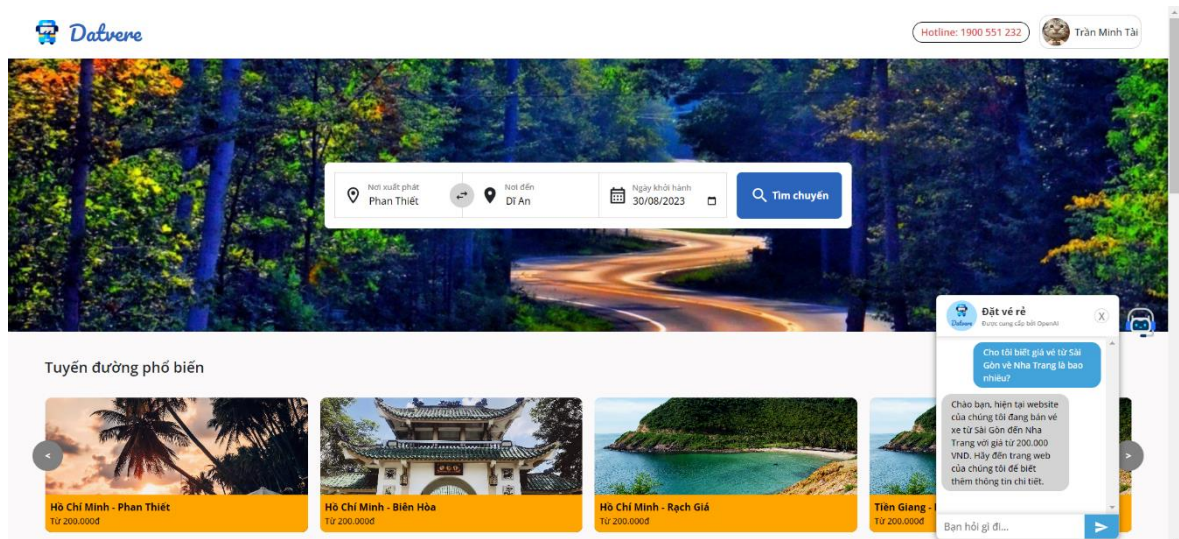
Họ và tên:	Trần Minh Tài
Email:	minhtai.py2017@gmail.com
Số điện thoại:	0338976478
Điểm đi:	Bến xe Phan Thiết
Điểm đến:	Bến xe Thủ Đức Một
Ngày, giờ đi:	14:50, 2023-08-30
Ghế ngồi:	A16 A5
Đơn giá:	175.000đ
Tổng tiền:	350.000đ

Hình thức thanh toán: ☐ Thanh toán trực tiếp ☒ Thanh toán qua Paypal

[Paypal](#)

[Thẻ ghi nợ hoặc tín dụng](#)

Hình 4.1.5 Trang thông tin chi tiết vé thanh toán



Hình 4.1.6 Trang chatbotAI

4.2 Hạn chế

Báo cáo bao gồm một số hạn chế ngoài những kết quả đã đạt được. Bài báo chủ yếu dựa trên khả năng đọc hiểu và chọn lọc thông tin từ các tài liệu tham khảo có liên quan. Do đó, báo cáo sẽ còn một số thiếu sót về kiến thức chọn lọc. Đề tài rất rộng lớn nên có những thiếu sót. Ngoài ra, một số chức năng của web vẫn chưa hoàn thành và chính xác. Đây là một trang web có nhiều người sử dụng. Để đáp ứng được điều này, hệ thống cần một máy chủ ổn định và mạnh mẽ. Mặt khác, việc xây dựng và vận hành một máy chủ sẽ tốn rất nhiều thời gian và tiền bạc. Đây thực sự là một thách thức.

4.3 Hướng phát triển tương lai

Công nghệ trí tuệ nhân tạo (AI) đã được ứng dụng rộng rãi trong nhiều lĩnh vực, bao gồm cả ngành vận tải. Các trang web đặt vé xe AI đang dần thay thế các trang web truyền thống.

Hướng phát triển tương lai của web đặt vé xe AI có thể được tóm tắt như sau:

- Tăng cường khả năng tự động hóa: Nhiều tính năng tự động hóa hơn sẽ được tích hợp vào các web đặt vé xe AI, giúp khách hàng đặt vé nhanh chóng và dễ dàng hơn. Khách hàng, ví dụ, có thể yêu cầu vé bằng giọng nói hoặc sử dụng các tính năng gợi ý thông minh.
- Nâng cao trải nghiệm người dùng: Để đáp ứng nhu cầu đa dạng của khách hàng, các web đặt vé xe AI sẽ được thiết kế thân thiện với người dùng hơn.
- Tăng cường tính bảo mật: Các tính năng bảo mật tiên tiến hơn sẽ được tích hợp vào các web đặt vé xe AI để bảo vệ thông tin cá nhân của khách hàng. Ví dụ, web có thể sử dụng các giải pháp mã hóa dữ liệu, nhận dạng khuôn mặt hoặc xác thực hai yếu tố.

Dưới đây là một số xu hướng cụ thể có thể được áp dụng trong các web đặt vé xe AI trong tương lai:

- Trí tuệ nhân tạo (AI): Việc tự động và cá nhân hóa trải nghiệm người dùng sẽ được cải thiện bằng cách sử dụng AI.
- Viễn thông di động (5G): 5G sẽ mang lại trải nghiệm khách hàng tốt hơn bằng cách cho phép các web đặt vé AI hoạt động nhanh chóng và ổn định hơn.
- Internet vạn vật (IoT): AI sẽ cải thiện chất lượng dịch vụ bằng cách hỗ trợ các web đặt vé xe bằng cách thu thập và phân tích dữ liệu từ nhiều nguồn khác nhau. Ví dụ, IoT có thể được sử dụng để theo dõi tình trạng xe, thời gian đến và các thông tin khác.

TÀI LIỆU THAM KHẢO

Tiếng Việt

- Hiếu, N. V. (2021). *Trí tuệ nhân tạo (AI) trong du lịch*. NXB Đại học Quốc gia Hà Nội.
- Hùng, P. N. (2022). *Ứng dụng AI trong du lịch và lễ hành*. NXB Thanh Niên.
- Linh, L. T. T. (2021). *AI trong du lịch: Thực trạng và triển vọng*. Tạp chí Du lịch.
- Anh, Đ. V. (2023). *AI trong du lịch: Cơ hội và thách thức*. Tạp chí Công nghệ thông tin.

Tiếng Anh

- Theobald, O. (2020). *Machine learning for absolute beginners*. Packt Publishing.
- Ng, A. (2017). *Deep learning for coders*. Manning Publications.
- Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with Python*. O'Reilly Media.
- Fuchs, T. (2021). *Artificial intelligence for travel and tourism*. Routledge.
- McKinsey & Company. (2020). *The future of AI in travel*. McKinsey & Company.
- Forbes. (2023, March 8). *How AI is changing the travel industry*. Forbes.
- Travel Weekly. (2022, January 20). *Chatbots in travel*. Travel Weekly.
- Skift. (2023, February 22). *Virtual reality and augmented reality in travel*. Skift.
- Google AI. (2022, April 20). *Machine learning for recommendation engines*. Google AI.