

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC QUẢN LÝ VÀ CÔNG NGHỆ HẢI PHÒNG



ĐỒ ÁN TỐT NGHIỆP
NGÀNH CÔNG NGHỆ THÔNG TIN

Sinh viên : Hoàng Việt Trung

Giảng viên hướng dẫn : Ths. Vũ Phạm Minh Hiền

HẢI PHÒNG – 2025

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC QUẢN LÝ VÀ CÔNG NGHỆ HẢI PHÒNG

XÂY DỰNG TRANG ĐẶT ĐỒ ĂN TRỰC TUYẾN

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY
NGÀNH CÔNG NGHỆ THÔNG TIN

Sinh viên: **Hoàng Việt Trung**

Giảng viên hướng dẫn: **Ths. Vũ Phạm Minh Hiền**

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC QUẢN LÝ VÀ CÔNG NGHỆ HẢI PHÒNG

NHIỆM VỤ ĐỀ TÀI TỐT NGHIỆP

Sinh viên : Hoàng Việt Trung - **Mã SV:** 2212102021

Lớp : CT2601

Ngành : Công nghệ thông tin

Tên đề tài: Xây dựng trang Đặt đồ ăn trực tuyến

NHIỆM VỤ ĐỀ TÀI

1. Nội dung và các yêu cầu cần giải quyết trong nhiệm vụ đề tài tốt nghiệp

a. Mô tả tóm tắt đề tài

- Đề tài tập trung vào việc xây dựng Xây dựng trang Đặt đồ ăn trực tuyến với ba vai trò người dùng chính: khách hàng, chủ nhà hàng và quản trị viên.

b. Nội dung hướng dẫn

- Khảo sát và Phân tích yêu cầu:

- Xây dựng một hệ thống website hỗ trợ nhiều nhà hàng hoạt động cùng lúc, cho phép người dùng đặt món ăn trực tuyến.
- Thiết kế giao diện thân thiện, hiện đại và dễ sử dụng cho cả người dùng, chủ nhà hàng và quản trị viên.
- Cung cấp đầy đủ các chức năng: quản lý tài khoản, giỏ hàng, đơn hàng, món ăn, thống kê và phê duyệt đăng ký nhà hàng.
- Đảm bảo tính bảo mật thông tin, toàn vẹn dữ liệu và khả năng mở rộng trong tương lai.

- Thiết kế hệ thống:

- Thiết kế kiến trúc tổng thể cho website.
- Thiết kế cơ sở dữ liệu
- Thiết kế giao diện người dùng cho các trang chức năng chính.

c. Kết quả cần đạt được

- Nắm được kiến thức về xây dựng website cũng như ngôn ngữ lập trình web.
- Xây dựng một trang web mang lại thuận tiện cho người dùng, tiết kiệm thời gian cũng như công sức.

2. Địa điểm thực tập tốt nghiệp

- Công ty TNHH máy tính Đông Phong

CÁN BỘ HƯỚNG DẪN ĐỀ TÀI TỐT NGHIỆP

Họ và tên : Ths. Vũ Phạm Minh Hiền

Học hàm, học vị : Thạc sĩ

Cơ quan công tác : Khoa Công nghệ thông tin - Trường Đại học Quản Lý và Công Nghệ Hải Phòng

Nội dung hướng dẫn:

- Ngôn ngữ lập trình
- Platform
- Các use case

Đề tài tốt nghiệp được giao ngày 06 tháng 10 năm 2025

Yêu cầu phải hoàn thành xong trước ngày 27 tháng 12 năm 2025

Đã nhận nhiệm vụ ĐTTN

Sinh viên

Đã giao nhiệm vụ ĐTTN

Giảng viên hướng dẫn

Ths. Vũ Phạm Minh Hiền

Hải Phòng, ngày tháng 12 năm 2025

TRƯỞNG KHOA

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

Độc lập - Tự do - Hạnh phúc

PHIẾU NHẬN XÉT CỦA GIÁNG VIÊN HƯỚNG DẪN TỐT NGHIỆP

Họ và tên giảng viên: Ths. Vũ Phạm Minh Hiền

Đơn vị công tác: Khoa Công nghệ thông tin - Trường Đại học Quản Lý và Công Nghệ Hải Phòng

Họ và tên sinh viên: Hoàng Việt Trung

Ngành: Công nghệ Thông tin

Nội dung hướng dẫn: Toàn bộ đề tài

1. Tinh thần thái độ của sinh viên trong quá trình làm đề tài tốt nghiệp

2. Đánh giá chất lượng của đồ án/khóa luận (so với nội dung yêu cầu đã đề ra trong nhiệm vụ Đ.T.T.N, trên các mặt lý luận, thực tiễn, tính toán số liệu...)

3. Ý kiến của giảng viên hướng dẫn tốt nghiệp

Được bảo vệ

Không được bảo vệ

Điểm hướng dẫn

Hải Phòng, ngày tháng năm 202...

Giảng viên chấm phản biện

(ký và ghi rõ họ tên)

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

Độc lập - Tự do - Hạnh phúc

PHIẾU NHẬN XÉT CỦA GIẢNG VIÊN CHẤM PHẢN BIỆN

Họ và tên giảng viên: ThS. Vũ Trọng Chiến

Đơn vị công tác: Khoa Công nghệ thông tin - Trường Đại học Quản Lý và Công Nghệ Hải Phòng

Họ và tên sinh viên: Hoàng Việt Trung

Ngành: Công nghệ Thông tin

Đề tài tốt nghiệp: Xây dựng trang đặt đồ ăn trực tuyến

1. Phần nhận xét của giảng viên chấm phản biện

- Đồ án nghiêm túc, đáp ứng yêu cầu với phân tích rõ ràng về thiết kế website đặt đồ ăn.
- Triển khai thành công các chức năng chính: đăng ký/đăng nhập, quản lý giỏ hàng, quản lý món ăn cho User, phê duyệt nhà hàng/món ăn cho Admin.
- Thiết kế cơ sở dữ liệu và giao diện cơ bản, kiểm thử chức năng tốt.

2. Những mặt còn hạn chế

- Chức năng đặt đơn hàng chưa hoàn thiện (thanh toán, đặt nhiều shop), thiếu tính năng đánh giá/bình luận.
- Tìm kiếm cơ bản, chưa có theo dõi đơn hàng.
- Giao diện chưa chuyên nghiệp, kiểm thử chưa toàn diện với dữ liệu lớn.

3. Ý kiến của giảng viên chấm phản biện: Được bảo vệ. Ý kiến của giảng viên chấm phản biện

Được bảo vệ Không được bảo vệ Điểm phản biện

Hải Phòng, ngày.....tháng năm 2025

Giảng viên chấm phản biện

(ký và ghi rõ họ tên)

LỜI CẢM ƠN

Lời đầu tiên, em xin được bày tỏ lòng biết ơn chân thành đến Quý thầy cô trong khoa Công nghệ Thông tin của Trường Đại học Quản lý và Công nghệ Hải Phòng, không chỉ là những người đã truyền đạt những kiến thức quý báu mà còn là những người thắp lên ngọn lửa đam mê, truyền cảm hứng và hướng dẫn chúng em trong suốt hành trình khám phá tri thức, không chỉ trong lĩnh vực công nghệ thông tin mà còn về kỹ năng mềm và phương pháp tư duy hiệu quả.

Em xin gửi lời cảm ơn sâu sắc đến **thầy Vũ Phạm Minh Hiểu**, người đã trực tiếp hướng dẫn và hỗ trợ em trong suốt quá trình thực hiện khóa luận. Sự tận tâm, kiên nhẫn và những góp ý quý báu của thầy đã giúp em vượt qua khó khăn và hoàn thiện đề án một cách tốt nhất.

Đặc biệt, em muốn gửi lời cảm ơn đến gia đình – những người luôn là hậu phương vững chắc. Sự yêu thương và hy sinh thầm lặng của bố mẹ, anh chị em là nguồn động lực to lớn giúp em vượt qua những thử thách trong quá trình học tập.

Dù đã cố gắng hết sức và dành nhiều thời gian để nghiên cứu, nhưng với kiến thức và kinh nghiệm thực tế còn hạn chế, đề án chắc chắn không tránh khỏi thiếu sót. Em rất mong nhận được những góp ý chân thành từ Quý thầy cô để có thể hoàn thiện hơn, đồng thời nâng cao kiến thức và kỹ năng của bản thân.

Cuối cùng, em xin kính chúc toàn thể Quý thầy cô trong khoa Công nghệ Thông tin cùng các thầy cô của Trường Đại học Quản lý và Công nghệ Hải Phòng luôn dồi dào sức khỏe, tràn đầy nhiệt huyết để tiếp tục dìu dắt và truyền đạt tri thức cho các thế hệ sinh viên mai sau.

Hải Phòng, ngày ... tháng năm 2025

Sinh viên thực hiện

LỜI CAM ĐOAN

Em cam đoan rằng nội dung và kết quả của đề tài “**Xây dựng trang Đặt đồ ăn trực tuyến**” trong chuyên đề này là sản phẩm do chính em thực hiện, không sao chép hay vi phạm bản quyền của bất kỳ cá nhân hoặc tổ chức nào khác.

Các tài liệu, hình ảnh và nội dung tham khảo được sử dụng trong đồ án đều được trích dẫn rõ ràng nguồn gốc, tuân thủ đúng quy định của **Luật Sở hữu trí tuệ** và **Luật Bản quyền tác giả**.

Em xin chịu hoàn toàn trách nhiệm trước **Nhà trường, Giảng viên hướng dẫn** và **pháp luật** về tính trung thực, chính xác của toàn bộ nội dung và kết quả nghiên cứu trong đồ án.

Em cũng xin cam đoan rằng trong quá trình thực hiện, em đã tuân thủ nghiêm túc quy định của Trường Đại học Quản lý và Công nghệ Hải Phòng, không có bất kỳ hành vi gian lận nào trong việc thu thập, xử lý dữ liệu hay trình bày kết quả.

Em rất mong nhận được sự đánh giá, góp ý quý báu của Quý thầy cô để hoàn thiện hơn nữa đề tài này.

Hải Phòng, ngày tháng năm 2025

SINH VIÊN

MỤC LỤC

LỜI CẢM ƠN.....	8
LỜI CAM ĐOAN.....	9
DANH MỤC TỪ VIẾT TẮT VÀ THUẬT NGỮ.....	5
DANH MỤC HÌNH ẢNH.....	7
DANH MỤC BẢNG.....	9
LỜI MỞ ĐẦU.....	10
CHƯƠNG I: CƠ SỞ LÝ THUYẾT	13
1.1. Mô hình Client – Server trong phát triển web.....	13
1.1.1. Khái niệm về mô hình Client – Server	14
1.1.2. Vai trò và chức năng của Client	14
1.1.3. Vai trò và chức năng của Server.....	15
1.1.4. Nguyên lý hoạt động.....	15
1.1.5. Ưu điểm của mô hình Client – Server.....	16
1.2. Frontend	16
1.2.1. React.js	16
1.2.2. React Router.....	18
1.2.3. Context API.....	19
1.3. Backend.....	20
1.3.1. Node.js.....	20
1.3.2. Express.js.....	21
1.3.3. Mô hình kiến trúc Backend	22
1.3.4. Middleware.....	24
1.3.5. Kết nối Frontend – Backend	25
1.4. Cơ sở dữ liệu MySQL.....	26
1.4.1. Giới thiệu MySQL	26
1.4.2. Đặc điểm nổi bật.....	27
1.4.3. Vai trò của MySQL trong dự án.....	27
1.5. Kiểm thử	27

1.5.1. Giới thiệu.....	27
1.5.2. Kiểm thử hộp đen (Black Box Testing)	28
CHƯƠNG II : PHÂN TÍCH ĐẶC TẢ CHỨC NĂNG CỦA HỆ THỐNG.....	30
2.1.1. Yêu cầu chức năng	30
2.1.2. Yêu cầu phi chức năng	31
2.2.1. Tác nhân hệ thống.....	33
2.2.2. Biểu đồ usecase tổng quát.....	34
2.2.3. Biểu đồ usecase phân rã	35
2.3.1. Use Case: Đăng nhập	41
2.3.2. Use Case: Đăng ký.....	42
2.3.3. Use Case: Đăng xuất	44
2.3.4. Use Case: Tìm kiếm nhà hàng	45
2.3.5. Use Case: Xem Chi tiết Nhà hàng và Thực đơn	46
2.3.6. Use Case: Quản lý Giỏ hàng	48
2.3.7. Use Case: Đặt hàng	50
2.3.8. Use Case: Xem và Quản lý Đơn hàng	51
2.3.9. Use Case: Cập nhật Thông tin Tài khoản.....	53
2.3.10. Use Case: Đăng ký làm Nhà hàng	55
2.3.11. Use Case: Quản lý Thực đơn (Owner)	57
2.3.12. Use Case: Quản lý Đơn hàng (Owner).....	59
2.3.13. Use Case: Kiểm duyệt Món ăn (Admin).....	61
2.3.14. Use Case: Quản lý Người dùng (Admin)	62
2.3.15. Use Case: Quản lý Nhà hàng (Admin).....	64
2.3.16. Use Case: Phê duyệt Nhà hàng (Admin)	65
CHƯƠNG III: THIẾT KẾ CHỨC NĂNG VÀ CƠ SỞ DỮ LIỆU.....	68
3.1. Biểu đồ tuần tự.....	68
3.1.1. Biểu đồ tuần tự chức năng Đăng nhập	68
3.1.2. Biểu đồ tuần tự chức năng Đăng ký.....	69
3.1.3. Biểu đồ tuần tự chức năng Tìm kiếm và Xem chi tiết Nhà hàng.....	70

3.1.4. Biểu đồ tuần tự chức năng quản lý giỏ hàng	71
3.1.5. Biểu đồ tuần tự chức năng Đặt hàng	71
3.1.6. Biểu đồ tuần tự chức năng xem và hủy đơn hàng	72
3.1.7. Biểu đồ tuần tự chức năng Cập nhật thông tin tài khoản	73
3.1.8. Biểu đồ tuần tự chức năng Đăng ký làm nhà hàng.....	74
3.1.9. Biểu đồ tuần tự chức năng Quản lý đơn hàng	75
3.1.10. Biểu đồ tuần tự chức năng Phê duyệt nhà hàng	76
3.1.11. Biểu đồ tuần tự chức năng Quản lý người dùng.....	77
3.2. Thiết kế cơ sở dữ liệu.....	78
3.2.1. Lược đồ cơ sở dữ liệu.....	78
3.2.2. Biểu đồ UML	79
3.2.3. Mô hình EER	80
3.2.4. Mô hình quan hệ.....	81
Danh sách các bảng	82
3.2.5. Chi tiết các bảng	82
CHƯƠNG IV: KIỂM THỬ VÀ TRIỂN KHAI WEBSITE.....	87
4.1. Kiểm thử	87
4.1.1. Phân tích các trường hợp kiểm thử.....	87
4.1.2. Danh sách Testcase chi tiết.....	89
4.1.3. Kết quả thực thi Test	89
4.2. Yêu cầu về cài đặt	89
4.2.1. Yêu cầu phần cứng:	89
4.2.2. Yêu cầu phần mềm:	90
4.2.3. Yêu cầu mạng Internet:	91
4.3. Kết quả cài đặt môi trường.....	92
4.3.1. Data Access Layer (MySQL).....	92
4.3.2. Bussiness Logic Layer (Express)	93
4.3.3. Presentation Layer (ReactJS)	94
4.4. Kết quả thực nghiệm	96

4.4.1. Giao diện đăng nhập Website	96
4.4.2. Giao diện trang Đăng ký.....	97
4.4.3. Giao diện trang chủ Website.....	97
4.4.4. Giao diện trang nhà hàng	98
4.4.5. Giao diện giỏ hàng.....	98
4.4.6. Giao diện chi tiết đơn hàng.....	99
4.4.7. Giao diện thống kê của nhà hàng	99
4.4.8. Giao diện trang đăng ký làm nhà hàng.....	100
4.4.9. Giao diện thống kê của nhà hàng	100
4.4.10. Giao diện quản lý thực đơn.....	101
4.4.11. Giao diện quản lý đơn hàng.....	102
4.4.12. Giao diện thống kê của Admin.....	102
4.4.13. Giao diện trang duyệt đơn nhà hàng	103
4.4.14. Giao diện trang quản lý nhà hàng	103
4.4.15. Giao diện trang quản lý user	104
4.4.16. Giao diện trang kiểm duyệt thực đơn.....	104
CHƯƠNG V: KẾT LUẬN	105
5.1. Kết quả đạt được.....	105
5.2. Hạn chế	105
5.3. Hướng phát triển	106
TÀI LIỆU THAM KHẢO.....	107

DANH MỤC TỪ VIẾT TẮT VÀ THUẬT NGỮ

Thuật Ngữ	Giải Thích
UI	User Interface (Giao diện người dùng)
DOM	Document Object Model
JSX	JavaScript XML
API	Application Programming Interface
RESTful API	Representational State Transfer API
HTTP/HTTPS	Hypertext Transfer Protocol / Hypertext Transfer Protocol Secure
JSON	JavaScript Object Notation
JWT	JSON Web Token
CRUD	Create, Read, Update, Delete
RDBMS	Relational Database Management System (Hệ quản trị cơ sở dữ liệu quan hệ)
SQL	Structured Query Language (Ngôn ngữ Truy vấn có Cấu trúc)
ORM	Object-Relational Mapping
MVC	Model - View - Controller
COD	Cash on Delivery (Thanh toán khi nhận hàng)
OOAD	Object-Oriented Analysis and Design (Phân tích và Thiết kế hướng đối tượng)
UML	Unified Modeling Language
CPU	Central Processing Unit (Bộ xử lý trung tâm)
RAM	Random Access Memory (Bộ nhớ truy cập ngẫu nhiên)
NPM	Node Package Manager
VSCoDe	Visual Studio Code
IDE	Integrated Development Environment (Môi trường phát triển tích hợp)

CORS	Cross-Origin Resource Sharing
DB / CSDL	Database / Cơ sở dữ liệu
Admin	Quản trị viên
Owner	Chủ nhà hàng
Member	Người dùng đã đăng nhập (Khách hàng)

DANH MỤC HÌNH ẢNH

ẢNH 1.1: MÔ HÌNH CLIENT – SERVER-DATA	13
ẢNH 1.2: MÔ HÌNH MVC	23
ẢNH 2.1: BIỂU ĐỒ USECASE TỔNG QUÁT.....	34
ẢNH 2.2: BIỂU ĐỒ PHÂN RÃ USECASE GIỎ HÀNG	35
ẢNH 2.3: BIỂU ĐỒ PHÂN RÃ USECASE ĐẶT MÓN.....	36
ẢNH 2.4: BIỂU ĐỒ PHÂN RÃ USECASE HỒ SƠ CÁ NHÂN.....	36
ẢNH 2.5: BIỂU ĐỒ PHÂN RÃ USECASE ĐĂNG KÝ LÀM NHÀ HÀNG.....	37
ẢNH 2.6: BIỂU ĐỒ PHÂN RÃ USECASE QUẢN LÝ ĐƠN HÀNG	37
ẢNH 2.7: BIỂU ĐỒ PHÂN RÃ USECASE QUẢN LÝ NGƯỜI DÙNG.....	38
ẢNH 2.8: BIỂU ĐỒ PHÂN RÃ USECSE QUẢN LÝ NHÀ HÀNG	39
ẢNH 2.9: BIỂU ĐỒ PHÂN RÃ USECSE DUYỆT ĐƠN ĐĂNG KÝ NHÀ HÀNG..	39
ẢNH 2.10: BIỂU ĐỒ PHÂN RÃ USECSE TÌM KIẾM VÀ XEM NHÀ HÀNG/THỰC ĐƠN.....	40
ẢNH 2.11: BIỂU ĐỒ PHÂN RÃ USECSE ĐĂNG NHẬP.....	40
ẢNH 2.12: BIỂU ĐỒ PHÂN RÃ USECSE ĐĂNG KÝ.....	40
ẢNH 3.1. BIỂU ĐỒ TUẦN TỰ CHỨC NĂNG ĐĂNG NHẬP	68
ẢNH 3.2. BIỂU ĐỒ TUẦN TỰ CHỨC NĂNG ĐĂNG KÝ	69
ẢNH 3.3. BIỂU ĐỒ TUẦN TỰ CHỨC NĂNG TÌM KIẾM VÀ XEM CHI TIẾT NHÀ HÀNG.....	70
ẢNH 3.4. BIỂU ĐỒ TUẦN TỰ CHỨC NĂNG QUẢN LÝ GIỎ HÀNG	71
ẢNH 3.5. BIỂU ĐỒ TUẦN TỰ CHỨC NĂNG ĐẶT HÀNG	71
ẢNH 3 .6. BIỂU ĐỒ TUẦN TỰ CHỨC NĂNG XEM VÀ HỦY ĐƠN HÀNG.....	72
ẢNH 3.7. BIỂU ĐỒ CHỨC NĂNG CẬP NHẬT THÔNG TIN TÀI KHOẢN.....	73
ẢNH 3.8. BIỂU ĐỒ TUẦN TỰ CHỨC NĂNG ĐĂNG KÝ LÀM NHÀ HÀNG.....	74
ẢNH 3.9. BIỂU ĐỒ TUẦN TỰ CHỨC NĂNG QUẢN LÝ ĐƠN HÀNG	75
ẢNH 3.10. BIỂU ĐỒ TUẦN TỰ CHỨC NĂNG PHÊ DUYỆT NHÀ HÀNG	76
ẢNH 3.11. BIỂU ĐỒ TUẦN TỰ CHỨC NĂNG QUẢN LÝ NGƯỜI DÙNG.....	77
ẢNH 3.12. LƯỢC ĐỒ CƠ SỞ DỮ LIỆU	78

ẢNH 3.13. BIỂU ĐỒ UML	79
ẢNH 3.14. MÔ HÌNH EER.....	80
ẢNH 3.15. MÔ HÌNH QUAN HỆ	81
ẢNH 4.1. CÀI ĐẶT CƠ SỞ DỮ LIỆU.....	92
ẢNH 4.2. CÀI ĐẶT BUSSINESS LOGIC LAYER (EXPRESS)	93
ẢNH 4.3. CÀI ĐẶT PRESENTATION LAYER (REACTJS)	95
ẢNH.4.4. GIAO DIỆN CHỨC NĂNG ĐĂNG NHẬP.....	96
ẢNH 4.5. GIAO DIỆN CHỨC NĂNG ĐĂNG KÝ	97
ẢNH 4.6. GIAO DIỆN TRANG CHỦ MÀN HÌNH TRANG CHỦ.....	97
ẢNH 4.7. GIAO DIỆN CHI TIẾT THỰC ĐƠN NHÀ HÀNG	98
ẢNH 4.8. GIAO DIỆN GIỎ HÀNG	98
ẢNH 4.9. GIAO DIỆN DANH SÁCH ĐƠN HÀNG ĐÃ ĐẶT.....	99
ẢNH 4.10. GIAO DIỆN CẬP NHẬT TÀI KHOẢN.....	99
ẢNH 4.11. GIAO DIỆN TRANG ĐĂNG KÝ LÀM NHÀ HÀNG	100
ẢNH 4.12. GIAO DIỆN THỐNG KÊ CỦA NHÀ HÀNG.....	100
ẢNH 4.13. GIAO DIỆN QUẢN LÝ THỰC ĐƠN.....	101
ẢNH 4.14. GIAO DIỆN THÊM MỚI MÓN ĂN.....	101
ẢNH 4.15. GIAO DIỆN QUẢN LÝ ĐƠN HÀNG.....	102
ẢNH 4.16. GIAO DIỆN THỐNG KÊ CỦA ADMIN	102
ẢNH 4.17. GIAO DIỆN TRANG DUYỆT NHÀ HÀNG	103
ẢNH 4.18. GIAO DIỆN TRANG QUẢN LÝ NHÀ HÀNG	103
ẢNH 4.19. GIAO DIỆN TRANG QUẢN LÝ USER	104
ẢNH 4.20. GIAO DIỆN TRANG KIỂM DUYỆT THỰC ĐƠN	104

DANH MỤC BẢNG

BẢNG 1.1: ƯU NHƯỢC ĐIỂM CỦA KIỂM THỬ HỘP ĐEN	28
BẢNG 2.1: DANH SÁCH TÁC NHÂN	33
BẢNG 2.2: UC ĐĂNG NHẬP	42
BẢNG 2.3: UC ĐĂNG KÝ	44
BẢNG 2.4: UC ĐĂNG XUẤT	45
BẢNG 2.5: UC TÌM KIẾM NHÀ HÀNG.....	46
BẢNG 2.6: UC CHI TIẾT NHÀ HÀNG VÀ THỰC ĐƠN	48
BẢNG 2.7: UC QUẢN LÝ GIỎ HÀNG	49
BẢNG 2.8: UC ĐẶT HÀNG.....	51
BẢNG 2.9: UC XEM VÀ QUẢN LÝ ĐƠN HÀNG.....	53
BẢNG 2.10: UC CẬP NHẬT THÔNG TIN TÀI KHOẢN.....	55
BẢNG 2.11: UC CẬP NHẬT THÔNG TIN TÀI KHOẢN.....	57
BẢNG 2.12: UC QUẢN LÝ THỰC ĐƠN (OWNER)	59
BẢNG 2.13: UC QUẢN LÝ ĐƠN HÀNG (OWNER).....	61
BẢNG 2.14: UC KIỂM DUYỆT MÓN ĂN (ADMIN)	62
BẢNG 2.15: UC QUẢN LÝ NGƯỜI DÙNG (ADMIN).....	64
BẢNG 2.16: UC QUẢN LÝ NHÀ HÀNG (ADMIN)	65
BẢNG 2.17: UC QUẢN LÝ NHÀ HÀNG (ADMIN)	67
BẢNG 3.1. DANH SÁCH CÁC BẢNG DỮ LIỆU	82
BẢNG 3.2. BẢNG USER.....	83
BẢNG 3.3. BẢNG MENU_ITEMS	83
BẢNG 3.4. BẢNG RESTAURANTS	84
BẢNG 3.5. BẢNG ORDER	85
BẢNG 3.6. BẢNG ORDER_ITEMS	86
BẢNG 3.7. BẢNG CART_ITEMS	86
BẢNG 4.1. PHÂN TÍCH CÁC TRƯỜNG HỢP KIỂM THỬ	88
BẢNG 4.2. BẢNG KẾT QUẢ THỰC THI TEST	89

LỜI MỞ ĐẦU

A. Lý do chọn đề tài

Trong thời đại công nghệ số phát triển mạnh mẽ, các hoạt động mua sắm và dịch vụ trực tuyến đang dần trở thành xu hướng tất yếu của xã hội hiện đại. Đặc biệt trong lĩnh vực ẩm thực, nhu cầu đặt đồ ăn online ngày càng tăng cao do sự tiện lợi và nhanh chóng mà nó mang lại. Tuy nhiên, hầu hết các nền tảng hiện nay chỉ phục vụ cho một nhà hàng hoặc chuỗi nhà hàng nhất định, gây hạn chế cho người dùng khi muốn so sánh và lựa chọn món ăn từ nhiều nhà hàng khác nhau.

Xuất phát từ thực tế đó, việc xây dựng **Xây dựng trang Đặt đồ ăn trực tuyến** là vô cùng cần thiết. Hệ thống này không chỉ giúp người dùng dễ dàng tìm kiếm, đặt món và theo dõi đơn hàng mà còn hỗ trợ các chủ nhà hàng quản lý thực đơn, đơn hàng, doanh thu một cách thuận tiện. Đồng thời, trang quản trị viên có thể giám sát toàn bộ hoạt động hệ thống, đảm bảo tính ổn định và minh bạch.

Với mong muốn áp dụng các công nghệ web hiện đại vào thực tế, em quyết định chọn đề tài **“Xây dựng trang Đặt đồ ăn trực tuyến”** làm đề án tốt nghiệp nhằm rèn luyện kỹ năng thiết kế, lập trình và triển khai một hệ thống web hoàn chỉnh, phục vụ nhu cầu thiết thực của xã hội.

B. Mục tiêu và ý nghĩa nghiên cứu

Mục tiêu của đề tài:

- Xây dựng một hệ thống website hỗ trợ nhiều nhà hàng hoạt động cùng lúc, cho phép người dùng đặt món ăn trực tuyến.
- Thiết kế giao diện thân thiện, hiện đại và dễ sử dụng cho cả người dùng, chủ nhà hàng và quản trị viên.
- Cung cấp đầy đủ các chức năng: quản lý tài khoản, giỏ hàng, đơn hàng, món ăn, thống kê và phê duyệt đăng ký nhà hàng.
- Đảm bảo tính bảo mật thông tin, toàn vẹn dữ liệu và khả năng mở rộng trong tương lai.

Ý nghĩa nghiên cứu:

- **Đối với người dùng:** Tăng trải nghiệm khi có thể đặt món ăn nhanh chóng, dễ dàng so sánh giá và chất lượng giữa các nhà hàng.

- **Đối với chủ nhà hàng:** Giúp quản lý thực đơn, đơn hàng và doanh thu hiệu quả hơn, giảm bớt chi phí nhân sự và sai sót.
- **Đối với sinh viên thực hiện:** Giúp vận dụng kiến thức đã học về phân tích – thiết kế hệ thống thông tin, lập trình web frontend (ReactJS) và backend (NodeJS, Express), cơ sở dữ liệu (MySQL) cùng với cơ chế bảo mật (JWT).

C. Phạm vi và phương pháp nghiên cứu

Phạm vi nghiên cứu:

- Đề tài tập trung vào việc xây dựng Xây dựng trang Đặt đồ ăn trực tuyến với ba vai trò người dùng chính: **khách hàng, chủ nhà hàng và quản trị viên.**
- Hệ thống cung cấp các tính năng chính như:
 - Tìm kiếm và xem thông tin nhà hàng, món ăn.
 - Đặt hàng và quản lý giỏ hàng.
 - Quản lý thực đơn, đơn hàng và thống kê doanh thu.
 - Quản trị toàn bộ hệ thống và phê duyệt đăng ký nhà hàng.
- Hệ thống hoạt động theo mô hình Client–Server, frontend giao tiếp backend thông qua RESTful API.

Phương pháp nghiên cứu:

- Khảo sát thực tế: Nghiên cứu các nền tảng đặt đồ ăn phổ biến (GrabFood, ShopeeFood, Baemin, v.v.) để rút ra ưu nhược điểm.
- Phân tích hệ thống: Áp dụng phương pháp hướng đối tượng (OOAD), sử dụng các biểu đồ UML (Use Case, Class, Sequence) để mô hình hóa.
- Lập trình & Kiểm thử: Xây dựng ứng dụng bằng ReactJS, NodeJS, Express, MySQL, sau đó tiến hành kiểm thử đơn vị và tích hợp để đảm bảo tính ổn định của hệ thống.

D. Cấu trúc đề án

Đề án tốt nghiệp được chia thành 5 chương chính:

- **Chương 1:** Cơ sở lý thuyết – Trình bày các công nghệ và công cụ được sử dụng trong quá trình phát triển website.

- **Chương 2:** Phân tích và đặc tả chức năng hệ thống – Xác định các yêu cầu, tác nhân và luồng chức năng chính.
- **Chương 3:** Thiết kế chức năng và cơ sở dữ liệu – Trình bày mô hình cơ sở dữ liệu, thiết kế giao diện và sơ đồ hệ thống.
- **Chương 4:** Kiểm thử và triển khai hệ thống – Trình bày các kết quả kiểm thử, mô tả quá trình cài đặt và vận hành thực tế.
- **Chương 5:** Kết luận và hướng phát triển – Tổng kết kết quả đạt được, nêu ra hạn chế và định hướng mở rộng trong tương lai.

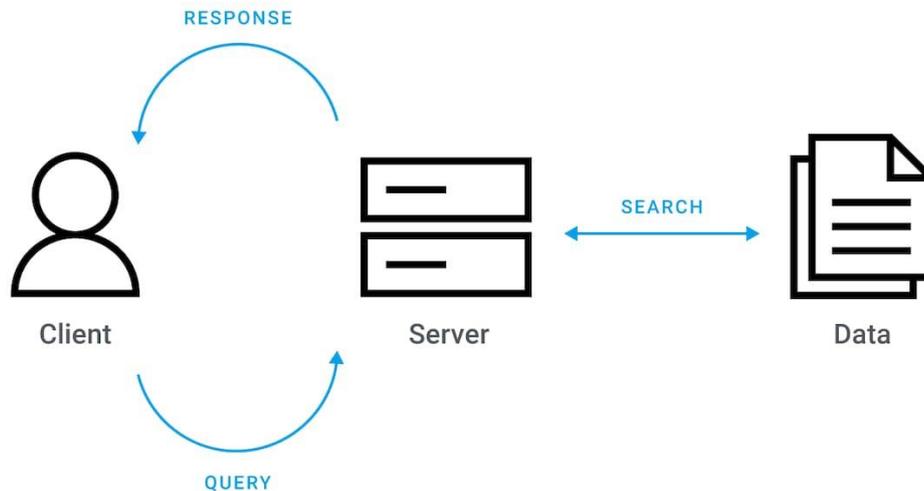
E. Tình hình nghiên cứu trong và ngoài nước

- Trong nước: Các nền tảng đặt đồ ăn như GrabFood, ShopeeFood, Baemin đã trở nên quen thuộc với người dùng Việt Nam. Tuy nhiên, đa số các hệ thống này mang tính thương mại quy mô lớn, không mở mã nguồn và không phù hợp cho mục tiêu học tập, nghiên cứu hoặc triển khai độc lập cho từng đơn vị. Việc xây dựng một hệ thống đặt đồ ăn online cho **hiều nhà hàng hoạt động song song** vẫn còn ít được triển khai trong môi trường giáo dục, nên đây là hướng nghiên cứu mới và thực tiễn..
- Ngoài nước: Trên thế giới, nhiều hệ thống đặt món trực tuyến đã ứng dụng trí tuệ nhân tạo (AI) và học máy (Machine Learning) để gợi ý món ăn, tối ưu quy trình giao hàng và cá nhân hóa trải nghiệm người dùng. Các mô hình kinh doanh như Uber Eats, DoorDash hay Zomato đã chứng minh hiệu quả và tiềm năng lớn của thị trường này. Việc nghiên cứu và tham khảo mô hình hoạt động của các nền tảng này giúp đề tài có cơ sở lý luận vững chắc trong việc thiết kế hệ thống đa nhà hàng, mở rộng quy mô và tối ưu hiệu năng..

CHƯƠNG I: CƠ SỞ LÝ THUYẾT

1.1. Mô hình Client – Server trong phát triển web

What is the client/server model?



Ảnh 1.1: Mô hình Client – Server-Data

Trong lĩnh vực phát triển phần mềm nói chung và lập trình web nói riêng, mô hình Client – Server là một kiến trúc cơ bản và phổ biến nhất. Kiến trúc này phân chia rõ ràng vai trò giữa hai thành phần chính: **Client** (máy khách) và **Server** (máy chủ), giúp hệ thống hoạt động hiệu quả, linh hoạt và dễ dàng mở rộng.

- **Client (Máy khách):** Là phía người dùng tương tác trực tiếp, trong dự án này chính là trình duyệt web mà người dùng sử dụng để truy cập website. Client có nhiệm vụ gửi yêu cầu (request) đến Server thông qua giao thức HTTP/HTTPS. Giao diện người dùng của dự án được phát triển bằng công nghệ **ReactJS**, kết hợp với HTML, CSS và JavaScript để tạo ra trải nghiệm tương tác và linh hoạt.
- **Server (Máy chủ):** Là nơi xử lý logic nghiệp vụ, tương tác với cơ sở dữ liệu và phản hồi lại các yêu cầu từ Client. Trong dự án này, Server được xây dựng bằng Node.js và framework Express.js. Server chịu trách nhiệm xử lý các yêu-cầu như xác thực người dùng, quản lý nhà hàng, xử lý đơn đặt hàng và truy vấn dữ liệu từ cơ sở dữ liệu.

1.1.1. Khái niệm về mô hình Client – Server

Mô hình Client – Server là một mô hình phân tán, trong đó Client là bên gửi yêu cầu dịch vụ và Server là bên cung cấp dịch vụ đó. Khi người dùng thực hiện một hành động trên website (ví dụ: **xem danh sách nhà hàng, đăng nhập, đặt món ăn**), các hành động đó sẽ được trình duyệt (Client) gửi dưới dạng yêu cầu HTTP tới Server.

Sau khi nhận và xử lý yêu cầu, Server sẽ gửi phản hồi (response) về cho Client, thường dưới dạng dữ liệu **JSON**. Client sẽ nhận dữ liệu này và sử dụng nó để cập nhật giao diện, hiển thị thông tin tương ứng cho người dùng.

Một đặc điểm quan trọng của mô hình này là Client không trực tiếp truy cập vào cơ sở dữ liệu. Toàn bộ tương tác với dữ liệu đều phải thông qua các API do Server cung cấp, giúp đảm bảo tính bảo mật, toàn vẹn và quản lý dữ liệu một cách tập trung.

1.1.2. Vai trò và chức năng của Client

Client là phần giao diện người dùng (UI – User Interface), nơi người dùng tương tác trực tiếp với hệ thống. Trong dự án này, phần Client được xây dựng bằng **React.js** – một thư viện JavaScript mạnh mẽ để xây dựng các giao diện người dùng có tính tương tác cao.

- Các chức năng chính của Client trong dự án bao gồm:
 - Hiển thị giao diện thân thiện, cho phép người dùng xem danh sách nhà hàng, thực đơn, và quản lý giỏ hàng.
 - Gửi các yêu cầu HTTP đến Server khi người dùng thực hiện các hành động như đăng nhập, đặt hàng, hoặc tìm kiếm. Các phương thức phổ biến được dùng là GET, POST, PUT, DELETE.
 - Nhận dữ liệu (thường ở dạng JSON) từ Server và cập nhật lại giao diện một cách linh hoạt mà không cần tải lại trang.
 - Thực hiện điều hướng giữa các trang (ví dụ: từ trang chủ đến trang chi tiết nhà hàng), quản lý trạng thái của ứng dụng (state), và xử lý các sự kiện từ người dùng.
- Với việc sử dụng công nghệ Single Page Application (SPA), ứng dụng chỉ tải trang một lần duy nhất và sau đó cập nhật nội dung một cách linh hoạt, mang đến trải nghiệm mượt mà và nhanh chóng cho người dùng.

1.1.3. Vai trò và chức năng của Server

Server là nơi tiếp nhận, xử lý các yêu cầu từ Client và gửi phản hồi tương ứng. Trong dự án này, Server được xây dựng bằng **Node.js** kết hợp với framework **Express.js**.

- Các chức năng chính của Server bao gồm:
 - Xử lý logic nghiệp vụ (business logic) như: xử lý đăng ký/đăng nhập, quản lý gi-ỏ hàng, tạo và quản lý đơn hàng, phê duyệt nhà hàng mới.
 - Quản lý và tương tác với cơ sở dữ liệu (thêm, sửa, xóa, truy vấn dữ liệu liên quan đến người dùng, nhà hàng, món ăn, đơn hàng).
 - Cung cấp các API theo tiêu chuẩn RESTful để Client có thể dễ dàng giao tiếp và sử dụng tài nguyên.
 - Bảo mật hệ thống bằng các phương pháp xác thực (authentication) để xác định danh tính người dùng và phân quyền (authorization) để kiểm soát quyền truy cập tài nguyên (ví dụ: chỉ Admin mới có quyền duyệt nhà hàng).
 - Xử lý việc tải lên và lưu trữ các tệp tin (uploads) như hình ảnh đại diện của nhà hàng hoặc món ăn.
- Server hoạt động trên một cổng (port) riêng, liên tục lắng nghe các yêu cầu HTTP đến và gửi phản hồi dữ liệu về cho Client.

1.1.4. Nguyên lý hoạt động

- Server hoạt động trên một cổng (port) riêng, liên tục lắng nghe các yêu cầu HTTP đến và gửi phản hồi dữ liệu về cho Client.
 - Người dùng thực hiện thao tác trên giao diện, ví dụ: nhấn nút "Thêm vào giỏ hàng" cho một món ăn.
 - Client (ReactJS) sẽ tạo một yêu cầu HTTP (cụ thể là phương thức POST) chứa thông tin về món ăn đó và gửi đến Server.
 - Server (Node.js/Express) nhận yêu cầu, xác thực người dùng, sau đó thực hiện xử lý logic: tìm thông tin người dùng và món ăn, rồi thêm món ăn đó vào giỏ hàng tương ứng trong cơ sở dữ liệu.

- Sau khi xử lý xong, Server gửi lại một phản hồi HTTP (HTTP response), thường kèm theo dữ liệu giỏ hàng đã được cập nhật ở định dạng JSON.
- Client nhận phản hồi này và cập nhật lại giao diện người dùng, ví dụ: cập nhật số lượng sản phẩm trên biểu tượng giỏ hàng.

1.1.5. Ưu điểm của mô hình Client – Server

- **Tính phân tách rõ ràng:** Tách biệt hoàn toàn giữa giao diện (Frontend) và phần xử lý logic (Backend), giúp cho việc phát triển, bảo trì và nâng cấp từng phần trở nên độc lập và dễ dàng hơn.
- **Tái sử dụng cao:** Backend API được xây dựng có thể phục vụ cho nhiều loại Client khác nhau trong tương lai (ví dụ: ứng dụng di động, ứng dụng desktop) mà không cần xây dựng lại logic.
- **Dễ mở rộng (Scalability):** Khi hệ thống có nhiều người dùng hơn, chúng ta có thể dễ dàng nâng cấp hiệu năng của Server (scale-up) hoặc triển khai thêm nhiều máy chủ (scale-out) mà không ảnh hưởng đến Client.
- **Bảo mật tập trung:** Toàn bộ dữ liệu và logic nghiệp vụ quan trọng được quản lý tập trung tại Server, giúp việc kiểm soát truy cập và bảo vệ dữ liệu trở nên hiệu quả hơn.

1.2. Frontend

1.2.1. React.js

Khái niệm

React.js (hay React) là một thư viện JavaScript mã nguồn mở do Facebook phát triển, ra mắt lần đầu vào năm 2013. React được sử dụng để xây dựng giao diện người dùng (UI) cho các ứng dụng web. Thay vì thao tác trực tiếp với DOM (Document Object Model) của trình duyệt, React sử dụng một cơ chế gọi là **Virtual DOM** để tối ưu hóa hiệu suất và tốc độ cập nhật giao diện.

React không phải là một framework toàn diện như Angular, mà chỉ tập trung vào lớp View (giao diện) trong mô hình MVC, cho phép lập trình viên dễ dàng tạo ra các thành phần giao diện có khả năng tái sử dụng cao.

Đặc điểm nổi bật của React.js

- **Virtual DOM:** React tạo ra một bản sao của DOM thật trong bộ nhớ. Khi trạng thái (state) của một component thay đổi, React sẽ tính toán sự khác biệt và chỉ cập nhật những phần cần thiết trên DOM thật, giúp ứng dụng hoạt động nhanh và mượt mà hơn.
- **Kiến trúc dựa trên Component:** Giao diện trong React được chia thành các thành phần (component) độc lập và có thể tái sử dụng. Mỗi component quản lý trạng thái riêng, giúp mã nguồn trở nên rõ ràng, dễ quản lý và bảo trì.
- **JSX (JavaScript XML):** Là một phần mở rộng của JavaScript, cho phép viết mã trông giống HTML ngay trong các tệp JavaScript. JSX giúp việc mô tả giao diện trở nên trực quan và dễ hiểu hơn.
- **Ràng buộc dữ liệu một chiều (One-Way Data Binding):** Dữ liệu trong React chảy theo một chiều, từ component cha xuống các component con. Luồng dữ liệu này giúp việc theo dõi và gỡ lỗi trở nên dễ dàng hơn.
- **Hooks:** Kể từ phiên bản 16.8, Hooks (ví dụ: useState, useEffect, useContext) cho phép sử dụng state và các tính năng khác của React trong các function component mà không cần viết class component.

Ứng dụng của React trong chuyên đề

Trong dự án website nhà hàng này, React.js được sử dụng để:

- Xây dựng toàn bộ giao diện người dùng, bao gồm các trang như: Trang chủ, Trang chi tiết nhà hàng, Giỏ hàng, Đăng nhập, Đăng ký, Trang thông tin người dùng, và các trang quản lý dành cho Chủ nhà hàng và Admin.
- Tạo ra các component có khả năng tái sử dụng cao như: Header, Footer, RestaurantCard (thẻ nhà hàng), MenuItemCard (thẻ món ăn), Sidebar (thanh điều hướng bên), Modal (hộp thoại)...
- Sử dụng thư viện **Axios** để gửi yêu cầu HTTP và nhận dữ liệu từ Backend API.
- Sử dụng **React Router DOM** (react-router-dom) để xử lý việc điều hướng, chuyển trang trong ứng dụng.

- Quản lý trạng thái toàn cục của ứng dụng, chẳng hạn như thông tin người dùng đang đăng nhập (AuthContext) và dữ liệu giỏ hàng (CartContext) bằng **Context API** và hook useContext.

1.2.2. React Router

Khái niệm

React Router là một thư viện định tuyến (routing) tiêu chuẩn cho các ứng dụng React. Nó cho phép xây dựng các ứng dụng đơn trang (SPA) với trải nghiệm điều hướng mượt mà như một trang web đa trang truyền thống. Bằng cách đồng bộ giao diện người dùng với URL của trình duyệt, React Router giúp quản lý việc di chuyển giữa các component khác nhau mà không cần tải lại trang.

Cách hoạt động

React Router theo dõi sự thay đổi của URL trong trình duyệt, sau đó ánh xạ các đường dẫn (path) này với các component React tương ứng đã được cấu hình để hiển thị. Nó sử dụng History API của trình duyệt để quản lý lịch sử điều hướng, cho phép người dùng sử dụng các nút back/forward một cách tự nhiên

Các thành phần chính được sử dụng trong dự án:

- `<BrowserRouter>`: Component bao bọc toàn bộ ứng dụng để kích hoạt chức năng định tuyến.
- `<Routes>`: Component dùng để nhóm các định nghĩa `<Route>`.
- `<Route>`: Định nghĩa một tuyến đường, ánh xạ một path (đường dẫn URL) tới một element (component) cụ thể.
- `<Link>`: Component thay thế cho thẻ `<a>` truyền thống, cho phép chuyển hướng giữa các trang mà không gây tải lại trang.
- Các Hooks: `useNavigate()` để điều hướng theo lập trình, `useParams()` để lấy các tham số động từ URL (ví dụ: ID của nhà hàng), và `useLocation()` để truy cập thông tin về URL hiện tại.

Ứng dụng trong đồ án

Trong dự án website nhà hàng, React Router được sử dụng để:

- Tạo hệ thống định tuyến cho toàn bộ ứng dụng với các trang chính như:
 - `/`: Trang chủ hiển thị danh sách nhà hàng.
 - `/restaurants/:id`: Trang chi tiết của một nhà hàng cụ thể.
 - `/cart`: Trang giỏ hàng.
 - `/login, /register`: Các trang đăng nhập, đăng ký.
 - `/profile, /orders`: Các trang quản lý thông tin cá nhân và lịch sử đơn hàng của người dùng.
 - Các tuyến đường được bảo vệ cho Admin và Chủ nhà hàng (ví dụ: `/admin/dashboard, /owner/menu-manager`).
- Sử dụng `useNavigate()` để tự động chuyển hướng người dùng sau khi thực hiện các hành động như **đăng nhập, đăng ký thành công, hoặc sau khi đặt hàng**.
- Sử dụng `useParams()` để lấy id của nhà hàng từ URL, từ đó gọi API để tải và hiển thị đúng thông tin chi tiết của nhà hàng đó.

1.2.3. Context API

Khái niệm

Context API là một cơ chế được tích hợp sẵn trong React, cho phép truyền dữ liệu giữa các component mà không cần phải sử dụng phương pháp truyền props qua từng cấp (một vấn đề thường được gọi là "prop drilling"). Thay vì truyền dữ liệu qua nhiều lớp component trung gian, Context API cho phép chia sẻ "trạng thái toàn cục" (global state) từ một nơi trung tâm đến bất kỳ component nào trong cây component có nhu cầu sử dụng.

Context API đặc biệt hữu ích để quản lý những dữ liệu cần được truy cập ở nhiều nơi trong ứng dụng, ví dụ như: thông tin người dùng đang đăng nhập, trạng thái giỏ hàng, chủ đề giao diện (theme), hoặc ngôn ngữ hiển thị.

Ưu điểm của Context API

- Giúp mã nguồn sạch hơn bằng cách loại bỏ việc truyền props không cần thiết.
- Cung cấp một giải pháp quản lý trạng thái đơn giản và hiệu quả cho các ứng dụng vừa và nhỏ.

- Là một tính năng có sẵn trong React, không cần cài đặt thêm thư viện bên ngoài như Redux.
- Dễ dàng kết hợp với React Hooks (cụ thể là hook useContext) để giúp việc truy cập dữ liệu từ Context trở nên ngắn gọn và trực quan.

Ứng dụng trong dự án

Trong dự án website nhà hàng, Context API đóng vai trò trung tâm trong việc quản lý trạng thái của ứng dụng, cụ thể qua hai context chính:

- **AuthContext (AuthContext.js):**
 - Quản lý trạng thái xác thực của người dùng trên toàn bộ ứng dụng (đã đăng nhập hay chưa, thông tin người dùng, vai trò - user/owner/admin).
 - Cung cấp thông tin này cho các component khác như Header (để hiển thị nút đăng nhập/đăng xuất) và ProtectedRoute (để kiểm tra quyền truy cập vào các trang quản trị).
- **CartContext (CartContext.js):**
 - Quản lý toàn bộ dữ liệu của giỏ hàng, bao gồm danh sách các món ăn, số lượng, và tổng tiền.
 - Cho phép các component khác nhau như MenuItemCard (nút "Thêm vào giỏ"), CartPage (trang giỏ hàng), và Header (hiển thị số lượng món trong giỏ) có thể cùng lúc truy cập và thay đổi dữ liệu giỏ hàng một cách nhất quán.

Việc áp dụng Context API giúp giảm độ phức tạp trong việc quản lý và đồng bộ hóa dữ liệu, làm cho mã nguồn trở nên dễ bảo trì và mở rộng hơn.

1.3. Backend

1.3.1. Node.js

Khái niệm

Node.js là một môi trường thực thi (runtime environment) cho phép chạy mã JavaScript ở phía máy chủ (server-side), được xây dựng trên V8 JavaScript Engine của Google. Khác với JavaScript truyền thống chỉ hoạt động trên trình duyệt, Node.js cho phép

lập trình viên sử dụng JavaScript để xây dựng các ứng dụng backend mạnh mẽ, ví dụ như RESTful API, ứng dụng thời gian thực, và microservices.

Node.js sử dụng mô hình I/O không chặn (non-blocking I/O) và kiến trúc hướng sự kiện (event-driven), giúp nó có khả năng xử lý hàng nghìn kết nối đồng thời một cách hiệu quả, rất phù hợp cho các ứng dụng web đòi hỏi hiệu suất cao.

Ứng dụng trong dự án

Trong dự án website nhà hàng này, Node.js đóng vai trò là nền tảng chính cho toàn bộ hệ thống backend, với các nhiệm vụ:

- Xây dựng một hệ thống RESTful API hoàn chỉnh để phía frontend (React) có thể giao tiếp, gửi và nhận dữ liệu.
- Xử lý toàn bộ logic nghiệp vụ của ứng dụng, bao gồm: **đăng ký, đăng nhập, quản lý người dùng, quản lý nhà hàng, món ăn, giỏ hàng và xử lý đơn hàng.**
- Tương tác với cơ sở dữ liệu **MySQL** thông qua một ORM (Object-Relational Mapping) là Sequelize để thực hiện các thao tác truy vấn và cập nhật dữ liệu.
- Quản lý các tuyến đường (routes) của API một cách rõ ràng và có tổ chức.

1.3.2. Express.js

Khái niệm

Express.js là một framework web tối giản và linh hoạt được xây dựng trên nền tảng Node.js. Nó cung cấp một bộ công cụ mạnh mẽ để phát triển các ứng dụng web và API một cách nhanh chóng. Express là thành phần chính trong ngăn xếp công nghệ **MERN/MEVN** (MongoDB/MySQL, Express, React/Vue, Node.js) và là một trong những framework phổ biến nhất trong hệ sinh thái Node.js

Ưu điểm của Express.js

- Đơn giản và linh hoạt: Cung cấp các tính năng cốt lõi mà không áp đặt một cấu trúc cứng nhắc, cho phép lập trình viên tự do tổ chức dự án.
- Quản lý Route mạnh mẽ: Dễ dàng định nghĩa các điểm cuối (endpoint) cho API, hỗ trợ tất cả các phương thức HTTP và các mẫu URL động.

- Hệ thống Middleware: Đây là tính năng nổi bật nhất của Express. Middleware là các hàm được thực thi tuần tự trong chu trình xử lý request, cho phép thực hiện các tác vụ như xác thực, ghi log, kiểm tra dữ liệu đầu vào, và xử lý lỗi.
- Cộng đồng lớn: Có một hệ sinh thái rộng lớn các thư viện và công cụ hỗ trợ.

Ứng dụng trong chuyên đề

Trong dự án này, Express.js được sử dụng để:

- Định nghĩa và tổ chức toàn bộ các API endpoint để quản lý người dùng, nhà hàng, thực đơn, và các đơn đặt hàng.
- Sử dụng các middleware để xử lý yêu cầu. Ví dụ:
 - Middleware `isAuth` để kiểm tra xem người dùng đã đăng nhập hay chưa bằng cách xác thực JSON Web Token (JWT).
 - Middleware `isAdmin` và `isOwner` để thực hiện phân quyền, đảm bảo chỉ những người dùng có vai trò phù hợp mới có thể truy cập các tài nguyên quản trị.
 - Kết nối với cơ sở dữ liệu MySQL để truy xuất và cập nhật dữ liệu theo yêu cầu từ client.
 - Xử lý việc tải lên các tệp tin như hình ảnh món ăn, hình ảnh nhà hàng.

1.3.3. Mô hình kiến trúc Backend

Dự án áp dụng một kiến trúc phần mềm phía backend dựa trên mô hình MVC (Model - View - Controller) được biến thể để phù hợp với việc xây dựng RESTful API. Cấu trúc này giúp phân tách rõ ràng các thành phần của ứng dụng, làm cho mã nguồn trở nên có tổ chức, dễ đọc và dễ bảo trì.

Các thành phần được tổ chức như sau:

- **Model:**
 - Vị trí: Thư mục `backend/models`.
 - Nhiệm vụ: Định nghĩa cấu trúc dữ liệu của các đối tượng trong ứng dụng (ví dụ: `User`, `Restaurant`, `Order`). Model chịu trách nhiệm cho tất cả các tương tác với cơ sở dữ liệu như tạo, đọc, cập nhật, và xóa (CRUD). Trong dự án này, các model của Sequelize định nghĩa bảng và các mối quan hệ trong DB.

- **Controller:**

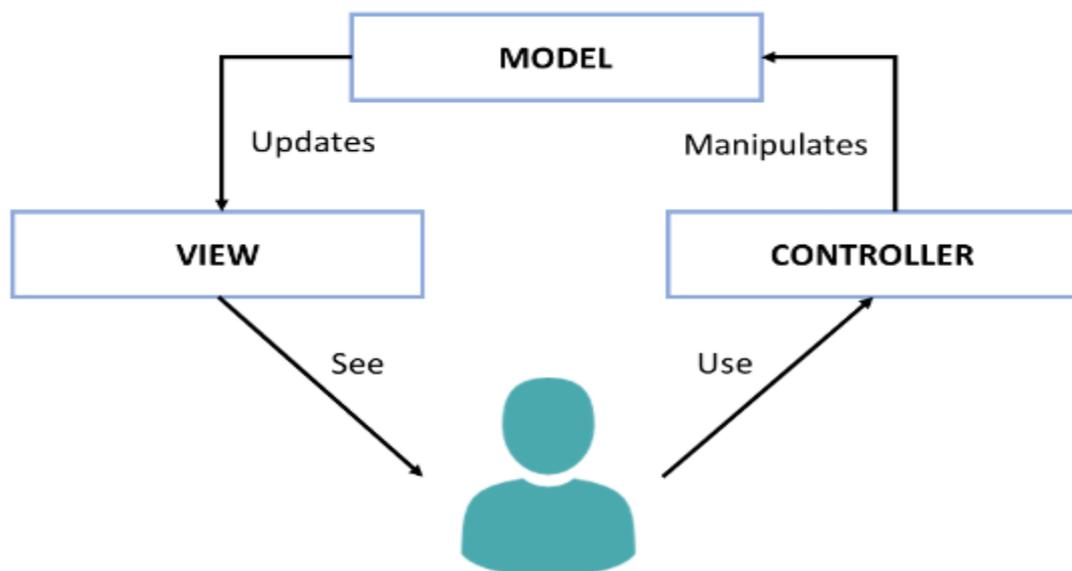
- **Vị trí:** Thư mục backend/controllers.
- **Nhiệm vụ:** Chứa đựng logic nghiệp vụ chính của ứng dụng. Mỗi controller nhận yêu cầu được chuyển tiếp từ router, sử dụng các hàm trong **Model** để xử lý dữ liệu, sau đó chuẩn bị và gửi phản hồi về cho client. Ví dụ, auth.controller.js xử lý logic đăng ký và đăng nhập.

- **View:**

- **Khái niệm:** Trong kiến trúc REST API, không có "View" truyền thống tạo ra giao diện HTML. Thay vào đó, View chính là dữ liệu được gửi về cho client, thường ở định dạng JSON. Controller sau khi xử lý xong sẽ gửi một phản hồi JSON, và phía client (React) sẽ chịu trách nhiệm "vẽ" ra giao diện người dùng dựa trên dữ liệu đó.

- **Routes:**

- **Vị trí:** Thư mục backend/routes.
- **Nhiệm vụ:** Đóng vai trò điều hướng. Nó định nghĩa các URL endpoint và các phương thức HTTP tương ứng (GET, POST, PUT, DELETE), sau đó ánh xạ mỗi yêu cầu đến một hàm xử lý (handler) cụ thể trong Controller.



Ảnh 1.2: Mô hình MVC

1.3.4. Middleware

Khái niệm

Middleware là một khái niệm phổ biến trong lập trình backend, đặc biệt trong các framework như Express.js. Middleware có thể hiểu là một hàm trung gian được gọi trong quá trình xử lý HTTP request và response. Nó có nhiệm vụ can thiệp, kiểm tra, xử lý dữ liệu trước khi request được chuyển tiếp đến route chính hoặc khi response được trả về client.

Trong Express.js, middleware có thể thực hiện nhiều chức năng như:

- Kiểm tra xác thực người dùng (authentication).
- Ghi log thông tin yêu cầu.
- Phân tích body của request (body parsing).
- Xử lý lỗi.
- Hạn chế số lượng truy cập (rate limiting), kiểm tra quyền (authorization), hoặc xác minh token (JWT).

Cơ chế hoạt động

Middleware hoạt động theo chuỗi: Mỗi middleware sau khi xử lý xong có thể gọi next() để chuyển sang middleware tiếp theo. Nếu không gọi next(), quá trình sẽ dừng lại tại middleware đó. Điều này giúp kiểm soát linh hoạt luồng xử lý của ứng dụng.

Ưu điểm của Middleware

Tái sử dụng cao: Một middleware có thể áp dụng cho nhiều route.

Tăng bảo mật và kiểm soát: Nhờ có thể kiểm tra token, quyền truy cập, và trạng thái request trước khi xử lý chính.

Giúp mã nguồn rõ ràng, dễ tổ chức: Tách biệt các bước xử lý request thành từng phần riêng biệt.

Dễ mở rộng và bảo trì: Có thể thêm mới chức năng mà không ảnh hưởng logic cũ.

Ứng dụng trong chuyên đề

Trong chuyên đề xây dựng website bán hàng, Middleware đóng vai trò quan trọng trong:

- Xác thực người dùng thông qua JWT để đảm bảo chỉ người đã đăng nhập mới được thực hiện các thao tác như đặt thuê, xem thông tin tài khoản.
- Phân quyền người dùng, ví dụ: chỉ admin mới được thêm/sửa sản phẩm.
- Kiểm tra dữ liệu đầu vào (validate) như kiểm tra email, mật khẩu, tên sản phẩm...
- Xử lý lỗi chung, trả về thông báo lỗi rõ ràng và có định dạng cho frontend.
- Bảo vệ API khỏi các cuộc tấn công như spam hoặc gửi nhiều request bất hợp pháp.

Middleware giúp tăng độ bảo mật, hiệu quả và khả năng mở rộng của hệ thống backend.

1.3.5. Kết nối Frontend – Backend

Khái niệm

Trong kiến trúc ứng dụng web hiện đại, Frontend và Backend là hai hệ thống độc lập nhưng giao tiếp chặt chẽ với nhau.

- **Frontend (Client-side):** Là giao diện người dùng mà người dùng tương tác, được xây dựng bằng React.
- **Backend (Server-side):** Là máy chủ xử lý logic nghiệp vụ, quản lý dữ liệu và cung cấp API.

Quá trình kết nối giữa chúng được thực hiện thông qua giao thức HTTP, nơi Frontend gửi yêu cầu đến các API do Backend cung cấp để trao đổi dữ liệu.

Cơ chế kết nối phổ biến

- **Giao tiếp qua RESTful API:** Backend xây dựng một hệ thống API theo tiêu chuẩn REST. Frontend sử dụng thư viện Axios để gửi các yêu cầu HTTP (GET, POST, PUT, DELETE) đến các API này.
- **Trao đổi dữ liệu bằng JSON:** Dữ liệu được trao đổi giữa hai bên chủ yếu dưới định dạng JSON, một định dạng nhẹ và dễ dàng cho JavaScript xử lý.
- **Xử lý CORS (Cross-Origin Resource Sharing):** Vì Frontend (chạy trên một port, ví dụ localhost:3000) và Backend (chạy trên port khác, ví dụ localhost:5000) là hai "nguồn" khác nhau, Backend phải cấu hình middleware CORS để cho phép trình duyệt chấp nhận các yêu cầu từ Frontend.

- **Xác thực bằng Token (JWT):** Sau khi người dùng đăng nhập thành công, Backend sẽ cấp một JWT (JSON Web Token). Frontend lưu token này lại (thường trong Local Storage) và gửi kèm nó trong header của mỗi yêu cầu cần xác thực sau đó.

Vai trò trong dự án chuyên đề

Việc kết nối này là xương sống của toàn bộ ứng dụng:

- **Tải dữ liệu động:** Khi người dùng truy cập trang chủ, Frontend gọi API để lấy và hiển thị **danh sách các nhà hàng** từ cơ sở dữ liệu.
- **Thực hiện hành động:** Khi người dùng **thêm một món ăn vào giỏ hàng** hoặc **tạo đơn hàng**, Frontend gửi thông tin đó đến API của Backend để xử lý và lưu trữ.
- **Đồng bộ trạng thái:** Thông tin người dùng, nội dung giỏ hàng, và trạng thái đơn hàng luôn được đồng bộ giữa giao diện và cơ sở dữ liệu thông qua các lượt gọi API.
- **Hiển thị thông báo và lỗi:** Các phản hồi từ Backend (thành công hay thất bại) giúp Frontend hiển thị các thông báo phù hợp cho người dùng.

Công cụ và công nghệ sử dụng

Frontend: React.js kết hợp với thư viện Axios để gửi HTTP requests.

Backend: Express.js xây dựng hệ thống API RESTful.

Token (JWT): Dùng để xác thực và phân quyền giữa frontend và backend.

CORS Middleware: Được sử dụng tại backend để cho phép frontend truy cập từ domain khác.

1.4. Cơ sở dữ liệu MySQL

1.4.1. Giới thiệu MySQL

MySQL là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) mã nguồn mở phổ biến, sử dụng Ngôn ngữ Truy vấn có Cấu trúc (SQL). Nó được sử dụng rộng rãi trong các ứng dụng web nhờ tính ổn định, hiệu suất cao và khả năng tương thích tốt với các ngôn ngữ phía server như Node.js.

1.4.2. Đặc điểm nổi bật

- Dữ liệu được lưu trữ dưới dạng bảng (table) có cấu trúc.
- Hỗ trợ chuẩn SQL với đầy đủ các thao tác CRUD (Create, Read, Update, Delete).
- Cho phép liên kết dữ liệu giữa các bảng thông qua khóa ngoại (foreign key).
- Tính bảo mật cao, hỗ trợ phân quyền người dùng.
- Tương thích đa nền tảng (Windows, macOS, Linux).

1.4.3. Vai trò của MySQL trong dự án

Trong dự án xây dựng website nhà hàng, MySQL được dùng làm cơ sở dữ liệu chính, có nhiệm vụ:

- **Lưu trữ thông tin người dùng:** Tên, email, mật khẩu (đã mã hóa), vai trò (quyền truy cập)...
- **Quản lý nhà hàng và thực đơn:** Bao gồm thông tin về nhà hàng (tên, địa chỉ, ảnh) và các món ăn (tên, mô tả, giá, ảnh)...
- **Xử lý đơn hàng:** Lưu thông tin giỏ hàng, chi tiết đơn hàng, trạng thái đơn hàng và lịch sử đặt món.
- Các bảng chính thường có trong hệ thống gồm: users, restaurants, menuItems, orders, orderItems và cartItems.

1.5. Kiểm thử

1.5.1. Giới thiệu

Định nghĩa: Kiểm thử phần mềm (Software Testing) là hoạt động nhằm tìm kiếm và phát hiện lỗi của phần mềm. Đây là một tiến trình hoặc một tập hợp các tiến trình được thiết kế để đảm bảo chương trình máy tính thực hiện đúng theo yêu cầu của khách hàng. Theo "The Art of Software Testing", kiểm thử phần mềm là quá trình thực thi một chương trình với mục đích tìm ra lỗi.

Mục tiêu: Phát hiện sớm các lỗi và tìm được càng nhiều lỗi càng tốt để tránh sai sót, sửa đi sửa lại nhiều lần. Ngăn chặn các lỗi phát sinh, hạn chế rủi ro cho hệ thống và chứng minh rằng sản phẩm phù hợp, hiệu quả và đúng với đặc tả của nó. Từ đó, chứng tỏ được rằng các trường hợp kiểm thử (test cases) và các báo cáo được tạo ra là đúng và hữu dụng.

1.5.2. Kiểm thử hộp đen (Black Box Testing)

Định nghĩa: Black box testing (Kiểm thử hộp đen) là một phương pháp kiểm thử phần mềm tập trung vào việc kiểm tra hành vi bên ngoài của một ứng dụng mà không quan tâm đến cấu trúc nội bộ hay chi tiết cài đặt của nó. Trong phương pháp này, người kiểm thử chỉ xem ứng dụng như một "hộp đen" và tập trung vào việc kiểm tra dữ liệu đầu vào, kết quả đầu ra, cũng như các tình huống và điều kiện biên của hệ thống.

Ưu điểm	Nhược điểm
<ul style="list-style-type: none">– Phát hiện các lỗi và hành vi không mong muốn từ góc độ người dùng.– Giúp đảm bảo tính tương tác mượt mà và đáp ứng nhanh chóng, mang lại trải nghiệm người dùng tốt.– Người kiểm thử không cần biết về ngôn ngữ lập trình hay cấu trúc mã nguồn.– Việc kiểm thử có thể được thực hiện bởi một nhóm độc lập, đảm bảo tính khách quan.– Có thể tiết kiệm thời gian và nguồn lực vì không cần phân tích mã nguồn phức tạp.	<ul style="list-style-type: none">– Không thể đảm bảo độ phủ kiểm thử cao như White box testing.– Có thể bỏ qua các lỗi nằm sâu trong hệ thống như lỗi logic hoặc lỗi thiết kế.– Khó kiểm tra được các tình huống hoặc các đường đi (path) phức tạp bên trong mã nguồn.– Chất lượng kiểm thử phụ thuộc nhiều vào khả năng và kinh nghiệm của người kiểm thử; nếu thiếu kỹ năng có thể dẫn đến bỏ sót lỗi.

Bảng.1: Ưu nhược điểm của kiểm thử hộp đen

Các phương pháp được sử dụng:

- **Equivalence Partitioning (Phân vùng tương đương):** Chia các giá trị đầu vào thành các nhóm tương đương và chỉ kiểm tra một giá trị đại diện trong mỗi nhóm. Điều này giúp giảm số lượng ca kiểm thử cần thực hiện mà vẫn đảm bảo bao quát các trường hợp quan trọng.
- **Boundary Value Analysis (Phân tích giá trị biên):** Tập trung vào việc kiểm tra các giá trị tại và gần các điểm biên của các vùng dữ liệu đầu vào, nơi thường xảy ra lỗi.

- **Decision Table Testing (Kiểm thử bảng quyết định):** Sử dụng các bảng quyết định để xác định các điều kiện và hành động tương ứng của ứng dụng. Các ca kiểm thử được tạo dựa trên các kết hợp của các điều kiện trong bảng.
- **State Transition Testing (Kiểm thử chuyển trạng thái):** Tập trung vào việc kiểm thử các quá trình chuyển đổi trạng thái của ứng dụng. Các ca kiểm thử được tạo dựa trên các trạng thái và các sự kiện gây ra sự chuyển đổi.
- **Error Guessing (Đoán lỗi):** Dựa trên kinh nghiệm và trực giác của người kiểm thử để dự đoán các lỗi có thể xảy ra trong ứng dụng.
- **Exploratory Testing (Kiểm thử khám phá):** Người kiểm thử đồng thời thiết kế và thực thi các bài kiểm thử để khám phá ứng dụng một cách tự do, không theo kịch bản cứng nhắc.
- **Regression Testing (Kiểm thử hồi quy):** Đảm bảo rằng các thay đổi hoặc sửa lỗi trong ứng dụng không gây ảnh hưởng tiêu cực đến các chức năng đã hoạt động tốt trước đó.

CHƯƠNG II : PHÂN TÍCH ĐẶC TẢ CHỨC NĂNG CỦA HỆ THỐNG

2.1. Yêu cầu hệ thống

2.1.1. Yêu cầu chức năng

Chức năng chung:

- Đối với khách truy cập (chưa đăng nhập): Có thể xem danh sách các nhà hàng, tìm kiếm nhà hàng hoặc món ăn, và xem thông tin chi tiết của một nhà hàng (bao gồm địa chỉ, thực đơn, đánh giá).
- Yêu cầu đăng nhập: Hệ thống yêu cầu người dùng phải đăng nhập hoặc đăng ký tài khoản để thực hiện các chức năng như đặt món, quản lý giỏ hàng và quản lý thông tin cá nhân.

Chức năng phía Quản trị viên (Admin):

- Quản lý nhà hàng:
 - Xem danh sách tất cả các nhà hàng trên hệ thống.
 - Phê duyệt hoặc từ chối các yêu cầu đăng ký nhà hàng mới từ các chủ nhà hàng.
 - Vô hiệu hóa hoặc kích hoạt tài khoản nhà hàng.
- Quản lý người dùng:
 - Xem danh sách tất cả tài khoản người dùng và chủ nhà hàng.
 - Quản lý tài khoản: có thể khóa hoặc mở khóa tài khoản khi cần thiết.
- Quản lý đơn hàng: Xem và theo dõi tất cả các đơn hàng trên toàn hệ thống.
- Báo cáo và thống kê: Xem báo cáo thống kê tổng quan về doanh thu, số lượng đơn hàng, và hiệu suất hoạt động của toàn bộ nền tảng.

Chức năng phía Chủ nhà hàng (Owner):

- Quản lý thông tin nhà hàng: Cập nhật các thông tin của nhà hàng mình sở hữu như tên, địa chỉ, giờ mở cửa, ảnh đại diện.
- Quản lý thực đơn (Menu):
 - Xem, thêm mới, chỉnh sửa, và xóa các món ăn trong thực đơn của nhà hàng.

- Cập nhật giá, mô tả, và hình ảnh cho từng món ăn.
- Quản lý đơn hàng:
- Xem danh sách các đơn hàng được đặt tại nhà hàng của mình.
 - Cập nhật trạng thái đơn hàng (ví dụ: "Đang chuẩn bị", "Đang giao", "Hoàn thành").
- Thống kê: Xem báo cáo doanh thu và số lượng đơn hàng của riêng nhà hàng mình để theo dõi hiệu suất kinh doanh.

Chức năng phía Người dùng (Khách hàng):

- Xem và tìm kiếm:
- Xem danh sách các nhà hàng hiện có.
 - Sử dụng công cụ tìm kiếm để tìm nhà hàng hoặc món ăn cụ thể.
 - Lọc nhà hàng theo các tiêu chí (ví dụ: loại ẩm thực, khu vực).
- Giỏ hàng:
- Thêm các món ăn yêu thích vào giỏ hàng.
 - Xem, chỉnh sửa số lượng, hoặc xóa các món ăn trong giỏ hàng.
- Đặt hàng và thanh toán:
- Thực hiện quy trình đặt hàng từ giỏ hàng.
 - Cung cấp địa chỉ giao hàng và chọn phương thức thanh toán.
- Quản lý tài khoản:
- Cập nhật thông tin cá nhân như tên, số điện thoại, địa chỉ mặc định.
 - Thay đổi mật khẩu.
 - Xem lại lịch sử các đơn hàng đã đặt.
 - Đăng xuất khỏi tài khoản.

2.1.2. Yêu cầu phi chức năng

Yêu cầu phi chức năng đưa ra các tiêu chí nhằm đánh giá hoạt động của hệ thống, đảm bảo hệ thống vận hành ổn định, an toàn và hiệu quả.

Hệ thống website đặt món ăn cần đáp ứng được các yêu cầu phi chức năng sau:

– **Yêu cầu về bảo mật (Security):**

- Hệ thống phải bảo vệ thông tin người dùng và dữ liệu kinh doanh khỏi các hành vi truy cập trái phép, rò rỉ thông tin và các hình thức tấn công mạng.
- Xác thực và cấp quyền truy cập vào hệ thống cho người dùng (Admin, Chủ nhà hàng, Khách hàng) dựa trên vai trò và quyền hạn tương ứng.
- Mật khẩu người dùng phải được mã hóa trước khi lưu vào cơ sở dữ liệu.

– **Yêu cầu về sao lưu và phục hồi (Backup & Recovery):**

- Dữ liệu quan trọng của hệ thống (thông tin người dùng, đơn hàng, nhà hàng) cần có cơ chế sao lưu dự phòng định kỳ để tránh mất mát dữ liệu.
- Hệ thống phải có khả năng phục hồi dữ liệu từ các bản sao lưu khi có sự cố xảy ra.

– **Yêu cầu về tính khả dụng (Availability):**

- Hệ thống phải hoạt động ổn định và sẵn sàng 24/7 để người dùng có thể truy cập và đặt món bất cứ lúc nào.

– **Yêu cầu về tính dễ sử dụng (Usability):**

- Giao diện người dùng phải được thiết kế đơn giản, trực quan, và dễ sử dụng, giúp người dùng mới có thể nhanh chóng làm quen và thực hiện các thao tác chính.
- Hệ thống phải tương thích và hoạt động tốt trên các trình duyệt web phổ biến như Google Chrome, Cốc Cốc, và Firefox.

– **Yêu cầu về hiệu năng (Performance):**

- Thời gian phản hồi của hệ thống cho các tác vụ thông thường (như tải trang, xem thực đơn, thêm vào giỏ hàng) phải nhanh, dưới 3 giây.
- Hệ thống phải đảm bảo phục vụ được khoảng 50 người dùng truy cập và thực hiện các tác vụ đồng thời mà không bị chậm hoặc quá tải.

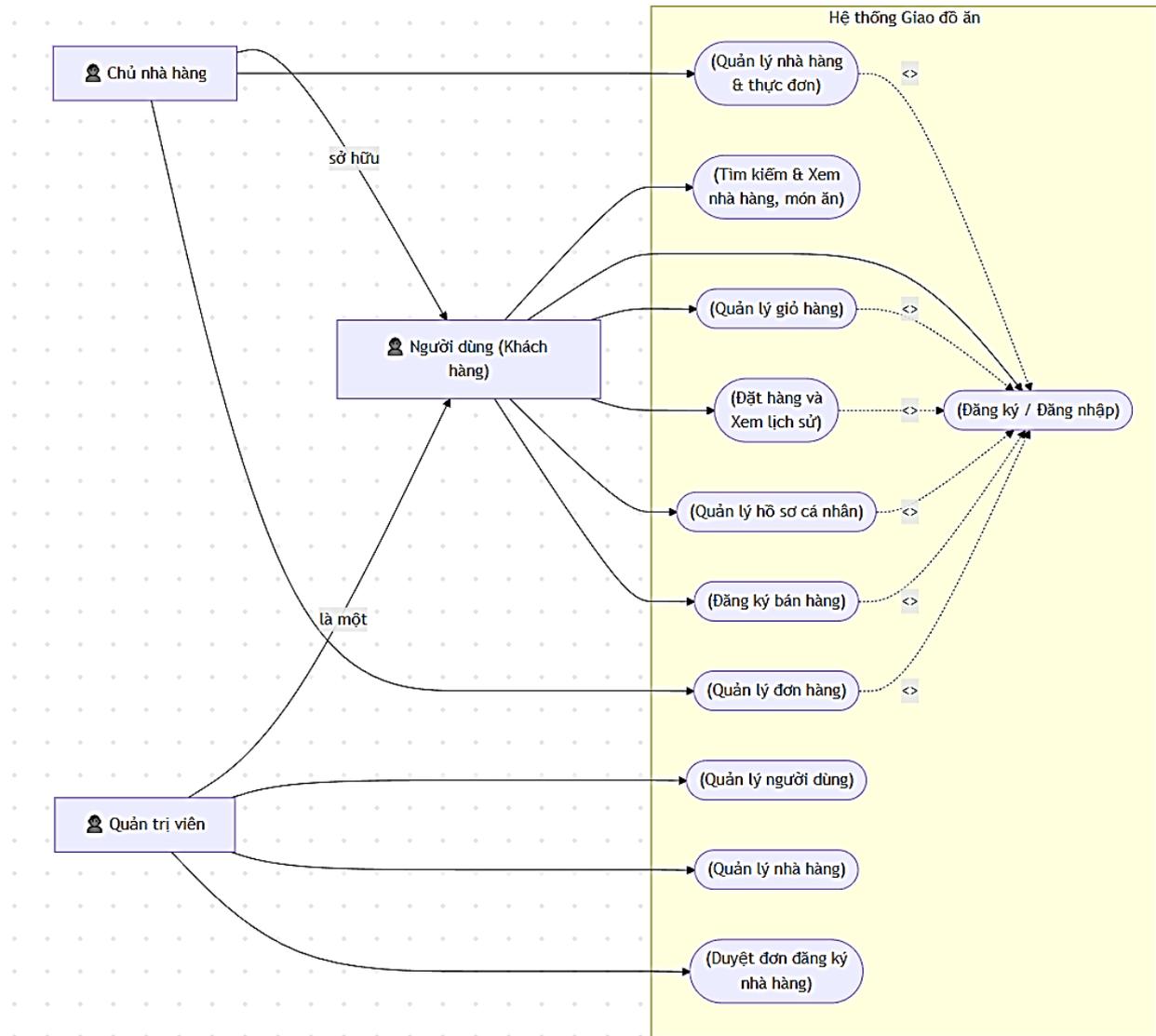
2.2. Usecase

2.2.1. Tác nhân hệ thống

STT	Tác nhân	Mô tả
1	Khách hàng vãng lai	Đây là tác nhân đại diện cho người dùng truy cập website nhưng chưa đăng nhập. Tác nhân này có thể xem danh sách nhà hàng, tìm kiếm nhà hàng hoặc món ăn, và xem thông tin chi tiết của một nhà hàng.
2	Người dùng	Đây là tác nhân đại diện cho khách hàng đã có tài khoản và đăng nhập vào hệ thống. Ngoài các quyền của Khách vãng lai, Người dùng có thể thực hiện các chức năng cốt lõi như: thêm món ăn vào giỏ hàng, đặt hàng, thanh toán, quản lý thông tin cá nhân và xem lại lịch sử đặt hàng.
3	Chủ nhà hàng	Đây là tác nhân đại diện cho người sở hữu và quản lý một hoặc nhiều nhà hàng trên nền tảng. Chủ nhà hàng có thể quản lý thông tin nhà hàng, quản lý thực đơn (thêm, sửa, xóa món ăn), xem và xử lý các đơn hàng thuộc nhà hàng của mình.
4	Quản trị viên	Đây là tác nhân đại diện cho người quản trị cấp cao nhất của hệ thống. Admin có toàn quyền quản lý, bao gồm: quản lý tất cả tài khoản (Người dùng, Chủ nhà hàng), phê duyệt nhà hàng mới, xem thống kê toàn hệ thống, và giám sát tất cả hoạt động trên nền tảng.

Bảng.1: Danh sách tác nhân

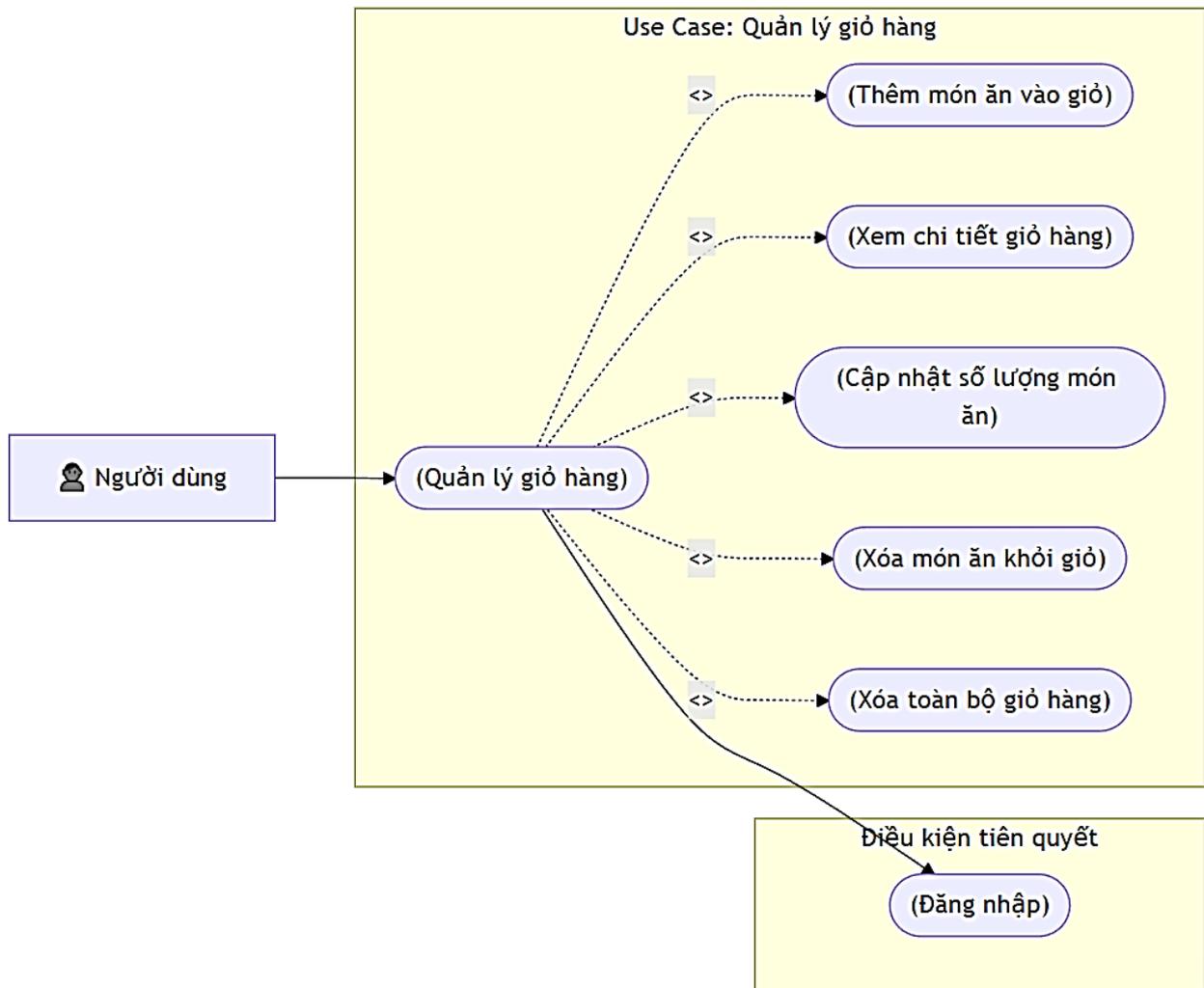
2.2.2. Biểu đồ usecase tổng quát



Ảnh 2.1: Biểu đồ usecase tổng quát

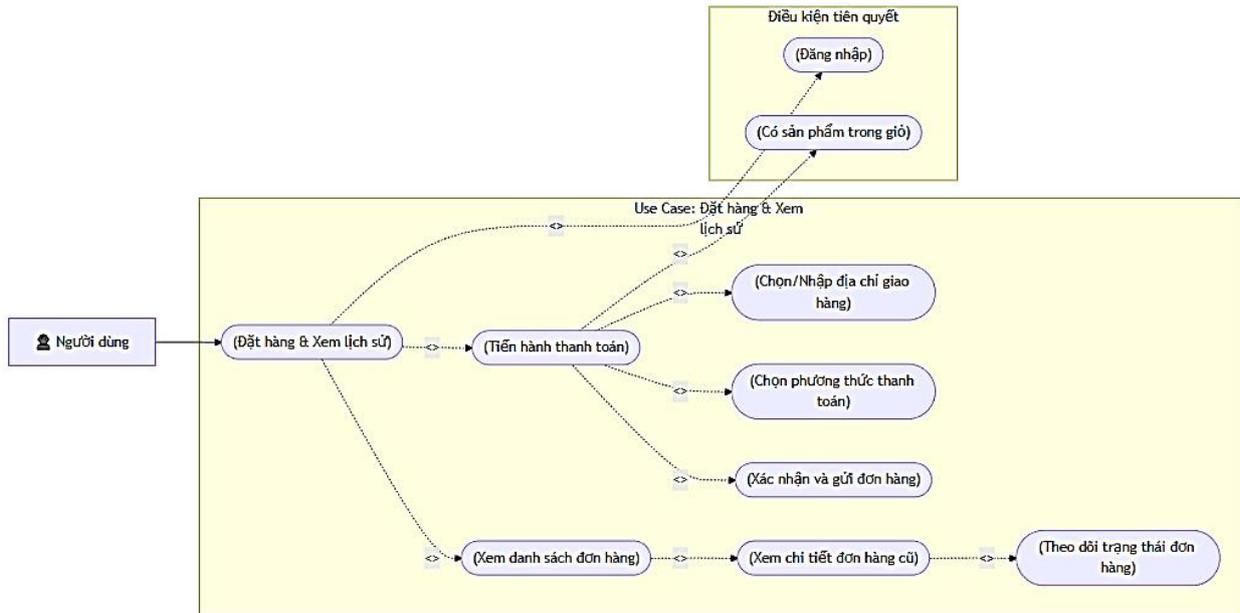
2.2.3. Biểu đồ usecase phân rã

Biểu đồ phân rã usecase Giỏ hàng



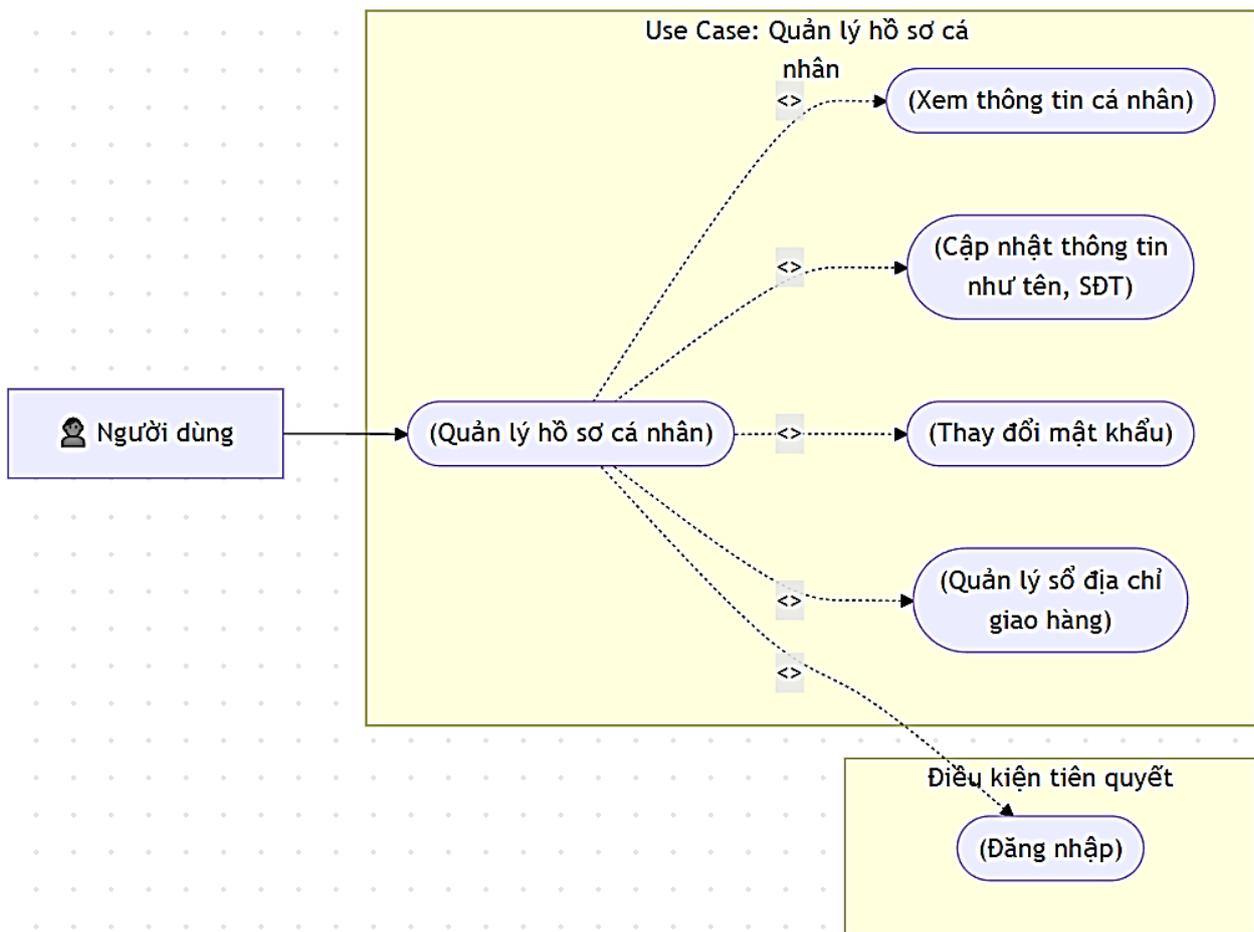
Ảnh 2.2: Biểu đồ phân rã usecase Giỏ hàng

Biểu đồ phân rã usecase Đặt hàng



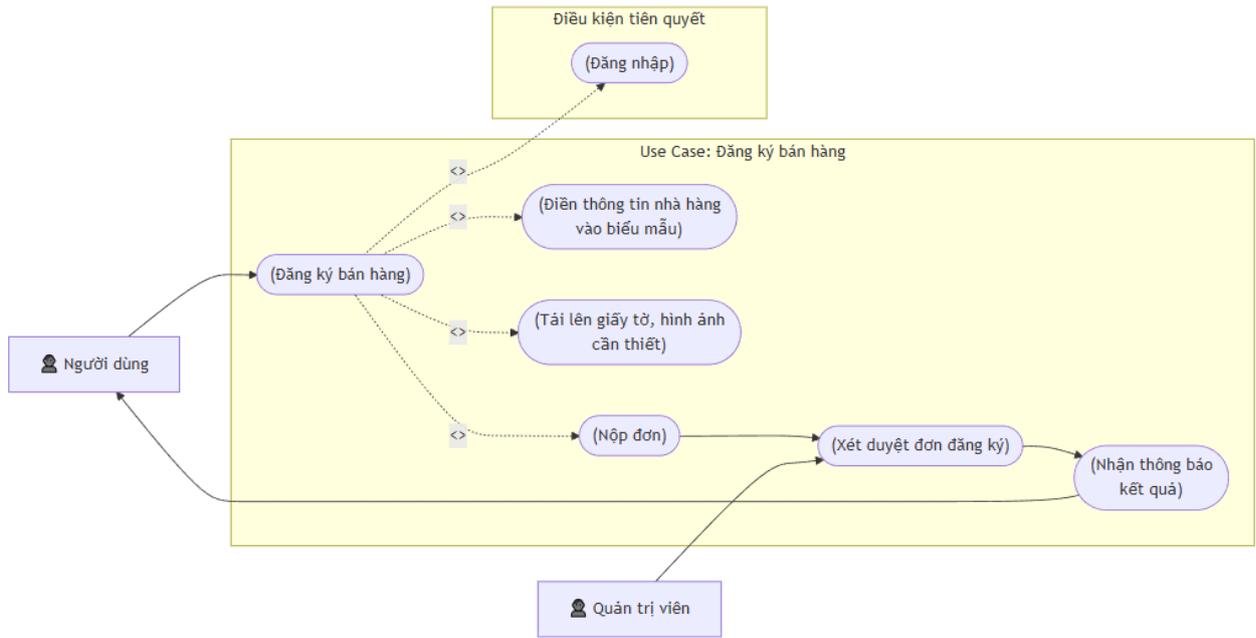
Ảnh 2.3: Biểu đồ phân rã usecase Đặt món

Biểu đồ phân rã usecase Quản lý hồ sơ cá nhân



Ảnh 2.4: Biểu đồ phân rã usecase Hồ sơ cá nhân

Biểu đồ phân rã usecase Đăng ký làm nhà hàng



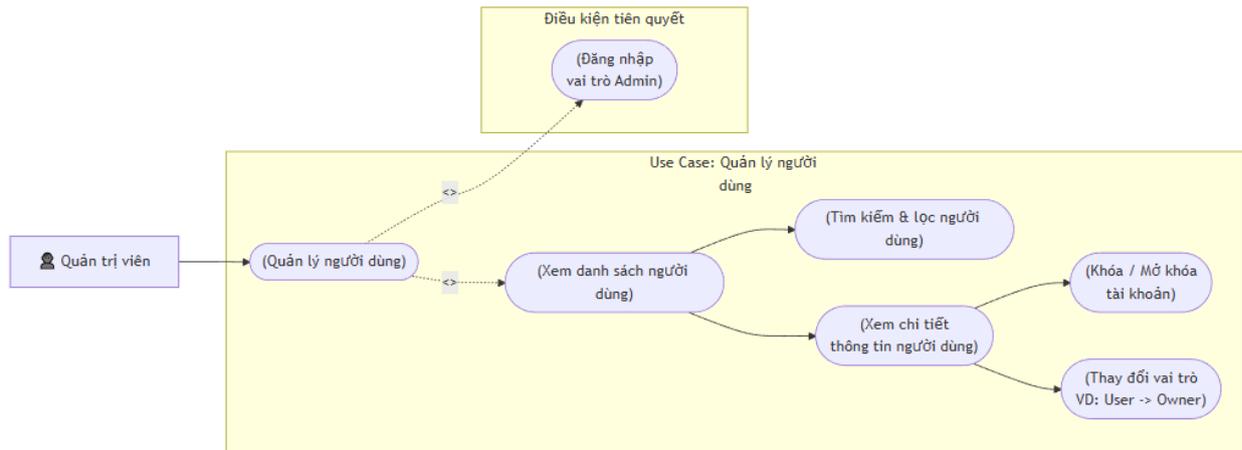
Ảnh 2.5: Biểu đồ phân rã usecase Đăng ký làm nhà hàng

Biểu đồ phân rã usecase Quản lý đơn hàng



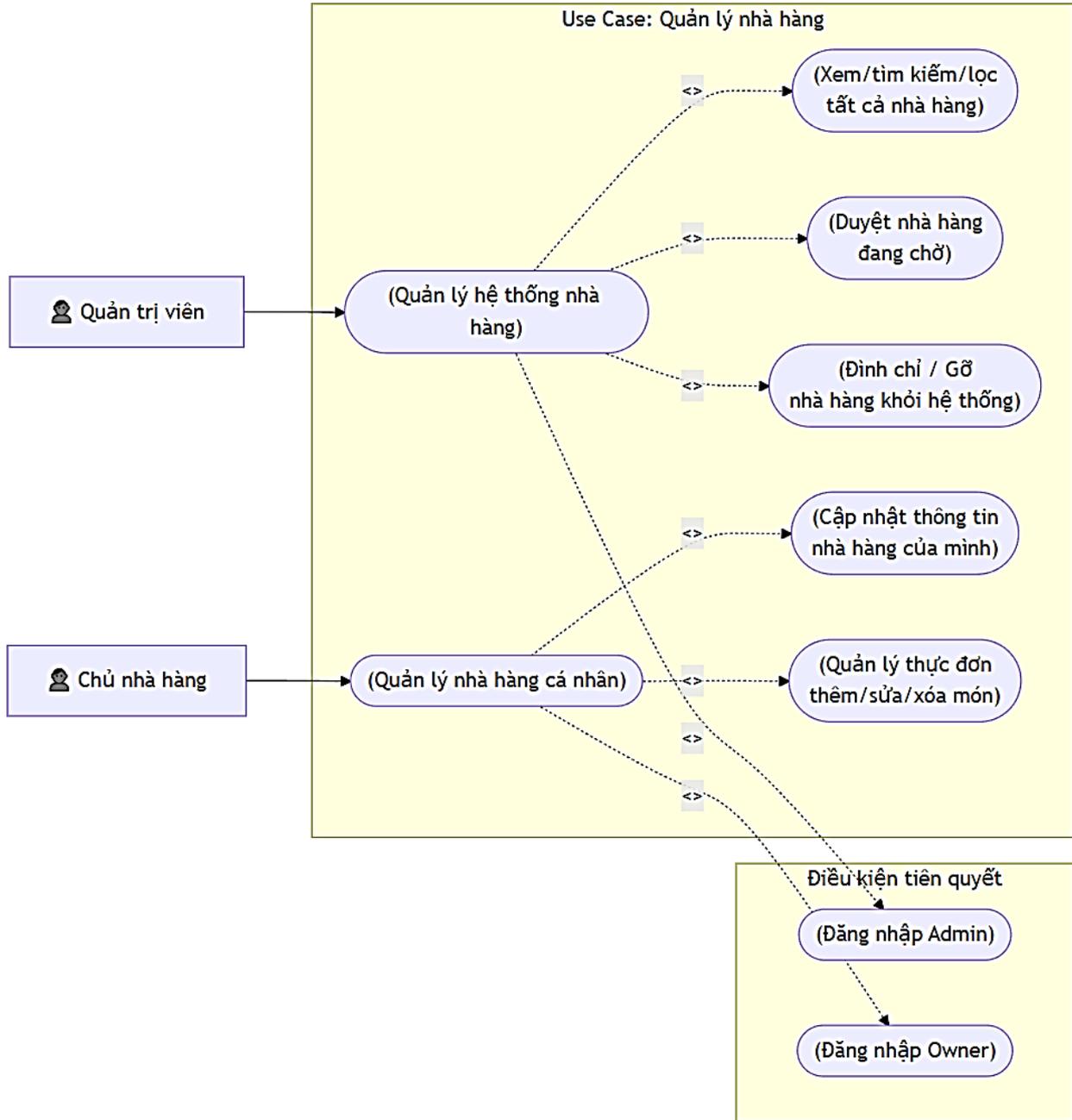
Ảnh 2.6: Biểu đồ phân rã usecase Quản lý đơn hàng

Biểu đồ phân rã usecase Quản lý người dùng



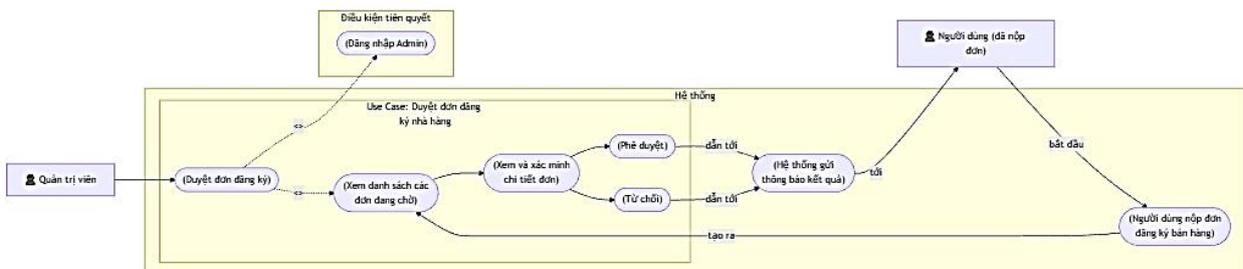
Ảnh 2.7: Biểu đồ phân rã usecase Quản lý người dùng

Biểu đồ phân rã usecase Quản lý nhà hàng



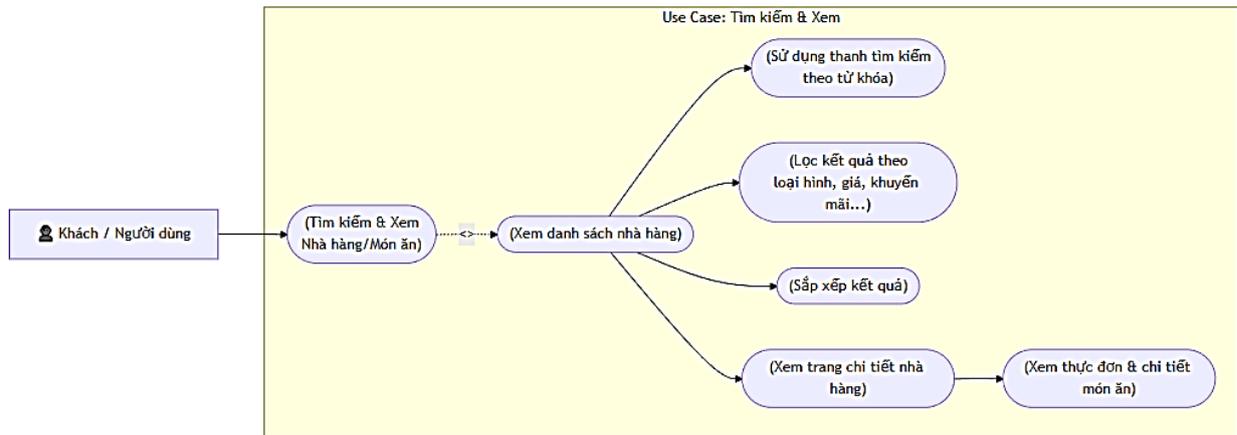
Ảnh 2.8: Biểu đồ phân rã usecase Quản lý nhà hàng

Biểu đồ phân rã usecase Duyệt đơn đăng ký nhà hàng



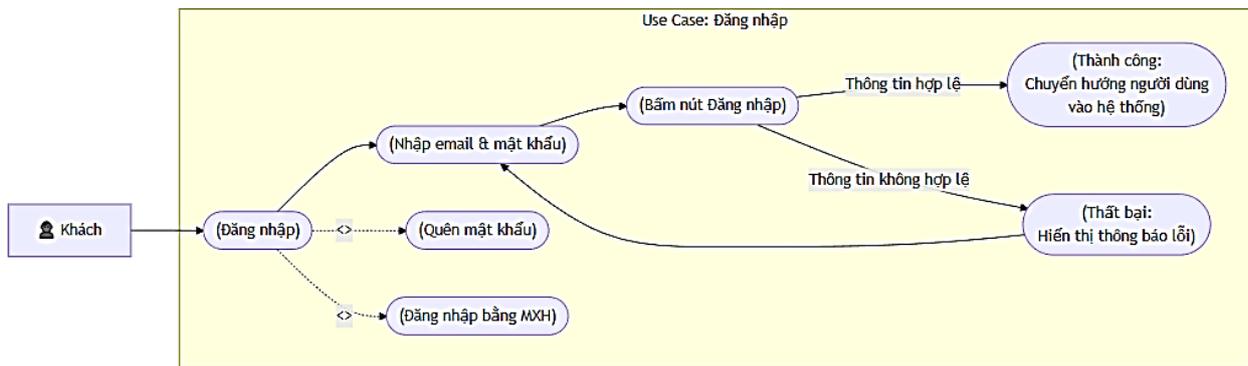
Ảnh 2.9: Biểu đồ phân rã usecase Duyệt đơn đăng ký nhà hàng

Biểu đồ phân rã usecase Tìm kiếm và xem nhà hàng/Thực đơn



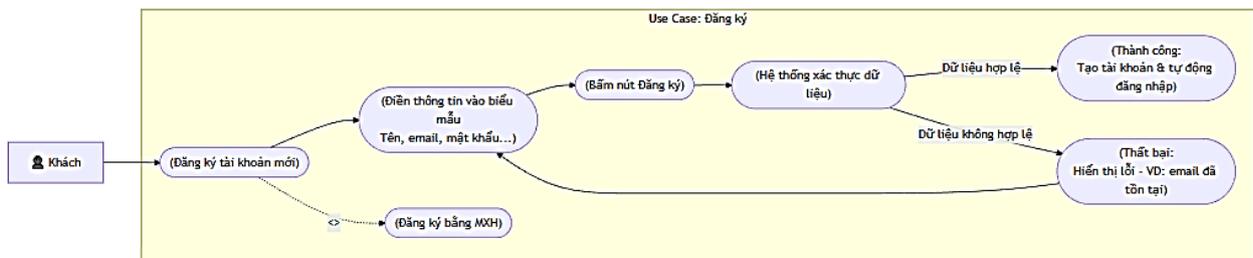
Ảnh 2.10: Biểu đồ phân rã usecase Tìm kiếm và xem nhà hàng/thực đơn

Biểu đồ phân rã usecase Đăng nhập



Ảnh 2.11: Biểu đồ phân rã usecase Đăng nhập

Biểu đồ phân rã usecase Đăng ký



Ảnh 2.12: Biểu đồ phân rã usecase Đăng ký

2.3. Đặc tả chức năng

2.3.1. Use Case: Đăng nhập

Tên Use Case	Đăng nhập	Độ phức tạp: Cao
Mô tả	Cho phép người dùng (Khách hàng, Chủ nhà hàng, Admin) đã có tài khoản truy cập vào hệ thống bằng thông tin xác thực của họ.	
Tác nhân	User	
Tiền điều kiện	Người dùng đã truy cập trang web và đã có một tài khoản được đăng ký trước đó trong hệ thống.	
Hậu điều kiện	Thành công	Người dùng được xác thực thành công. Hệ thống tạo một phiên làm việc (session/token), chuyển hướng người dùng đến trang phù hợp với vai trò của họ (Trang chủ cho Khách hàng, Dashboard cho Admin/Chủ nhà hàng) và cập nhật giao diện (ví dụ: hiển thị tên người dùng và nút đăng xuất). Thất bại: Người dùng không được xác thực, hệ thống hiển thị thông báo lỗi và người dùng vẫn ở lại trang đăng nhập.
	Lỗi	Đăng nhập thất bại
ĐẶC TẢ CHỨC NĂNG		
Luồng sự kiện chính/Kịch bản chính		
Usecase này bắt đầu khi user cần đăng nhập vào trang web. <ul style="list-style-type: none">- Use case bắt đầu khi Người dùng chọn chức năng "Đăng nhập" trên giao diện.- Hệ thống hiển thị trang Đăng nhập, yêu cầu Người dùng nhập Email và Mật khẩu.- Người dùng nhập Email và Mật khẩu đã đăng ký, sau đó nhấn nút "Đăng nhập".- Hệ thống kiểm tra tính hợp lệ của dữ liệu đầu vào ở phía client (frontend). Nếu có lỗi, chuyển đến Luồng A.- Hệ thống gửi thông tin Email và Mật khẩu đến máy chủ (backend) để xác thực.- Máy chủ kiểm tra thông tin đăng nhập với cơ sở dữ liệu.		

<ul style="list-style-type: none"> - Thông tin đăng nhập chính xác. Hệ thống tạo token xác thực, trả về cho client và chuyển hướng người dùng đến trang tương ứng với vai trò của họ. Use case kết thúc thành công.
Luồng sự kiện phát sinh/Kịch bản phát sinh
Luồng A: Người dùng không nhập đủ thông tin
<ul style="list-style-type: none"> - Tại bước 4 của luồng chính, nếu Người dùng không nhập Email, hệ thống sẽ hiển thị thông báo lỗi ngay trên form, ví dụ: "Vui lòng nhập email". - Tại bước 4 của luồng chính, nếu Người dùng không nhập Mật khẩu, hệ thống sẽ hiển thị thông báo lỗi ngay trên form, ví dụ: "Vui lòng nhập mật khẩu". - Use case quay trở lại bước 3.
Luồng B: Thông tin đăng nhập không chính xác
<ul style="list-style-type: none"> - Tại bước 6 của luồng chính, nếu Email không tồn tại trong cơ sở dữ liệu hoặc Mật khẩu không khớp. - Hệ thống trả về một thông báo lỗi cho client. - Client hiển thị thông báo lỗi cho Người dùng, ví dụ: "Email hoặc mật khẩu không chính xác". - Use case quay trở lại bước 3.

Bảng 2.2: UC Đăng nhập

2.3.2. Use Case: Đăng ký

Tên Use Case		Đăng ký	Độ phức tạp: Cao
Mô tả		Cho phép người dùng (User) đăng ký tài khoản mới trong website để có thể truy cập và sử dụng dịch vụ.	
Tác nhân		Khách vãng lai (Guest)	
Tiền điều kiện		Người dùng chưa có tài khoản và đang truy cập website.	
Hậu điều kiện	Thành công	Đăng ký thành công, tài khoản được tạo và dữ liệu được lưu vào cơ sở dữ liệu. Hệ thống tự động đăng nhập và điều hướng người dùng đến trang mà họ đang truy cập trước đó, hoặc về trang chủ.	

	Lỗi	Đăng ký không thành công do lỗi từ phía người dùng hoặc hệ thống. Người dùng ở lại trang đăng ký và nhận được thông báo lỗi.
ĐẶC TẢ CHỨC NĂNG		
Luồng sự kiện chính/ Kịch bản chính		
<p>Use case bắt đầu khi người dùng muốn đăng ký tài khoản mới:</p> <ul style="list-style-type: none"> - Sau khi truy cập website, người dùng chọn "Đăng ký". - Hệ thống hiển thị giao diện đăng ký tài khoản, yêu cầu nhập các trường thông tin sau (tất cả đều bắt buộc): <ul style="list-style-type: none"> + Họ và tên + Email + Số điện thoại + Mật khẩu + Xác nhận mật khẩu - Người dùng nhập đầy đủ thông tin vào các trường. - Khi người dùng nhấn nút "Đăng ký", hệ thống tiến hành kiểm tra: <ul style="list-style-type: none"> + Tất cả các trường bắt buộc đã được điền hay chưa. Nếu chưa, thực thi Luồng A. + Thông tin nhập vào có hợp lệ theo các quy tắc đã định hay không. Nếu không, thực thi Luồng B. + Email đã tồn tại trong hệ thống hay chưa. Nếu đã tồn tại, thực thi Luồng C. - Nếu tất cả thông tin hợp lệ, hệ thống tạo một tài khoản mới cho người dùng, mã hóa mật khẩu và lưu vào cơ sở dữ liệu. - Hệ thống thông báo đăng ký thành công và tự động đăng nhập cho người dùng. 		
Luồng sự kiện phát sinh/ Kịch bản phát sinh		
Luồng A: Không nhập các trường bắt buộc		
<p>Khi người dùng không điền thông tin vào một hoặc nhiều trường bắt buộc và cố gắng đăng ký:</p> <ul style="list-style-type: none"> - Hệ thống hiển thị thông báo lỗi ngay dưới trường bị bỏ trống: 		

<ul style="list-style-type: none"> + Tên không được để trống + Email không được để trống + Số điện thoại không được để trống + Mật khẩu không được để trống
Luồng B: Dữ liệu nhập không hợp lệ
<p>Khi người dùng nhập thông tin không tuân thủ các quy tắc:</p> <ul style="list-style-type: none"> + Hệ thống hiển thị thông báo lỗi tương ứng: + Số điện thoại: Số điện thoại phải có 10 chữ số và bắt đầu bằng 0. + Email: Định dạng email không hợp lệ. + Mật khẩu: Mật khẩu phải có ít nhất 6 ký tự. + Xác nhận mật khẩu: Mật khẩu không khớp.

Bảng 2.3: UC Đăng ký

2.3.3. Use Case: Đăng xuất

Tên Use Case		Đăng xuất	Độ phức tạp: thấp
Mô tả		Cho phép người dùng đã đăng nhập (User) kết thúc phiên làm việc và đăng xuất khỏi tài khoản một cách an toàn.	
Tác nhân		Member, Owner, Admin	
Tiền điều kiện		Người dùng đã đăng nhập thành công vào hệ thống.	
Hậu điều kiện	Thành công	Thành công: Người dùng đăng xuất thành công, phiên làm việc kết thúc, thông tin xác thực phía client bị xóa. Hệ thống điều hướng người dùng về trang chủ ở trạng thái khách vắng lai.	
	Lỗi	Không có	
ĐẶC TẢ CHỨC NĂNG			
Luồng sự kiện chính/Kịch bản chính			
Use case này bắt đầu khi người dùng muốn đăng xuất khỏi tài khoản:			

<ul style="list-style-type: none"> - Trên thanh điều hướng (header) của trang web, người dùng nhấp vào tên tài khoản của mình. - Một menu thả xuống (dropdown menu) xuất hiện với các tùy chọn tài khoản. - Người dùng chọn mục "Đăng xuất" từ menu. - Hệ thống thực hiện các hành động sau: <ul style="list-style-type: none"> + Xóa thông tin người dùng và token xác thực khỏi bộ nhớ cục bộ (Local Storage) của trình duyệt. + Cập nhật trạng thái xác thực trong toàn bộ ứng dụng thành "chưa đăng nhập". - Hệ thống tự động chuyển hướng người dùng về trang chủ của website. - Người dùng giờ đây có thể tiếp tục duyệt web với tư cách là khách hoặc chọn đăng nhập lại.
Luồng sự kiện phát sinh/Kịch bản phát sinh

Bảng 2.4: UC Đăng xuất

2.3.4. Use Case: Tìm kiếm nhà hàng

Tên Use Case		Tìm kiếm nhà hàng	Độ phức tạp: trung bình
Mô tả		Cho phép người dùng tìm kiếm và lọc các nhà hàng dựa trên tên, và sắp xếp kết quả hiển thị.	
Tác nhân		Guest, Member	
Tiền điều kiện		Người dùng đã truy cập vào trang chủ của website.	
Hậu điều kiện	Thành công	Hiển thị danh sách các nhà hàng phù hợp với từ khóa tìm kiếm và tiêu chí sắp xếp.	
	Lỗi	Hiển thị thông báo không tìm thấy kết quả phù hợp.	
ĐẶC TẢ CHỨC NĂNG			
Luồng sự kiện chính/ Kịch bản chính			
Use case bắt đầu khi người dùng muốn tìm một nhà hàng cụ thể.			

<ul style="list-style-type: none"> - Trên trang chủ, người dùng nhập từ khóa (tên nhà hàng) vào ô tìm kiếm trong thanh "Lọc và Tìm kiếm". - Khi người dùng nhập, hệ thống tự động lọc và cập nhật danh sách nhà hàng hiển thị trên trang theo thời gian thực để khớp với từ khóa tìm kiếm. - Kết quả trả về là danh sách các thẻ nhà hàng (Restaurant Card), mỗi thẻ hiển thị thông tin cơ bản: <ul style="list-style-type: none"> + Hình ảnh đại diện của nhà hàng + Tên nhà hàng + Địa chỉ + Trạng thái (Mở cửa / Đóng cửa) - Nếu không tìm thấy nhà hàng nào phù hợp, thực thi Luồng A. - Người dùng có thể nhấp vào một thẻ nhà hàng bất kỳ để xem thông tin chi tiết và thực đơn của nhà hàng đó.
Luồng sự kiện phát sinh/ Kịch bản phát sinh
Luồng A: Không tìm thấy nhà hàng
<ul style="list-style-type: none"> - Nếu không có nhà hàng nào phù hợp với từ khóa tìm kiếm, hệ thống sẽ hiển thị một thông báo rõ ràng trên trang: "Không tìm thấy nhà hàng nào." - Người dùng có thể xóa hoặc thay đổi từ khóa trong ô tìm kiếm để thử lại.

Bảng 2.5: UC Tìm kiếm nhà hàng

2.3.5. Use Case: Xem Chi tiết Nhà hàng và Thực đơn

Tên Use Case		Chi tiết Nhà hàng và Thực đơn	Độ phức tạp: Cao
Mô tả		Cho phép người dùng xem thông tin chi tiết về một nhà hàng cụ thể và thực đơn các món ăn mà nhà hàng đó cung cấp, đồng thời cho phép thêm món ăn vào giỏ hàng.	
Tác nhân		Guest, Member	
Tiền điều kiện		Người dùng đang duyệt trang web.	
Hậu	Thành	Người dùng xem được thông tin chi tiết của nhà hàng và có	

điều kiện	công	thêm thêm các món ăn vào giỏ hàng.
	Lỗi	Thêm món ăn vào giỏ hàng thất bại do chưa đăng nhập.

ĐẶC TẢ CHỨC NĂNG

Luồng sự kiện chính/ Kịch bản chính

Usecase bắt đầu khi User muốn xem chi tiết sản phẩm.

- Từ trang chủ hoặc trang kết quả tìm kiếm, người dùng nhấp vào một thẻ nhà hàng (restaurant card).
- Hệ thống điều hướng đến trang chi tiết nhà hàng, hiển thị các thông tin sau:
 - + **Thông tin nhà hàng:**
 - Ảnh bìa (banner) của nhà hàng.
 - Tên nhà hàng.
 - Địa chỉ.
 - Trạng thái hoạt động (Mở cửa / Đóng cửa).
 - + **Danh sách món ăn (Thực đơn):**
 - Các món ăn được liệt kê dưới dạng thẻ (card), mỗi thẻ gồm: Tên món ăn, hình ảnh, và giá bán.
- Người dùng có thể xem qua thực đơn. Khi nhấp vào một món ăn, một cửa sổ (modal) chi tiết món ăn sẽ hiện lên.
- Trong cửa sổ chi tiết món ăn, người dùng có thể:
 - + Xem hình ảnh lớn hơn, tên, mô tả chi tiết và giá của món ăn.
 - + Điều chỉnh số lượng muốn đặt.
- **Luồng con: Thêm món ăn vào giỏ hàng**
 - + Người dùng nhấn nút "Thêm vào giỏ hàng" trong cửa sổ chi tiết món ăn.
 - + Nếu người dùng chưa đăng nhập, thực thi **Luồng A**.
 - + Nếu đã đăng nhập, hệ thống sẽ thêm món ăn với số lượng đã chọn vào giỏ hàng và hiển thị thông báo thành công.
- Người dùng có thể đóng cửa sổ chi tiết món ăn để tiếp tục duyệt thực đơn hoặc điều hướng đến các trang khác.

Luồng sự kiện phát sinh/ Kịch bản phát sinh
Luồng A: Người dùng chưa đăng nhập
<ul style="list-style-type: none"> - Khi người dùng chưa đăng nhập nhấn nút "Thêm vào giỏ hàng": + Hệ thống sẽ hiển thị một thông báo yêu cầu đăng nhập, ví dụ: "Bạn cần đăng nhập để thêm sản phẩm vào giỏ hàng". + Hệ thống có thể sẽ điều hướng người dùng đến trang Đăng nhập. Món ăn sẽ không được thêm vào giỏ.

Bảng 2.6: UC Chi tiết nhà hàng và thực đơn

2.3.6. Use Case: Quản lý Giỏ hàng

Tên Use Case		Quản lý Giỏ hàng	Độ phức tạp: Trung bình
Mô tả		Cho phép người dùng đã đăng nhập xem lại các món ăn đã chọn, điều chỉnh số lượng, xóa món ăn và tiến hành đặt hàng.	
Tác nhân		Member	
Tiền điều kiện		Người dùng đã đăng nhập vào website và đã thêm ít nhất một món ăn vào giỏ hàng.	
Hậu điều kiện	Thành công	Người dùng cập nhật giỏ hàng thành công hoặc tiến hành đến trang đặt hàng.	
	Lỗi	Cập nhật giỏ hàng không thành công (ví dụ: do lỗi hệ thống).	
ĐẶC TẢ CHỨC NĂNG			
Luồng sự kiện chính/ Kịch bản chính			
Use case bắt đầu khi người dùng truy cập vào trang giỏ hàng. <ul style="list-style-type: none"> - Người dùng nhấp vào biểu tượng giỏ hàng trên thanh điều hướng (header) của website. - Hệ thống hiển thị giao diện trang giỏ hàng, bao gồm hai phần chính: <ul style="list-style-type: none"> + Danh sách món ăn: Liệt kê tất cả các món ăn người dùng đã thêm, với mỗi món ăn hiển thị: 			

<ul style="list-style-type: none"> ▪ Hình ảnh và tên món ăn. ▪ Đơn giá. ▪ Trình điều chỉnh số lượng (nút + và -). ▪ Giá tổng theo số lượng của món ăn đó. ▪ Nút "Xóa" để loại bỏ món ăn khỏi giỏ hàng. <p>+ Tóm tắt đơn hàng:</p> <ul style="list-style-type: none"> ▪ Tổng cộng tiền hàng. ▪ Nút "Tiên hành đặt hàng". <p>- Người dùng có thể thực hiện các thao tác quản lý giỏ hàng:</p> <p>+ Thay đổi số lượng: Nhấn nút + để tăng hoặc - để giảm số lượng của một món ăn. Hệ thống sẽ tự động cập nhật lại giá tổng của món ăn đó và "Tổng cộng tiền hàng".</p> <p>+ Xóa món ăn: Nhấn nút "Xóa" (biểu tượng thùng rác) bên cạnh một món ăn để loại bỏ hoàn toàn món ăn đó khỏi giỏ hàng.</p> <p>- Khi người dùng đã hài lòng với giỏ hàng của mình, họ nhấn nút "Tiên hành đặt hàng".</p> <p>- Hệ thống lưu lại thông tin giỏ hàng và chuyển hướng người dùng đến trang điền thông tin đặt hàng (Order Details).</p>
Luồng sự kiện phát sinh/ Kịch bản phát sinh
Luồng A: Giỏ hàng trống
<ul style="list-style-type: none"> - Nếu người dùng truy cập giỏ hàng mà chưa thêm món ăn nào, hoặc sau khi đã xóa hết các món ăn. - Hệ thống sẽ hiển thị thông báo "Giỏ hàng của bạn đang trống" cùng với một nút khuyến khích người dùng quay lại mua sắm (ví dụ: "Quay lại trang chủ").

Bảng 2.7: UC Quản lý Giỏ hàng

2.3.7. Use Case: Đặt hàng

Tên Use Case		Đặt hàng	Độ phức tạp: Trung bình
Mô tả		Cho phép người dùng (Member) xác nhận thông tin và hoàn tất việc đặt hàng các món ăn từ giỏ hàng của mình.	
Tác nhân		Member	
Tiền điều kiện		Người dùng đã đăng nhập và có ít nhất một món ăn trong giỏ hàng.	
Hậu điều kiện	Thành công	Đơn hàng được tạo thành công và lưu vào hệ thống. Giỏ hàng của người dùng được xóa.	
	Lỗi	Đặt hàng không thành công do lỗi.	
ĐẶC TẢ CHỨC NĂNG			
Luồng sự kiện chính/ Kịch bản chính			
<p>Use case bắt đầu khi người dùng quyết định đặt hàng từ trang giỏ hàng.</p> <ul style="list-style-type: none"> - Tại trang "Giỏ hàng", sau khi đã kiểm tra các món ăn, người dùng nhấn nút "Tiến hành đặt hàng". - Hệ thống điều hướng người dùng đến trang "Đặt hàng" (Order Details). - Trên trang này, hệ thống hiển thị các thông tin sau: <ul style="list-style-type: none"> + Thông tin người dùng: Tên, Số điện thoại, và Địa chỉ giao hàng (lấy từ thông tin tài khoản của người dùng). + Phương thức thanh toán: Mặc định là "Thanh toán khi nhận hàng" (Cash on Delivery). Lưu ý: Đây là phương thức duy nhất được hỗ trợ. + Tóm tắt đơn hàng: Liệt kê các món ăn, số lượng, và tổng số tiền phải thanh toán. - Người dùng kiểm tra lại tất cả thông tin. - Người dùng nhấn nút "Đặt hàng" để xác nhận. - Hệ thống thực hiện các việc sau: <ul style="list-style-type: none"> + Tạo một đơn hàng mới trong cơ sở dữ liệu với trạng thái "Pending". 			

<ul style="list-style-type: none"> + Xóa tất cả các món ăn khỏi giỏ hàng của người dùng. + Hiện thị thông báo "Đặt hàng thành công!". - Hệ thống điều hướng người dùng đến trang "Lịch sử đơn hàng" (Profile Page), nơi họ có thể xem lại đơn hàng vừa tạo.
Luồng sự kiện phát sinh/ Kịch bản phát sinh
Luồng A: Lỗi trong quá trình đặt hàng
<ul style="list-style-type: none"> - Nếu có lỗi xảy ra ở phía máy chủ trong quá trình tạo đơn hàng (ví dụ: lỗi kết nối cơ sở dữ liệu). - Hệ thống sẽ hiển thị một thông báo lỗi cho người dùng, ví dụ: "Đã xảy ra lỗi. Vui lòng thử lại." - Người dùng vẫn ở lại trang đặt hàng và giỏ hàng không bị ảnh hưởng.

Bảng 2.8: UC Đặt hàng

2.3.8. Use Case: Xem và Quản lý Đơn hàng

Tên Use Case		Xem và Quản lý Đơn hàng	Độ phức tạp: Trung bình
Mô tả		Cho phép người dùng (Member) xem lại lịch sử các đơn hàng đã đặt và thực hiện hủy đơn hàng nếu điều kiện cho phép.	
Tác nhân		Member	
Tiền điều kiện		Người dùng đã đăng nhập vào website và đã có ít nhất một đơn hàng được đặt trước đó.	
Hậu điều kiện	Thành công	Hiển thị chi tiết đơn hàng. Nếu hủy, trạng thái đơn hàng được cập nhật thành "Cancelled".	
	Lỗi	Thao tác thất bại (ví dụ: hủy đơn hàng không hợp lệ).	
ĐẶC TẢ CHỨC NĂNG			
Luồng sự kiện chính/ Kịch bản chính			
Use case bắt đầu khi người dùng muốn xem lại các đơn hàng của mình.			

- Người dùng nhấp vào tên tài khoản trên thanh điều hướng và chọn "Thông tin cá nhân" (Profile).
- Hệ thống hiển thị trang cá nhân, trong đó có một tab/phần "Lịch sử đơn hàng".
- Phần này liệt kê tất cả các đơn hàng người dùng đã đặt, mỗi đơn hàng hiển thị thông tin tóm tắt như:
 - + ID đơn hàng
 - + Ngày đặt
 - + Tổng số tiền
 - + Trạng thái đơn hàng (ví dụ: Pending, Confirmed, Delivering, Delivered, Cancelled)
- Người dùng nhấp vào một đơn hàng cụ thể để xem chi tiết.
- Trang chi tiết đơn hàng hiển thị đầy đủ thông tin:
 - + Thông tin giao hàng: Tên người nhận, địa chỉ, số điện thoại.
 - + Danh sách các món ăn đã đặt, số lượng và giá.
 - + Tổng tiền và phương thức thanh toán.
 - + Trạng thái hiện tại của đơn hàng.
- Luồng con: Hủy đơn hàng
 - + Nếu trạng thái của đơn hàng là "Pending", một nút "Hủy đơn hàng" sẽ được hiển thị.
 - + Người dùng nhấn nút "Hủy đơn hàng".
 - + Hệ thống gửi yêu cầu hủy. Nếu thành công, trạng thái của đơn hàng được cập nhật thành "Cancelled" trong cơ sở dữ liệu.
 - + Giao diện cập nhật lại, hiển thị trạng thái mới là "Cancelled" và nút "Hủy đơn hàng" biến mất.
 - + Nếu trạng thái đơn hàng không phải là "Pending", thực thi Luồng A.

Luồng sự kiện phát sinh/ Kịch bản phát sinh

Luồng A: Hủy đơn hàng không thành công

- Nếu người dùng cố gắng hủy một đơn hàng đã được xác nhận hoặc đang giao (trạng thái khác "Pending"), nút "Hủy đơn hàng" sẽ không hiển thị.

- Do đó, người dùng không thể thực hiện hành động hủy cho các đơn hàng không hợp lệ.

Bảng 2.9: UC Xem và Quản lý Đơn hàng

2.3.9. Use Case: Cập nhật Thông tin Tài khoản

Tên Use Case		Cập nhật Thông tin Tài khoản	Độ phức tạp: Trung bình
Mô tả		Cho phép người dùng (Member) xem và chỉnh sửa thông tin cá nhân hoặc thay đổi mật khẩu của mình.	
Tác nhân		Member	
Tiền điều kiện		Người dùng đã đăng nhập vào website.	
Hậu điều kiện	Thành công	Thông tin cá nhân hoặc mật khẩu của người dùng được cập nhật thành công trong cơ sở dữ liệu.	
	Lỗi	Không thay đổi được thông tin do lỗi nhập liệu hoặc lỗi hệ thống.	
ĐẶC TẢ CHỨC NĂNG			
Luồng sự kiện chính/Kịch bản chính			
Use case bắt đầu khi người dùng muốn cập nhật thông tin tài khoản. <ul style="list-style-type: none"> – Người dùng nhấp vào tên tài khoản trên thanh điều hướng và chọn "Thông tin cá nhân" (Profile). – Hệ thống hiển thị trang cá nhân, bao gồm hai biểu mẫu (form) riêng biệt: "Cập nhật thông tin" và "Thay đổi mật khẩu". Luồng con 1: Chỉnh sửa thông tin cá nhân <ul style="list-style-type: none"> – Biểu mẫu "Cập nhật thông tin" hiển thị các thông tin hiện tại của người dùng: <ul style="list-style-type: none"> + Họ và tên (Name) + Email (chỉ hiển thị, không cho phép chỉnh sửa) + Số điện thoại (Phone) 			

- + Địa chỉ (Address)
- Người dùng chỉnh sửa các thông tin mong muốn (Họ và tên, Số điện thoại, Địa chỉ).
- Sau khi chỉnh sửa, người dùng nhấn nút "Cập nhật".
- Hệ thống kiểm tra dữ liệu đầu vào. Nếu có lỗi (ví dụ: tên để trống), thực thi **Luồng A**.
- Nếu dữ liệu hợp lệ, hệ thống gửi thông tin mới lên máy chủ để cập nhật vào cơ sở dữ liệu.
- Hệ thống hiển thị thông báo "Cập nhật thông tin thành công!".

Luồng con 2: Thay đổi mật khẩu

- Biểu mẫu "Thay đổi mật khẩu" yêu cầu người dùng nhập:
 - + Mật khẩu hiện tại (Current Password)
 - + Mật khẩu mới (New Password)
 - + Xác nhận mật khẩu mới (Confirm New Password)
- Người dùng điền đầy đủ thông tin và nhấn nút "Thay đổi mật khẩu".
- Hệ thống kiểm tra dữ liệu:
 - + Nếu có trường bị bỏ trống hoặc mật khẩu mới không khớp, thực thi **Luồng B**.
 - + Hệ thống xác thực "Mật khẩu hiện tại". Nếu không đúng, thực thi **Luồng C**.
- Nếu tất cả thông tin hợp lệ, hệ thống cập nhật mật khẩu mới đã được mã hóa vào cơ sở dữ liệu.
- Hệ thống hiển thị thông báo "Thay đổi mật khẩu thành công!".

Luồng sự kiện phát sinh/Kịch bản phát sinh

Luồng A: Lỗi cập nhật thông tin

- Nếu người dùng để trống tên hoặc nhập số điện thoại không hợp lệ, hệ thống sẽ hiển thị thông báo lỗi tương ứng ngay dưới trường nhập liệu, ví dụ: "Tên không được để trống".

Luồng B: Lỗi nhập liệu mật khẩu

<ul style="list-style-type: none"> - Nếu người dùng để trống trường nào đó hoặc mật khẩu mới và xác nhận không khớp, hệ thống hiển thị thông báo lỗi, ví dụ: "Mật khẩu mới không khớp".
Luồng C: Sai mật khẩu hiện tại
<ul style="list-style-type: none"> - Nếu người dùng nhập sai mật khẩu hiện tại, hệ thống hiển thị thông báo lỗi: "Mật khẩu hiện tại không đúng".

Bảng 2.10: UC Cập nhật Thông tin Tài khoản

2.3.10. Use Case: Đăng ký làm Nhà hàng

Tên Use Case	Đăng ký làm Nhà hàng		Độ phức tạp: cao
Mô tả	Cho phép người dùng đã có tài khoản (Member) gửi đơn đăng ký để trở thành đối tác (Owner) và quản lý nhà hàng của riêng họ trên nền tảng.		
Tác nhân	Member		
Tiền điều kiện	<ul style="list-style-type: none"> - Người dùng đã đăng nhập vào hệ thống với vai trò "customer". - Người dùng chưa có nhà hàng nào hoặc chưa có đơn đăng ký nào đang ở trạng thái "chờ duyệt". 		
Hậu điều kiện	Thành công	Một yêu cầu đăng ký nhà hàng mới được tạo trong hệ thống với trạng thái "Pending" (Chờ duyệt). Người dùng nhận được thông báo đã gửi đơn thành công.	
	Lỗi	Đơn đăng ký không được gửi do lỗi nhập liệu hoặc lỗi hệ thống.	
ĐẶC TẢ CHỨC NĂNG			
Luồng sự kiện chính/Kịch bản chính			
Use case bắt đầu khi một người dùng muốn đăng ký mở một nhà hàng. <ul style="list-style-type: none"> - Người dùng tìm và nhấp vào liên kết "Trở thành đối tác" hoặc "Đăng ký nhà hàng" trên trang web (thường ở footer hoặc trang cá nhân). 			

- Hệ thống điều hướng người dùng đến trang "Đăng ký Nhà hàng".
- Một biểu mẫu (form) xuất hiện, yêu cầu người dùng cung cấp thông tin chi tiết về nhà hàng của họ:
 - + Tên nhà hàng (Name) - Bắt buộc
 - + Địa chỉ (Address) - Bắt buộc
 - + Loại hình ẩm thực (Cuisine) - Bắt buộc
 - + Số điện thoại nhà hàng (Phone) - Bắt buộc
 - + Mô tả (Description)
 - + Ảnh đại diện nhà hàng (Image) - Tải file lên, bắt buộc
- Người dùng điền đầy đủ thông tin vào biểu mẫu.
- Người dùng nhấn nút "Gửi đơn đăng ký" (Submit).
- Hệ thống kiểm tra dữ liệu đầu vào:
 - + Nếu có bất kỳ trường bắt buộc nào bị bỏ trống, thực thi Luồng A.
- Nếu tất cả thông tin đều hợp lệ, hệ thống sẽ:
 - + Gửi thông tin đăng ký đến máy chủ.
 - + Tạo một bản ghi nhà hàng mới trong cơ sở dữ liệu với trạng thái là "Pending".
 - + Liên kết bản ghi này với tài khoản của người dùng đã gửi đơn.
- Hệ thống hiển thị thông báo thành công, ví dụ: "Đơn đăng ký của bạn đã được gửi thành công và đang chờ quản trị viên phê duyệt."
- Hệ thống có thể vô hiệu hóa nút gửi đơn hoặc chuyển hướng người dùng sang trang khác để tránh gửi lại.

Luồng sự kiện phát sinh/Kịch bản phát sinh

Luồng A: Không nhập trường bắt buộc

- Khi người dùng nhấn "Gửi" nhưng chưa điền đủ các trường bắt buộc (Tên, Địa chỉ, Ẩm thực, Điện thoại, Ảnh).
- Hệ thống sẽ hiển thị thông báo lỗi ngay dưới các trường còn trống, ví dụ: "Tên nhà hàng không được để trống."

Luồng B: Người dùng đã có nhà hàng hoặc đơn đang chờ

- Nếu người dùng đã là Owner hoặc đã có một đơn đăng ký đang ở trạng thái "Pending".
- Hệ thống sẽ không hiển thị biểu mẫu đăng ký mà thay vào đó hiển thị một thông báo về trạng thái hiện tại, ví dụ: "Bạn đã có một nhà hàng đang hoạt động" hoặc "Đơn đăng ký của bạn đang chờ được duyệt."

Bảng 2.11: UC Cập nhật Thông tin Tài khoản

2.3.11. Use Case: Quản lý Thực đơn (Owner)

Tên Use Case		Quản lý Thực đơn (Owner)	Độ phức tạp: Cao
Mô tả		Cho phép Chủ nhà hàng (Owner) tự quản lý thực đơn của nhà hàng mình, bao gồm việc thêm, sửa và xóa các món ăn.	
Tác nhân		Admin	
Tiền điều kiện		<ul style="list-style-type: none"> - Owner đã đăng nhập vào hệ thống. - Tài khoản Owner đã được liên kết với một nhà hàng đã được Admin phê duyệt. 	
Hậu điều kiện	Thành công	Thêm, sửa, hoặc xóa món ăn thành công. Thực đơn của nhà hàng được cập nhật.	
	Lỗi	Thao tác thất bại do lỗi nhập liệu hoặc lỗi hệ thống.	
ĐẶC TẢ CHỨC NĂNG			
Luồng sự kiện chính/Kịch bản chính			
<p>Use case bắt đầu khi Owner muốn quản lý thực đơn của mình.</p> <ul style="list-style-type: none"> - Từ trang quản lý của Owner, họ chọn mục "Quản lý Thực đơn" (Menu Manager). - Hệ thống tự động nhận diện nhà hàng của Owner và hiển thị danh sách các món ăn hiện có, với các thông tin: <ul style="list-style-type: none"> + Tên món ăn + Hình ảnh + Giá 			

- + Mô tả
- + Các nút hành động: "Sửa" (Edit) và "Xóa" (Delete).
- Phía trên danh sách là nút "Thêm món ăn mới" (Add New Item).

Luồng con 1: Thêm món ăn mới

- Owner nhấn nút "Thêm món ăn mới".
- Một biểu mẫu (form) xuất hiện, yêu cầu nhập các thông tin cho món ăn mới:
 - + Tên món ăn (Name) - Bắt buộc
 - + Giá (Price) - Bắt buộc, phải là số
 - + Mô tả (Description)
 - + Hình ảnh (Image) - Tải file lên
- Owner điền thông tin và nhấn "Lưu" (Save).
- Hệ thống kiểm tra dữ liệu:
 - + Nếu có trường bắt buộc bị trống, thực thi **Luồng A**.
 - + Nếu giá không phải là số hợp lệ, thực thi **Luồng B**.
- Nếu dữ liệu hợp lệ, hệ thống tạo món ăn mới và lưu vào cơ sở dữ liệu.
- Hệ thống hiển thị thông báo "Thêm món ăn thành công" và cập nhật lại danh sách món ăn.

Luồng con 2: Sửa thông tin món ăn

- Owner nhấn nút "Sửa" bên cạnh món ăn muốn cập nhật.
- Biểu mẫu chỉnh sửa xuất hiện, với các thông tin hiện tại của món ăn đã được điền sẵn.
- Owner thay đổi thông tin cần thiết và nhấn "Lưu".
- Hệ thống kiểm tra và lưu các thay đổi tương tự như luồng thêm mới.
- Hệ thống hiển thị thông báo "Cập nhật món ăn thành công" và cập nhật lại danh sách.

Luồng con 3: Xóa món ăn

- Owner nhấn nút "Xóa" bên cạnh món ăn muốn loại bỏ.

<ul style="list-style-type: none"> - Hệ thống hiển thị một hộp thoại xác nhận: "Bạn có chắc chắn muốn xóa món ăn này không?" - Nếu Owner chọn "Hủy", không có gì xảy ra. - Nếu Owner chọn "Xóa", hệ thống sẽ xóa món ăn khỏi cơ sở dữ liệu. - Hệ thống hiển thị thông báo "Xóa món ăn thành công" và loại bỏ món ăn khỏi danh sách hiển thị.
Luồng sự kiện phát sinh/Kịch bản phát sinh
Luồng A: Không nhập trường bắt buộc
<ul style="list-style-type: none"> - Khi Owner lưu mà không nhập Tên hoặc Giá, hệ thống sẽ hiển thị thông báo lỗi: "Vui lòng điền đầy đủ các trường bắt buộc."
Luồng B: Sai điều kiện của các trường
<ul style="list-style-type: none"> - Khi Owner nhập giá không phải là một con số, hệ thống hiển thị thông báo: "Giá phải là một con số."

Bảng 2.12: UC Quản lý Thực đơn (Owner)

2.3.12. Use Case: Quản lý Đơn hàng (Owner)

Tên Use Case		Quản lý Đơn hàng (Owner)	Độ phức tạp: Cao
Mô tả		Cho phép Chủ nhà hàng (Owner) xem và quản lý các đơn hàng mà khách hàng đã đặt tại nhà hàng của mình, bao gồm việc cập nhật trạng thái đơn hàng.	
Tác nhân		Owner.	
Tiền điều kiện		<ul style="list-style-type: none"> - Owner đã đăng nhập vào hệ thống. - Nhà hàng của Owner có ít nhất một đơn hàng đã được đặt. 	
Hậu điều kiện	Thành công	Trạng thái của đơn hàng được cập nhật thành công trong hệ thống.	
	Lỗi	Cập nhật trạng thái thất bại do lỗi hệ thống.	

ĐẶC TẢ CHỨC NĂNG

Luồng sự kiện chính/ Kịch bản chính

Use case bắt đầu khi Owner muốn xem và xử lý các đơn hàng.

- Từ trang quản lý của Owner, họ chọn mục "Quản lý Đơn hàng" (Order Manager).
- Hệ thống hiển thị một danh sách tất cả các đơn hàng thuộc về nhà hàng của họ. Mỗi đơn hàng trong danh sách hiển thị các thông tin chính:
 - + ID Đơn hàng
 - + Tên khách hàng
 - + Ngày đặt
 - + Tổng tiền
 - + Trạng thái hiện tại (ví dụ: Pending, Confirmed, Delivering, Delivered, Cancelled).
- Owner có thể nhấp vào một đơn hàng để xem chi tiết các món ăn trong đơn hàng đó.
- Luồng con: Cập nhật trạng thái đơn hàng
 - + Đối với mỗi đơn hàng, có một menu thả xuống (dropdown) hoặc các nút cho phép Owner thay đổi trạng thái.
 - + Các lựa chọn trạng thái tiếp theo phụ thuộc vào trạng thái hiện tại của đơn hàng. Quy trình tiêu biểu như sau:
 - Nếu đơn hàng đang ở trạng thái "Pending", Owner có thể chuyển nó sang "Confirmed" (Xác nhận).
 - Nếu đơn hàng đang ở trạng thái "Confirmed", Owner có thể chuyển nó sang "Delivering" (Đang giao).
 - Nếu đơn hàng đang ở trạng thái "Delivering", Owner có thể chuyển nó sang "Delivered" (Đã giao).
 - + Owner chọn trạng thái mới cho đơn hàng.
 - + Hệ thống gửi yêu cầu cập nhật lên máy chủ.
 - + Sau khi cập nhật thành công, hệ thống hiển thị thông báo "Cập nhật trạng thái thành công" và làm mới danh sách, hiển thị đơn hàng với trạng thái mới.

Luồng sự kiện phát sinh/ Kịch bản phát sinh
Luồng A: Không có đơn hàng
<ul style="list-style-type: none"> – Nếu nhà hàng chưa có đơn hàng nào, hệ thống sẽ hiển thị một thông báo như: "Bạn chưa có đơn hàng nào."
Luồng B: Lỗi khi cập nhật
<ul style="list-style-type: none"> – Nếu có lỗi xảy ra ở phía máy chủ trong quá trình cập nhật trạng thái. – Hệ thống sẽ hiển thị một thông báo lỗi cho Owner, ví dụ: "Đã xảy ra lỗi, không thể cập nhật trạng thái." Trạng thái của đơn hàng sẽ không thay đổi.

Bảng 2.13: UC Quản lý Đơn hàng (Owner)

2.3.13. Use Case: Kiểm duyệt Món ăn (Admin)

Tên Use Case		Kiểm duyệt Món ăn (Admin)	Độ phức tạp: Trung bình
Mô tả		Cho phép Quản trị viên (Admin) ẩn (vô hiệu hóa) hoặc hiện (kích hoạt lại) các món ăn từ thực đơn của bất kỳ nhà hàng nào để kiểm duyệt nội dung, mà không xóa vĩnh viễn.	
Tác nhân		Admin.	
Tiền điều kiện		Admin đã đăng nhập và truy cập vào trang "Quản lý Thực đơn" trong khu vực quản trị.	
Hậu điều kiện	Thành công	Trạng thái kích hoạt (is_active) của món ăn được thay đổi thành công trong cơ sở dữ liệu.	
	Lỗi	Thao tác thất bại do lỗi hệ thống.	
ĐẶC TẢ CHỨC NĂNG			
Luồng sự kiện chính/ Kịch bản chính			
Use case bắt đầu khi Admin muốn kiểm duyệt thực đơn của một nhà hàng. <ul style="list-style-type: none"> – Tại trang "Quản lý Thực đơn", Admin chọn một nhà hàng từ menu thả xuống. 			

- Hệ thống hiển thị danh sách các món ăn của nhà hàng đó. Mỗi món ăn sẽ có một chỉ báo trực quan về trạng thái hiện tại (ví dụ: món ăn bị ẩn sẽ được làm mờ đi).
- Bên cạnh mỗi món ăn, có một nút để thay đổi trạng thái:
 - + Nút "Ẩn" (Hide) nếu món ăn đang được kích hoạt (is_active: true).
 - + Nút "Hiện" (Show) nếu món ăn đang bị ẩn (is_active: false).

Luồng con: Thay đổi trạng thái món ăn

- + Admin nhấn nút "Ẩn" hoặc "Hiện" trên một món ăn cụ thể.
- + Hệ thống gửi yêu cầu đến máy chủ để cập nhật trường is_active của món ăn đó.
- + Sau khi cập nhật thành công, hệ thống hiển thị thông báo (ví dụ: "Đã ẩn món ăn thành công").
- + Giao diện được làm mới, chỉ báo trạng thái và nút hành động của món ăn đó được cập nhật ngay lập tức để phản ánh trạng thái mới.

Bảng 2.14: UC Kiểm duyệt Món ăn (Admin)

2.3.14. Use Case: Quản lý Người dùng (Admin)

Tên Use Case		Quản lý Người dùng (Admin)	Độ phức tạp: trung bình
Mô tả		Cho phép Quản trị viên (Admin) xem danh sách tất cả người dùng trong hệ thống và quản lý trạng thái tài khoản của họ (khóa hoặc mở khóa).	
Tác nhân		Admin.	
Tiền điều kiện		Admin đã đăng nhập và truy cập vào trang "Quản lý Người dùng".	
Hậu điều kiện	Thành công	Trạng thái hoạt động (is_active) của tài khoản người dùng được thay đổi thành công.	
	Lỗi	Thao tác thất bại do lỗi hệ thống.	
ĐẶC TẢ CHỨC NĂNG			

Luồng sự kiện chính/ Kịch bản chính

Use case bắt đầu khi Admin muốn xem và quản lý danh sách người dùng.

- Từ trang quản trị, Admin chọn mục "Người dùng" (Users).
- Hệ thống hiển thị một danh sách tất cả các tài khoản người dùng đã đăng ký, với mỗi người dùng hiển thị các thông tin sau:
 - + ID Người dùng
 - + Tên
 - + Email
 - + Số điện thoại
 - + Vai trò (Role: customer, owner, admin)
 - + Trạng thái (Status: Active/Inactive)
- Luồng con: Thay đổi trạng thái tài khoản
 - + Bên cạnh mỗi tài khoản người dùng, có một nút để thay đổi trạng thái:
 - Nút "Khóa" (Deactivate) nếu tài khoản đang hoạt động (is_active: true).
 - Nút "Mở khóa" (Activate) nếu tài khoản đang bị khóa (is_active: false).
 - + Admin nhấn vào nút hành động tương ứng với tài khoản họ muốn quản lý.
 - + Hệ thống gửi yêu cầu đến máy chủ để cập nhật trường is_active của tài khoản đó.
 - + Sau khi cập nhật thành công, hệ thống hiển thị thông báo (ví dụ: "Đã khóa tài khoản thành công").

Luồng sự kiện phát sinh/ Kịch bản phát sinh

Luồng A: Admin tự khóa tài khoản của chính mình

- Hệ thống được thiết kế để ngăn chặn Admin tự khóa tài khoản của chính mình.
- Nếu Admin cố gắng thực hiện hành động này, nút "Khóa" có thể bị vô hiệu hóa hoặc hệ thống sẽ hiển thị một thông báo lỗi: "Bạn không thể tự khóa tài khoản của chính mình."

Luồng B: Lỗi hệ thống

- Nếu có lỗi xảy ra ở phía máy chủ trong quá trình cập nhật.
- Hệ thống sẽ hiển thị một thông báo lỗi chung và trạng thái tài khoản sẽ không thay đổi.

Bảng 2.15: UC Quản lý Người dùng (Admin)

2.3.15. Use Case: Quản lý Nhà hàng (Admin)

Tên Use Case		Quản lý Nhà hàng (Admin)	Độ phức tạp: trung bình
Mô tả		Cho phép Quản trị viên (Admin) xem danh sách tất cả các nhà hàng đã được phê duyệt và quản lý trạng thái hoạt động của chúng (ẩn hoặc hiện trên trang web).	
Tác nhân		Admin.	
Tiền điều kiện		Admin đã đăng nhập và truy cập vào trang "Quản lý Nhà hàng".	
Hậu điều kiện	Thành công	Trạng thái hoạt động của nhà hàng được thay đổi thành công.	
	Lỗi	Thao tác thất bại do lỗi hệ thống.	
ĐẶC TẢ CHỨC NĂNG			
Luồng sự kiện chính/ Kịch bản chính			
<p>Use case bắt đầu khi Admin muốn xem và quản lý các nhà hàng đang hoạt động.</p> <ul style="list-style-type: none"> – Từ trang quản trị, Admin chọn mục "Nhà hàng" (Restaurants). – Hệ thống hiển thị một danh sách tất cả các nhà hàng đã được phê duyệt(status:approved). Mỗi nhà hàng trong danh sách hiển thị các thông tin: <ul style="list-style-type: none"> + Tên nhà hàng + Địa chỉ + Loại hình ẩm thực (Cuisine) + Tên chủ sở hữu (Owner) + Trạng thái hoạt động (Status: Active/Inactive) 			

<ul style="list-style-type: none"> - Luồng con: Thay đổi trạng thái hoạt động của nhà hàng + Bên cạnh mỗi nhà hàng, có một nút để thay đổi trạng thái: <ul style="list-style-type: none"> ▪ Nút "Vô hiệu hóa" (Deactivate) nếu nhà hàng đang hoạt động (is_active: true). ▪ Nút "Kích hoạt lại" (Activate) nếu nhà hàng đang bị vô hiệu hóa (is_active: false). + Admin nhấn vào nút hành động tương ứng với nhà hàng họ muốn quản lý. + Hệ thống gửi yêu cầu đến máy chủ để cập nhật trường is_active của nhà hàng đó. + Khi một nhà hàng bị vô hiệu hóa, nó sẽ không còn hiển thị cho khách hàng trên trang chủ hoặc trong kết quả tìm kiếm. + Sau khi cập nhật thành công, hệ thống hiển thị thông báo (ví dụ: "Đã vô hiệu hóa nhà hàng thành công"). + Giao diện được làm mới, trạng thái và nút hành động của nhà hàng đó được cập nhật để phản ánh trạng thái mới.
Luồng sự kiện phát sinh/ Kịch bản phát sinh
Luồng A: Admin tự khóa tài khoản của chính mình
<ul style="list-style-type: none"> - Nếu có lỗi xảy ra ở phía máy chủ trong quá trình cập nhật. - Hệ thống sẽ hiển thị một thông báo lỗi chung và trạng thái của nhà hàng sẽ không thay đổi.

Bảng 2.16: UC Quản lý Nhà hàng (Admin)

2.3.16. Use Case: Phê duyệt Nhà hàng (Admin)

Tên Use Case	Phê duyệt Nhà hàng (Admin)	Độ phức tạp: Cao
Mô tả	Cho phép Quản trị viên (Admin) xem xét và đưa ra quyết định phê duyệt hoặc từ chối các đơn đăng ký mở nhà hàng mới từ người dùng.	
Tác nhân	Admin.	

Tiền điều kiện		<ul style="list-style-type: none"> - Admin đã đăng nhập vào hệ thống. - Có ít nhất một đơn đăng ký nhà hàng đang ở trạng thái "Pending" (Chờ duyệt).
Hậu điều kiện	Phê duyệt thành công	Trạng thái hoạt động của nhà hàng được thay đổi thành công.
	Từ chối thành công	Trạng thái của nhà hàng được chuyển thành "Rejected". Vai trò của người dùng không thay đổi.
	Thất bại	Thao tác thất bại do lỗi hệ thống.

ĐẶC TẢ CHỨC NĂNG

Luồng sự kiện chính/ Kịch bản chính

Use case bắt đầu khi Admin muốn xử lý các đơn đăng ký nhà hàng mới.

- Từ trang quản trị, Admin chọn mục "Phê duyệt Nhà hàng" (Pending Restaurants).
- Hệ thống hiển thị một danh sách các đơn đăng ký nhà hàng đang ở trạng thái "Pending". Mỗi đơn trong danh sách hiển thị đầy đủ thông tin mà người dùng đã cung cấp:
 - + Tên nhà hàng, địa chỉ, loại hình ẩm thực, số điện thoại, mô tả.
 - + Thông tin người dùng đăng ký (tên, email).
- Với mỗi đơn đăng ký, có hai nút hành động rõ ràng: "**Duyệt**" (Approve) và "**Từ chối**" (Reject).

Luồng con 1: Phê duyệt đơn đăng ký

- Admin xem xét thông tin của một đơn đăng ký và quyết định phê duyệt.
- Admin nhấn nút "Duyệt".
- Hệ thống có thể hiển thị một hộp thoại để xác nhận lại quyết định.
- Khi xác nhận, hệ thống gửi yêu cầu "phê duyệt" đến máy chủ. Máy chủ sẽ thực hiện hai việc đồng thời:

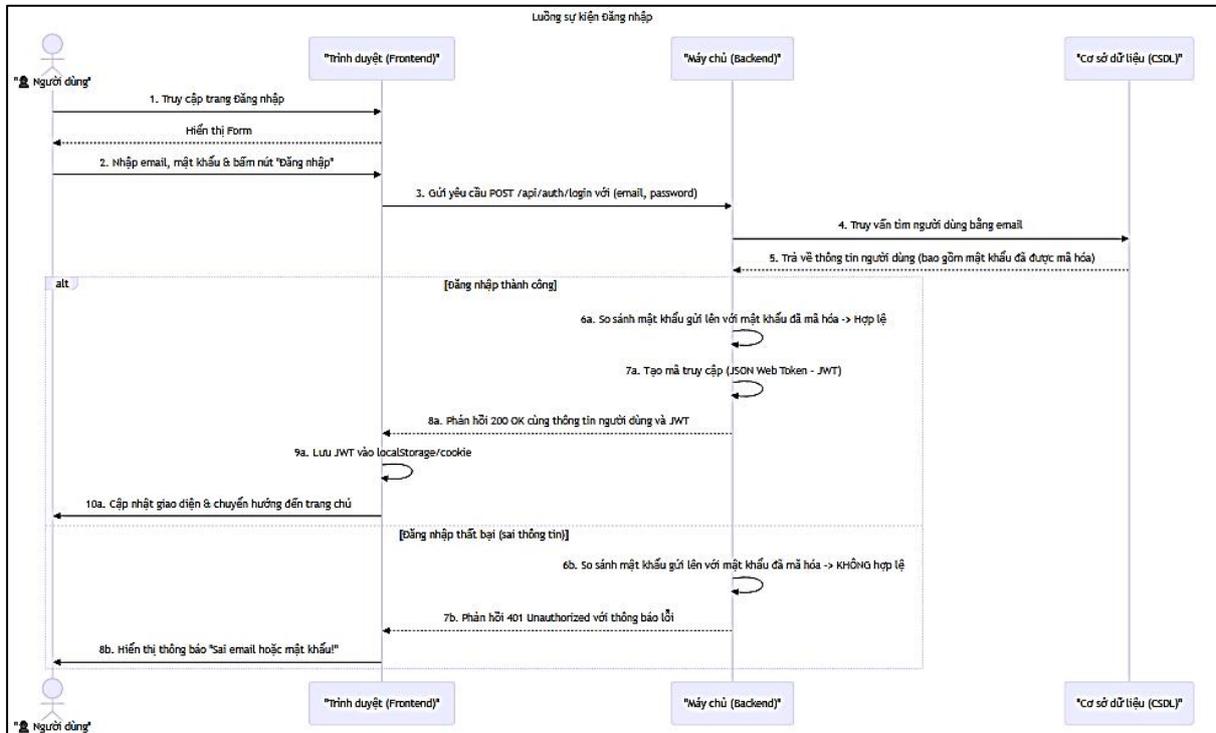
<ul style="list-style-type: none"> + Cập nhật trạng thái (status) của nhà hàng từ "Pending" thành "Approved". + Cập nhật vai trò (role) của người dùng đã gửi đơn từ "customer" thành "owner". – Hệ thống hiển thị thông báo "Phê duyệt nhà hàng thành công". – Đơn đăng ký vừa được xử lý sẽ biến mất khỏi danh sách chờ duyệt. <p>Luồng con 2: Từ chối đơn đăng ký</p> <ul style="list-style-type: none"> – Admin xem xét thông tin của một đơn đăng ký và quyết định từ chối. – Admin nhấn nút "Từ chối". – Hệ thống có thể hiển thị một hộp thoại để xác nhận lại quyết định. – Khi xác nhận, hệ thống gửi yêu cầu "từ chối" đến máy chủ. Máy chủ sẽ: <ul style="list-style-type: none"> + Cập nhật trạng thái (status) của nhà hàng từ "Pending" thành "Rejected". + (Vai trò của người dùng không thay đổi). – Hệ thống hiển thị thông báo "Đã từ chối đơn đăng ký". – Đơn đăng ký vừa được xử lý sẽ biến mất khỏi danh sách chờ duyệt.
Luồng sự kiện phát sinh/ Kịch bản phát sinh
Luồng A: Không có đơn đăng ký chờ duyệt
<ul style="list-style-type: none"> – Nếu không có đơn đăng ký nào ở trạng thái "Pending", hệ thống sẽ hiển thị một thông báo: "Không có đơn đăng ký nào đang chờ xử lý."

Bảng 2.17: UC Quản lý Nhà hàng (Admin)

CHƯƠNG III: THIẾT KẾ CHỨC NĂNG VÀ CƠ SỞ DỮ LIỆU

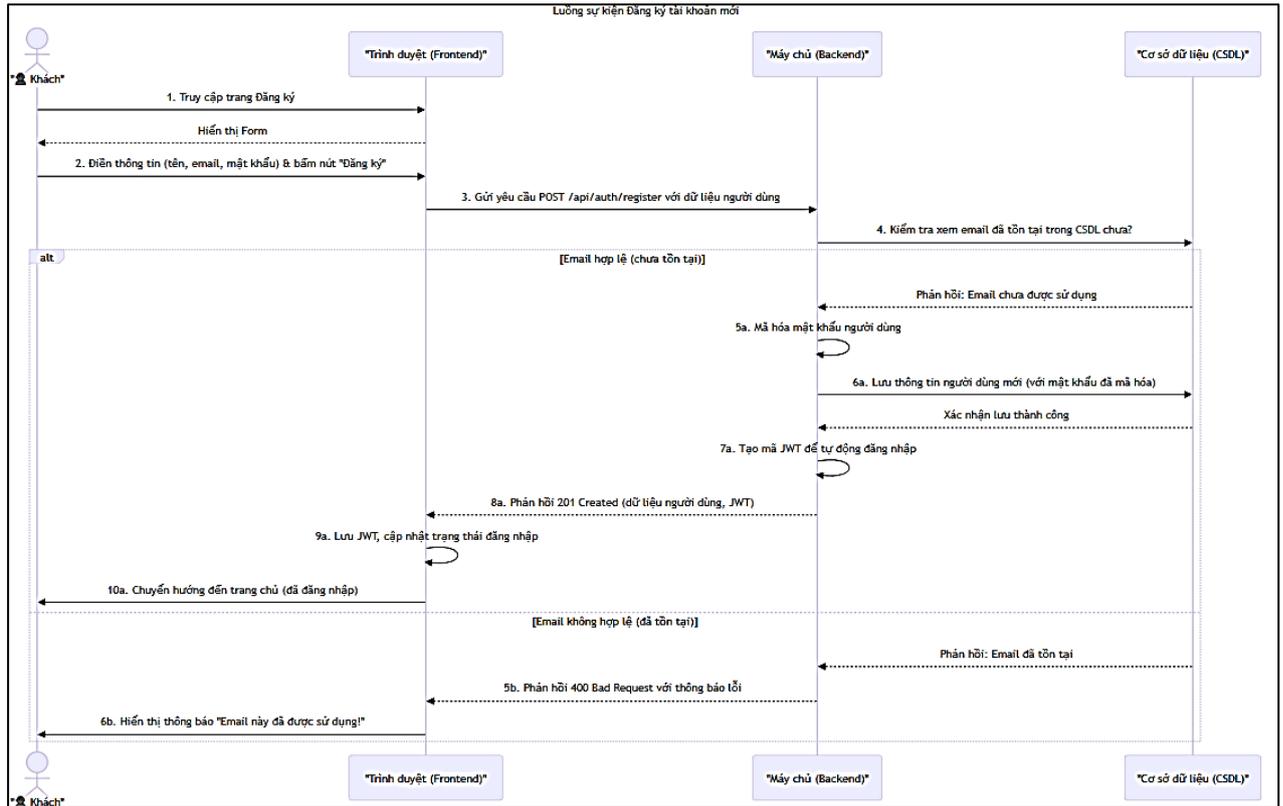
3.1. Biểu đồ tuần tự

3.1.1. Biểu đồ tuần tự chức năng Đăng nhập



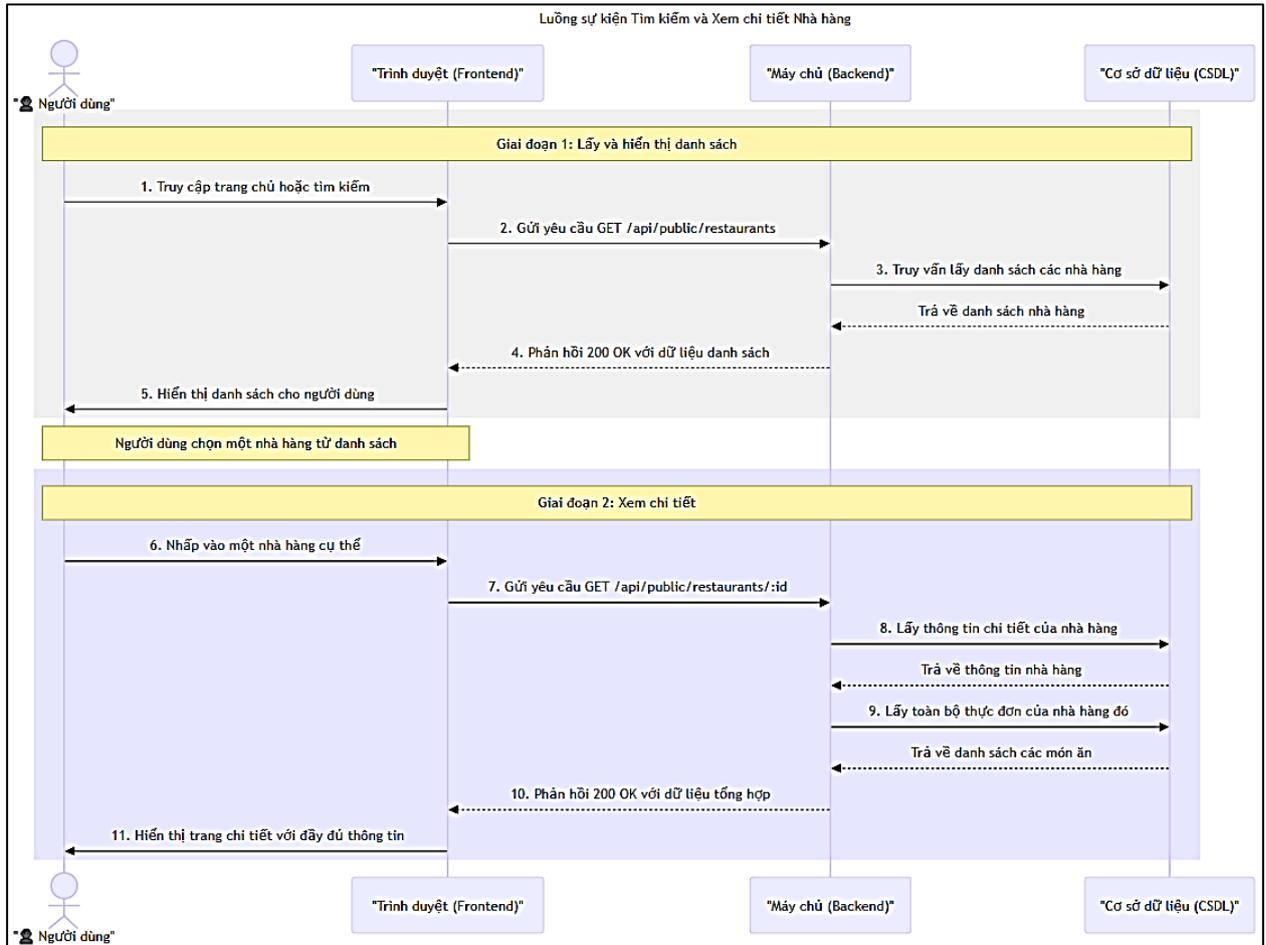
Ảnh 3.1. Biểu đồ tuần tự chức năng Đăng nhập

3.1.2. Biểu đồ tuần tự chức năng Đăng ký



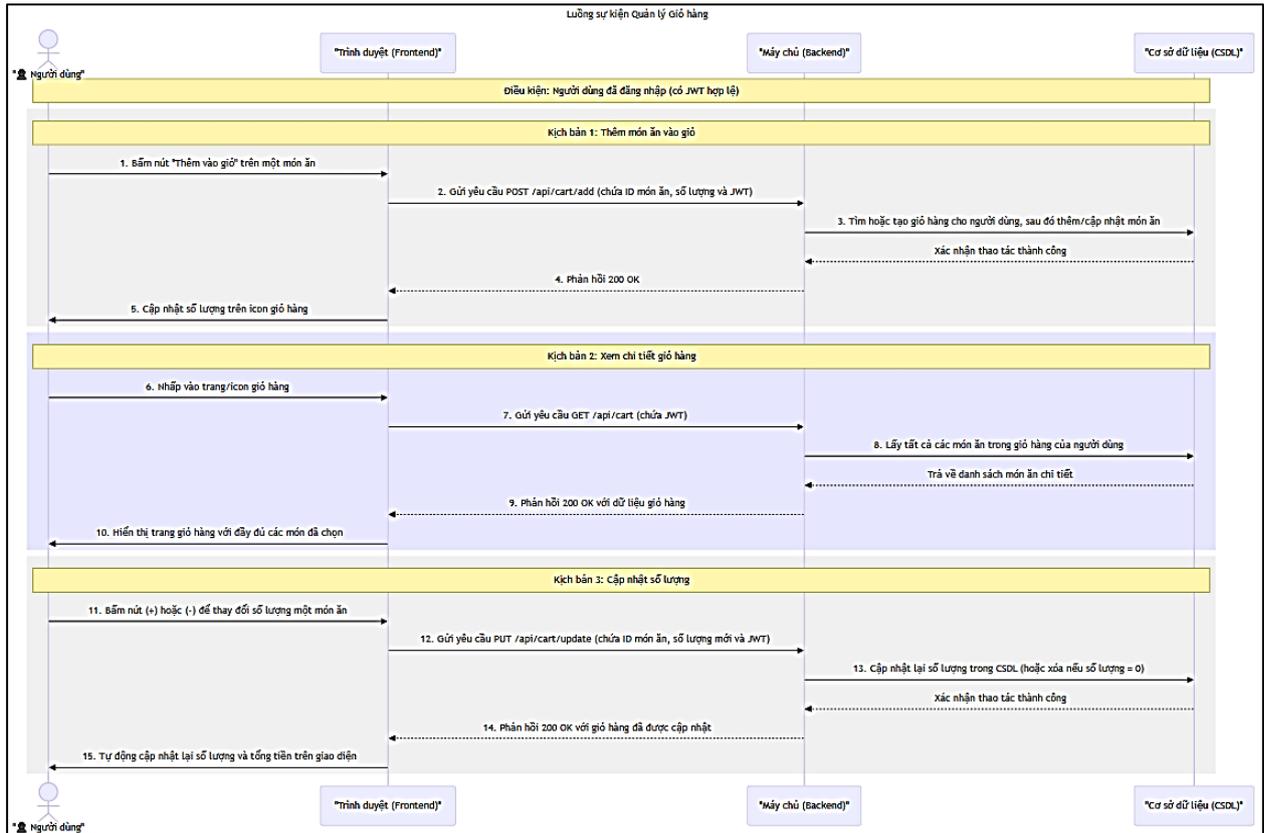
Ảnh 3.2. Biểu đồ tuần tự chức năng Đăng ký

3.1.3. Biểu đồ tuần tự chức năng Tìm kiếm và Xem chi tiết Nhà hàng



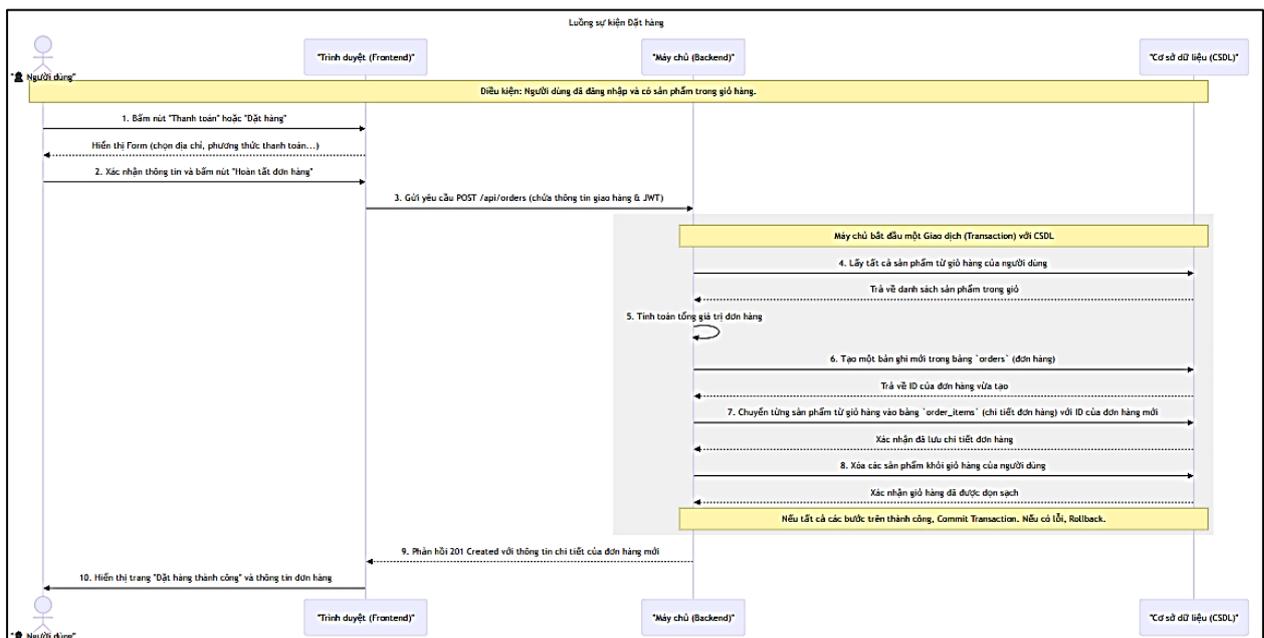
Ảnh 3.3. Biểu đồ tuần tự chức năng Tìm kiếm và Xem chi tiết Nhà hàng

3.1.4. Biểu đồ tuần tự chức năng quản lý giỏ hàng



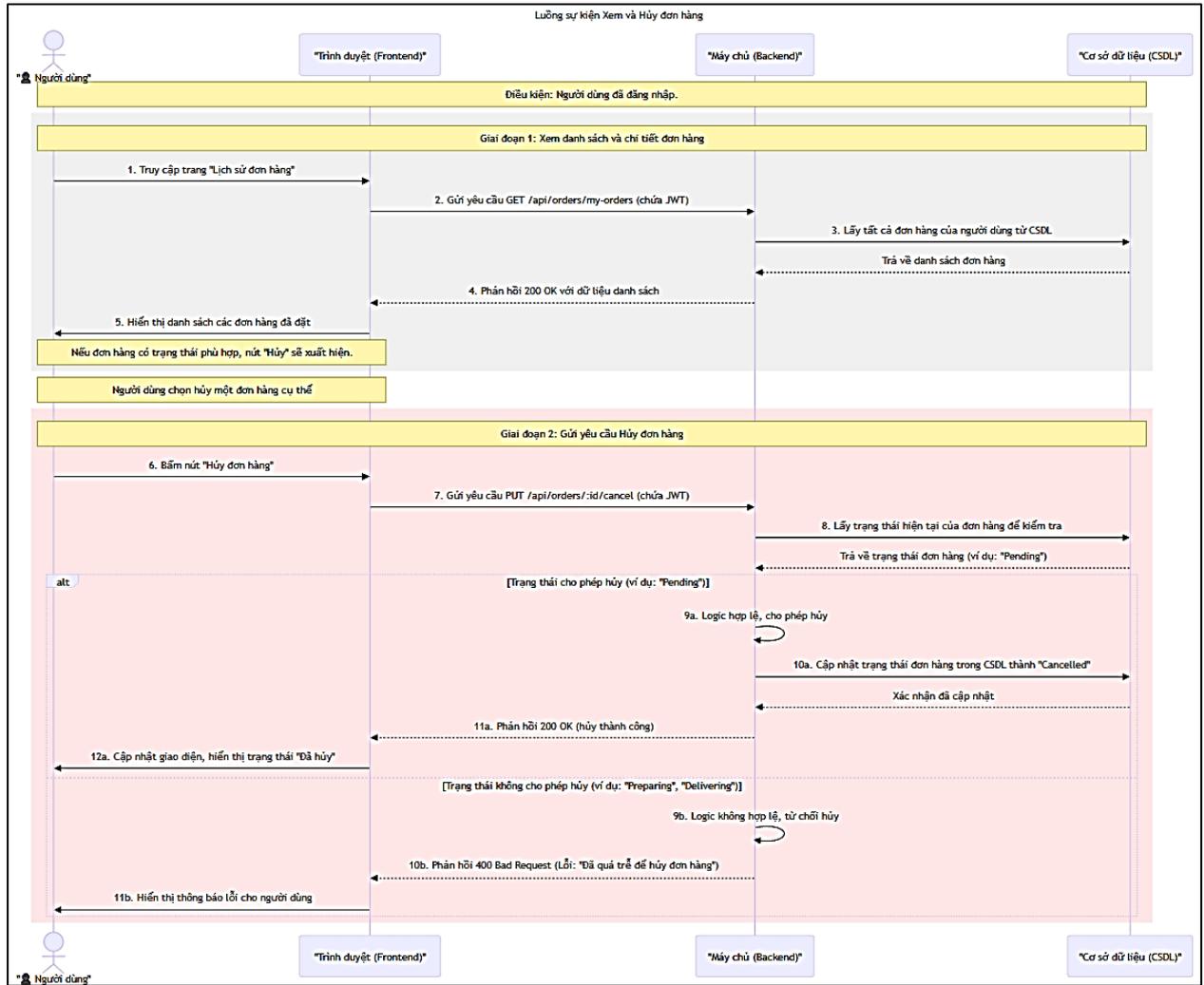
Ảnh 3.4. Biểu đồ tuần tự chức năng quản lý giỏ hàng

3.1.5. Biểu đồ tuần tự chức năng Đặt hàng



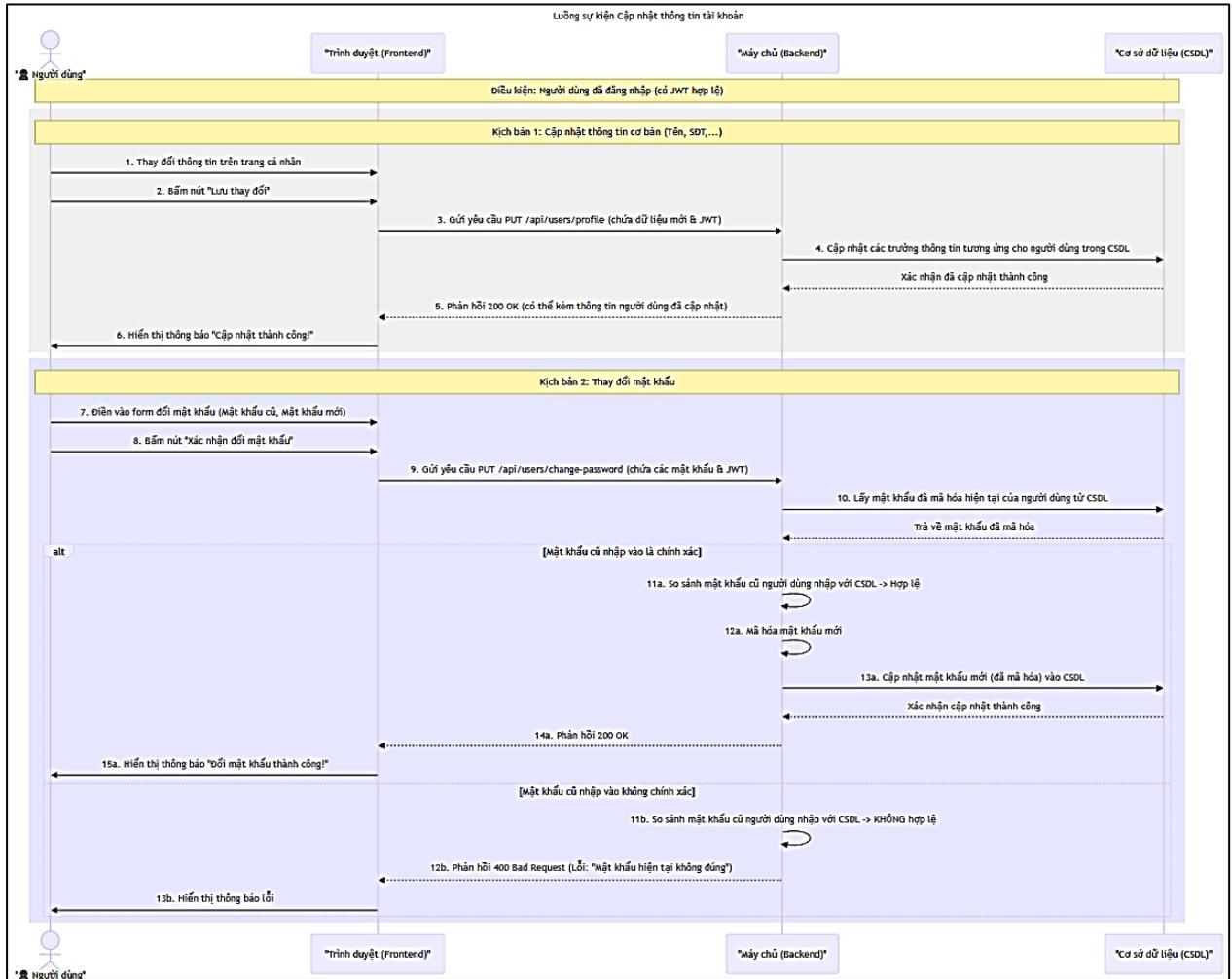
Ảnh 3.5. Biểu đồ tuần tự chức năng Đặt hàng

3.1.6. Biểu đồ tuần tự chức năng xem và hủy đơn hàng



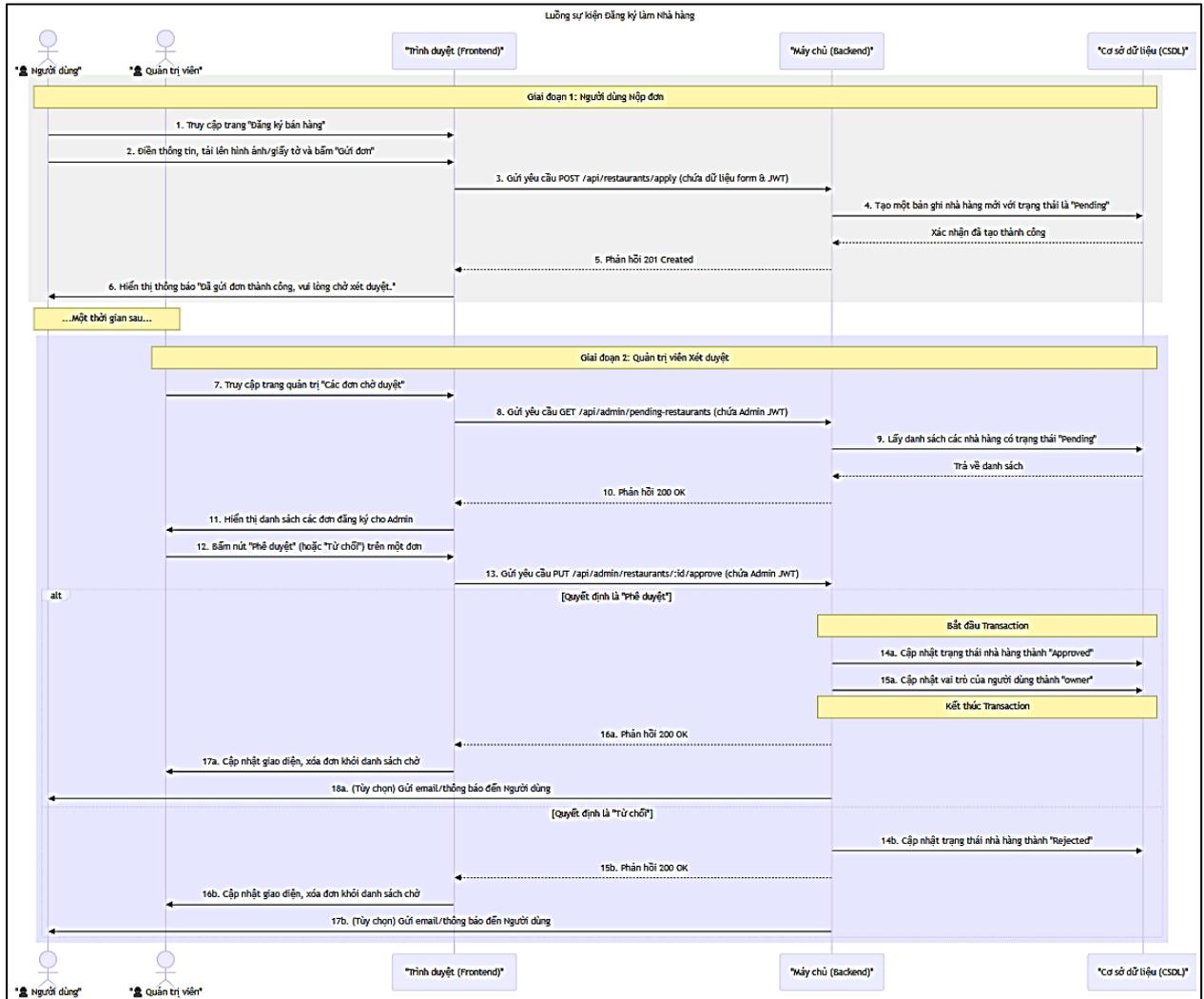
Ảnh 3.6. Biểu đồ tuần tự chức năng xem và hủy đơn hàng

3.1.7. Biểu đồ tuần tự chức năng Cập nhật thông tin tài khoản



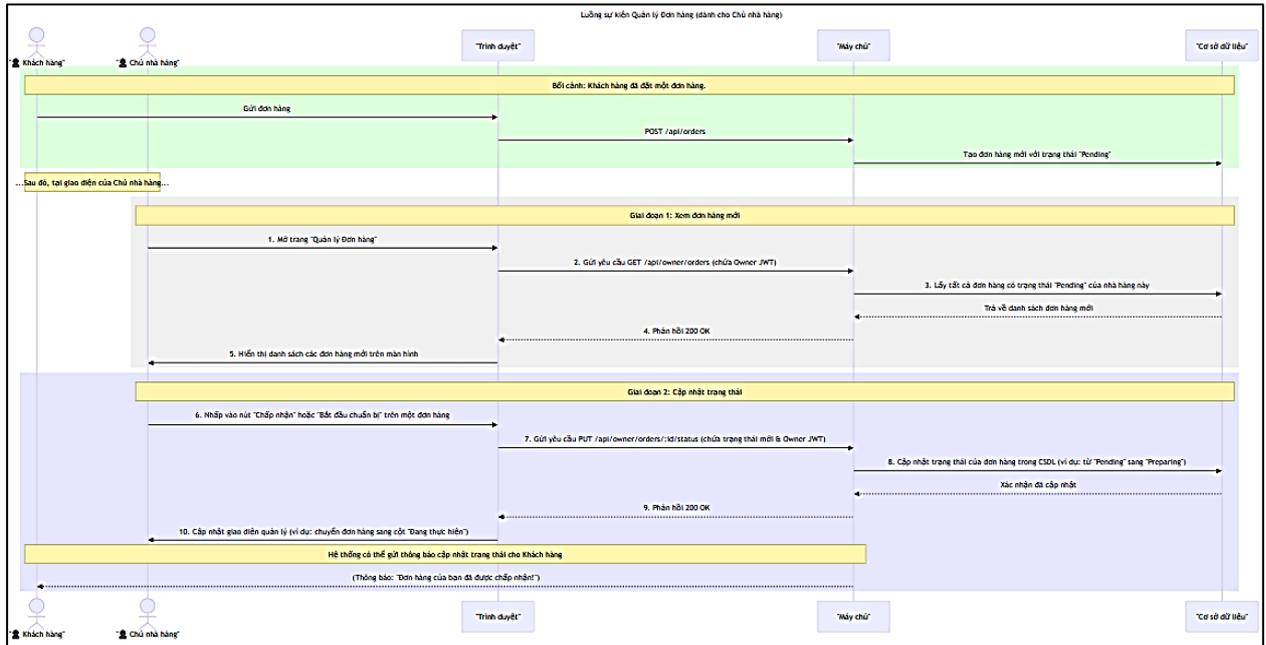
Ảnh 3.7. Biểu đồ chức năng cập nhật thông tin tài khoản

3.1.8. Biểu đồ tuần tự chức năng Đăng ký làm nhà hàng



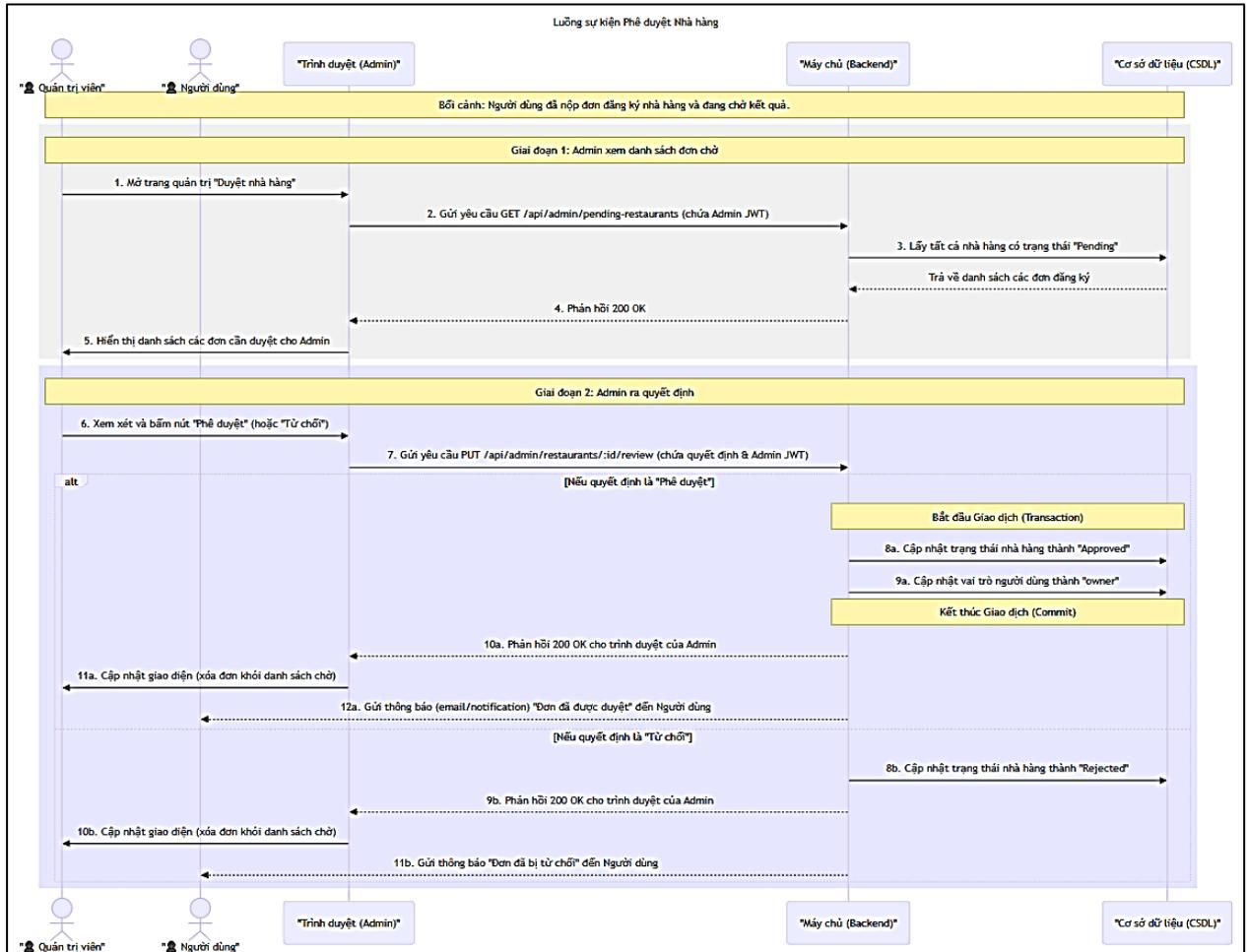
Ảnh 3.8. Biểu đồ tuần tự chức năng Đăng ký làm nhà hàng

3.1.9. Biểu đồ tuần tự chức năng Quản lý đơn hàng



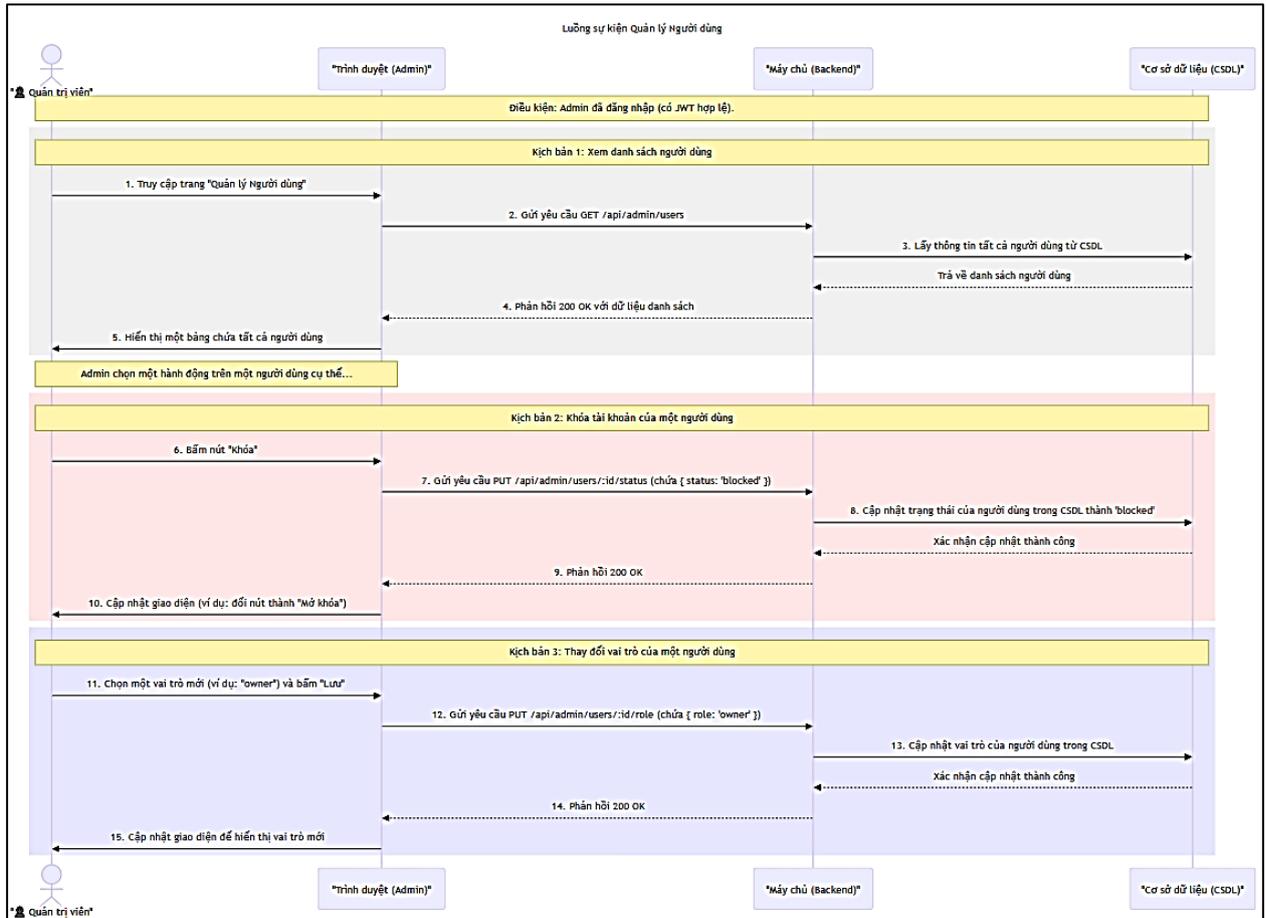
Ảnh 3.9. Biểu đồ tuần tự chức năng quản lý đơn hàng

3.1.10. Biểu đồ tuần tự chức năng Phê duyệt nhà hàng



Ảnh 3.10. Biểu đồ tuần tự chức năng phê duyệt nhà hàng

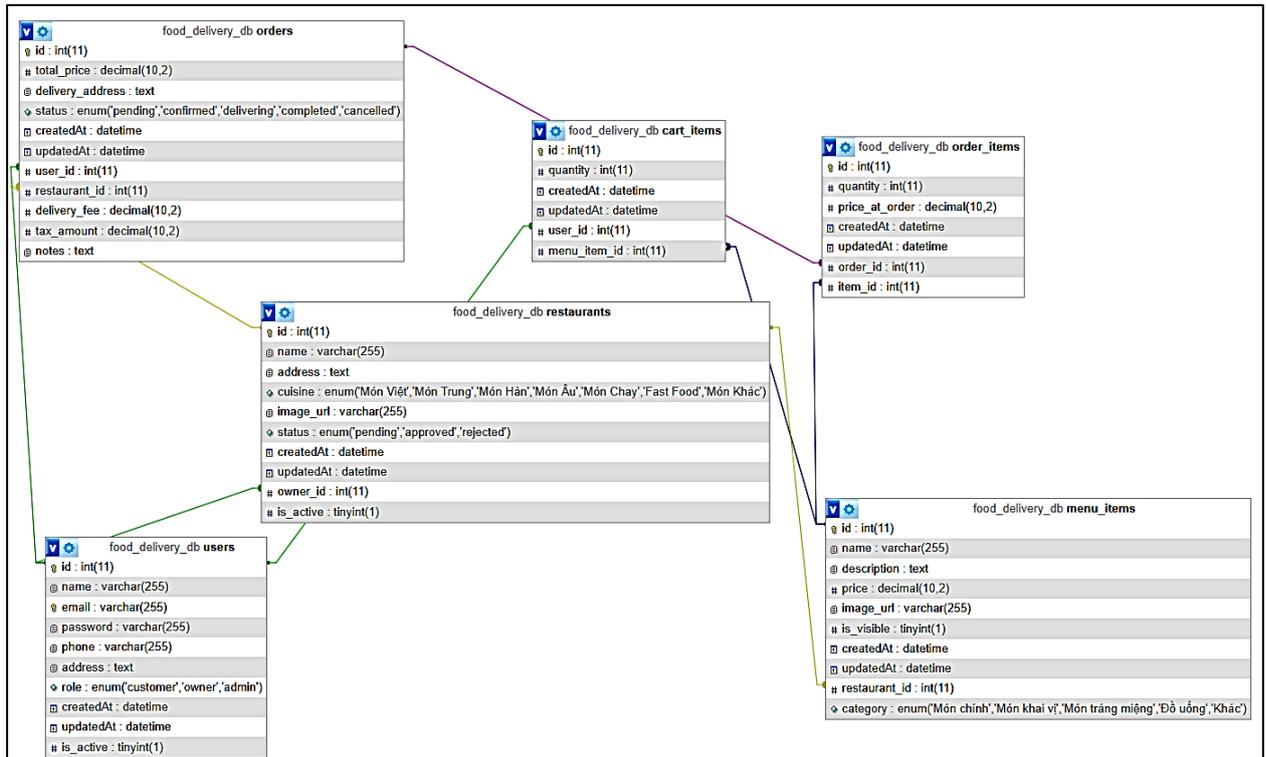
3.1.11. Biểu đồ tuần tự chức năng Quản lý người dùng



Ảnh 3.11. Biểu đồ tuần tự chức năng quản lý người dùng

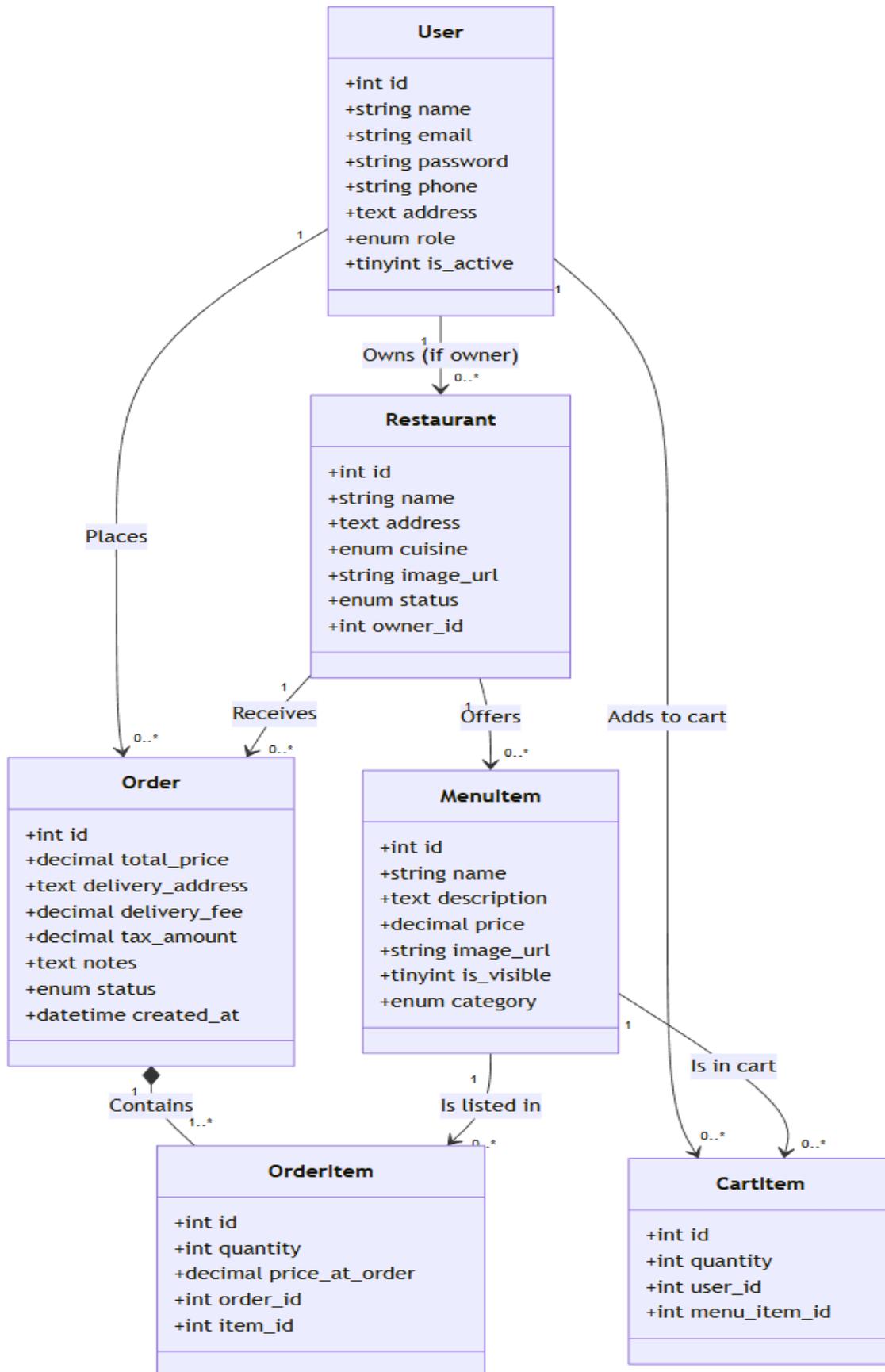
3.2. Thiết kế cơ sở dữ liệu

3.2.1. Lược đồ cơ sở dữ liệu



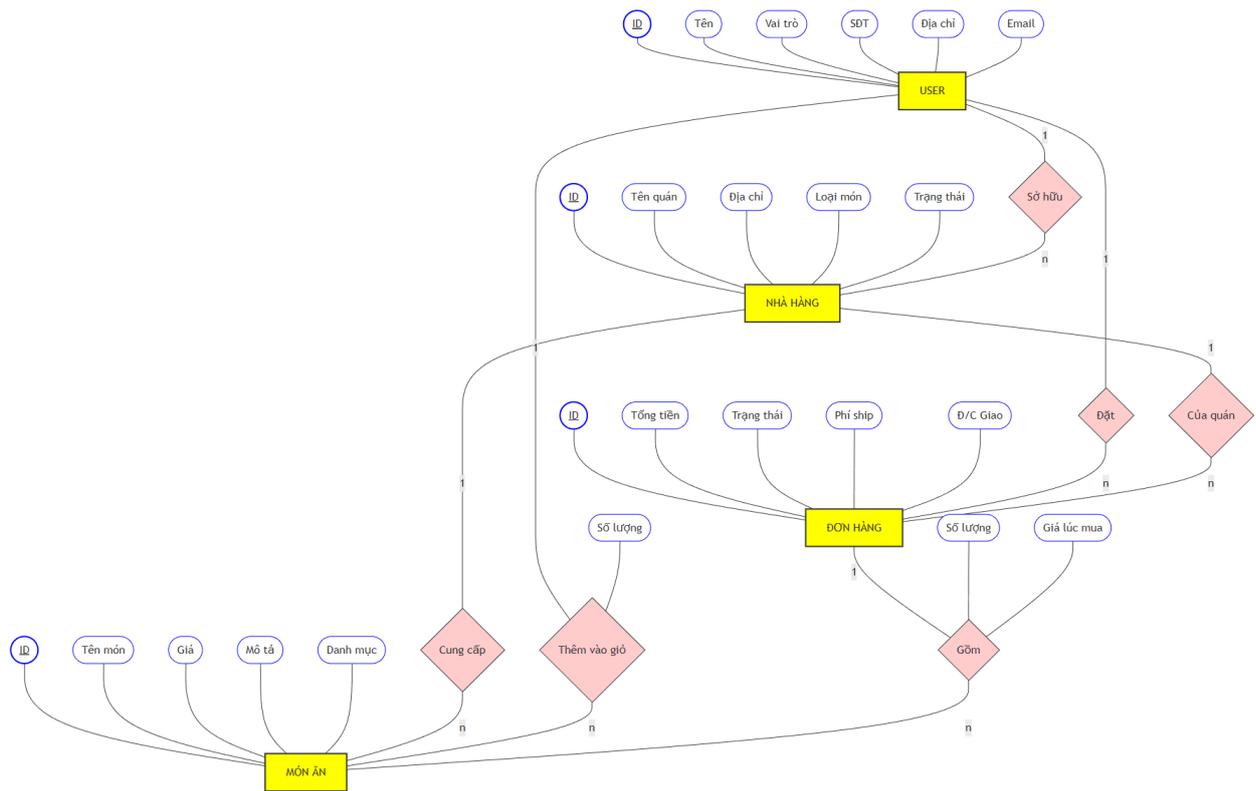
Ảnh 3.12. Lược đồ cơ sở dữ liệu

3.2.2. Biểu đồ UML



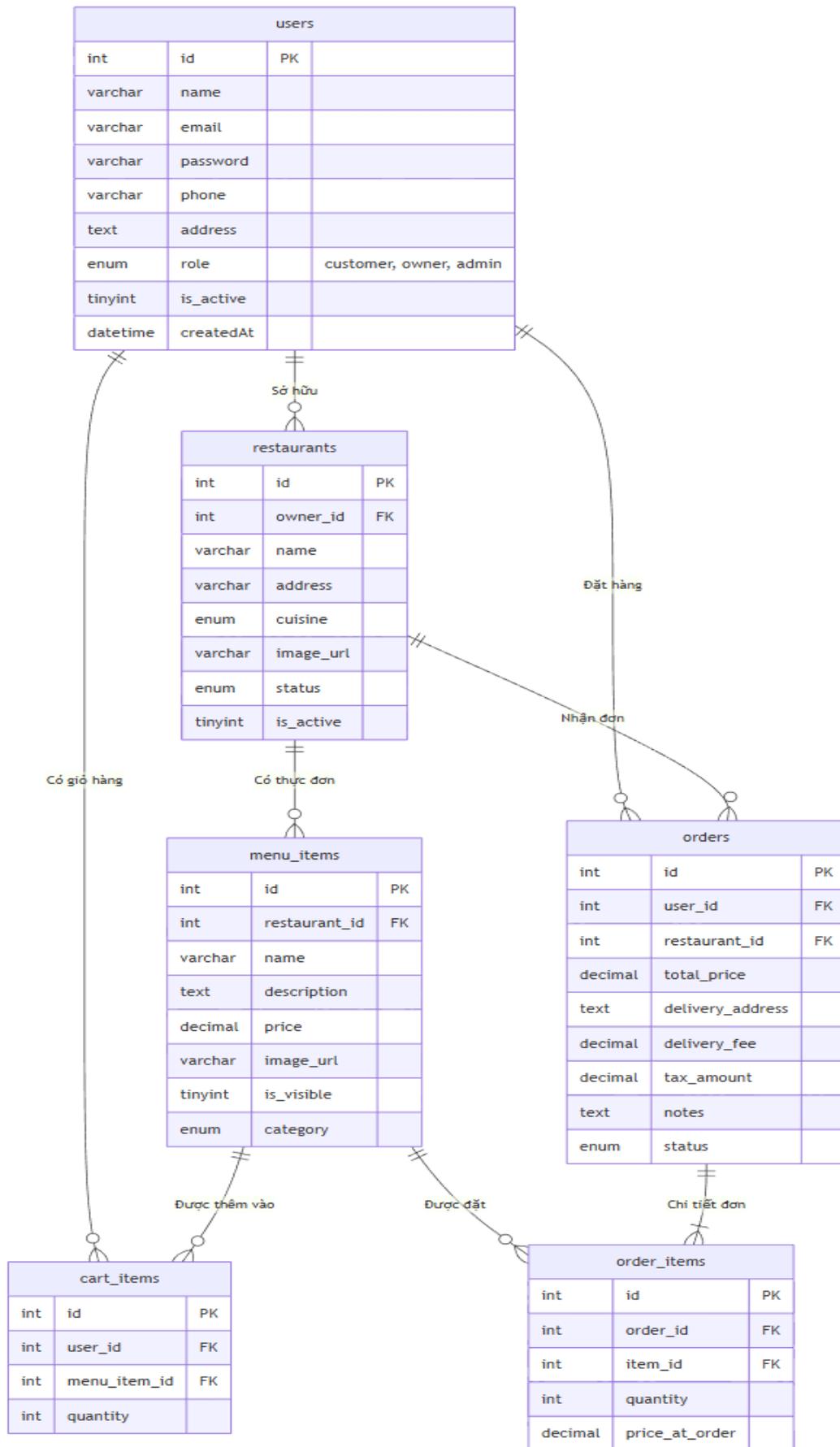
Ảnh 3.13. Biểu đồ UML

3.2.3. Mô hình EER



Ảnh 3.14. Mô hình EER

3.2.4. Mô hình quan hệ



Ảnh 3.15. Mô hình Quan hệ

Danh sách các bảng

STT	Tên bảng	Mô tả
1	Users	Lưu thông tin tài khoản của tất cả người dùng (khách hàng, chủ nhà hàng, admin).
2	Restaurants	Lưu thông tin các nhà hàng (tên, địa chỉ, trạng thái, chủ sở hữu).
3	Menu_Items	Lưu thông tin các món ăn (tên, giá, mô tả, thuộc nhà hàng nào).
4	Orders	Lưu thông tin chung của mỗi đơn hàng (tổng tiền, địa chỉ giao, trạng thái, người đặt).
5	Order_items	Lưu chi tiết các món ăn có trong một đơn hàng cụ thể
6	Cart_Items	Lưu các món ăn đang có trong giỏ hàng của từng người dùng.

Bảng.2. Danh sách các bảng dữ liệu

3.2.5. Chi tiết các bảng

Bảng Users

STT	Tên trường	Kiểu dữ liệu	Ghi chú
1	Id	INT	Khóa chính, tự động tăng (Mã User)
2	name	VARCHAR(255)	Tên người dùng
3	email	VARCHAR(255)	Email đăng nhập, không được trùng.
4	password	VARCHAR(255)	Mật khẩu đã được mã hóa.
5	phone	VARCHAR(255)	Số điện thoại
6	address	TEXT	Địa chỉ người dùng.

7	role	ENUM	Vai trò ('customer', 'owner', 'admin')
8	is_active	BOOLEAN	Trạng thái tài khoản (1: hoạt động, 0: bị khóa).
9	createdAt	DATETIME	Thời gian tạo tài khoản.
10	updatedAt	DATETIME	Thời gian cập nhật lần cuối.

Bảng.3. Bảng User

Bảng menu_items

STT	Tên trường	Kiểu dữ liệu	Ghi chú
1	id	INT	Khóa chính, tự động tăng (Mã món ăn)
2	name	VARCHAR(255)	Tên món ăn
3	description	TEXT	Mô tả chi tiết về món ăn
4	price	DECIMAL(10,2)	Giá bán của món ăn
5	image_url	VARCHAR(255)	Đường dẫn đến file hình ảnh của món ăn
6	is_visible	BOOLEAN	Trạng thái hiển thị (1: hiển thị, 0: ẩn)
7	restaurant_id	INT	Khóa ngoại liên kết với bảng restaurants
8	category	ENUM	Danh mục món ăn (VD: 'Món chính', 'Món khai vị', 'Đồ uống'...)
9	createdAt	DATETIME	Thời gian tạo
10	updatedAt	DATETIME	Thời gian cập nhật lần cuối

Bảng.4. Bảng menu_items

Bảng restaurants

STT	Tên trường	Kiểu dữ liệu	Ghi chú
1	id	INT	Khóa chính, tự động tăng (Mã nhà hàng)
2	name	VARCHAR(255)	Tên của nhà hàng
3	address	TEXT	Địa chỉ của nhà hàng
4	cuisine	ENUM	Loại hình ẩm thực (VD: 'Món Việt', 'Món Hàn', 'Món Chay'...)
5	image_url	VARCHAR(255)	Đường dẫn ảnh đại diện của nhà hàng
6	status	ENUM	Trạng thái đăng ký ('pending', 'approved', 'rejected')
7	owner_id	INT	Khóa ngoại liên kết với bảng users
8	is_active	BOOLEAN	Trạng thái hoạt động (1: hoạt động, 0: bị vô hiệu hóa)
9	createdAt	DATETIME	Thời gian tạo
10	updatedAt	DATETIME	Thời gian cập nhật lần cuối

Bảng 5. Bảng restaurants

Bảng Order

STT	Tên trường	Kiểu dữ liệu	Ghi chú
1	id	INT	Khóa chính, tự động tăng (Mã đơn hàng)
2	total_price	DECIMAL(10,2)	Tổng giá trị cuối cùng của đơn hàng (đã bao gồm phí, thuế)

3	delivery_address	TEXT	Địa chỉ giao hàng của khách
4	status	ENUM	Trạng thái đơn hàng ('pending', 'confirmed', 'delivering', 'completed', 'cancelled')
5	user_id	INT	Khóa ngoại liên kết với bảng users
6	restaurant_id	INT	Khóa ngoại liên kết với bảng restaurants
7	delivery_fee	DECIMAL(10,2)	Phí giao hàng
8	tax_amount	DECIMAL(10,2)	Số tiền thuế
9	notes	TEXT	Ghi chú của khách (VD: "ít cay")
10	createdAt	DATETIME	Thời gian tạo
11	updatedAt	DATETIME	Thời gian cập nhật trạng thái lần cuối

Bảng 5. Bảng Order

Bảng order_items

STT	Tên trường	Kiểu dữ liệu	Ghi chú
1	id	INT	Khóa chính, tự động tăng
2	quantity	INT	Số lượng của món ăn trong đơn hàng
3	price_at_order	DECIMAL(10,2)	Giá món ăn tại thời điểm đặt hàng
4	order_id	INT	Khóa ngoại liên kết với bảng orders

5	item_id	INT	Khóa ngoại liên kết với bảng menu_items
6	createdAt	DATETIME	Thời gian tạo
7	updatedAt	DATETIME	Thời gian cập nhật

Bảng 6. Bảng order_items

Bảng cart_items

STT	Tên trường	Kiểu dữ liệu	Ghi chú
1	id	INT	Khóa chính, tự động tăng
2	quantity	INT	Số lượng của món ăn trong giỏ hàng
3	user_id	INT	Khóa ngoại liên kết với bảng users
4	menu_item_id	INT	Khóa ngoại liên kết với bảng menu_items
5	createdAt	DATETIME	Thời gian thêm vào giỏ
6	updatedAt	DATETIME	Thời gian cập nhật lần cuối

Bảng 7. Bảng cart_items

CHƯƠNG IV: KIỂM THỬ VÀ TRIỂN KHAI WEBSITE

4.1. Kiểm thử

4.1.1. Phân tích các trường hợp kiểm thử

STT	Chức năng	Mục tiêu kiểm thử
1	Đăng nhập	Kiểm tra giao diện, hiển thị đủ các trường và nút bấm. Kiểm tra tính hợp lệ và thông báo lỗi cho Email, Mật khẩu. Kiểm tra đăng nhập thành công với tài khoản hợp lệ và thất bại với tài khoản không hợp lệ.
2	Đăng ký	Kiểm tra giao diện đầy đủ trường và nút bấm. Kiểm tra tính hợp lệ và thông báo lỗi cho Họ và tên, Email, Số điện thoại, Mật khẩu, Xác nhận mật khẩu. Kiểm tra đăng ký thành công và kiểm tra lỗi khi email đã tồn tại.
3	Cập nhật tài khoản	Kiểm tra giao diện, hiển thị đúng thông tin người dùng. Kiểm tra thay đổi và lưu Họ và tên, Địa chỉ, Số điện thoại. Kiểm tra chức năng thay đổi mật khẩu.
4	Tìm kiếm Nhà hàng	Kiểm tra giao diện trang chủ, ô tìm kiếm. Kiểm tra tìm kiếm theo tên nhà hàng và hiển thị kết quả. Kiểm tra chức năng sắp xếp danh sách nhà hàng.
5	Giỏ hàng	Kiểm tra giao diện, hiển thị đúng các món đã thêm. Kiểm tra tăng/giảm số lượng, xóa món. Kiểm tra cập nhật tổng tiền.

6	Xem chi tiết Nhà hàng	Kiểm tra hiển thị thông tin nhà hàng và thực đơn. Kiểm tra mở chi tiết món ăn, thay đổi số lượng. Kiểm tra thêm món ăn vào giỏ hàng.
7	Đặt hàng	Kiểm tra luồng từ giỏ hàng sang trang đặt hàng. Kiểm tra hiển thị thông tin người nhận và tóm tắt đơn hàng. Kiểm tra đặt hàng thành công và giỏ hàng được xóa.
8	Xem & Hủy đơn hàng (User)	Kiểm tra hiển thị lịch sử đơn hàng. Kiểm tra chỉ cho phép hủy đơn ở trạng thái Pending .
9	Quản lý Thực đơn (Owner)	Kiểm tra giao diện quản lý thực đơn. Kiểm tra thêm, sửa, xóa món ăn của nhà hàng.
10	Quản lý Đơn hàng (Owner)	Kiểm tra giao diện quản lý đơn hàng. Kiểm tra thay đổi trạng thái đơn: Pending → Confirmed → Delivering → Delivered.
11	Đăng ký Nhà hàng (User)	Kiểm tra giao diện form đăng ký nhà hàng. Kiểm tra các trường bắt buộc. Kiểm tra gửi đơn thành công và thông báo chờ duyệt.
12	Phê duyệt Nhà hàng (Admin)	Kiểm tra danh sách nhà hàng chờ duyệt. Kiểm tra chức năng Duyệt (user trở thành owner). Kiểm tra chức năng Từ chối.
13	Quản lý Người dùng (Admin)	Kiểm tra hiển thị danh sách người dùng. Kiểm tra khóa và mở khóa tài khoản.

Bảng 6. Phân tích các trường hợp kiểm thử

4.1.2. Danh sách Testcase chi tiết

- Cột trái phần Pass: Đây là số lượng test case Đạt. Tức là các trường hợp kiểm thử đã chạy thành công đúng như mong đợi.
- Cột phải phần Pass: Đây là số lượng Test case TRƯỢT/LỖI (Fail). Tức là các trường hợp kiểm thử phát hiện ra lỗi hoặc không chạy đúng mong đợi.

4.1.3. Kết quả thực thi Test

STT	Chức năng	Pass	
1	Đăng nhập	14	0
2	Đăng ký	38	4
3	Cập nhật tài khoản	30	4
4	Tìm kiếm & Chi tiết Nhà hàng	29	1
5	Giỏ hàng & Đặt hàng	33	1
6	Đăng ký Nhà hàng (User)	15	1
7	Quản lý Thực đơn (Owner)	46	1
8	Quản lý Đơn hàng (Owner)	20	1
9	Phê duyệt Nhà hàng (Admin)	16	0
10	Quản lý Người dùng (Admin)	12	0
Tổng		253	13

Bảng 7. Bảng kết quả thực thi Test

4.2. Yêu cầu về cài đặt

4.2.1. Yêu cầu phần cứng:

- Để đảm bảo hệ thống hoạt động ổn định, máy tính phát triển và triển khai cần đáp ứng các yêu cầu tối thiểu sau:
 - + CPU: Intel Core i3 hoặc tương đương.
 - + RAM: Tối thiểu 8GB.

+ Ổ cứng: Tối thiểu 20GB dung lượng trống.

4.2.2. Yêu cầu phần mềm:

- Để cài đặt và vận hành dự án, cần có các phần mềm sau:

+ 1. Hệ điều hành

- Hệ điều hành Windows 10 trở lên, hoặc các hệ điều hành khác như macOS, Linux.

+ 2. Node.js và NPM

- **Node.js:** Là môi trường thực thi (runtime) cho cả backend (Express.js) và frontend (React.js). Phiên bản đề nghị là 16.x trở lên.
- **NPM (Node Package Manager):** Đi kèm với Node.js, được sử dụng để quản lý các thư viện và gói phụ thuộc của dự án.

+ 3. Môi trường Cơ sở dữ liệu

- **XAMPP:** Là một gói phần mềm tích hợp, cung cấp môi trường máy chủ web cần thiết. Trong dự án này, XAMPP được sử dụng chủ yếu để cung cấp:
- **Cơ sở dữ liệu MariaDB (hoặc MySQL):** Hệ quản trị cơ sở dữ liệu để lưu trữ toàn bộ dữ liệu của ứng dụng.
- **phpMyAdmin:** Giao diện web để quản lý cơ sở dữ liệu một cách trực quan.

+ 4. Trình soạn thảo mã nguồn (Code Editor)

- **Visual Studio Code (VSCode):** Là một trình soạn thảo mã nguồn đa nền tảng, được phát triển bởi Microsoft. Nhờ vào hệ sinh thái tiện ích mở rộng (extensions) phong phú, VSCode có thể hoạt động như một IDE nhẹ hỗ trợ nhiều ngôn ngữ lập trình như JavaScript, Python, C++, Java, PHP, Go và nhiều ngôn ngữ khác.
- VSCode cung cấp các tính năng nổi bật như:
- Tự động hoàn thiện mã (IntelliSense)

- Gỡ lỗi trực tiếp trong trình soạn thảo (debugging)
- Tích hợp Git để quản lý mã nguồn
- Terminal tích hợp
- Hỗ trợ lập trình từ xa (Remote Development)
- Hàng nghìn tiện ích mở rộng để hỗ trợ framework, ngôn ngữ và công cụ phát triển.

+ 5. Git

- Hệ thống quản lý phiên bản phân tán, được sử dụng để theo dõi các thay đổi trong mã nguồn trong quá trình phát triển.

+ 6. Trình duyệt Web

- Một trình duyệt web hiện đại như Google Chrome, Mozilla Firefox, hoặc Microsoft Edge để chạy và kiểm thử ứng dụng web phía người dùng.

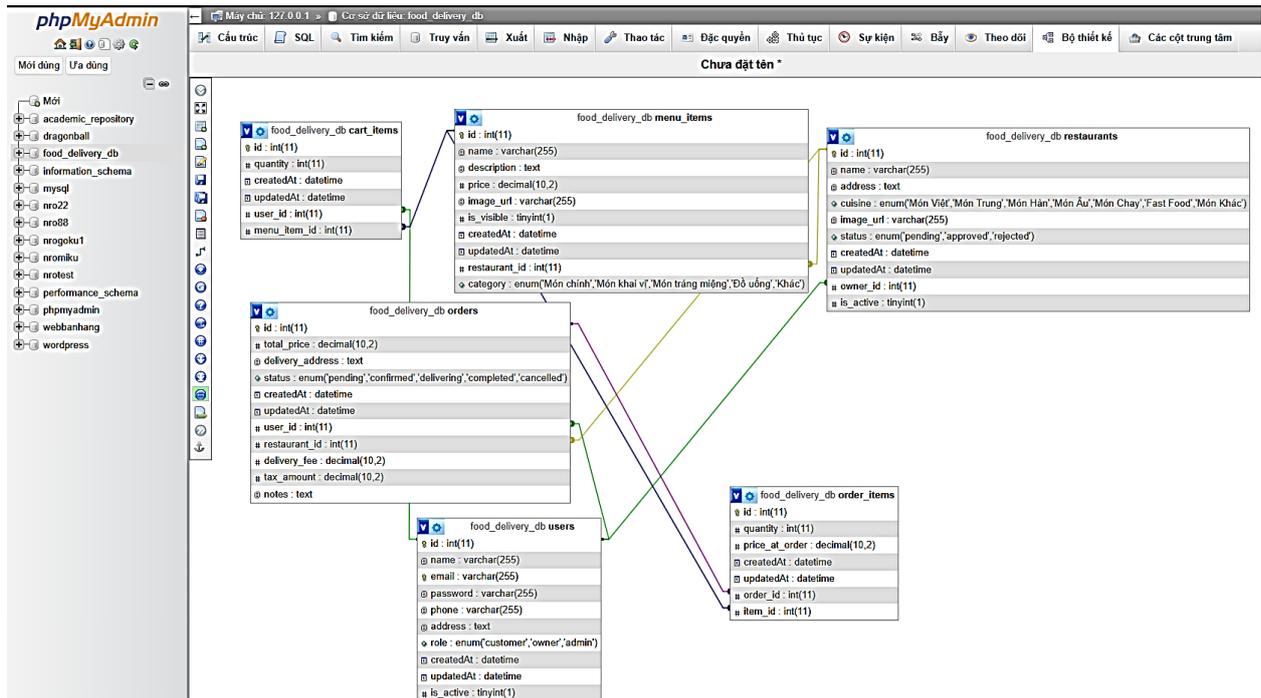
4.2.3. Yêu cầu mạng Internet:

Thiết bị cần kết nối mạng để:

- Tải và cập nhật tiện ích mở rộng (extensions)
- Cập nhật phần mềm và tính năng mới
- Đồng bộ hóa cài đặt nếu dùng tài khoản Microsoft
- Làm việc với các dịch vụ như GitHub, Live Share, Remote SSH, v.v.

4.3. Kết quả cài đặt môi trường

4.3.1. Data Access Layer (MySQL)



Ảnh 4.1. Cài đặt cơ sở dữ liệu

MySQL là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) phổ biến và mạnh mẽ, thường được sử dụng trong các ứng dụng web, doanh nghiệp và Website lớn. Khi triển khai trên các nền tảng đám mây như Amazon RDS, Google Cloud SQL hoặc Azure Database for MySQL, User có thể tận dụng một môi trường MySQL được quản lý hoàn toàn, giúp đơn giản hóa quá trình cài đặt, vận hành và mở rộng Website cơ sở dữ liệu.

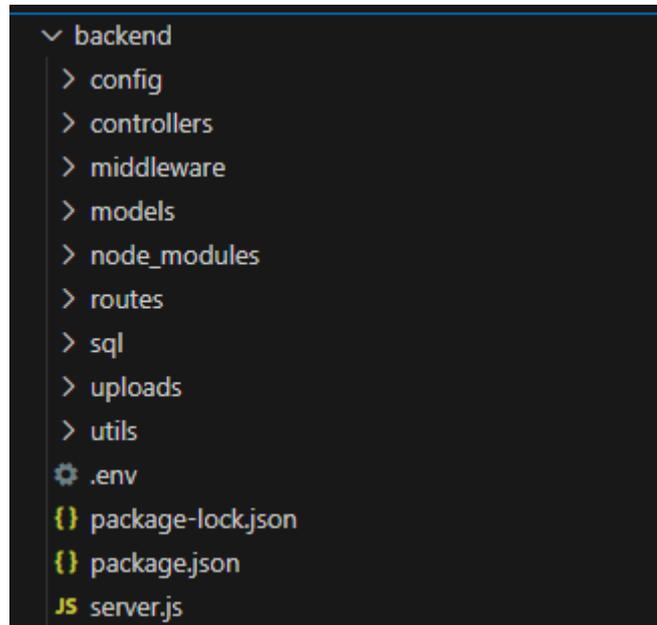
Với MySQL trên nền tảng đám mây, User có thể dễ dàng:

- Thiết lập và triển khai cơ sở dữ liệu chỉ trong vài bước
- Tự động sao lưu và phục hồi dữ liệu
- Quản lý hiệu năng và bảo mật với các công cụ tích hợp sẵn
- Mở rộng Website theo nhu cầu mà không cần can thiệp sâu vào hạ tầng

Giải pháp này đặc biệt phù hợp cho những doanh nghiệp và lập trình viên muốn tập trung vào phát triển ứng dụng mà không phải lo lắng về các vấn đề kỹ thuật như bảo trì máy chủ, cập nhật bảo mật hay tối ưu hiệu suất Website.

4.3.2. *Business Logic Layer (Express)*

- Lớp Business Logic của dự án được xây dựng bằng **Express.js**, một framework phổ biến của Node.js. Lớp này chịu trách nhiệm xử lý các yêu cầu (requests) từ phía frontend, thực thi các nghiệp vụ logic, tương tác với cơ sở dữ liệu (thông qua Lớp Data Access), và gửi phản hồi (responses) trở lại.



Ảnh 4.2. Cài đặt Business Logic Layer (Express)

Cấu trúc thư mục của backend được tổ chức như sau:

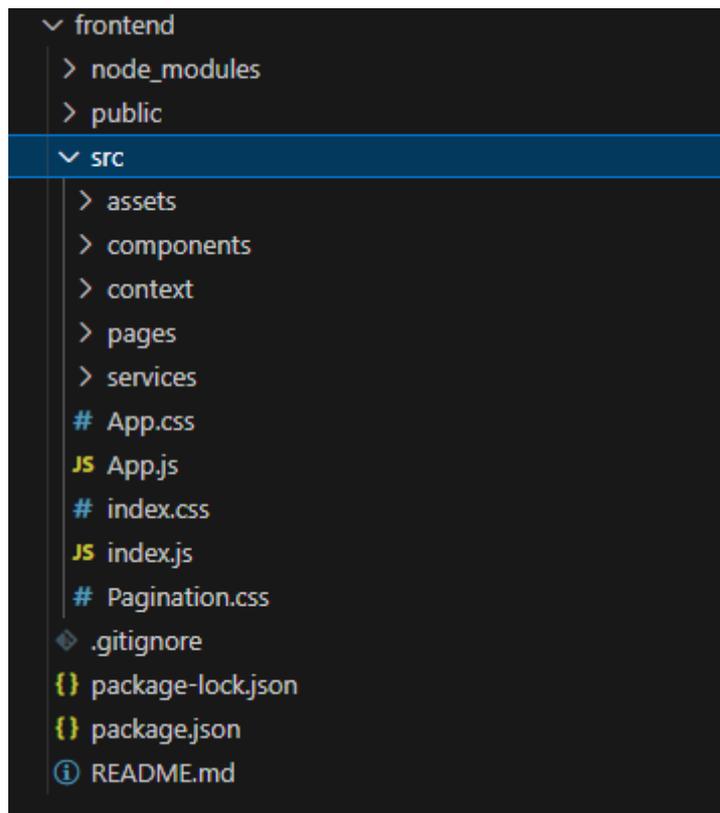
- **server.js**: Là file khởi động của toàn bộ backend. Nó có nhiệm vụ:
 - Khởi tạo Express app.
 - Kết nối với cơ sở dữ liệu qua Sequelize.
 - Áp dụng các middleware cần thiết (như cors, express.json).
 - Lắng nghe các request từ một cổng (port) nhất định.
- **config/**: Chứa các file cấu hình cho dự án.
 - **db.config.js**: Chứa thông tin cấu hình để kết nối với cơ sở dữ liệu MySQL/MariaDB (thông qua XAMPP).
- **routes/**: Định nghĩa các tuyến đường (API endpoints) của ứng dụng.

- Mỗi file trong này (ví dụ: `auth.routes.js`, `order.routes.js`) sẽ ánh xạ một URL và một phương thức HTTP (GET, POST, PUT) tới một hàm xử lý cụ thể trong controller.
- **controllers/**: Là nơi chứa đựng phần lớn logic nghiệp vụ.
 - Mỗi hàm trong controller chịu trách nhiệm xử lý một yêu cầu duy nhất (ví dụ: `registerUser`, `createOrder`).
 - Nó nhận dữ liệu từ request, sử dụng các Model của Sequelize để thao tác với database, và cuối cùng là trả về response (dữ liệu JSON hoặc mã lỗi) cho client.
- **middleware/**: Chứa các hàm trung gian.
 - Các hàm này sẽ được thực thi trước khi request đến được controller.
 - `auth.middleware.js`: Một middleware quan trọng, dùng để kiểm tra token xác thực của người dùng, bảo vệ các route cần đăng nhập và xác định quyền hạn (role) của họ.
- **models/**: (Thuộc lớp Data Access nhưng làm việc trực tiếp với Business Logic)
 - Định nghĩa cấu trúc của các đối tượng dữ liệu (User, Restaurant, Order,...) mà controller sẽ sử dụng.
- **utils/**: Chứa các hàm tiện ích có thể tái sử dụng ở nhiều nơi.
 - `generateToken.js`: Chứa hàm để tạo ra JSON Web Token (JWT) cho người dùng sau khi họ đăng nhập hoặc đăng ký thành công.
- **uploads/**: Thư mục dùng để lưu trữ các file tĩnh được người dùng tải lên, chủ yếu là hình ảnh của nhà hàng và món ăn.
- **.env**: File này lưu trữ các biến môi trường và các thông tin nhạy cảm như mật khẩu database, khóa bí mật để tạo JWT. Việc này giúp bảo mật thông tin và tách biệt cấu hình ra khỏi mã nguồn.

4.3.3. *Presentation Layer (ReactJS)*

- Lớp Presentation (Giao diện) được xây dựng dưới dạng một **Ứng dụng trang đơn (Single Page Application - SPA)** bằng thư viện **React.js**. Lớp này chịu trách

nhiệm hiển thị toàn bộ giao diện người dùng, xử lý các tương tác và giao tiếp với Business Logic Layer (backend) thông qua các lời gọi API.



Ảnh 4.3. Cài đặy Presentation Layer (ReactJS)

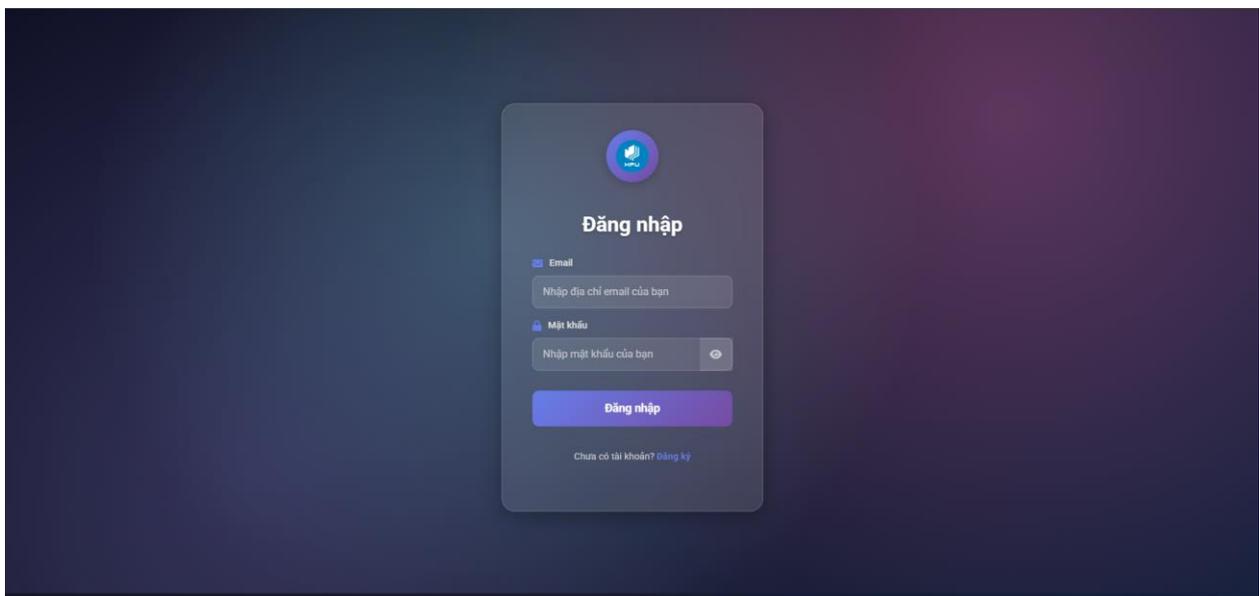
Cấu trúc thư mục src của frontend được tổ chức như sau:

- **App.js**: Là component gốc của ứng dụng. File này chịu trách nhiệm chính cho việc **định tuyến (routing)**, quyết định component page nào sẽ được hiển thị tương ứng với một URL trên trình duyệt (sử dụng thư viện react-router-dom).
- **pages/**: Chứa các component lớn, đại diện cho một trang hoàn chỉnh của ứng dụng.
 - Ví dụ: HomePage.js, LoginPage.js, ProfilePage.js, AdminDashboard.js.
 - Mỗi page sẽ tự chịu trách nhiệm về bố cục và logic của trang đó, đồng thời gọi và sắp xếp các component nhỏ hơn.
- **components/**: Chứa các component nhỏ, có khả năng **tái sử dụng** ở nhiều trang khác nhau để đảm bảo tính nhất quán và tránh lặp code.
 - Ví dụ: Header.js, Footer.js, RestaurantCard.js, MenuItemModal.js.
- **context/**: Quản lý trạng thái toàn cục (global state) của ứng dụng bằng React Context API.

- AuthContext.js: Quản lý trạng thái đăng nhập của người dùng (thông tin user, token, vai trò).
- CartContext.js: Quản lý trạng thái của giỏ hàng (danh sách các món ăn, số lượng) trên toàn bộ ứng dụng.
- **services/**: Là lớp chịu trách nhiệm giao tiếp với backend API.
 - Mỗi file trong này (ví dụ: auth.service.js, order.service.js) đóng gói logic để thực hiện các lời gọi HTTP (GET, POST, PUT) đến server Express, thường sử dụng thư viện axios.
- **assets/**: Chứa các tài nguyên tĩnh như font chữ, icon, hoặc các file CSS toàn cục.
- **public/**: Chứa file index.html gốc của ứng dụng, là điểm vào đầu tiên khi người dùng truy cập trang web, cùng với các tài sản tĩnh khác như favicon.ico.

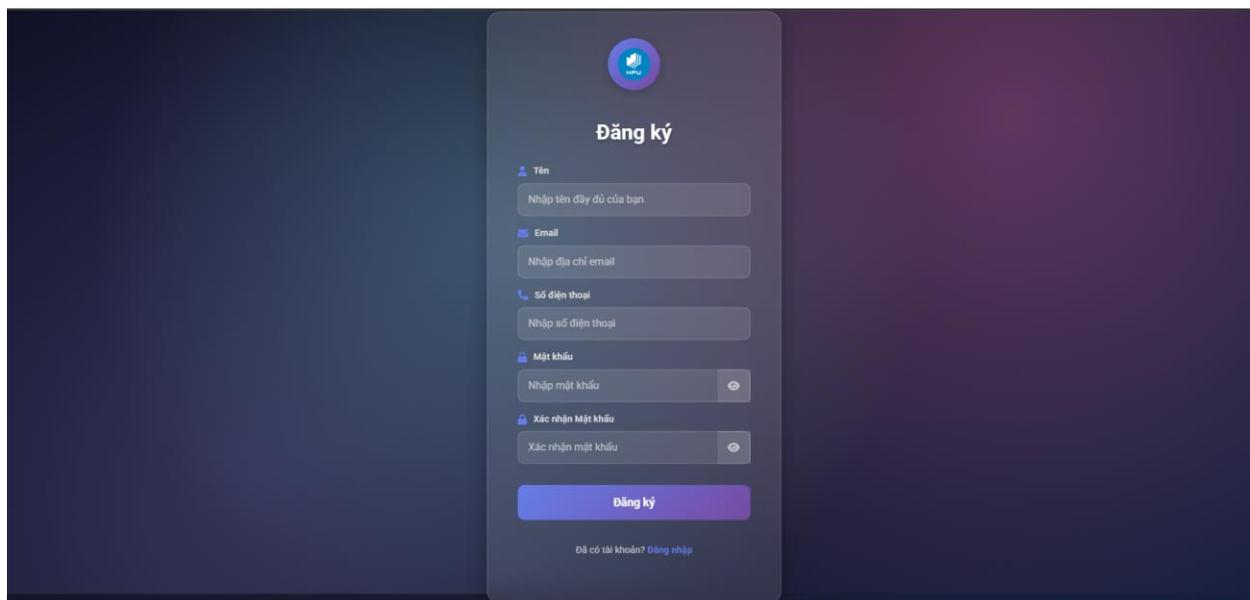
4.4. Kết quả thực nghiệm

4.4.1. Giao diện đăng nhập Website



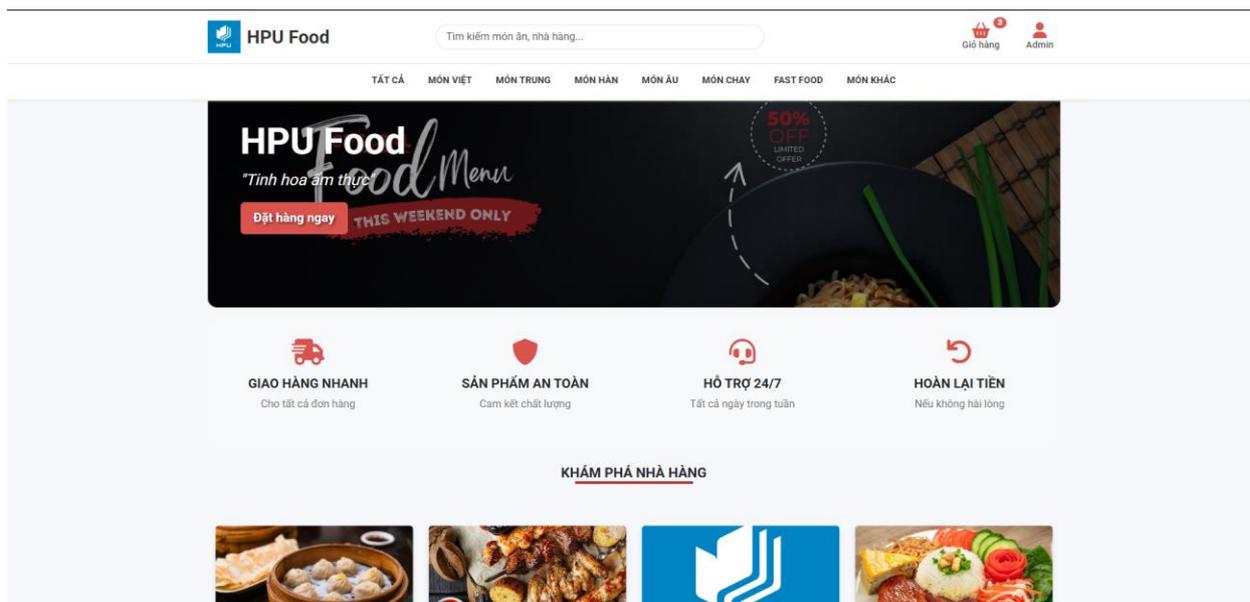
Ảnh.4.4. Giao diện chức năng đăng nhập

4.4.2. Giao diện trang Đăng ký



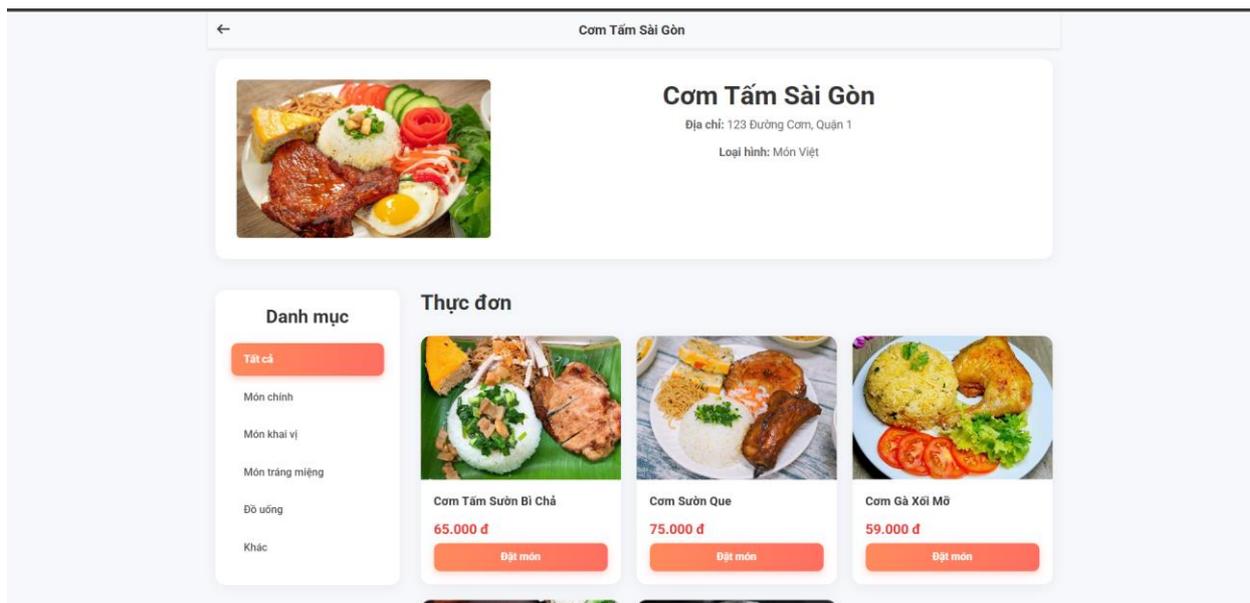
Ảnh 4.5. Giao diện chức năng đăng ký

4.4.3. Giao diện trang chủ Website



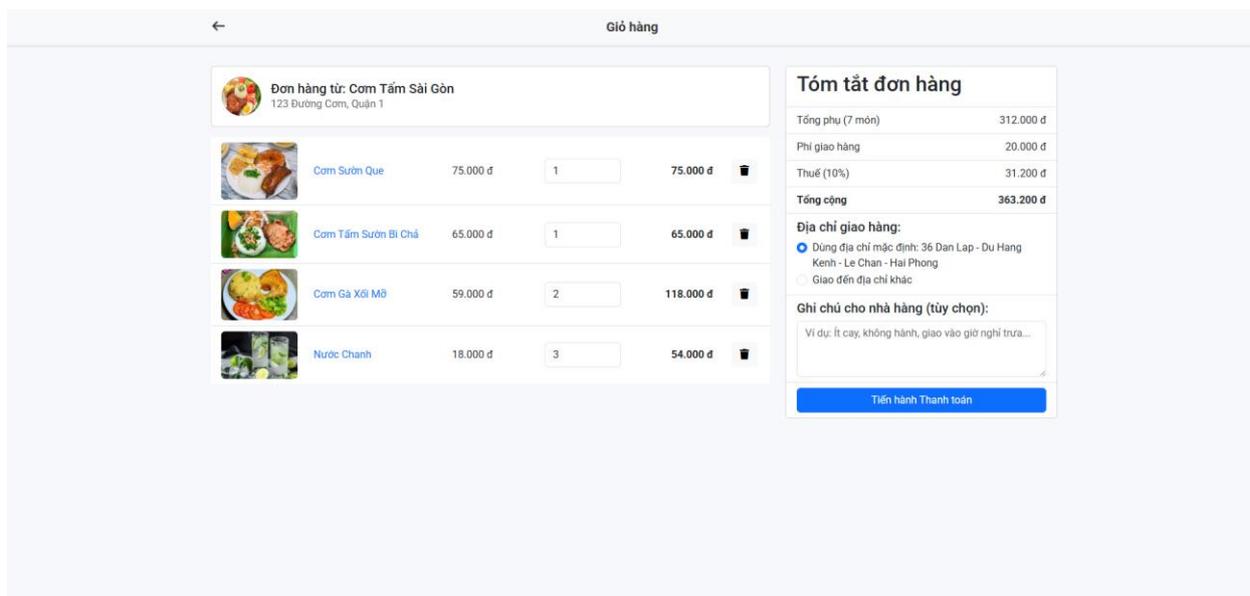
Ảnh 4.6. Giao diện trang chủ màn hình trang chủ

4.4.4. Giao diện trang nhà hàng



Ảnh 4.7. Giao diện chi tiết thực đơn nhà hàng

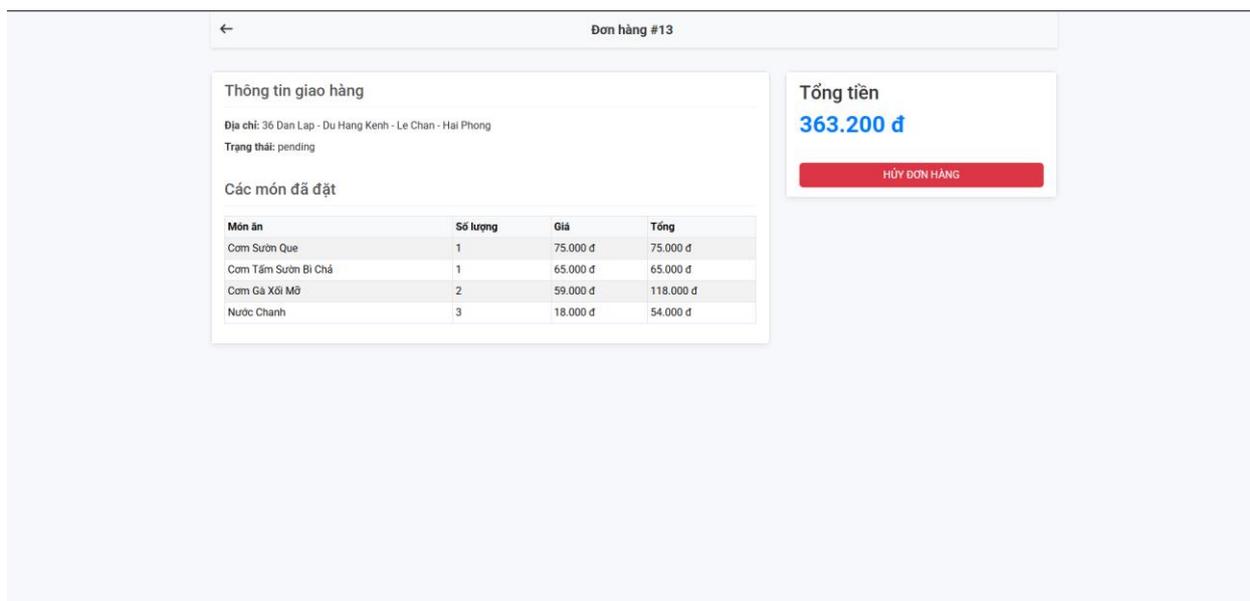
4.4.5. Giao diện giỏ hàng



Ảnh 4.8. Giao diện giỏ hàng

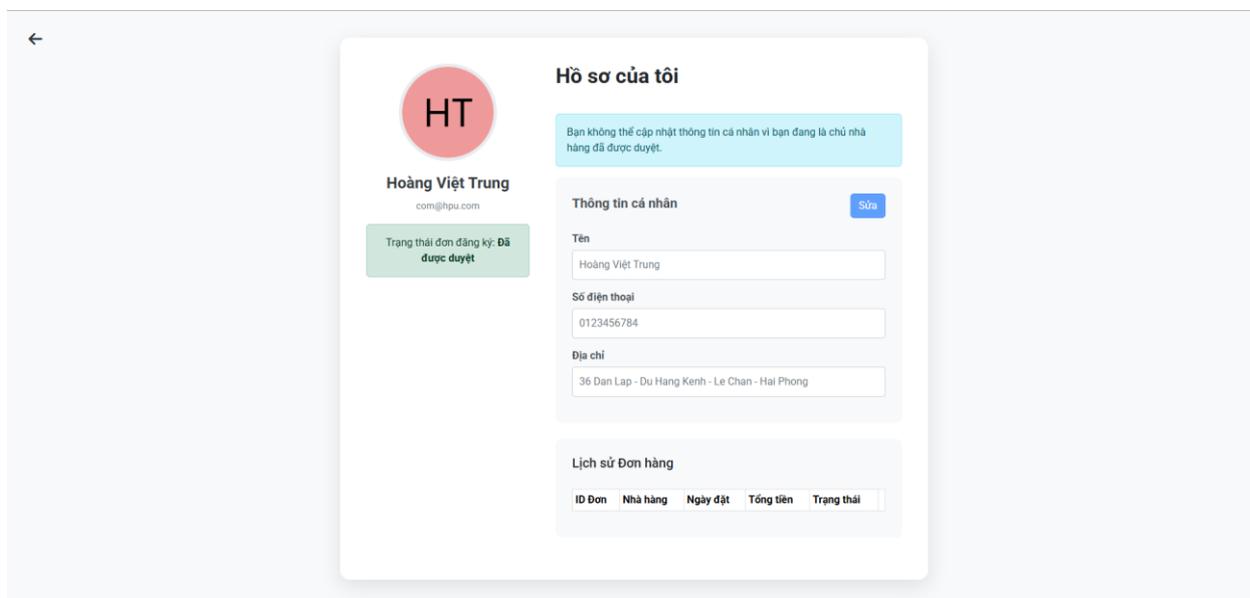
4.4.6. Giao diện chi tiết đơn hàng

Giao diện thông tin các đơn hàng đã đặt:



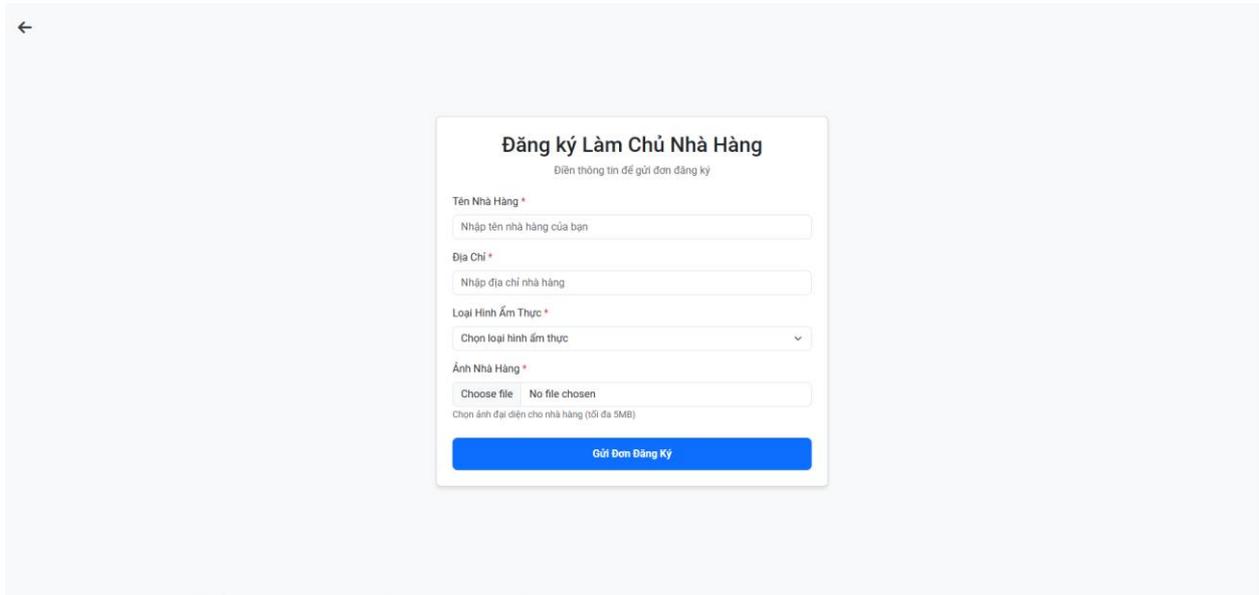
Ảnh 4.9. Giao diện danh sách đơn hàng đã đặt

4.4.7. Giao diện thống kê của nhà hàng



Ảnh 4.10. Giao diện cập nhật tài khoản

4.4.8. Giao diện trang đăng ký làm nhà hàng



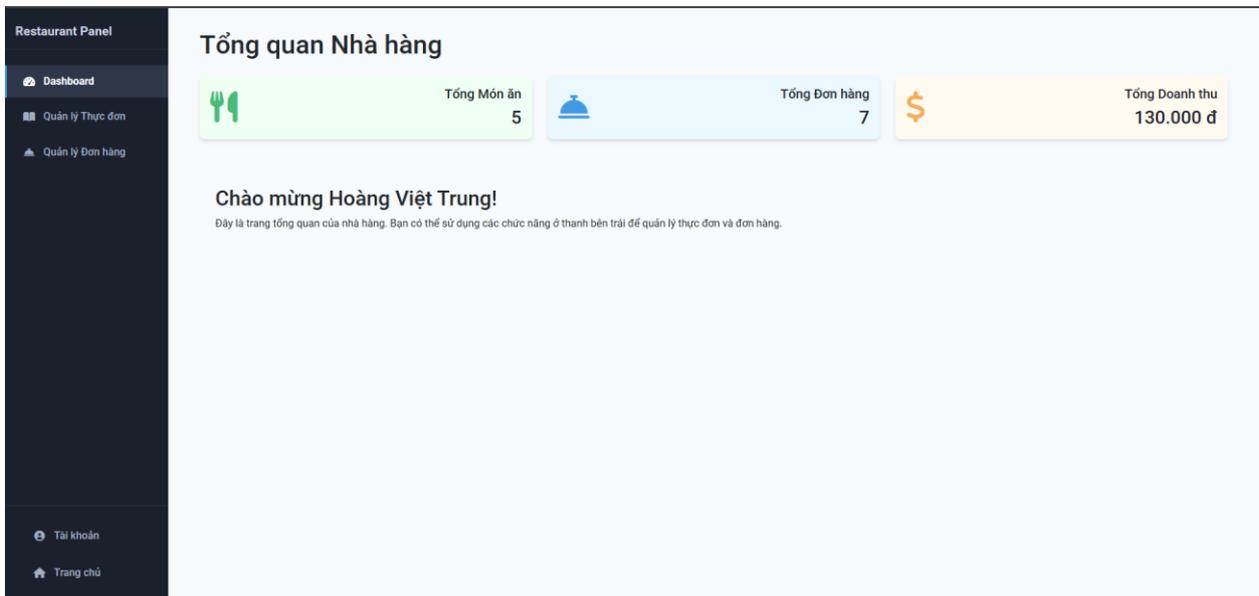
The screenshot shows a registration form titled "Đăng ký Làm Chủ Nhà Hàng" (Register as Restaurant Owner). The form includes the following fields and options:

- Tên Nhà Hàng ***: Nhập tên nhà hàng của bạn (Enter the name of your restaurant)
- Địa Chỉ ***: Nhập địa chỉ nhà hàng (Enter the restaurant address)
- Loại Hình Ẩm Thực ***: Chọn loại hình ẩm thực (Choose the type of cuisine)
- Ảnh Nhà Hàng ***: Choose file | No file chosen. Chon ảnh đại diện cho nhà hàng (tối đa 5MB) (Choose a representative image for the restaurant, up to 5MB)

A blue button labeled "Giữ Đơn Đăng Ký" (Save Registration Form) is located at the bottom of the form.

Ảnh 4.11. Giao diện trang đăng ký làm nhà hàng

4.4.9. Giao diện thống kê của nhà hàng



The screenshot displays the "Restaurant Panel" dashboard. The main content area is titled "Tổng quan Nhà hàng" (Restaurant Overview) and features three summary cards:

- Tổng Món ăn**: 5 (Total Dishes)
- Tổng Đơn hàng**: 7 (Total Orders)
- Tổng Doanh thu**: 130.000 đ (Total Revenue)

Below the summary cards, there is a greeting: "Chào mừng Hoàng Việt Trung!" (Welcome Hoàng Việt Trung!) and a brief description: "Đây là trang tổng quan của nhà hàng. Bạn có thể sử dụng các chức năng ở thanh bên trái để quản lý thực đơn và đơn hàng." (This is the restaurant overview page. You can use the functions in the left sidebar to manage the menu and orders.)

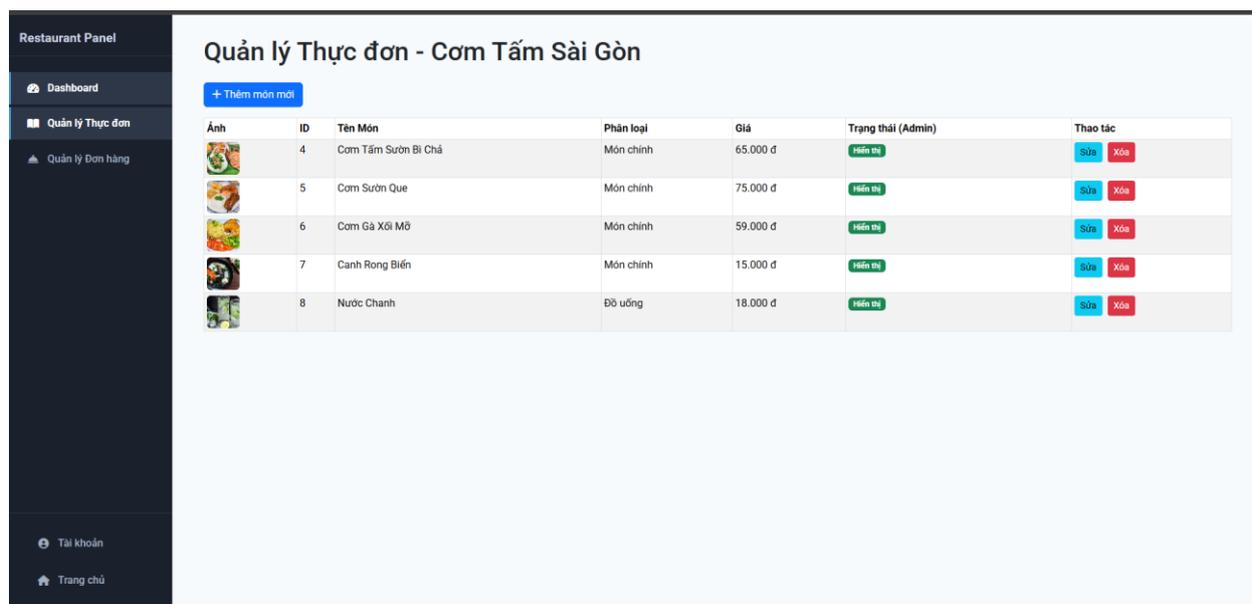
The left sidebar contains the following navigation items:

- Dashboard
- Quản lý Thực đơn (Menu Management)
- Quản lý Đơn hàng (Order Management)
- Tài khoản (Account)
- Trang chủ (Home)

Ảnh 4.12. Giao diện thống kê của nhà hàng

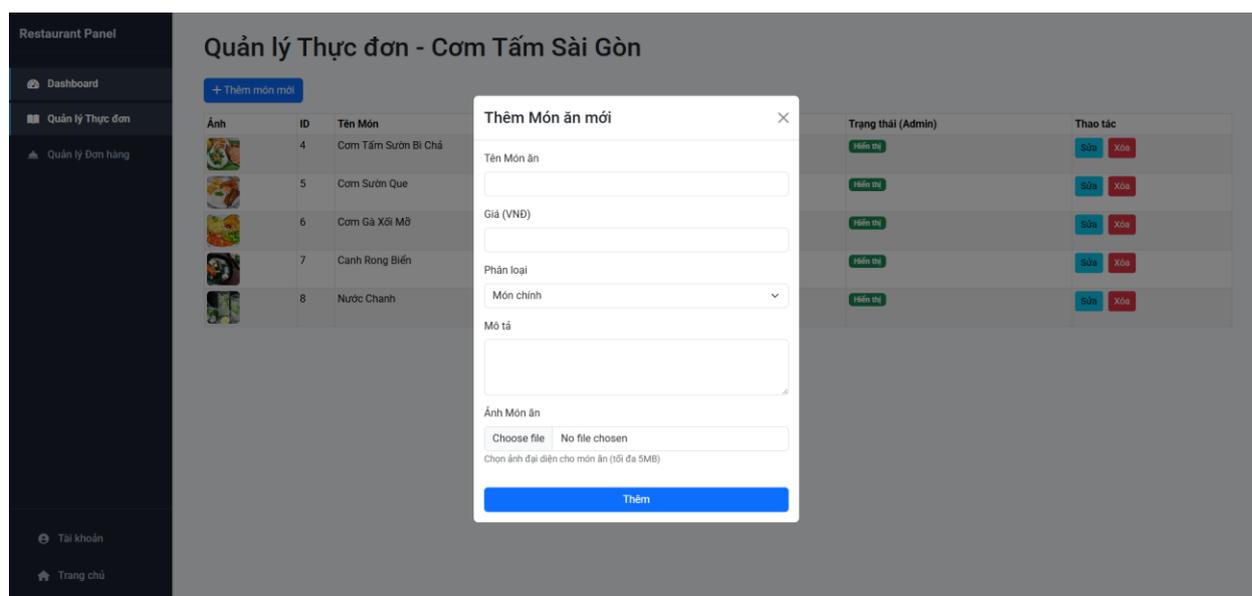
4.4.10. Giao diện quản lý thực đơn

Giao diện quản lý thực đơn:



Ảnh 4.13. Giao diện quản lý thực đơn

Giao diện thêm mới món ăn:



Ảnh 4.14. Giao diện thêm mới món ăn

4.4.11. Giao diện quản lý đơn hàng

Giao diện quản lý đơn hàng:

ID	Khách hàng	Địa chỉ	Tổng tiền	Phí giao hàng	Thuế	Ngày đặt	Trạng thái	Ghi chú	Thao tác
#13	Admin 0123456784	36 Đan Lap - Du Hang Kenh - Le Chan - Hai Phong	363.200 đ	20.000 đ	31.200 đ	07:20:09 10/11/2025	pending	Không có	Chuyển trạng thái
#12	User2 0123456783	số nhà 28 Lê Hồng Phong	321.400 đ	20.000 đ	27.400 đ	13:19:06 9/11/2025	pending	Không có	Chuyển trạng thái
#11	User2 0123456783	Tòa 10 Văn Phòng ABC	238.900 đ	20.000 đ	19.900 đ	15:58:55 6/11/2025	delivered	giao vào giờ nghỉ trưa	Chuyển trạng thái
#10	User2 0123456783	số nhà 28 Lê Hồng Phong	269.700 đ	20.000 đ	22.700 đ	06:46:10 6/11/2025	confirmed	ít cay	Chuyển trạng thái
#9	User2 0123456783	số nhà 28 Lê Hồng Phong	75.000 đ	0 đ	0 đ	06:29:14 6/11/2025	pending	Không có	Chuyển trạng thái
#8	User2 0123456783	Số nhà 28 Lê Hồng Phong	258.000 đ	0 đ	0 đ	06:13:10 6/11/2025	pending	Không có	Chuyển trạng thái
#7	Admin 0123456784	36 Đan Lap - Du Hang Kenh - Le Chan - Hai Phong	130.000 đ	0 đ	0 đ	05:32:21 3/11/2025	completed	Không có	Hoàn thành

Ảnh 4.15. Giao diện quản lý đơn hàng

4.4.12. Giao diện thống kê của Admin

Tổng quan Hệ thống		
	Tổng Người dùng 16	
	Tổng Món ăn 61	
	Tổng Doanh thu 175.000 đ	

Chào mừng Admin!
Đây là trang tổng quan của hệ thống. Bạn có thể sử dụng các chức năng ở thanh bên trái để quản lý nhà hàng, người dùng và các hoạt động khác.

Ảnh 4.16. Giao diện thống kê của admin

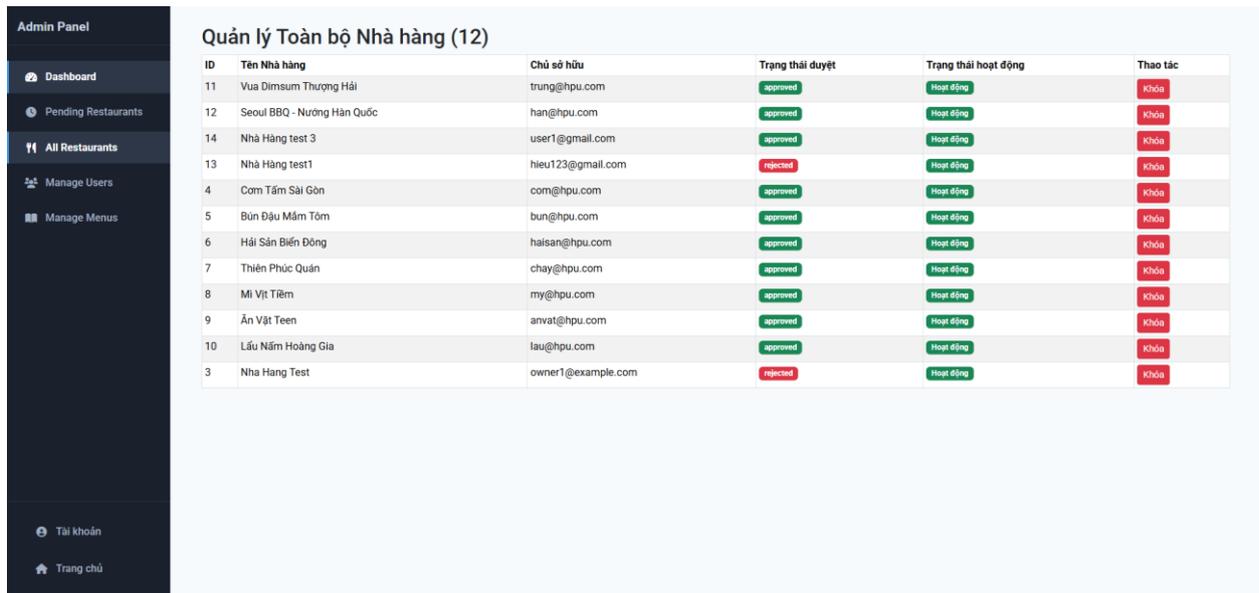
4.4.13. Giao diện trang duyệt đơn nhà hàng



ID	Tên Nhà hàng	Địa chỉ	Loại ẩm thực	Ảnh	Tên chủ sở hữu	SĐT chủ sở hữu	Ngày đăng ký	Trạng thái	Thao tác
1	Nha Hang So 1	456 Duong XYZ, Q2	Món Khác	 Restaurant	Chu Nha Hang Tuong Lai	0123456785	2/11/2025	pending	Duyệt Từ chối

Ảnh 4.17. Giao diện trang duyệt nhà hàng

4.4.14. Giao diện trang quản lý nhà hàng



ID	Tên Nhà hàng	Chủ sở hữu	Trạng thái duyệt	Trạng thái hoạt động	Thao tác
11	Vua Dimsum Thượng Hải	trung@hpu.com	approved	Hoạt động	Khóa
12	Seoul BBQ - Nướng Hàn Quốc	han@hpu.com	approved	Hoạt động	Khóa
14	Nhà Hàng test 3	user1@gmail.com	approved	Hoạt động	Khóa
13	Nhà Hàng test1	hieu123@gmail.com	rejected	Hoạt động	Khóa
4	Cơm Tấm Sài Gòn	com@hpu.com	approved	Hoạt động	Khóa
5	Bún Đậu Mắm Tôm	bun@hpu.com	approved	Hoạt động	Khóa
6	Hải Sản Biển Đông	haisan@hpu.com	approved	Hoạt động	Khóa
7	Thiên Phúc Quán	chay@hpu.com	approved	Hoạt động	Khóa
8	Mì Vịt Tiềm	my@hpu.com	approved	Hoạt động	Khóa
9	Ăn Vịt Teen	anvat@hpu.com	approved	Hoạt động	Khóa
10	Lẩu Nấm Hoàng Gia	lau@hpu.com	approved	Hoạt động	Khóa
3	Nha Hang Test	owner1@example.com	rejected	Hoạt động	Khóa

Ảnh 4.18. Giao diện trang quản lý nhà hàng

4.4.15. Giao diện trang quản lý user

ID	Tên	Email	Vai trò	Trạng thái	Ngày tạo	Hành động
18	chủ nhà hàng 1	owner3@example.com	Khách hàng	Hoạt động	9/11/2025	Khóa
17	Hoàng Việt Trung	user2@gmail.com	Khách hàng	Hoạt động	6/11/2025	Khóa
12	Chủ Quán Món Trung	trung@hpu.com	Chủ nhà hàng	Hoạt động	6/11/2025	Khóa
13	Chủ Quán Món Hàn	han@hpu.com	Chủ nhà hàng	Hoạt động	6/11/2025	Khóa
14	Nguyễn Minh Hiếu	hieu123@gmail.com	Khách hàng	Hoạt động	6/11/2025	Khóa
5	Hoàng Việt Trung	com@hpu.com	Chủ nhà hàng	Hoạt động	3/11/2025	Khóa
6	Chủ Bún	bun@hpu.com	Chủ nhà hàng	Hoạt động	3/11/2025	Khóa
7	Chủ Hải Sản	haisan@hpu.com	Chủ nhà hàng	Hoạt động	3/11/2025	Khóa
8	Chủ Chay	chay@hpu.com	Chủ nhà hàng	Hoạt động	3/11/2025	Khóa
9	Chủ Mỹ	my@hpu.com	Chủ nhà hàng	Hoạt động	3/11/2025	Khóa
10	Chủ Ăn Việt	anvat@hpu.com	Chủ nhà hàng	Hoạt động	3/11/2025	Khóa
11	Chủ Lẩu	lau@hpu.com	Chủ nhà hàng	Hoạt động	3/11/2025	Khóa
4	Phạm Thanh Lam	owner1@example.com	Khách hàng	Hoạt động	2/11/2025	Khóa
3	Chu Nhà Hàng Tuong Lai	owner@example.com	Khách hàng	Hoạt động	2/11/2025	Khóa
2	Admin	admin@example.com	Admin	Hoạt động	2/11/2025	Khóa
1	User2	user1@gmail.com	Chủ nhà hàng	Hoạt động	2/11/2025	Khóa

Ảnh 4.19. Giao diện trang quản lý user

4.4.16. Giao diện trang kiểm duyệt thực đơn

ID	Tên Món ăn	Giá	Phân loại	Nhà hàng	Trạng thái	Thao tác
39	Há cảo tôm	65.000 đ		Vua Dimsum Thượng Hải	Đang hiển thị	Ẩn
40	Xiu mại hấp	55.000 đ		Vua Dimsum Thượng Hải	Đang hiển thị	Ẩn
41	Vịt quay Bắc Kinh (nửa con)	350.000 đ		Vua Dimsum Thượng Hải	Đang hiển thị	Ẩn
42	Cơm chiên Dương Châu	75.000 đ		Vua Dimsum Thượng Hải	Đang hiển thị	Ẩn
43	Sủi cảo chiên	60.000 đ		Vua Dimsum Thượng Hải	Đang hiển thị	Ẩn
44	Đậu hũ Tứ Xuyên	85.000 đ		Vua Dimsum Thượng Hải	Đang hiển thị	Ẩn
45	Gà Kung Pao	120.000 đ		Vua Dimsum Thượng Hải	Đang hiển thị	Ẩn
46	Canh chua cay	70.000 đ		Vua Dimsum Thượng Hải	Đang hiển thị	Ẩn
47	Trà Ô Long	25.000 đ		Vua Dimsum Thượng Hải	Đang hiển thị	Ẩn
48	Chè trôi nước (vùng đen)	35.000 đ	Món tráng miệng	Vua Dimsum Thượng Hải	Đang hiển thị	Ẩn
49	Ba chỉ bò nướng (Set)	159.000 đ		Seoul BBQ - Nướng Hàn Quốc	Đang hiển thị	Ẩn
50	Gà rán sốt cay	129.000 đ		Seoul BBQ - Nướng Hàn Quốc	Đang hiển thị	Ẩn
51	Cơm trộn Bibimbap	89.000 đ		Seoul BBQ - Nướng Hàn Quốc	Đang hiển thị	Ẩn
52	Canh Kim Chi	60.000 đ		Seoul BBQ - Nướng Hàn Quốc	Đang hiển thị	Ẩn
53	Tokbokki phô mai	79.000 đ		Seoul BBQ - Nướng Hàn Quốc	Đang hiển thị	Ẩn
54	Miến trộn Japchae	95.000 đ		Seoul BBQ - Nướng Hàn Quốc	Đang hiển thị	Ẩn
55	Kimbap truyền thống	50.000 đ		Seoul BBQ - Nướng Hàn Quốc	Đang hiển thị	Ẩn
56	Panchan (Thêm)	20.000 đ		Seoul BBQ - Nướng Hàn Quốc	Đang hiển thị	Ẩn
57	Rượu Soju	70.000 đ		Seoul BBQ - Nướng Hàn Quốc	Đang hiển thị	Ẩn
58	Kem Bingsu (Nhỏ)	65.000 đ	Món tráng miệng	Seoul BBQ - Nướng Hàn Quốc	Đang hiển thị	Ẩn

Ảnh 4.20. Giao diện trang kiểm duyệt thực đơn

CHƯƠNG V: KẾT LUẬN

5.1. Kết quả đạt được

Trong quá trình triển khai và phát triển dự án, em đã đạt được những kết quả quan trọng, bao gồm:

- Hoàn thiện tài liệu: Phân tích và đặc tả chi tiết các yêu cầu chức năng (Use Cases), thiết kế luồng người dùng và kiến trúc hệ thống.
- Xây dựng cơ sở dữ liệu: Thiết kế và triển khai thành công cơ sở dữ liệu bằng MariaDB (thông qua XAMPP), đảm bảo lưu trữ và quản lý dữ liệu cho các đối tượng chính: người dùng, nhà hàng, món ăn, và đơn hàng.
- Xây dựng Backend: Phát triển thành công các API phía máy chủ bằng Node.js và Express.js, xử lý logic nghiệp vụ cho tất cả các chức năng cốt lõi.
- Xây dựng Frontend: Tạo dựng giao diện người dùng bằng React.js cho cả luồng khách hàng, trang quản lý của chủ nhà hàng và trang quản trị của admin.
- Hoàn thiện các chức năng chính:
 - **Người dùng:** Đăng ký, đăng nhập, cập nhật thông tin, tìm kiếm nhà hàng, xem thực đơn, quản lý giỏ hàng, đặt hàng và xem lịch sử/hủy đơn hàng.
 - **Chủ nhà hàng (Owner):** Gửi đơn đăng ký nhà hàng, quản lý thực đơn (thêm, sửa, xóa món ăn), quản lý và cập nhật trạng thái đơn hàng.
 - **Quản trị viên (Admin):** Phê duyệt/từ chối nhà hàng, quản lý tài khoản người dùng (khóa/mở khóa), và kiểm duyệt món ăn.

5.2. Hạn chế

Bên cạnh những kết quả đạt được, dự án vẫn còn một số hạn chế cần được cải thiện:

- Chức năng thanh toán còn đơn giản: Hệ thống mới chỉ hỗ trợ phương thức thanh toán khi nhận hàng (COD), chưa tích hợp các cổng thanh toán trực tuyến.
- Thiếu tính năng tương tác: Chưa có hệ thống đánh giá, bình luận cho nhà hàng hoặc món ăn, làm giảm sự tương tác và thông tin tham khảo cho người dùng.

- Tìm kiếm cơ bản: Chức năng tìm kiếm mới chỉ hỗ trợ tìm theo tên nhà hàng, chưa có các bộ lọc và tìm kiếm nâng cao (ví dụ: theo địa chỉ, loại ẩm thực, món ăn).
- Giao diện người dùng: Giao diện đã đáp ứng đủ các chức năng nhưng có thể tiếp tục được cải tiến để trở nên chuyên nghiệp và thu hút hơn.

5.3. Hướng phát triển

Dựa trên nền tảng đã xây dựng, dự án có thể được mở rộng và phát triển trong tương lai với các hướng sau:

- **Tích hợp thanh toán trực tuyến:** Tích hợp các cổng thanh toán phổ biến như MoMo, ZaloPay, VNPAY để tăng sự tiện lợi cho khách hàng.
- **Xây dựng hệ thống đánh giá, bình luận:** Cho phép người dùng đánh giá nhà hàng và món ăn (chấm sao, viết bình luận) để tăng tính cộng đồng và độ tin cậy.
- **Nâng cấp tìm kiếm và gợi ý:** Phát triển bộ lọc nâng cao, tìm kiếm theo vị trí, và xây dựng một hệ thống gợi ý nhà hàng/món ăn dựa trên lịch sử đặt hàng của người dùng.
- **Theo dõi đơn hàng thời gian thực:** Tích hợp bản đồ để người dùng có thể theo dõi vị trí của đơn hàng trong quá trình giao.
- **Phát triển ứng dụng di động:** Xây dựng phiên bản ứng dụng cho Android và iOS để cải thiện trải nghiệm người dùng trên thiết bị di động.
- **Mở rộng chức năng cho Owner/Admin:** Bổ sung các biểu đồ và báo cáo thống kê chi tiết hơn về doanh thu, món ăn bán chạy, v.v.

TÀI LIỆU THAM KHẢO

- [1] Mozilla Developer Network (MDN). *Client–Server overview*. https://developer.mozilla.org/en-US/docs/Learn/Server-side/First_steps/Client-Server_overview
- [2] React Official Documentation – <https://react.dev>
- [3] Vite Official Documentation – <https://vitejs.dev>
- [4] React Router – Official Documentation: <https://reactrouter.com>
- [5] Nguyễn Thanh Hưng (2023). *Ứng dụng SCSS và CSS Modules trong phát triển giao diện React*, Trường Đại học Công nghệ Thông tin TP.HCM.
- [6] MDN Web Docs – *React Context API Overview*: <https://developer.mozilla.org/>
- [7] Node.js Official Website – <https://nodejs.org/en>
- [8] Express.js Official Website – <https://expressjs.com/>
- [9] Express.js Official Website – <https://expressjs.com/>
- [10] Phạm Quang Huy (2022). *Ứng dụng kiến trúc MVC trong phát triển web với Node.js và Express*, Trường Đại học FPT TP.HCM.
- [11] Trần Đức Tài (2023). *Sử dụng Middleware trong Express.js để bảo vệ Website API*, Đại học Công nghệ Thông tin - ĐHQG TP.HCM.
- [12] Nguyễn Văn Minh (2022). *Xây dựng và kết nối frontend-backend trong ứng dụng thương mại điện tử*, Trường Đại học Bách Khoa Hà Nội.
- [13] W3Schools – MySQL Tutorial:
<https://www.w3schools.com/mysql/>