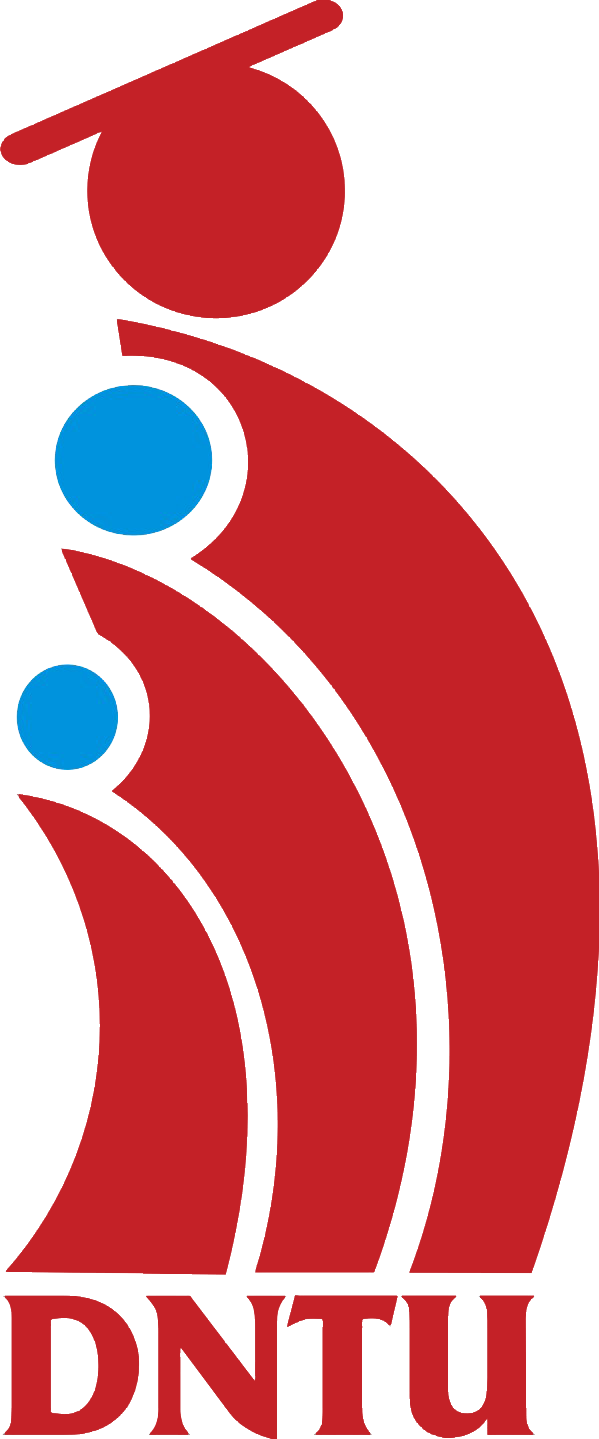
BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ ĐỒNG NAI**



Trần Tiến

Huỳnh Minh Đạt

Lê Quang Nhựt

Lê Quang Thái

**XÂY DỰNG WEBSITE ĐẶT XE TAXI BẰNG NODEJS**

**KHÓA LUẬN TỐT NGHIỆP**

**Chuyên ngành: CÔNG NGHỆ THÔNG TIN**

**Khóa học: 2020 - 2025**

**Người hướng dẫn: LÊ THỊ TRANG**

**ĐỒNG NAI - 2025**

THỪA THIÊN , năm

HÀ NỘI - năm

BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ ĐỒNG NAI



Trần Tiến

Huỳnh Minh Đạt

Lê Quang Nhựt

Lê Quang Thái

XÂY DỰNG WEBSITE ĐẶT XE TAXI BẰNG NODEJS

KHÓA LUẬN TỐT NGHIỆP

Chuyên ngành: CÔNG NGHỆ THÔNG TIN

Khóa học: 2020-2024

Người hướng dẫn: Lê Thị Trang

Đồng Nai-2025

|  |  |
| --- | --- |
| TRƯỜNG ĐẠI HỌC CÔNG NGHỆ ĐỒNG NAI  **KHOA CÔNG NGHỆ** | **CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**  **Độc lập – Tự do – Hạnh phúc** |

NHIỆM VỤ KHÓA LUẬN TỐT NGHIỆP

Nhóm Sinh Viên Thực Hiện:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| STT | MSSV | Họ | Tên | Lớp | Khóa học | Ghi chú |
|  | 162000131 | Trần | Tiến | 20DTH4 | 2020 - 2024 |  |
|  | 162001644 | Huỳnh Minh | Đạt | 20DTH5 | 2020 - 2024 |  |
|  | 162000614 | Lê Quang | Nhựt | 20DTH2 | 2020 - 2024 |  |
|  | 162000316 | Lê Quang | Thái | 20DTH5 | 2020- 2024 |  |

- Chuyên ngành: CÔNG NGHỆ THÔNG TIN

**1.Tên Đề Tài:**

Xây dựng website đặt xe taxi bằng nodejs

**2.Cấu trúc của khóa luận tốt nghiệp:**

Chương 1. Giới thiệu đề tài

* [Bối cảnh và lý do chọn đề tài](#_Toc195115322)
* Bài toán đặt ra
* Mục tiêu nghiên cứu
* Phạm vi và giới hạn nghiên cứu

Chương 2. Cơ sở lý thuyết

* Khái niệm về website đặt xe taxi
* Các vấn đề và thách thức
* Các công nghệ và công cụ hỗ trợ

Chương 3. Phân tích và thiết kế hệ thống

Chương 4. Kết luận và hướng phát triển

**3. Ngày giao nhiệm vụ khóa luận: .../.../20....**

**4. Ngày hoàn thành nhiệm vụ: ......../20....**

**5. Họ tên người hướng dẫn: ThS/TS.**

Nội dung và yêu cầu khóa luận tốt nghiệp đã được thông qua bộ môn và khoa.

|  |  |
| --- | --- |
|  | *Đồng Nai, ngày…...tháng…….năm 20....* |
| NGƯỜI HƯỚNG DẪN  *(Ký và ghi rõ họ tên)* | TRƯỞNG KHOA  *(Ký và ghi rõ họ tên)* |
|  |  |
| TỔ TRƯỞNG BỘ MÔN  *(Ký và ghi rõ họ tên)* | PHẦN GHI KẾT QUẢ KLTN  Ngày bảo vệ: …………………  Điểm tổng kết: ………………. |

LỜI CAM ĐOAN

Cả nhóm xin cam đoan rằng đề tài "Xây dựng hệ thống Web Booking Taxi bằng Node.js" là một công trình nghiên cứu độc lập, được thực hiện dưới sự hướng dẫn của giảng viên Lê Thị Trang.

Toàn bộ quá trình nghiên cứu, khảo sát và triển khai hệ thống đều dựa trên những số liệu thực tế, không sao chép từ bất kỳ công trình nghiên cứu nào khác. Chúng tôi cam kết chịu hoàn toàn trách nhiệm về tính trung thực và nội dung của đề tài này.

Đồng Nai, ngày …... tháng …... năm …...

Nhóm thực hiện

LỜI CẢM ƠN

Chúng tôi xin gửi lời cảm ơn chân thành đến quý thầy cô khoa Công nghệ Thông tin, Trường Đại học Công nghệ Đồng Nai đã tạo điều kiện thuận lợi để nhóm có cơ hội nghiên cứu và thực hiện luận án "Xây dựng hệ thống Web Booking Taxi bằng Node.js".

Trong suốt quá trình thực hiện, chúng tôi đã có cơ hội học hỏi, nâng cao kỹ năng làm việc nhóm, thu thập và xử lý thông tin một cách khoa học. Đặc biệt, chúng tôi xin gửi lời tri ân sâu sắc đến giảng viên Lê Thị Trang, người đã tận tình hướng dẫn, hỗ trợ và đóng góp những ý kiến quý báu giúp nhóm hoàn thiện luận án này.

Mặc dù đã nỗ lực hết mình, nhưng do hạn chế về kinh nghiệm, nhóm khó tránh khỏi những thiếu sót. Chúng tôi rất mong nhận được những góp ý quý báu từ quý thầy cô để có thể cải thiện và hoàn thiện hơn trong những dự án nghiên cứu tiếp theo.

Trân trọng cảm ơn!

|  |  |
| --- | --- |
| BỘ GIÁO DỤC VÀ ĐÀO TẠO  **TRƯỜNG ĐH CÔNG NGHỆ ĐỒNG NAI** | **CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**  **Độc lập – Tự do – Hạnh phúc**    *Đồng Nai, ngày……tháng ….. năm 20…..* |

**NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN**

**Họ tên sinh viên**:

**Về đề tài**:

**Họ và tên giảng viên nhận xét :**

**Học hàm, học vị:**

**Đơn vị:**

1. **Về tính cấp thiết của đề tài**

1. **Về nội dung**
   1. ***Những ưu điểm***

* 1. ***Những nhược điểm***

1. **Về thái độ, tinh thần thực hiện việc trong quá trình thực hiện khóa luận tốt nghiệp**

1. **Kết luận**

|  |  |
| --- | --- |
|  | **NGƯỜI HƯỚNG DẪN**  *(ký và ghi rõ họ tên)* |

Mục lục

[CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI 15](#_Toc195115918)

[1.1 Bối cảnh và lý do chọn đề tài 15](#_Toc195115919)

[1.2 Bài toán đặt ra 15](#_Toc195115920)

[1.3 Mục tiêu nghiên cứu 15](#_Toc195115921)

[1.4 Phạm vi và giới hạn nghiên cứu 16](#_Toc195115922)

[CHƯƠNG 2. CƠ SỞ LÝ THUYẾT 17](#_Toc195115923)

[2.1 Khái niệm về website đặt xe taxi 17](#_Toc195115924)

[2.2 Các vấn đề và thách thức 17](#_Toc195115925)

[2.3 Các công nghệ và công cụ sử dụng 18](#_Toc195115926)

[2.3.1 Mô hình MVC 18](#_Toc195115927)

[2.3.2 Node JS 20](#_Toc195115928)

[2.3.3 Express 22](#_Toc195115929)

[2.4 Thời gian thực: 24](#_Toc195115930)

[2.4.1 Web socket 24](#_Toc195115931)

[2.4.2 Socket.io 26](#_Toc195115932)

[2.5 Cơ sở liệu 29](#_Toc195115933)

[2.6 Tích hợp bản đồ OpenMap 30](#_Toc195115934)

[CHƯƠNG 3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 33](#_Toc195115935)

[3.1. Khảo sát và xây dựng đặc tả 33](#_Toc195115936)

[3.1.1. Xác định các đối tượng 34](#_Toc195115937)

[3.2 Thiết kế sơ đồ hệ thống 35](#_Toc195115938)

[3.2.1 Sơ đồ phần cấp chức năng 35](#_Toc195115939)

[3.2.2 Sơ đồ usecase 35](#_Toc195115940)

[3.2.3 Database 52](#_Toc195115941)

[3.3.4 Biểu đồ hoạt động 52](#_Toc195115942)

[3.3 Giao diện phần mềm 53](#_Toc195115943)

[3.3.1 Web cho user 53](#_Toc195115944)

[3.3.2 Web cho Admin 59](#_Toc195115945)

[CHƯƠNG 4. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 63](#_Toc195115946)

[4.1. Kết luận 63](#_Toc195115947)

[4.2. Ưu và nhược điểm 63](#_Toc195115948)

[4.2.1 Ưu điểm 63](#_Toc195115949)

[4.2.2 Nhược điểm 63](#_Toc195115950)

[4.3. Hướng phát triển 64](#_Toc195115951)

DANH MỤC HÌNH ẢNH

[Hình 2.1: Thành phần của mô hình MVC 13](#_Toc195016963)

[Hình 2.2: Luồng giao tiếp WebSocket 21](#_Toc195016964)

[Hình 2.5: Ví dụ về cách tích hợp OpenMap GL VN 25](#_Toc195016965)

[Hình 3.1: Sơ đồ chức năng của hệ thống 28](#_Toc195016966)

[Hình 3.2: Sơ đồ use case đăng nhập 28](#_Toc195016967)

[Hình 3.3: Sơ đồ use case quản lý / phân quyền tài khoản 30](#_Toc195016968)

[Hình 3.4: Sơ đồ use case báo cáo thống kê 31](#_Toc195016969)

[Hình 3.5: Sơ đồ use case quản lý hình ảnh, banner và logo 33](#_Toc195016970)

[Hình 3.6: Sơ đồ use case quản lý dịch vụ 34](#_Toc195016971)

[Bảng 3.5: Sơ đồ use case quản lý dịch vụ 36](#_Toc195016972)

[Hình 3.7: Sơ đồ use case quản lý đơn chuyến 36](#_Toc195016973)

[Hình 3.8: Sơ đồ use case theo dõi doanh thu 38](#_Toc195016974)

[Hình 3.9: Sơ đồ use case của người đặt xe 39](#_Toc195016975)

[Hình 3.11: Biểu đồ database 41](#_Toc195016976)

[Hình 3.12: Biểu đồ hoạt động quá trình đặt xe 42](#_Toc195016977)

[Hình 3.13: Giao diện trang đăng nhập 43](#_Toc195016978)

[Hình 3.14: Giao diện trang đăng ký 43](#_Toc195016979)

[Hình 3.14: Trang thông tin người dùng 44](#_Toc195016980)

[Hình 3.13: Trang chủ 45](#_Toc195016981)

[Hinh 3.34: Trang dịch vụ 46](#_Toc195016982)

[Hình 3.15: Trang liên hệ 46](#_Toc195016983)

[Hình 3.16: Trang tin tức 47](#_Toc195016984)

[Hình :Trang Dashboard 48](#_Toc195016985)

[Hình 3.19: Hiển thị các chuyến đang được thực hiện 49](#_Toc195016986)

[Hình 3.19: Hiễn thị lịch sử chuyến thành công 50](#_Toc195016987)

[Hình 3.20: Chi tiết chuyến đi 50](#_Toc195016988)

[Hình 3.12: Trang báo cáo thống kê 51](#_Toc195016989)

DANH MỤC BẢNG

[Bảng 2.1: Ưu nhược điểm của mô hình MVC 18](#_Toc195018962)

[Bảng 2.1: Ưu nhược điểm của Websocket 24](#_Toc195018963)

[Bảng 3.1: Đặc tả sơ đồ usecase đăng nhập 36](#_Toc195018964)

[Bảng 3.2: Đặc tả sơ đồ usecase quản lý / phân quyền tài khoản 38](#_Toc195018965)

[Bảng 3.3: Đặc tả sơ đồ usecase báo cáo thống kê 40](#_Toc195018966)

[Bảng 3.4: Đặc tả sơ đồ usecase quản lý hình ảnh, banner và logo 42](#_Toc195018967)

[Bảng 3.6: Bảng đặc tả usecase quản lý đơn chuyến 46](#_Toc195018968)

[Bảng 3.7: Bảng đặc tả usecase quản lý đơn chuyến 48](#_Toc195018969)

[Bảng 3.8: Bảng đặc tả usecase của người đặt xe 51](#_Toc195018970)

LỜI MỞ ĐẦU

Trong thời đại công nghệ phát triển mạnh mẽ, việc ứng dụng công nghệ vào đời sống ngày càng phổ biến, đặc biệt là trong lĩnh vực giao thông vận tải. Với nhu cầu di chuyển ngày càng cao, việc đặt xe truyền thống dần bộc lộ nhiều hạn chế như mất thời gian, khó kiểm soát lộ trình và giá cả không minh bạch. Do đó, hệ thống **Web Booking Taxi** ra đời nhằm mang đến một giải pháp tối ưu, giúp người dùng dễ dàng đặt xe trực tuyến, đồng thời hỗ trợ quản lý hoạt động vận tải hiệu quả hơn.

Báo cáo luận văn này trình bày quá trình **nghiên cứu và phát triển hệ thống đặt xe trực tuyến sử dụng Node.js.** Hệ thống không chỉ cung cấp giao diện thân thiện, giúp người dùng tìm kiếm, đặt xe nhanh chóng mà còn hỗ trợ tài xế và quản trị viên trong việc quản lý chuyến xe, theo dõi thu nhập và tối ưu hóa vận hành.

Mục tiêu chính của dự án là **xây dựng một nền tảng đặt xe trực tuyến hiện đại, đảm bảo tính chính xác, bảo mật và dễ sử dụng**. Bên cạnh đó, hệ thống còn tích hợp các công nghệ web tiên tiến như **React, Vite, và Node.js**, giúp tối ưu hiệu suất và trải nghiệm người dùng.

Báo cáo này sẽ trình bày chi tiết về **mô hình hệ thống, các công nghệ sử dụng, quy trình phát triển phần mềm,** cũng như đánh giá hiệu quả của hệ thống sau khi triển khai. Hy vọng rằng dự án sẽ góp phần cải thiện chất lượng dịch vụ vận tải, tạo ra một nền tảng đặt xe thông minh và đáng tin cậy cho người dùng.

# GIỚI THIỆU ĐỀ TÀI

## Bối cảnh và lý do chọn đề tài

Nhu cầu di chuyển bằng taxi tại các đô thị lớn ngày càng tăng do tốc độ đô thị hóa và xu hướng sử dụng dịch vụ trực tuyến. Tuy nhiên, hình thức gọi xe truyền thống vẫn tồn tại nhiều bất cập như khó khăn trong việc gọi xe giờ cao điểm, giá cả thiếu minh bạch và quy trình đặt xe thủ công gây bất tiện.

Sự ra đời của các nền tảng đặt xe công nghệ như Grab, Uber đã góp phần thay đổi thói quen người dùng, mở ra cơ hội cải thiện trải nghiệm và tối ưu hóa hoạt động kinh doanh. Đề tài hướng tới xây dựng một website đặt xe trực tuyến sử dụng Node.js và các công nghệ web hiện đại, giúp khách hàng dễ dàng tìm kiếm, đặt xe, theo dõi hành trình và thanh toán một cách tiện lợi, an toàn và minh bạch.

## Bài toán đặt ra

Các phương thức đặt xe truyền thống hiện nay còn tồn tại nhiều hạn chế như quy trình thủ công, thiếu minh bạch về giá cước và lộ trình, gây bất tiện cho cả hành khách lẫn nhà cung cấp dịch vụ. Đặc biệt, trong giờ cao điểm, việc gọi xe thường gặp khó khăn do nhu cầu tăng cao nhưng hệ thống không đủ khả năng xử lý hiệu quả.

Trước những bất cập đó, vấn đề đặt ra là cần phát triển một hệ thống đặt xe taxi trực tuyến có thể khắc phục các điểm yếu của phương pháp cũ. Hệ thống cần cho phép người dùng đặt xe theo thời gian thực, theo dõi vị trí tài xế trên bản đồ, biết trước giá cước và thanh toán linh hoạt qua nhiều hình thức.

## 1.3 Mục tiêu nghiên cứu

Mục tiêu của đề tài là xây dựng một hệ thống đặt xe taxi trực tuyến có giao diện thân thiện, dễ sử dụng, cho phép người dùng đặt xe theo thời gian thực, theo dõi hành trình di chuyển và thanh toán một cách nhanh chóng. Hệ thống nhằm tạo cầu nối hiệu quả giữa khách hàng và tài xế, góp phần cải thiện trải nghiệm người dùng và nâng cao hiệu suất vận hành cho doanh nghiệp.

Để đạt được mục tiêu này, đề tài tập trung vào một số nội dung cụ thể:

* Phân tích yêu cầu hệ thống nhằm xác định rõ các chức năng cốt lõi như đặt xe, theo dõi vị trí, thanh toán và quản lý thông tin chuyến đi.
* Thiết kế kiến trúc tổng thể của ứng dụng, bao gồm sơ đồ cơ sở dữ liệu, luồng xử lý và bố cục giao diện người dùng.
* Triển khai hệ thống bằng các công nghệ hiện đại: sử dụng Node.js cho phần back-end, React cho front-end và MongoDB làm cơ sở dữ liệu.
* Cuối cùng, tiến hành kiểm thử toàn diện để đánh giá tính ổn định, hiệu quả và mức độ thân thiện với người dùng.

## 1.4 Phạm vi và giới hạn nghiên cứu

Đề tài tập trung phát triển một hệ thống đặt xe taxi trực tuyến với các chức năng cốt lõi như đặt xe theo thời gian thực, quản lý tài khoản người dùng và tài xế, cùng khả năng thanh toán trực tuyến thông qua ví điện tử hoặc thẻ ngân hàng.

Về mặt công nghệ, hệ thống sử dụng Node.js cho phía máy chủ, React để xây dựng giao diện người dùng và MongoDB làm cơ sở dữ liệu chính. Việc triển khai được giới hạn trong môi trường web, chưa mở rộng sang nền tảng di động hay các công nghệ phức tạp như trí tuệ nhân tạo dự đoán giá cước hoặc tối ưu tuyến đường.

Đối tượng áp dụng chính là người dùng tại các khu vực đô thị lớn như Hà Nội và TP.HCM – nơi nhu cầu di chuyển bằng taxi cao và có điều kiện tiếp cận công nghệ thuận lợi.

## 1.5 Kết quả dự kiến

# CƠ SỞ LÝ THUYẾT

## 2.1 Khái niệm về website đặt xe taxi

Website đặt xe taxi là nền tảng trực tuyến kết nối khách hàng với dịch vụ vận tải hành khách, cho phép người dùng dễ dàng đăng ký, tìm kiếm và đặt xe theo nhu cầu cá nhân. Hệ thống này không chỉ cung cấp các chức năng quan trọng như theo dõi tình trạng xe, quản lý lịch trình và lưu trữ lịch sử chuyến đi, mà còn giúp tối ưu hóa quy trình vận hành thông qua tự động hóa các tác vụ như báo cáo, phân tích dữ liệu và định vị GPS. Sự phát triển của hệ thống đặt xe taxi gắn liền với sự bùng nổ của công nghệ di động và Internet, khi các cải tiến về giao diện người dùng, tính năng định vị chính xác và tích hợp thanh toán trực tuyến đã góp phần cải thiện trải nghiệm khách hàng và nâng cao hiệu quả quản lý vận tải.

Đối với hệ thống quản lý và quảng bá dịch vụ đặt xe, nhà quản lý không chỉ cần giám sát các hoạt động vận hành hàng ngày mà còn phải thu thập và phân tích dữ liệu liên quan đến tình trạng xe, lộ trình, cũng như hợp đồng, thỏa thuận với tài xế. Hệ thống quản lý sẽ giúp tối ưu hóa quy trình bằng cách tự động hóa các tác vụ như theo dõi lịch sử chuyến đi, báo cáo tình trạng xe, và cải thiện trải nghiệm người dùng thông qua giao diện thân thiện, dễ sử dụng.

## 2.2 Các vấn đề và thách thức

Trong quá trình vận hành hệ thống đặt xe taxi, các nhà quản lý phải đối mặt với nhiều thách thức ở nhiều mặt khác nhau, từ quản lý nội bộ đến việc tuân thủ các quy định của pháp luật và đảm bảo an toàn cho người dùng.

Quản lý xe: Việc theo dõi tình trạng, bảo dưỡng và phân phối xe một cách hợp lý là điều cần thiết để đảm bảo hiệu suất vận hành. Hệ thống cần phải tích hợp các công nghệ định vị và giám sát thời gian thực nhằm hỗ trợ việc quản lý số lượng xe, lịch trình hoạt động và xử lý các sự cố kỹ thuật kịp thời.

Quản lý doanh thu: Theo dõi thu chi, báo cáo doanh thu và phân tích chi phí là những yếu tố quan trọng để đảm bảo tính bền vững của doanh nghiệp. Điều này không chỉ giúp tối ưu hóa lợi nhuận mà còn hỗ trợ việc lập kế hoạch tài chính, bảo trì xe và đầu tư phát triển hệ thống.

Vấn đề công nghệ và hạ tầng: Sự ổn định và bảo mật của hệ thống đặt xe taxi phụ thuộc vào cơ sở hạ tầng công nghệ hiện đại. Việc tích hợp các hệ thống định vị, thanh toán trực tuyến và lưu trữ dữ liệu đòi hỏi hạ tầng mạng vững chắc, đồng thời phải luôn được cập nhật để chống lại các mối đe dọa từ tấn công mạng và các sự cố kỹ thuật.

Quy định pháp lý và quản lý: Các quy định pháp lý liên quan đến vận tải hành khách, cấp phép hoạt động, bảo hiểm và các vấn đề an toàn giao thông đòi hỏi hệ thống phải luôn tuân thủ. Nhà quản lý cần nắm vững và cập nhật thường xuyên các quy định pháp luật để tránh rủi ro pháp lý và bảo đảm hoạt động kinh doanh diễn ra suôn sẻ.

Tuân thủ quy định pháp luật: Việc thực hiện đầy đủ và nghiêm túc các quy định pháp luật không chỉ giúp tăng tính minh bạch mà còn xây dựng lòng tin của khách hàng và đối tác. Điều này bao gồm cả việc đảm bảo các tiêu chuẩn về an toàn, bảo vệ quyền lợi người dùng và thực hiện đúng các thủ tục hành chính theo quy định.

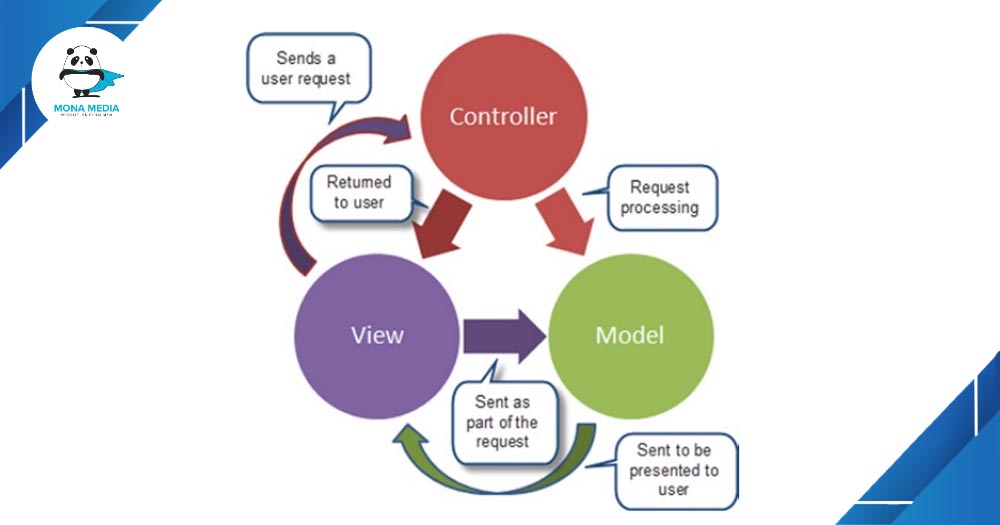
Tính an toàn: An toàn luôn là yếu tố hàng đầu trong dịch vụ vận tải. Hệ thống cần đảm bảo các tiêu chuẩn an toàn cho cả hành khách và tài xế thông qua đào tạo, giám sát liên tục và sử dụng các công nghệ tiên tiến nhằm theo dõi tình trạng xe và xử lý sự cố một cách nhanh chóng, hiệu quả.

## 2.3 Các công nghệ và công cụ sử dụng

### 2.3.1. Mô hình MVC

MVC (Model-View-Controller) là một mẫu kiến trúc phần mềm được sử dụng rộng rãi trong việc phát triển ứng dụng để tạo ra các giao diện người dùng trực quan và có khả năng tương tác cao. MVC chia một ứng dụng thành 3 phần chính và mỗi phần có một vai trò riêng biệt:

* Model đại diện cho dữ liệu và quy tắc nghiệp vụ của ứng dụng.
* View chịu trách nhiệm hiển thị dữ liệu cho người dùng một cách trực quan và tương tác.
* Controller đóng vai trò là cầu nối giữa Model và View, xử lý các yêu cầu từ người dùng và cập nhật giao diện tương ứng.



Hình 2.1: Thành phần của mô hình MVC

Lợi ích của việc sử dụng mô hình MVC:

* Phân tách rõ ràng: Giúp tách biệt logic nghiệp vụ, giao diện người dùng và xử lý đầu vào, tạo sự rõ ràng trong cấu trúc ứng dụng.
* Dễ bảo trì và mở rộng: Mỗi thành phần có thể được cập nhật hoặc thay thế độc lập mà không ảnh hưởng đến các phần khác.
* Hỗ trợ phát triển đồng thời: Nhiều lập trình viên có thể làm việc trên các thành phần khác nhau cùng lúc, tăng hiệu suất phát triển.

Ưu và nhược điểm của mô hình MVC:

|  |  |
| --- | --- |
| Ưu điểm | Nhược điểm |
| Tách biệt rõ ràng giữa các thành phần, giúp mã nguồn dễ bảo trì và mở rộng.  Hỗ trợ phát triển ứng dụng theo hướng module, cho phép nhiều lập trình viên làm việc đồng thời trên các phần khác nhau.  Dễ dàng thay thế hoặc cập nhật từng thành phần mà không ảnh hưởng đến toàn bộ hệ thống. | Đối với ứng dụng nhỏ, việc áp dụng MVC có thể gây thêm phức tạp không cần thiết.  Cần có sự hiểu biết sâu về kiến trúc MVC để triển khai hiệu quả. |

Bảng 2.1: Ưu nhược điểm của mô hình MVC

### 2.3.2. Backend API

Node.js

Node.js là một nền tảng chạy JavaScript phía máy chủ, ra đời nhằm mở rộng khả năng của JavaScript ra ngoài trình duyệt. Việc sử dụng JavaScript cho cả frontend và backend giúp đơn giản hóa quy trình phát triển, tăng hiệu quả và giảm độ phức tạp của ứng dụng web.



Hình 2.2: Node.JS logo

Kiến trúc và hoạt động

Node.js hoạt động dựa trên mô hình Event-Driven (hướng sự kiện) và Non-blocking I/O (bất đồng bộ), cho phép xử lý hàng loạt tác vụ mà không làm nghẽn luồng chính. Với kiến trúc single-threaded event loop, Node.js chỉ sử dụng một luồng chính để xử lý, nhưng vẫn có thể quản lý hàng nghìn kết nối đồng thời mà không làm tăng chi phí tài nguyên.

Node.js sử dụng V8 Engine – bộ máy JavaScript của Google Chrome – giúp biên dịch mã JavaScript thành mã máy nhanh chóng qua cơ chế Just-In-Time (JIT). Nhờ đó, Node.js đạt hiệu suất cao, phù hợp với các ứng dụng cần phản hồi nhanh như trò chuyện trực tuyến, hệ thống theo dõi hoặc API tốc độ cao.

Hệ sinh thái và npm

Node.js đi kèm với npm (Node Package Manager) – hệ thống quản lý gói lớn nhất thế giới. Lập trình viên có thể dễ dàng tái sử dụng các thư viện như Express.js, Socket.IO, hoặc Axios để tăng tốc phát triển ứng dụng. npm còn hỗ trợ kiểm soát phiên bản, quét bảo mật (npm audit), và tích hợp CI/CD hiệu quả.

Khả năng mở rộng

Node.js hỗ trợ mở rộng theo hai hướng:

* Chiều ngang: Sử dụng Cluster Module để khai thác đa lõi CPU hoặc triển khai theo kiến trúc Microservices trên các nền tảng như AWS hoặc Google Cloud.
* Chiều dọc: Tối ưu hiệu suất ứng dụng bằng cách tối ưu mã nguồn, phân tách chức năng hợp lý.

Node.js cũng tương thích tốt với các công nghệ hiện đại như Docker, Kubernetes, cho phép container hóa ứng dụng và mở rộng linh hoạt.

Ứng dụng thực tiễn

Node.js phù hợp với nhiều loại ứng dụng yêu cầu xử lý thời gian thực và hiệu năng cao:

* Ứng dụng chat: Sử dụng WebSocket và Socket.IO để giao tiếp theo thời gian thực.
* Thông báo thời gian thực: Theo dõi đơn hàng, vận chuyển, push notifications.
* Dịch vụ phát trực tuyến: Truyền tải media hiệu quả bằng Stream API.
* API backend: Kết hợp với Express.js để xây dựng các RESTful API hoặc GraphQL.
* Thương mại điện tử: Xử lý đơn hàng, thanh toán và giỏ hàng động.

Node.js hiện đang được ứng dụng rộng rãi tại các tập đoàn lớn như Netflix, PayPal, Walmart, LinkedIn... nhờ khả năng tối ưu tài nguyên và nâng cao hiệu suất hệ thống.

Express.js

Express.js là một framework web linh hoạt và nhẹ, được xây dựng trên nền tảng Node.js, giúp đơn giản hóa quá trình phát triển các ứng dụng web và API. Được phát triển bởi TJ Holowaychuk từ năm 2010, Express cung cấp cấu trúc tối giản, cho phép lập trình viên xây dựng ứng dụng phía máy chủ hiệu quả với ít mã hơn nhưng vẫn đảm bảo hiệu suất và khả năng mở rộng.

Một trong những đặc điểm nổi bật của Express là hệ thống middleware, cho phép xử lý các tác vụ như xác thực, ghi log, xử lý lỗi... trước khi gửi phản hồi về phía người dùng. Ngoài ra, Express hỗ trợ hệ thống routing mạnh mẽ để quản lý và định tuyến các yêu cầu HTTP một cách linh hoạt.

Framework này cũng dễ dàng tích hợp với các template engines như Pug, EJS hoặc Handlebars để tạo ra giao diện động. Hơn thế nữa, Express có thể kết hợp hiệu quả với các thư viện khác trong hệ sinh thái Node.js như MongoDB (xử lý cơ sở dữ liệu), JWT (xác thực người dùng), và đặc biệt là Socket.IO trong các ứng dụng thời gian thực.

Express.js thường được sử dụng để xây dựng:

* Các API RESTful nhờ khả năng xử lý dữ liệu JSON dễ dàng.
* Các ứng dụng web động có giao diện người dùng.
* Các hệ thống giao tiếp thời gian thực như ứng dụng đặt xe, hệ thống chat hay giám sát.

### 2.3.4. Tích hợp thơi gian thật

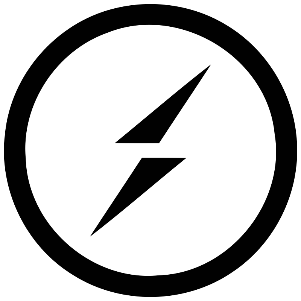
WebSocket

**WebSocket** là một giao thức mạng hỗ trợ kết nối hai chiều liên tục giữa client và server. Không giống với giao thức HTTP truyền thống vốn dựa trên mô hình yêu cầu - phản hồi, WebSocket cho phép cả hai bên chủ động gửi và nhận dữ liệu bất kỳ lúc nào sau khi kết nối được thiết lập. Điều này giúp giảm độ trễ và tối ưu hiệu suất truyền tải trong các ứng dụng thời gian thực.

WebSocket thường được ứng dụng trong các hệ thống trò chuyện trực tuyến, cập nhật thông báo, giám sát thiết bị, hoặc trong trò chơi cần phản hồi nhanh chóng.

Socket.IO

**Socket.IO** là một thư viện JavaScript xây dựng trên WebSocket, hỗ trợ thêm các tính năng cần thiết để phát triển ứng dụng thời gian thực như tự động kết nối lại, phân kênh (rooms), truyền dữ liệu nhị phân, và tương thích với nhiều trình duyệt.



Hình 2.4: Socket.io logo

Socket.IO được sử dụng rộng rãi để xây dựng các ứng dụng như:

* Hệ thống nhắn tin trực tuyến
* Game đa người chơi (multiplayer)
* Cập nhật dữ liệu liên tục (giá xe, vị trí tài xế…)
* Công cụ giám sát và dashboard theo thời gian thực

### 2.3.5. Cơ sở liệu

XAMPP

XAMPP là một gói phần mềm miễn phí và mã nguồn mở, bao gồm Apache (máy chủ web), MariaDB (hoặc MySQL), PHP và Perl. Nó được thiết kế để giúp các nhà phát triển nhanh chóng thiết lập một máy chủ web cục bộ, lý tưởng để kiểm tra và phát triển ứng dụng trước khi triển khai lên máy chủ thực tế. XAMPP tương thích với nhiều nền tảng như Windows, Linux và macOS, và thường đi kèm với phpMyAdmin để quản lý cơ sở dữ liệu dễ dàng.



Hình 2.5: XAMPP logo

MySQL

MySQL là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) mã nguồn mở, được phát triển bởi Oracle Corporation. Nó sử dụng ngôn ngữ SQL để tạo, truy xuất và quản lý dữ liệu, tổ chức thông tin thành các bảng có mối quan hệ, giúp dễ dàng quản lý và truy vấn. MySQL được biết đến với tốc độ nhanh, độ tin cậy cao và khả năng mở rộng, thường được sử dụng trong các ứng dụng web như trang thương mại điện tử hoặc hệ thống doanh nghiệp lớn.



Hình 2.6: MySQL logo

Cách XAMPP và MySQL hoạt động cùng nhau:

* 1. Cài đặt và khởi động XAMPP:
* Sau khi tải và cài đặt XAMPP, bạn có thể sử dụng XAMPP Control Panel để quản lý các dịch vụ.​
* Mở XAMPP Control Panel và khởi động các module cần thiết như Apache và MySQL bằng cách nhấp vào nút "Start" tương ứng.​
  1. Truy cập phpMyAdmin:
* Sau khi khởi động MySQL, bạn có thể quản lý cơ sở dữ liệu thông qua phpMyAdmin.​
* Mở trình duyệt web và nhập địa chỉ http://localhost/phpmyadmin để truy cập giao diện quản lý phpMyAdmin.​
  1. Tạo và quản lý cơ sở dữ liệu:
* Trong phpMyAdmin, bạn có thể tạo cơ sở dữ liệu mới bằng cách:​
  + Nhấp vào tab "Databases".
  + Nhập tên cơ sở dữ liệu vào ô "Create database" và nhấn "Create".
* Sau khi tạo cơ sở dữ liệu, bạn có thể tạo bảng, chèn dữ liệu và thực hiện các truy vấn SQL thông qua phpMyAdmin.​
  1. Kết nối PHP với MySQL:
* Để ứng dụng PHP của bạn có thể tương tác với cơ sở dữ liệu MySQL trong XAMPP, bạn cần sử dụng các hàm kết nối như MySQLi hoặc PDO.

### 2.3.6. Tích hợp bản đồ

OpenMap GL VN là một thư viện mã nguồn mở được phát triển từ MapLibre GL JS – bản phân nhánh tự do của Mapbox GL JS – nhằm phục vụ nhu cầu hiển thị bản đồ tương tác trên nền tảng web tại Việt Nam. Thư viện này được nội địa hóa với dữ liệu địa lý, địa danh và giao diện tiếng Việt, giúp tăng tính tương thích với các ứng dụng bản đồ trong nước.

Sử dụng công nghệ WebGL, OpenMap GL VN cho phép kết xuất bản đồ vector hiệu năng cao, mang lại trải nghiệm mượt mà và trực quan với các thao tác như phóng to, thu nhỏ hoặc xoay bản đồ. Thư viện hỗ trợ nhiều lớp bản đồ khác nhau như bản đồ hành chính, giao thông, địa hình và có khả năng tùy biến linh hoạt qua JavaScript.

Việc tích hợp OpenMap GL VN vào website khá đơn giản, chỉ cần thêm các tệp thư viện JavaScript và CSS, sau đó cấu hình đối tượng bản đồ với các thông số như tọa độ, mức zoom hoặc kiểu bản đồ. Điều này tạo điều kiện thuận lợi cho việc xây dựng các ứng dụng web định vị, thống kê hoặc phân tích theo khu vực.

Ứng dụng thực tiễn của OpenMap GL VN rất đa dạng, bao gồm:

* Hệ thống bản đồ hành chính hiển thị ranh giới xã, huyện, tỉnh.
* Quy hoạch hạ tầng đô thị, giao thông và điện lực.
* Ứng dụng GIS trong doanh nghiệp, cơ quan quản lý nhà nước.
* Các nền tảng web hoặc di động liên quan đến định vị, dẫn đường, thống kê vị trí.

Nhờ tính mở và khả năng tùy biến cao, OpenMap GL VN là lựa chọn phù hợp cho các tổ chức trong nước muốn xây dựng hệ thống bản đồ riêng mà không phụ thuộc vào các dịch vụ bản đồ quốc tế.

### 2.3.7 Các Công Cụ Khác

Git & GitHub

Git là một hệ thống quản lý phiên bản phân tán (Distributed Version Control System – DVCS) phổ biến, cho phép theo dõi mọi thay đổi của mã nguồn trong quá trình phát triển phần mềm. Nhờ Git, lập trình viên có thể:

* Ghi lại lịch sử thay đổi theo từng lần commit.
* Quản lý nhiều nhánh phát triển (branch) độc lập.
* Dễ dàng hoàn tác hoặc so sánh các phiên bản mã.

GitHub là một nền tảng lưu trữ mã nguồn trực tuyến dựa trên Git, cung cấp các công cụ cộng tác như:

* Pull Request: Cho phép xem xét và hợp nhất các thay đổi từ các nhánh khác nhau.
* Issue & Project Boards: Quản lý lỗi và tiến độ công việc nhóm.
* Actions: Tích hợp CI/CD để tự động hóa kiểm thử và triển khai ứng dụng.

Sự kết hợp giữa Git và GitHub giúp các nhóm phát triển phối hợp hiệu quả, đảm bảo tính nhất quán, an toàn và dễ kiểm soát trong quá trình làm việc nhóm, đặc biệt trong các dự án phần mềm quy mô lớn.

**Thanh toán MoMo**

MoMo là một **ví điện tử phổ biến tại Việt Nam**, được tích hợp trong nhiều ứng dụng để hỗ trợ người dùng thanh toán nhanh chóng và tiện lợi. Trong ứng dụng web, MoMo có thể được tích hợp thông qua **API cổng thanh toán MoMo** cho phép:

* Tạo yêu cầu thanh toán (payment request) với thông tin đơn hàng.
* Chuyển hướng người dùng đến giao diện MoMo để xác nhận thanh toán.
* Nhận thông báo kết quả giao dịch thông qua webhook để xử lý đơn hàng.

Tích hợp MoMo giúp đơn giản hóa quá trình thanh toán, tăng trải nghiệm người dùng và phù hợp với thị trường nội địa.

AWS SNS (Simple Notification Service)

AWS SNS là một dịch vụ nhắn tin theo mô hình pub/sub (publish-subscribe) từ Amazon, cho phép gửi tin nhắn (SMS) đến người dùng nhanh chóng. Trong ứng dụng web, SNS thường được sử dụng để gửi mã xác thực OTP (One-Time Password) qua SMS với mục đích:

* Xác thực đăng nhập hoặc đăng ký tài khoản.
* Bảo vệ truy cập thông tin cá nhân và giao dịch.

Quy trình gửi OTP qua AWS SNS:

* 1. Hệ thống tạo mã OTP ngẫu nhiên.
  2. Sử dụng AWS SDK để gửi OTP đến số điện thoại người dùng.
  3. Người dùng nhập lại mã để xác minh.

AWS SNS đảm bảo độ tin cậy cao, thời gian gửi nhanh và khả năng mở rộng khi cần gửi hàng loạt OTP trong các hệ thống quy mô lớn.

# PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

## 3.1. Khảo sát và xây dựng đặc tả

Tổng quan thị trường:

Trong những năm gần đây, thị trường dịch vụ gọi xe trực tuyến tại Việt Nam đã trở thành điểm nóng với sự tăng trưởng mạnh mẽ. Theo số liệu từ các báo cáo, vào năm 2024, thị trường gọi xe của Việt Nam được định giá khoảng 1,7 tỷ đô la với tốc độ tăng trưởng đạt 9%. Sự phục hồi nhanh chóng sau đại dịch COVID-19 đã giúp ngành phát triển mạnh, với tỷ lệ tăng trưởng kép (CAGR) trên 8%, từ mức 1,3 tỷ USD vào năm 2021 được dự báo sẽ tăng lên khoảng 2,5 tỷ USD vào năm 2029.

Số lượng người dùng dịch vụ gọi xe cũng tăng đều đặn, từ 22,8 triệu vào năm 2017 lên 28,1 triệu vào năm 2024, và dự kiến đạt khoảng 37 triệu người dùng vào năm 2029, tương ứng với tỷ lệ thâm nhập đạt 36%. Đặc biệt, các thành phố lớn như Thành phố Hồ Chí Minh và Hà Nội là những trung tâm phát triển chính của ngành, nơi mà dịch vụ gọi xe trực tuyến đã trở thành một phần không thể thiếu trong đời sống hàng ngày của người dân.

Những số liệu này cho thấy rằng, ngoài việc cải thiện trải nghiệm người dùng, thị trường đặt xe taxi trực tuyến còn mở ra nhiều cơ hội kinh doanh và phát triển, đòi hỏi các nhà quản lý và nhà phát triển cần nắm bắt xu hướng để xây dựng hệ thống đáp ứng đúng nhu cầu và tạo ra giá trị gia tăng cho cả khách hàng lẫn các đối tác vận tải.

Khảo sát người sử dụng dịch vụ đặt xe taxi

Theo các khảo sát và báo cáo thị trường gần đây, người dùng tại các đô thị lớn như Hà Nội và TP.HCM ngày càng ưa chuộng dịch vụ đặt xe taxi trực tuyến. Khoảng 80-85% người dùng đã chuyển sang sử dụng các ứng dụng đặt xe thay vì gọi xe truyền thống. Họ đánh giá cao tính tiện lợi, khả năng theo dõi vị trí tài xế theo thời gian thực và hệ thống thanh toán trực tuyến nhanh chóng. Ngoài ra, mức độ hài lòng chung của người dùng được ghi nhận ở mức trên 85%, nhờ giao diện thân thiện và thông tin giá cước minh bạch. Một số phản hồi cho biết, mặc dù dịch vụ thường đáp ứng tốt nhu cầu di chuyển, nhưng vẫn có một số bất cập về độ ổn định khi xử lý lượng yêu cầu lớn vào giờ cao điểm và vấn đề bảo mật thông tin cá nhân. Những kết quả khảo sát này cho thấy, dịch vụ đặt xe taxi trực tuyến đang phát triển mạnh mẽ và có tiềm năng mở rộng thị trường trong tương lai.

### 3.1.1. Xác định các đối tượng

Đối tượng "Khách hàng"

Thuộc tính chính:

* Mã khách hàng: Mã định danh duy nhất cho mỗi khách hàng.
* Tên khách hàng: Tên của khách hàng.
* Số điện thoại, email và địa chỉ: Thông tin liên hệ để liên lạc và gửi thông báo.
* Lịch sử đặt xe: Danh sách các chuyến đi đã thực hiện.

Phương thức hoạt động:

Hệ thống cho phép khách hàng đăng ký và đăng nhập tài khoản, sau đó thực hiện đặt xe theo thời gian thực. Khi đặt xe, khách hàng có thể nhập điểm đón, điểm đến và chọn loại xe phù hợp. Hệ thống sẽ theo dõi hành trình và lưu trữ thông tin chuyến đi. Ngoài ra, khách hàng cũng có thể thanh toán trực tuyến qua các phương thức như ví điện tử hoặc thẻ ngân hàng, đồng thời đánh giá chất lượng dịch vụ sau mỗi chuyến đi.

Đối tượng "Tài xế"

Thuộc tính chính:

* Mã tài xế: Mã định danh duy nhất cho mỗi tài xế.
* Tên tài xế và thông tin xe: Bao gồm tên, biển số xe, loại xe và tình trạng hoạt động (sẵn sàng, đang bận).
* Vị trí hiện tại: Thông tin định vị giúp hệ thống cập nhật lộ trình và định vị xe theo thời gian thực.
* Lịch sử chuyến đi: Danh sách các chuyến xe đã nhận và hoàn thành.

Phương thức hoạt động:

Tài xế đăng nhập vào hệ thống để nhận thông báo đặt xe. Khi có đơn đặt, tài xế có thể chấp nhận chuyến đi, sau đó cập nhật trạng thái (đang di chuyển, đã đến điểm đón, hoàn thành, hoặc hủy). Hệ thống cũng lưu lại thông tin lịch sử và phản hồi từ khách hàng, giúp đánh giá hiệu quả và chất lượng dịch vụ của tài xế.

## 3.2 Thiết kế sơ đồ hệ thống

### 3.2.1 Sơ đồ phần cấp chức năng

A screenshot of a computer screen

Description automatically generated

Hình 3.1: Sơ đồ chức năng của hệ thống

### Sơ đồ usecase

1. Usecase đăng nhập

A diagram of a person's work flow

Description automatically generated

Hình 3.2: Sơ đồ use case đăng nhập

Đăc tả usecase đăng nhập

|  |  |  |  |
| --- | --- | --- | --- |
| Mã Usecase | UC001 | | |
| Tên Usecase | Use case đăng nhập | | |
| Tác nhân | Khách hàng, tài xế và quản trị viên | | |
| Mô tả | Cho phép người có tài khoản cập vào hệ thống | | |
| Điều kiện tiên quyết | Người dùng chưa đăng nhập vào hệ thống | | |
| Luồng sự kiện | STT | Tác nhân | Phản hồi |
| 1 | Mở trang đăng nhập | Hiện ra giao diện |
| 2 | nhập thông tin cần thiết |  |
| 3 | Nhấn nút đăng nhập | Kiểm tra thông tin đăng nhập. |
| 4 |  | Nếu là **Client**, chuyển đến trang **Home (Client)**. |
| 5 |  | Nếu là **Driver**, chuyển đến trang **Home (Driver)**. |
| 6 |  | Nếu là **Admin**, chuyển đến **Dashboard (Admin)**. |
| Luồng ngoại lệ |  |  | Báo lỗi khi không tìm thấy dữ liệu |
| Điều kiện | Đăng nhập thành công | | |

Bảng 3.1: Đặc tả sơ đồ usecase đăng nhập

1. Usecae quản lý tài khoản

A diagram with text and circles

Description automatically generated with medium confidence

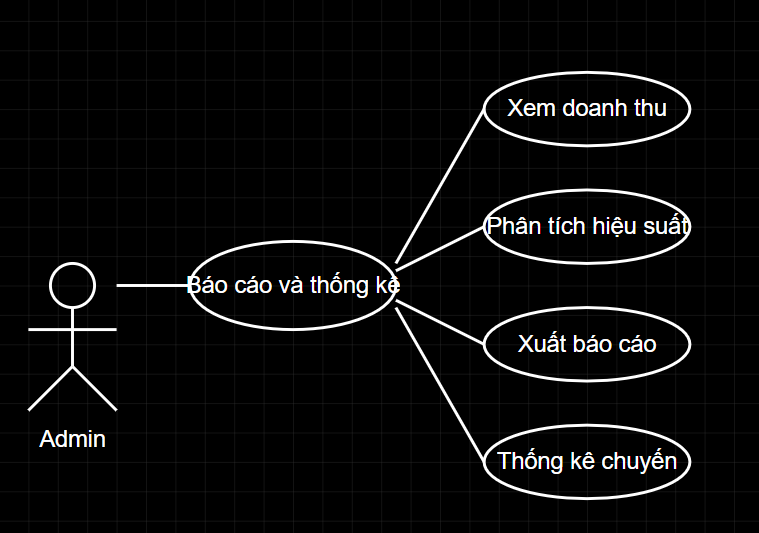
Hình 3.3: Sơ đồ use case quản lý / phân quyền tài khoản

Đặc tả sơ đồ usecase quản lý tài khoản

|  |  |  |  |
| --- | --- | --- | --- |
| Mã Usecase | UC002 | | |
| Tên Usecase | Use case quản lý / phân quyền tài khoản | | |
| Tác nhân | Quản trị viên (Admin) | | |
| Mô tả | Cho phép quản trị viên quản lý tài khoản và phân quyền | | |
| Điều kiện tiên quyết | Hệ thống kết nối thành công đến cơ sở dữ liệu. | | |
| Luồng sự kiện | STT | Tác nhân | Phản hồi |
| 1 | Admin chọn “Quản lý tài khoản” | Hiện ra giao diện |
| 2 |  | Hiển thị các tùy chọn quản lý:  Tìm kiếm tài khoản  Thêm tài khoản  Xóa tài khoản  Thay đổi vai trò |
| 3 | Admin sử dụng chức năng theo nhu cầu |  |
| 4 |  | Nếu sử dụng chức năng “Thay đổi vai trò” được thực hiện luồng sự kiện con |
| 5 | Sau khi kết thúc quá trình, admin được đưa lại giao diện quản lý |  |
| 6 |  | Hệ thống hiện thay đổi và tiếp tục sử dụng |
| Luồng ngoại lệ |  |  |  |
| Điều kiện | Tài khoản và vai trò được cập nhật trong cơ sở dữ liệu | | |

Bảng 3.2: Đặc tả sơ đồ usecase quản lý / phân quyền tài khoản

Usecae báo cáo thống kê



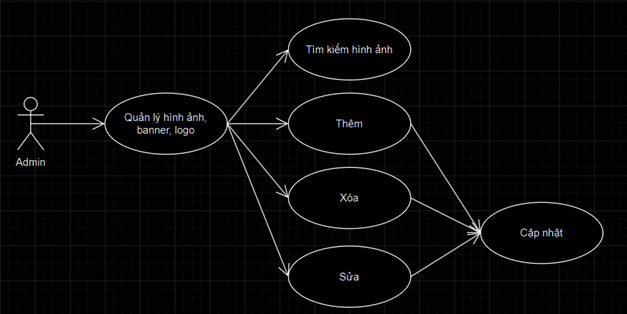
Hình 3.4: Sơ đồ use case báo cáo thống kê

Đặc tả usecase báo cáo thống kê

|  |  |  |  |
| --- | --- | --- | --- |
| Mã Usecase | UC003 | | |
| Tên Usecase | Use case báo cáo thống kê | | |
| Tác nhân | Quản trị viên (Admin) | | |
| Mô tả | Cho phép Admin tạo và quản lý các báo cáo dựa trên dữ liệu thống kê | | |
| Điều kiện tiên quyết | Hệ thống có sẵn dữ liệu cập nhật để phục vụ cho việc báo cáo. | | |
| Luồng sự kiện | STT | Tác nhân | Phản hồi |
| 1 | Admin chọn chức năng báo cáo | Hiển thị giao diện module báo cáo. |
| 2 | Chọn một hoặc nhiều tùy chọn từ danh sách | Hiển thị các tác vụ:  Xem doanh thu.  Phân tích hiệu suất  Xuất báo cáo  Thống kê chuyên |
| 3 | Xem xét kết quả, có thể điều chỉnh tham số hoặc thực hiện thêm tác vụ nếu cần | Cập nhật kết quả theo yêu cầu (nếu có) |
| Luồng ngoại lệ |  |  | Dữ liệu cần thiết không có sẵn hoặc bị lỗi |
| Điều kiện | Báo cáo được lưu trữ hoặc gửi đi | | |

Bảng 3.3: Đặc tả sơ đồ usecase báo cáo thống kê

Usecase quản lý hình ảnh, banner và logo



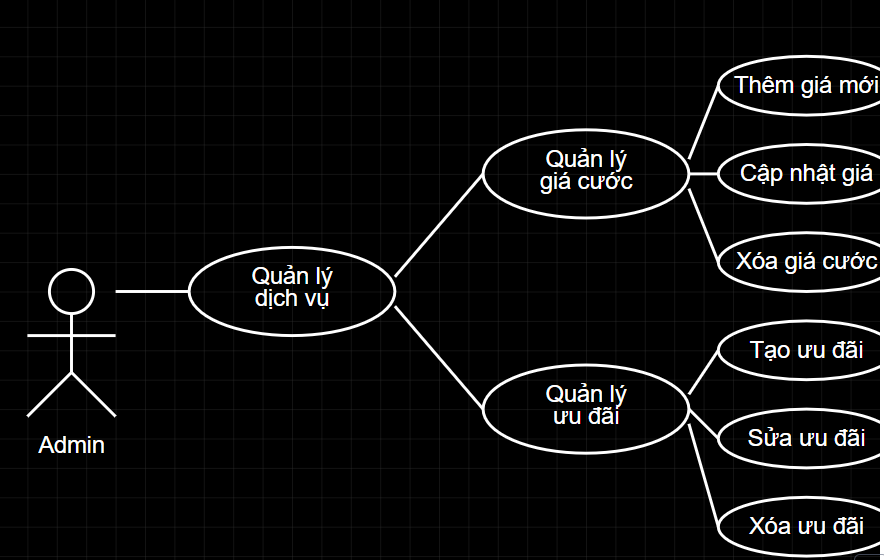
Hình 3.5: Sơ đồ use case quản lý hình ảnh, banner và logo

Đặc tả use case quản lý hình ảnh, banner, logo

|  |  |  |  |
| --- | --- | --- | --- |
| Mã Usecase | UC004 | | |
| Tên Usecase | Use case quản lý hình ảnh, banner và logo | | |
| Tác nhân | Quản trị viên (Admin) | | |
| Mô tả | Cho phép Admin tạo và quản lý các tài sản hình ảnh | | |
| Điều kiện tiên quyết | Hệ thống có thiết lập kho chứa tài sản ảnh | | |
| Luồng sự kiện | STT | Tác nhân | Phản hồi |
| 1 | Admin truy cập chức năng quản lý hình ảnh |  |
| 2 |  | Hiển thị các tác vụ:  Tìm kiếm hình ảnh  Thêm, xóa sửa, hình ảnh |
| 3 | Admin sẻ thực hiện chọn tác vụ cần thực hiện |  |
|  | 4 |  | Nếu chọn "Thêm hình ảnh" thực hiện luồng sự kiện của use case con.  Nếu chọn các tác vụ còn lại sẻ thục hiện tương ứng |
| 5 | Admin chọn cập nhật những thay đổi | Hệ thống sẻ ghi đè lên file cũ thành file mới |
| Luồng ngoại lệ |  |  |  |
| Điều kiện | Dữ liệu hình ảnh được cập nhật kịp thời và chính xác. | | |

Bảng 3.4: Đặc tả sơ đồ usecase quản lý hình ảnh, banner và logo

Usecase quản lý dịch vụ



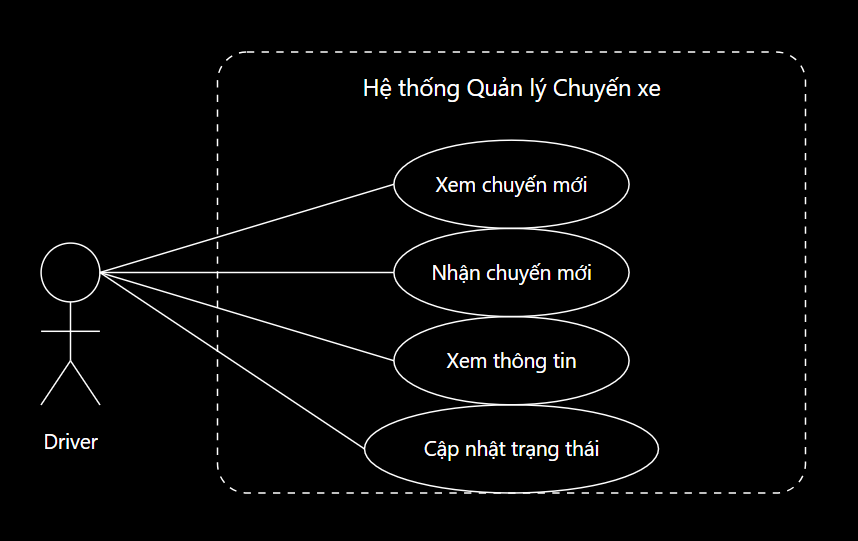
Hình 3.6: Sơ đồ use case quản lý dịch vụ

Đặc tả usecase quản lý dịch vụ

|  |  |  |  |
| --- | --- | --- | --- |
| Mã Usecase | UC005 | | |
| Tên Usecase | Use case quản lý dịch vụ | | |
| Tác nhân | Quản trị viên (Admin) | | |
| Mô tả | Cho phép Admin thực hiện các thao tác quản lý quản lý giá cước và ưu đãi | | |
| Điều kiện tiên quyết | Admim có quyền quản trị hệ thống | | |
| Luồng sự kiện | STT | Tác nhân | Phản hồi |
| 1 | Admin truy cập chức năng quản lý dịch vụ | Hiển thị giao diện có 2 phân hệ: "Giá cước", "Ưu đãi" |
| 2 | Chọn “Quản lý giá cước” | Hiển thị danh sách giá hiện tại và các tùy chọn thao tác:  Thêm giá mới  Xóa giá cước  Cập nhật giá mới |
| 3 | Admin thực hiện chọn tác vụ cần thực hiện |  |
| 4 |  | Nếu chọn "Thêm giá cước" thực hiện luồng sự kiện của use case con.  Nếu chọn các tác vụ còn lại sẻ thục hiện tương ứng |
| 5 | Chọn “Quản lý ưu đãi” | Hiển thị danh sách giá hiện tại và các tùy chọn thao tác:  Tạo ưu đãi  Sửa ưu đãi  Xóa ưu đãi |
| 6 | Admin làm việc với chức năng của use case được chọn |  |
| 7 | Chọn “Quản lý giá cước” |  |
| 8 |  | Nếu chọn "Thêm giá cước" thực hiện luồng sự kiện của use case con.  Nếu chọn các tác vụ còn lại sẻ thục hiện tương ứng |
| 9 |  | Hệ thống luôn sẵn sàng để tiếp tục thực hiện các chức năng khác. |
| Luồng ngoại lệ |  |  |  |
| Điều kiện | Giá cước/ưu đãi được cập nhật thành công | | |

Bảng 3.5: Sơ đồ use case quản lý dịch vụ

Usecase quản lý đơn chuyến



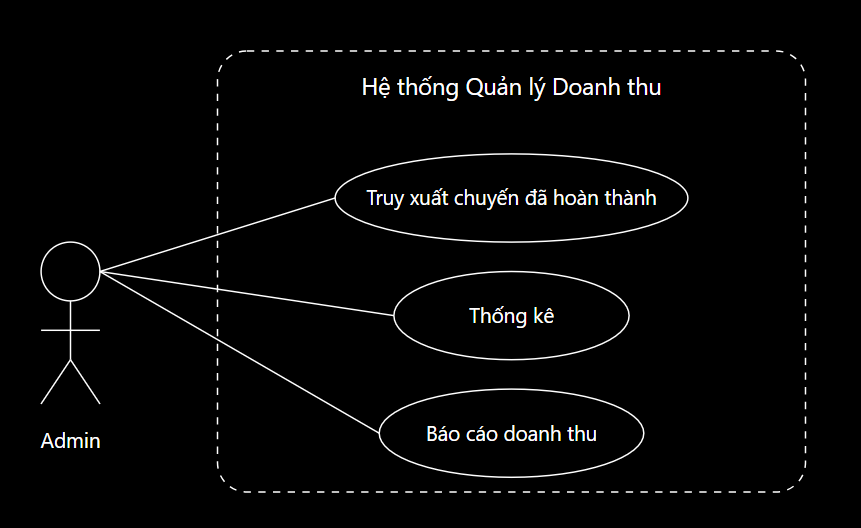
Hình 3.7: Sơ đồ use case quản lý đơn chuyến

Đặc tả usecase quản lý đơn chuyến

|  |  |  |  |
| --- | --- | --- | --- |
| Mã Usecase | UC006 | | |
| Tên Usecase | Use case quản lý đơn chuyến | | |
| Tác nhân | Tài xế (Driver) | | |
| Mô tả | Cho phép tài xế quản lý các đơn chuyến | | |
| Điều kiện tiên quyết | Tài xế có quyền truy cập vào module  Hệ thống đã có dữ liệu về các đơn chuyến (mới hoặc đang quản lý). | | |
| Luồng sự kiện | STT | Tác nhân | Phản hồi |
| 1 | Tài xế truy cập chức năng quản lý đơn chuyến |  |
| 2 |  | Hiển thị giao diện quản lý đơn chuyến với các tùy chọn:  Xem chuyến mới  Nhận chuyến mới  Xem thông tin  Cập nhật trạng thái. |
| 3 | Tài xế thực hiện chọn tác vụ cần thực hiện |  |
| 4 |  | Nếu chọn " Nhận chuyến mới" thực hiện luồng sự kiện của use case con.  Nếu chọn các tác vụ còn lại sẻ thục hiện tương ứng |
| 5 |  | Hệ thống luôn sẵn sàng để tiếp tục thực hiện các chức năng khác. |
| Luồng ngoại lệ |  |  |  |
| Điều kiện | Dữ liệu đơn chuyến đã được tài xế cập nhật trong hệ thống | | |

Bảng 3.6: Bảng đặc tả usecase quản lý đơn chuyến

Usecase theo dõi doanh thu



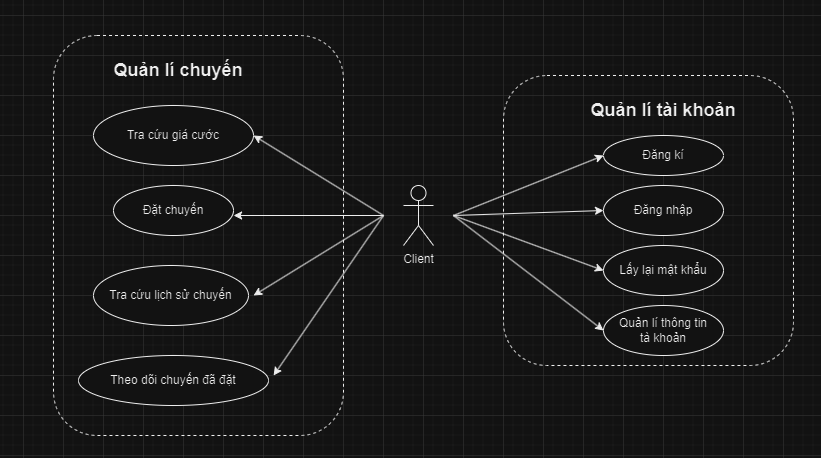
Hình 3.8: Sơ đồ use case theo dõi doanh thu

Đặc tả usecase theo dõi doanh thu

|  |  |  |  |
| --- | --- | --- | --- |
| Mã Usecase | UC007 | | |
| Tên Usecase | Use case theo dõi doanh thu | | |
| Tác nhân | Admin (Quản trị viên) | | |
| Mô tả | Cho phép Admin theo dõi doanh thu từ các đơn chuyến trên hệ thống đặt xe taxi | | |
| Điều kiện tiên quyết | Admin có quyền truy cập vào module "Theo dõi doanh thu" và "Quản lý đơn chuyến".  Hệ thống đã có dữ liệu về các đơn chuyến | | |
| Luồng sự kiện | STT | Tác nhân | Phản hồi |
| 1 | Admin truy cập module “Theo dõi doanh thu” |  |
| 2 |  | Hiển thị giao diện theo dõi doanh thu với các tùy chọn:  Truy xuất chuyễn đã hoàn thành  Thống kê  Báo cáo doanh thu |
| 3 | Admin thực hiện chọn tác vụ cần thực hiện |  |
| 4 |  | Nếu chọn " Truy xuất chuyễn đã hoàn thành" thực hiện luồng sự kiện của use case con.  Nếu chọn các tác vụ còn lại sẻ thục hiện tương ứng |
| 5 |  | Hệ thống luôn sẵn sàng để tiếp tục thực hiện các chức năng khác. |
| Luồng ngoại lệ |  |  |  |
| Điều kiện | Hệ thống hiển thị đầy đủ dữ liệu  Thao tác của Admin được ghi log (nếu cần). | | |

Bảng 3.7: Bảng đặc tả usecase quản lý đơn chuyến

Use case của người đặt xe

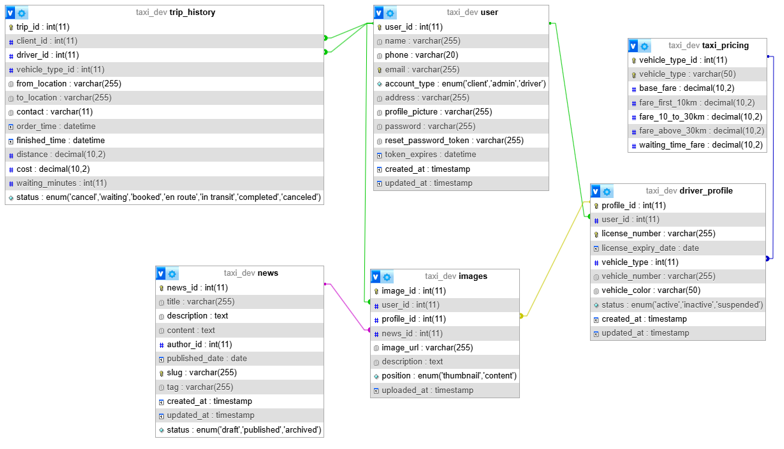


Hình 3.9: Sơ đồ use case của người đặt xe

|  |  |  |  |
| --- | --- | --- | --- |
| Mã Usecase | UC008 | | |
| Tên Usecase | Use case của người đặt xe | | |
| Tác nhân | Khách hàng (Client) | | |
| Mô tả | Cho phép Client sử dụng các chức năng trong hệ thống đặt xe trực tuyến cho khách hàng | | |
| Điều kiện tiên quyết | Người đặt xe phải có quyền truy cập vào hệ thống qua website  Hệ thống có thiết lập sẵn với các dịch vụ xe và phương thức thanh toán để người dùng sử dụng. | | |
| Luồng sự kiện | STT | Tác nhân | Phản hồi |
| 1 | Nếu Client truy cập module "Quản lý chuyến" |  |
| 2 |  | Hiển thị các tùy chọn:  Đặt chuyến  Theo dõi chuyến đã đặt  Tra cứu lịch sử chuyến"  Trả góp giá cước" |
| 3 | Client thực hiện chọn tác vụ cần thực hiện |  |
| 4 |  | Nếu chọn "Đặt chuyến" thực hiện luồng sự kiện của use case con.  Nếu chọn các tác vụ còn lại sẻ thục hiện tương ứng |
| 5 | Nếu Client truy cập module "Quản tài khoản" |  |
| 6 |  | Hiển thị các tùy chọn:  Đăng ký  Đăng nhập  Quản lý thông tin tài khoản  Lấy lại mật khẩu |
| 7 | Client thực hiện chọn tác vụ cần thực hiện |  |
| 8 |  | Nếu chọn "Đăng ký" thực hiện luồng sự kiện của use case con.  Nếu chọn các tác vụ còn lại sẻ thục hiện tương ứng |
| Luồng ngoại lệ |  |  |  |
| Điều kiện | Người đặt xe hoàn tất các tác  Dữ liệu chuyến đi và tài khoản được cập nhật trong hệ thống. | | |

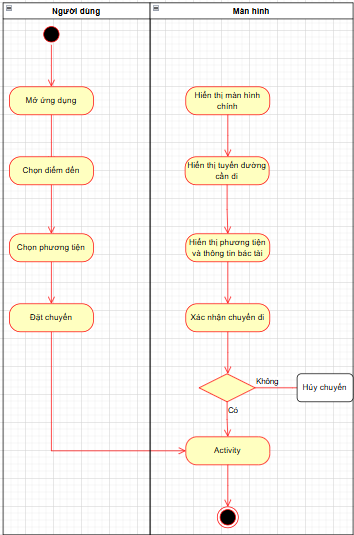
Bảng 3.8: Bảng đặc tả usecase của người đặt xe

### 3.2.3 Database



Hình 3.11: Biểu đồ database

### 3.3.4 Biểu đồ hoạt động

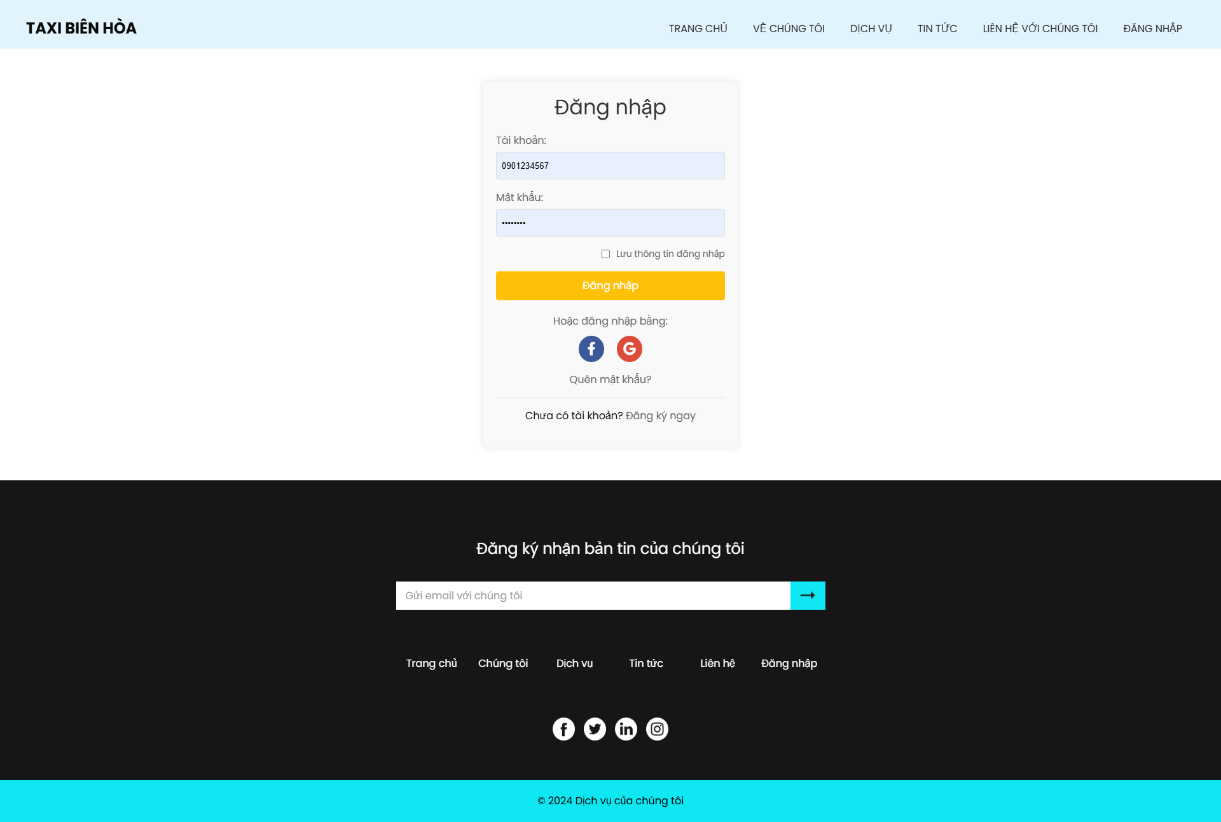


Hình 3.12: Biểu đồ hoạt động quá trình đặt xe

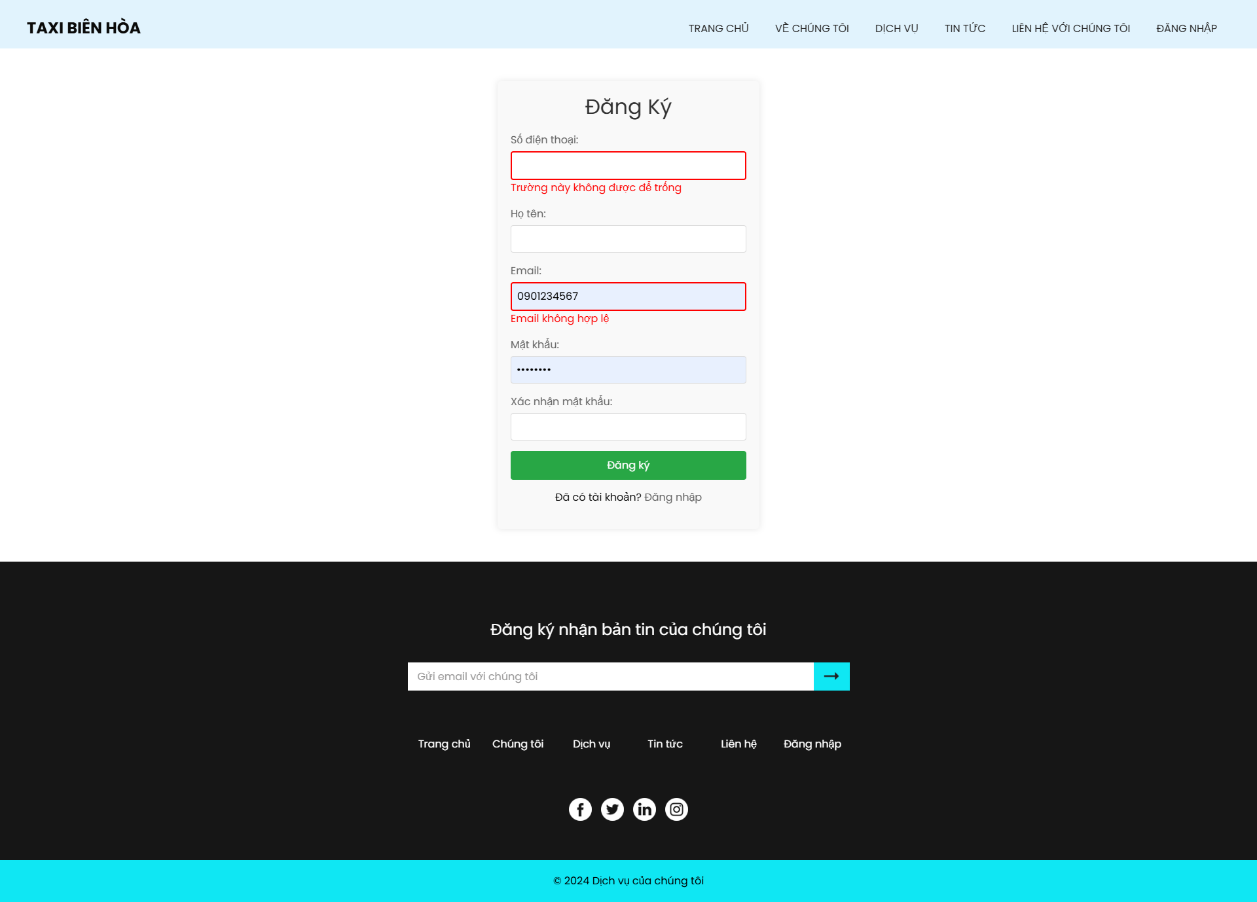
## 3.3 Giao diện phần mềm

### 3.3.1 Web cho user

* Giao diện đăng ký / đăng nhập

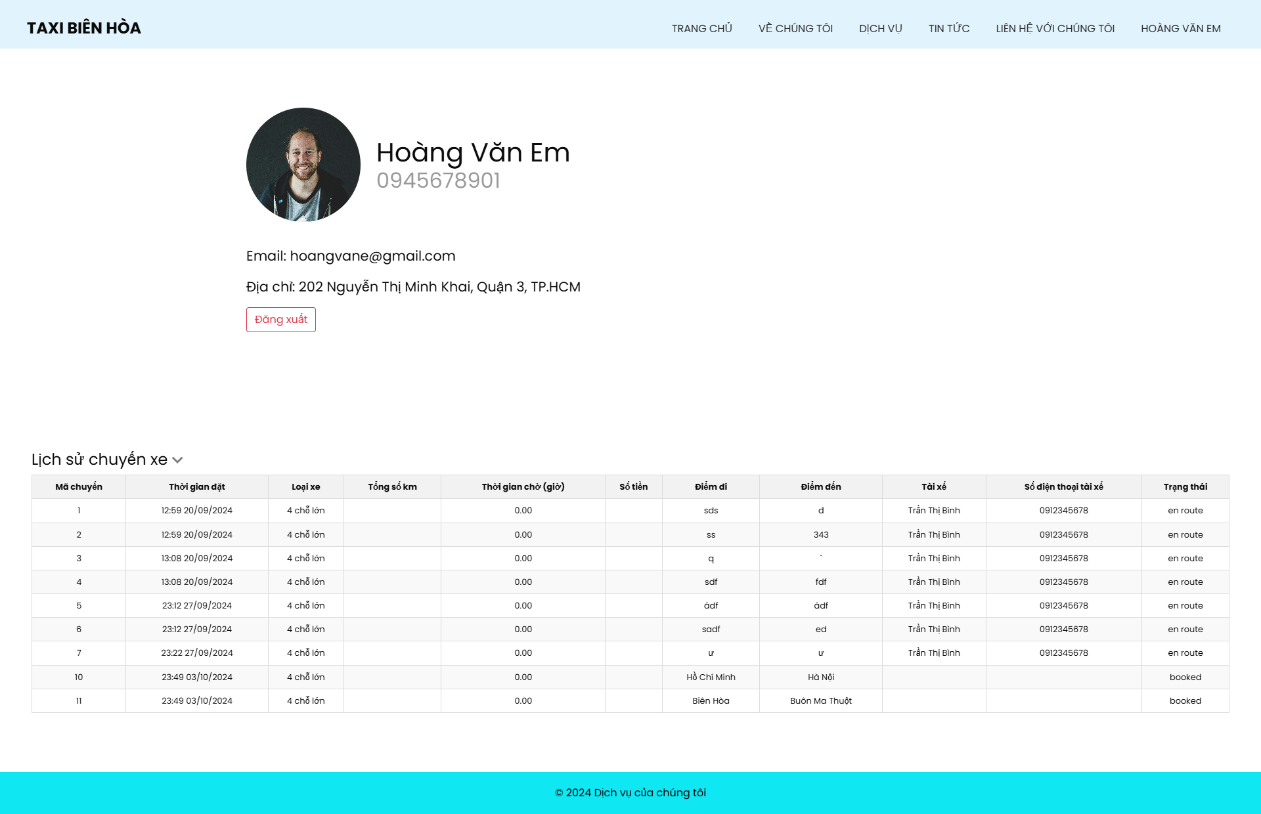


Hình 3.13: Giao diện trang đăng nhập



Hình 3.14: Giao diện trang đăng ký

Trang cá nhân của người dùng



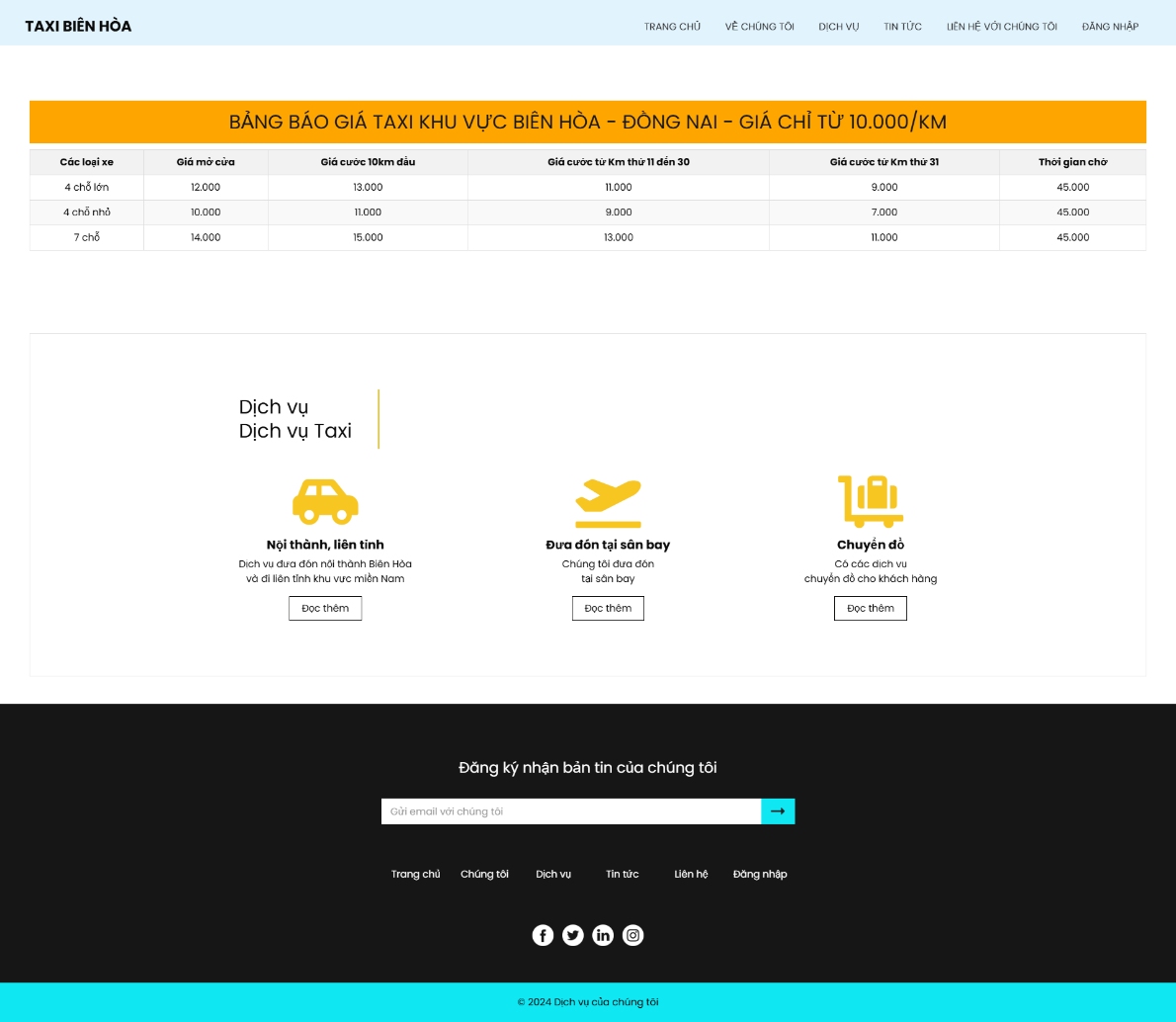
Hình 3.14: Trang thông tin người dùng

Giao diện chính khi đăng nhập



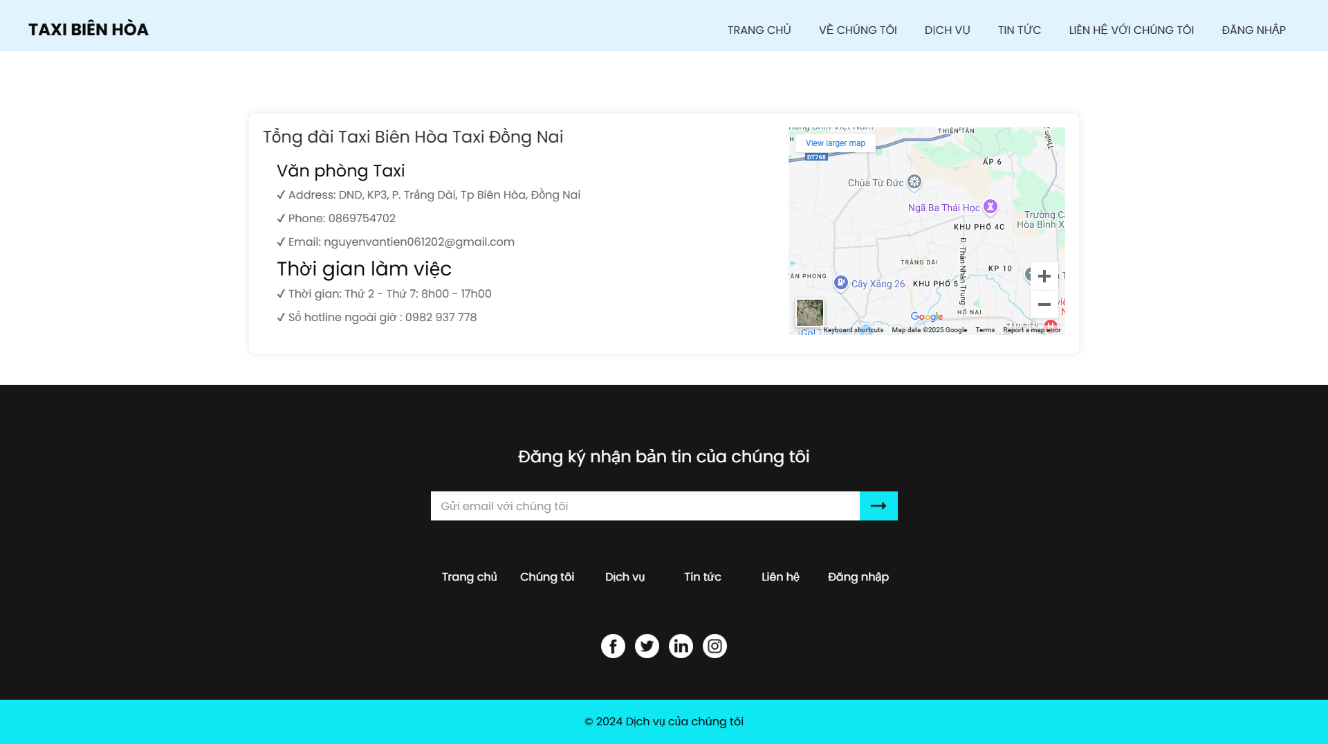
Hình 3.13: Trang chủ

Người dùng có thể vào trang dịch vụ để xem giá cước



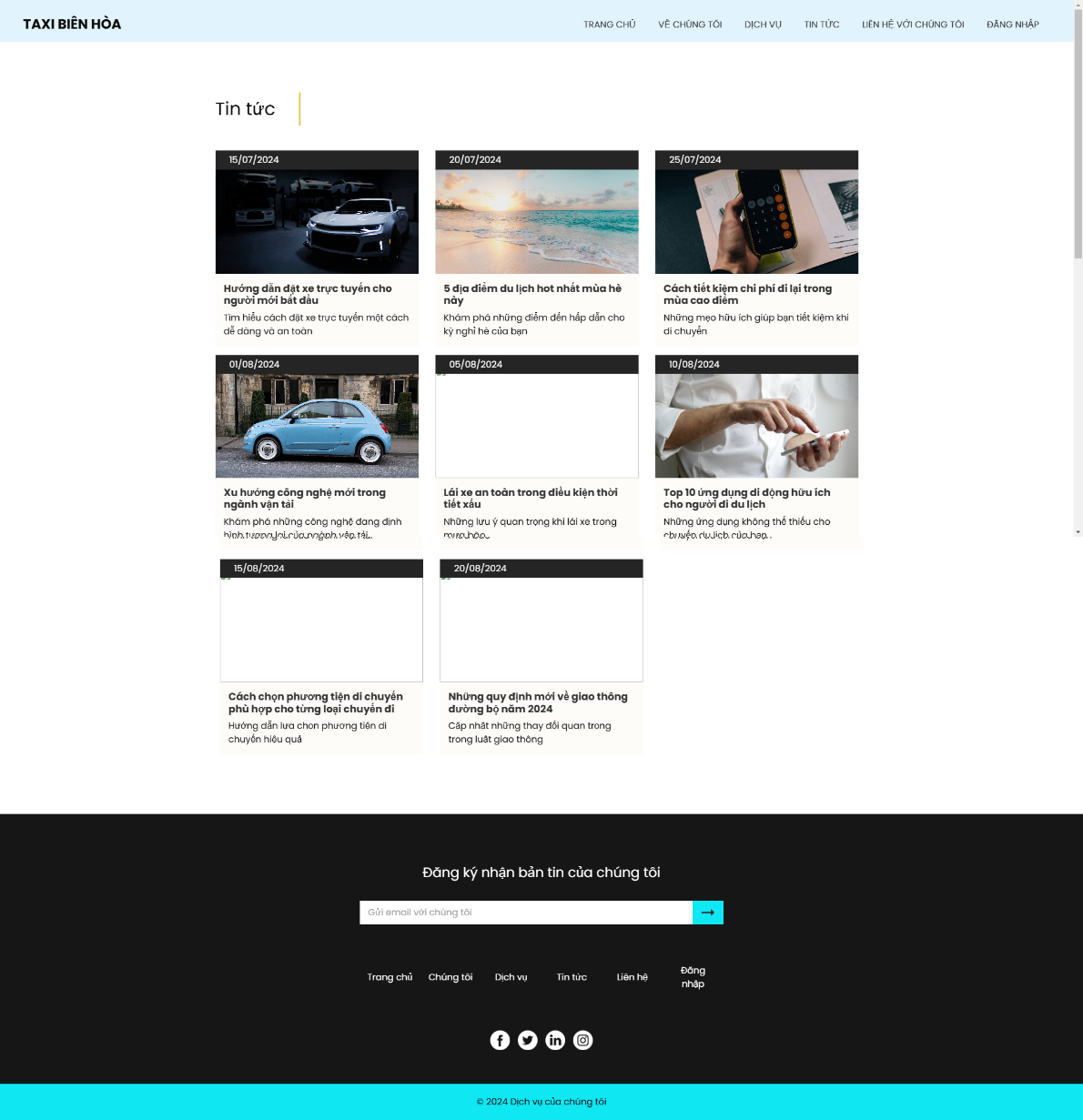
Hinh 3.34: Trang dịch vụ

Người dùng có thể vào trang liên hệ để có thể hỏi những thắc mắc tại đây



Hình 3.15: Trang liên hệ

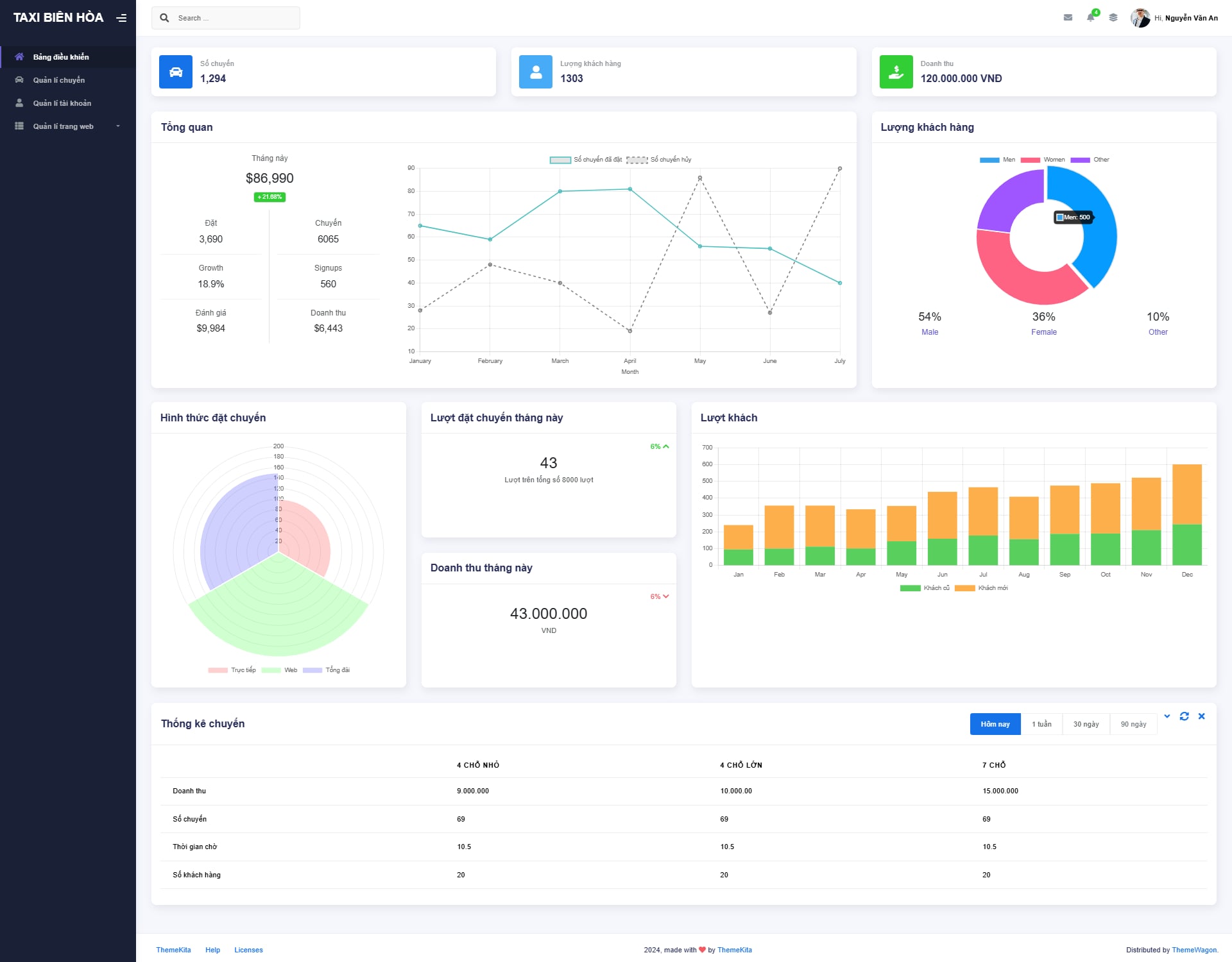
Các bạn có thể xem những tin tức nóng hổi tại web của chúng mà không cần phải sang web khác



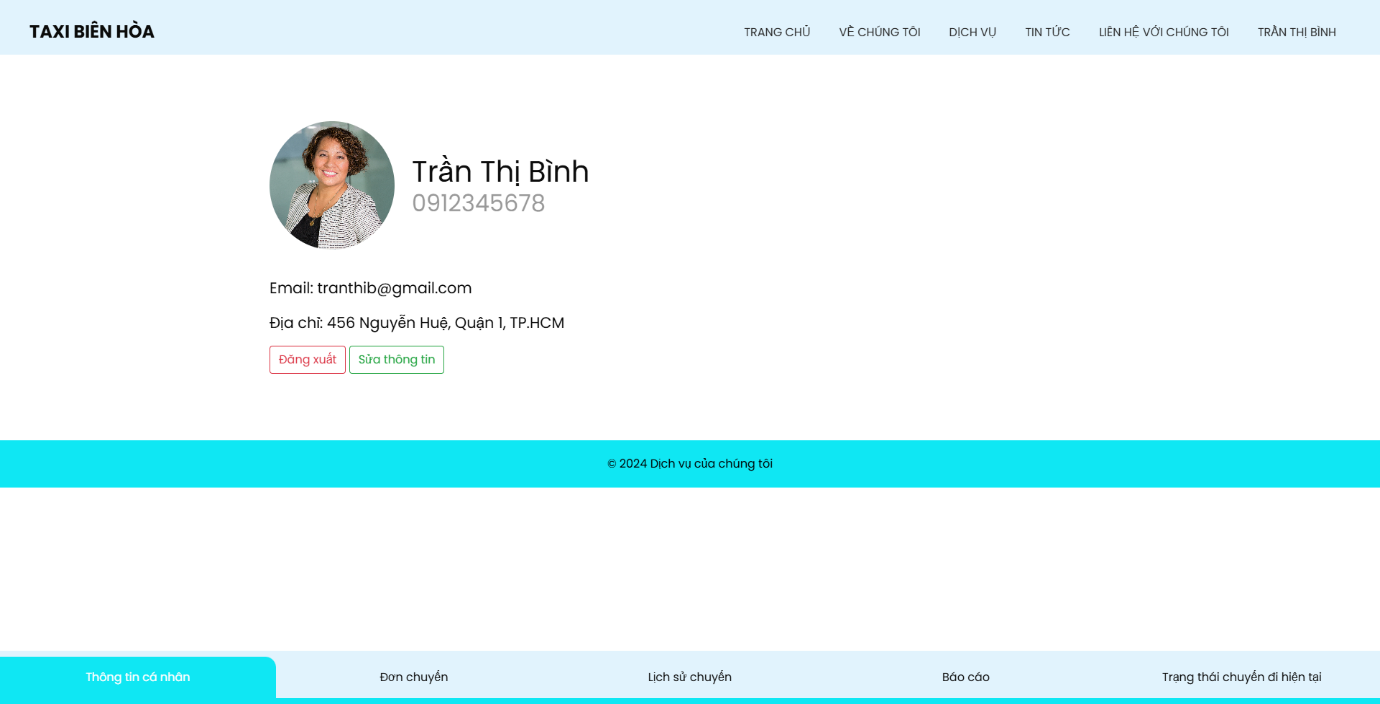
Hình 3.16: Trang tin tức

### 3.3.2 Web cho Admin

Giao diện chính của trang admin

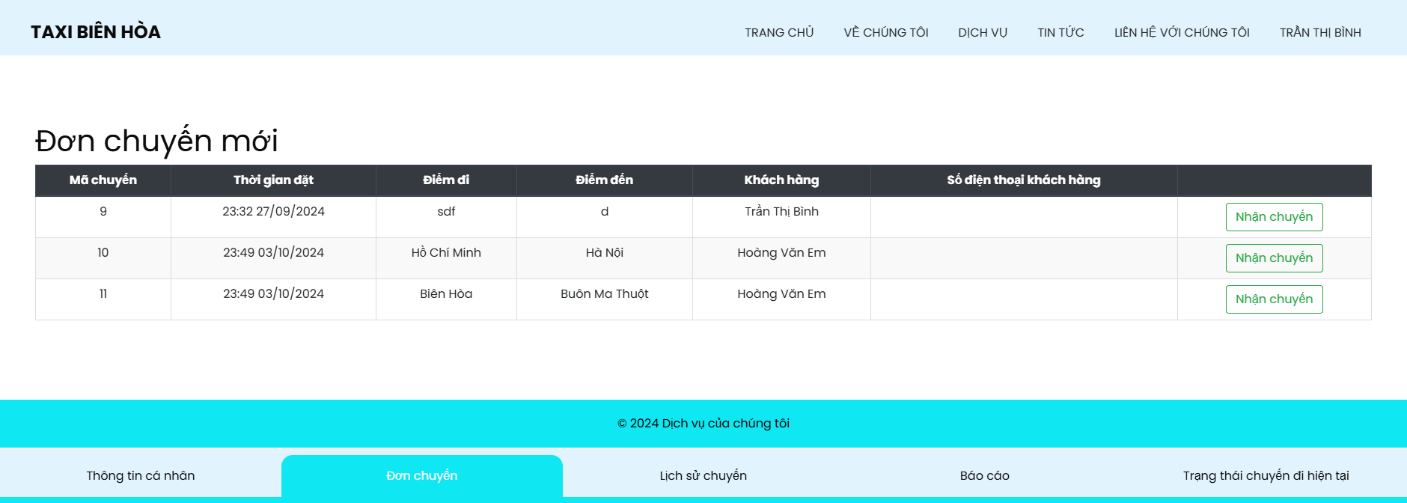


Hình :Trang Dashboard



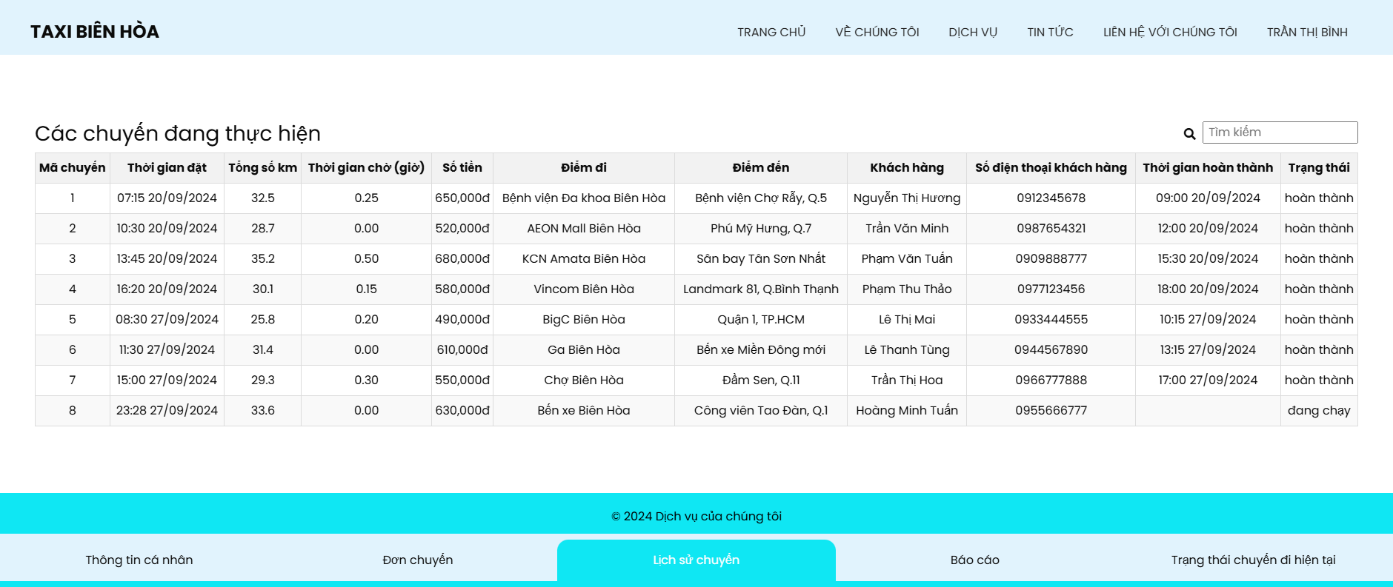
Hình 3.17: Giao diện trang thông tin Admin

3.3.7 Đơn chuyến vừa đặt



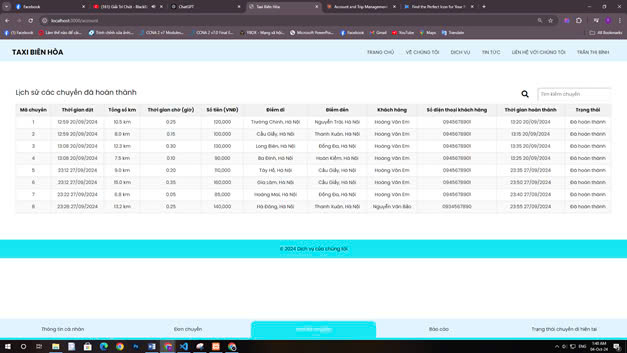
Hình 3.18: Hiển thị những chuyến mới và chờ nhận chuyến

3.3.8 Chuyến đang được thực hiện



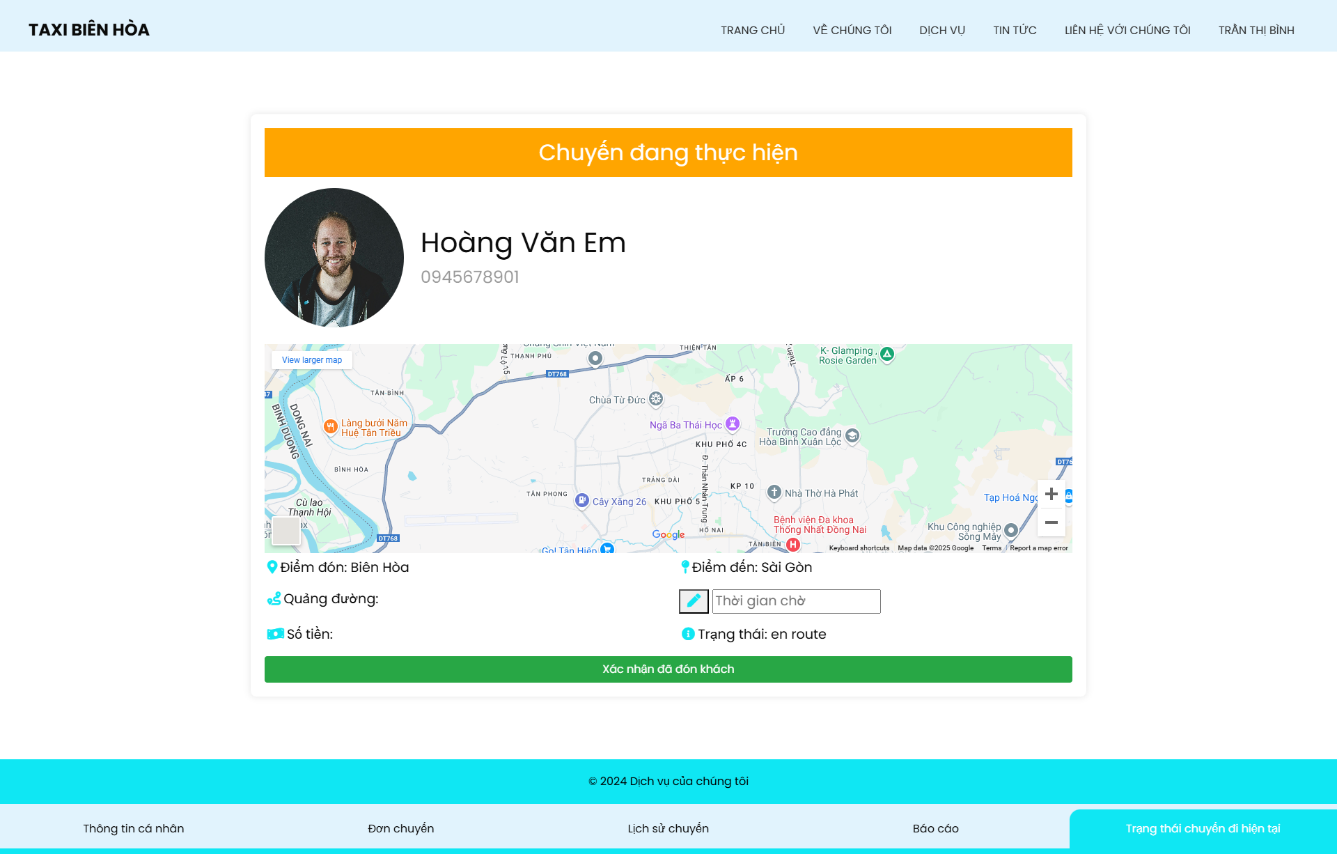
Hình 3.19: Hiển thị các chuyến đang được thực hiện

3.3.7 giao diện lịch sử đặt xe

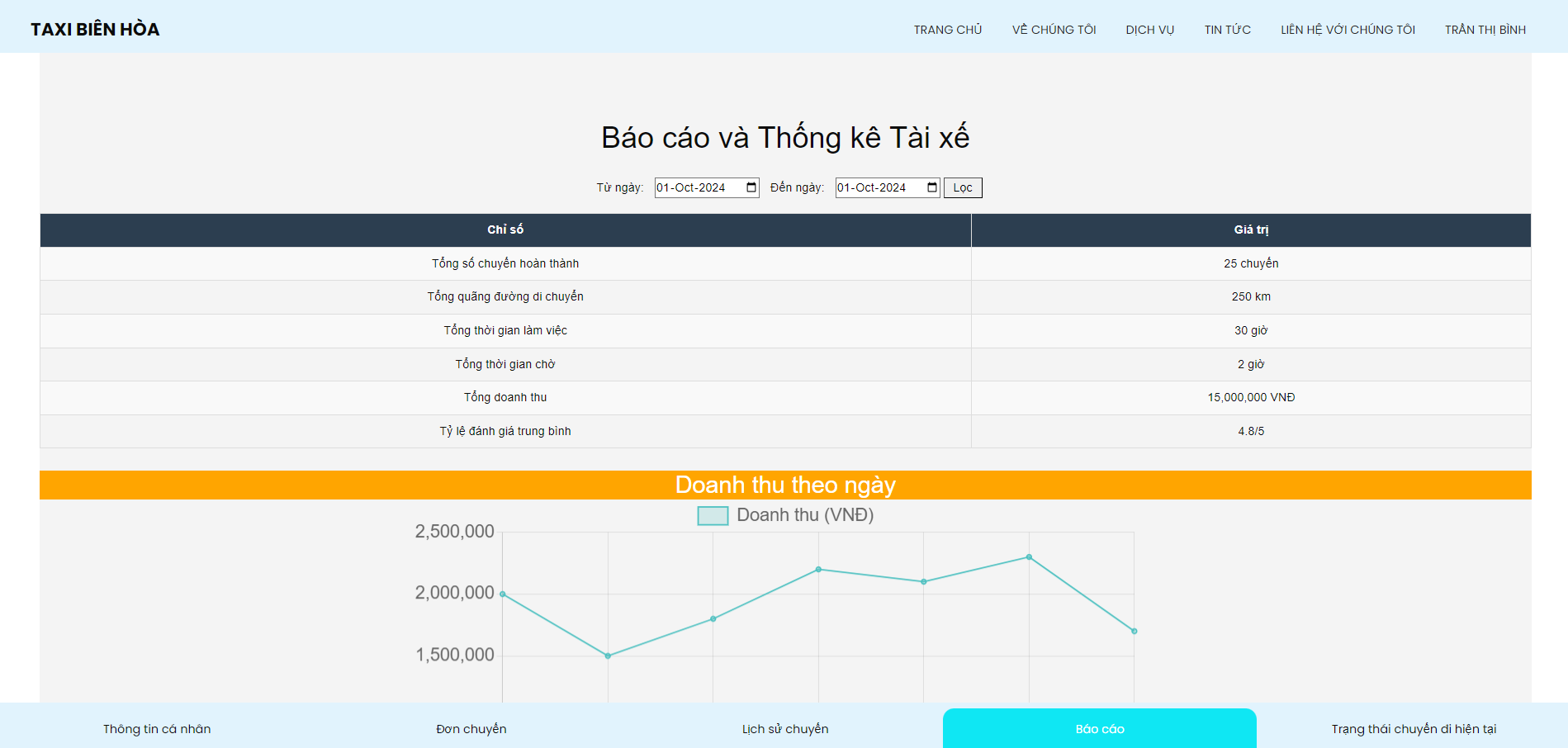


Hình 3.19: Hiễn thị lịch sử chuyến thành công

Giao diện sẽ hiện chi tiết chuyến



Hình 3.20: Chi tiết chuyến đi



Hình 3.12: Trang báo cáo thống kê

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## 4.1. Kết luận

Trong thời đại công nghệ số phát triển mạnh mẽ, việc xây dựng một hệ thống đặt xe taxi trực tuyến không chỉ giúp tối ưu hóa trải nghiệm người dùng mà còn nâng cao hiệu quả quản lý cho các doanh nghiệp vận tải. Đề tài "Xây dựng website đặt xe taxi bằng Node.js" đã giải quyết được nhiều yêu cầu thiết yếu trong việc đặt xe như: tạo đơn đặt xe nhanh chóng, định vị phương tiện, theo dõi trạng thái chuyến đi, và quản lý thông tin người dùng, tài xế một cách có hệ thống.

Việc lựa chọn Node.js làm công nghệ nền tảng mang lại nhiều lợi ích, đặc biệt là về tốc độ xử lý và khả năng phản hồi thời gian thực. Hệ thống có khả năng xử lý nhiều yêu cầu đồng thời mà vẫn giữ được độ ổn định, giúp nâng cao trải nghiệm cho người dùng khi thực hiện các thao tác đặt xe hay theo dõi hành trình. Bên cạnh đó, việc sử dụng cơ sở dữ liệu MongoDB và các thư viện hỗ trợ như Express.js, Socket.io đã góp phần tạo nên một kiến trúc linh hoạt, dễ bảo trì và phát triển thêm tính năng mới.

## 4.2. Ưu và nhược điểm

### 4.2.1 Ưu điểm

Một trong những ưu điểm lớn nhất của hệ thống là khả năng phản hồi nhanh và hỗ trợ xử lý đồng thời nhiều yêu cầu nhờ vào cơ chế non-blocking của Node.js. Điều này đặc biệt quan trọng với các ứng dụng thời gian thực như đặt xe, nơi thông tin cần được cập nhật liên tục và tức thời. Giao diện người dùng được thiết kế đơn giản, dễ sử dụng, phù hợp với nhiều nhóm đối tượng, kể cả những người không rành về công nghệ. Ngoài ra, hệ thống cũng đã xây dựng được mô hình quản lý tài xế, khách hàng, và đơn hàng khá chặt chẽ, giúp hỗ trợ quản lý hoạt động kinh doanh hiệu quả hơn.

### 4.2.2 Nhược điểm

Dù hệ thống đã đáp ứng được nhiều yêu cầu cơ bản, nhưng vẫn còn tồn tại một số điểm hạn chế cần được cải thiện. Giao diện hiện tại còn khá đơn giản, chưa thật sự hiện đại và chưa tối ưu cho các thiết bị di động. Một số tính năng nâng cao như đánh giá tài xế, theo dõi chuyến đi bằng bản đồ tương tác hay thanh toán trực tuyến vẫn chưa được tích hợp. Về mặt bảo mật, hệ thống vẫn cần được tăng cường thêm các cơ chế kiểm tra và mã hóa dữ liệu để đảm bảo an toàn thông tin người dùng.

## 4.3. Hướng phát triển

Trong tương lai, hệ thống cần được mở rộng và cải tiến để đáp ứng nhu cầu ngày càng cao của người dùng. Trước hết, cần nâng cấp giao diện để thân thiện và hiện đại hơn, đồng thời tối ưu tốc độ tải trang để tăng trải nghiệm người dùng. Về mặt chức năng, hệ thống có thể bổ sung thêm tính năng thanh toán trực tuyến, tích hợp bản đồ tương tác để theo dõi xe theo thời gian thực, và hệ thống phản hồi người dùng (như đánh giá tài xế, ghi chú chuyến đi). Bên cạnh đó, việc tăng cường bảo mật bằng các kỹ thuật như mã hóa JWT, xác thực hai lớp (2FA), và kiểm tra lỗ hổng thường xuyên sẽ giúp hệ thống an toàn hơn. Cuối cùng, việc cập nhật và phát hành phiên bản mới định kỳ cũng là một hướng đi cần thiết để duy trì hiệu năng và tính ổn định của hệ thống trong dài hạn.

TÀI LIỆU THAM KHẢO

Rà format