

TRƯỜNG ĐẠI HỌC KIẾN TRÚC ĐÀ NẴNG
KHOA CÔNG NGHỆ THÔNG TIN



ĐỒ ÁN TỐT NGHIỆP
NGÀNH: CÔNG NGHỆ THÔNG TIN

ĐỀ TÀI:

**Xây dựng website đặt bàn, đặt món cho nhà hàng thực
dưỡng Hoa Ban sử dụng Vuejs, Spring Boot và
PostgreSQL**

Người hướng dẫn: **ThS. LÝ QUỲNH TRÂN**
Sinh viên thực hiện: **NGUYỄN QUANG MINH**

Mã số sinh viên: **2151220144**

Lớp: **21CT3**

Đà Nẵng,/2025

TRƯỜNG ĐẠI HỌC KIẾN TRÚC ĐÀ NẴNG
KHOA CÔNG NGHỆ THÔNG TIN



ĐỒ ÁN TỐT NGHIỆP

NGÀNH: CÔNG NGHỆ THÔNG TIN

ĐỀ TÀI:

Xây dựng website đặt bàn, đặt món cho nhà hàng thực
dưỡng Hoa Ban sử dụng Vuejs, Spring Boot và
PostgreSQL

Người hướng dẫn: ThS. LÝ QUỲNH TRÂN
Sinh viên thực hiện: NGUYỄN QUANG MINH
Mã số sinh viên: 2151220144
Lớp: 21CT3

Đà Nẵng,/2025

{Trang trắng này dùng để dán bản Nhận xét của người hướng dẫn, hoặc thay trang này bằng Nhận xét của người hướng dẫn}

{Trang trắng này dùng để dán bản Nhận xét của người phản biện, hoặc thay trang này bằng Nhận xét của người phản biện}

TÓM TẮT

Tên đề tài: **Xây dựng website đặt bàn, đặt món cho nhà hàng thực dưỡng Hoa Ban**
sử dụng Vue.js, Spring Boot và PostgreSQL

Sinh viên thực hiện: Nguyễn Quang Minh

Mã số SV: 2151220144 Lớp: 21CT3

Đề tài “**Xây dựng website đặt bàn, đặt món cho nhà hàng thực dưỡng Hoa Ban sử dụng Vue.js, Spring Boot và PostgreSQL**” được thực hiện với mục tiêu tạo ra một nền tảng trực tuyến hiện đại, thân thiện và đáp ứng tốt nhu cầu sử dụng thực tế của khách hàng. Hệ thống hướng đến việc kết hợp công nghệ thông tin tiên tiến với lĩnh vực dịch vụ ẩm thực, giúp tối ưu vận hành cho nhà hàng và nâng cao trải nghiệm người dùng.

Về công nghệ, Vue.js được sử dụng để xây dựng giao diện người dùng năng động với tốc độ phản hồi nhanh; Spring Boot đảm nhận phát triển backend với tính bảo mật cao và khả năng mở rộng mạnh mẽ; PostgreSQL đóng vai trò là hệ quản trị cơ sở dữ liệu đáng tin cậy, đảm bảo tính toàn vẹn và hiệu suất lưu trữ.

Hệ thống cung cấp đầy đủ các chức năng phục vụ hai nhóm đối tượng chính:

Phía khách hàng:

- Đăng ký, đăng nhập, quên mật khẩu, cập nhật thông tin và đổi mật khẩu.
 - Xem trang chủ, duyệt menu theo danh mục, xem combo, xem chi tiết món ăn và tìm kiếm nhanh.
 - Thêm món vào giỏ, cập nhật/xóa món, đặt món qua mã QR không cần đăng nhập, thanh toán tiền mặt hoặc VNPay, áp dụng voucher và xem kết quả đơn hàng.
 - Đặt bàn trước và theo dõi lịch trạng thái đơn đặt bàn.
 - Xem lịch sử, xem chi tiết và theo dõi trạng thái xử lý của từng đơn.
 - Xem danh sách voucher, tích điểm loyalty từ đơn hàng và sử dụng điểm thưởng.
 - Đánh giá nhà hàng sau khi sử dụng dịch vụ đặt món.
 - Chatbot: hỗ trợ tư vấn món.
 - Đặt món và xem thực đơn bằng mã QR tại bàn.

Phía quản trị:

- Quản lý danh mục món ăn: thêm, sửa và điều chỉnh trạng thái hoạt động.
- Quản lý món ăn với đầy đủ CRUD, phân trang, sắp xếp và tìm kiếm nâng cao.
- Quản lý combo: tạo combo khuyến mãi bao gồm nhiều món.
- Quản lý bàn và phân khu vực nhà hàng
- Theo dõi và cập nhật trạng thái đơn hàng.
- Xử lý đơn đặt bàn của khách hàng và gán bàn
- Quản lý người dùng và tài khoản khách hàng.
- Tạo và quản lý voucher giảm giá theo phần trăm hoặc số tiền.
- Quản lý đánh giá của khách hàng và thống kê chi tiết.

Các tính năng nổi bật

- Giao diện responsive tương thích mọi thiết bị: máy tính, máy tính bảng và điện thoại.
- Bảo mật với JWT authentication và phân quyền Customer – Admin.
- Thông báo realtime bằng toast notifications.
- Trải nghiệm UX/UI hiện đại với hiệu ứng mượt mà.

Dự án đã xây dựng thành công một website đặt bàn – đặt món có giao diện đẹp mắt, đầy đủ chức năng và hiệu suất ổn định. Kết quả đạt được không chỉ đáp ứng mục tiêu đề ra mà còn giúp sinh viên củng cố kiến thức và kỹ năng lập trình web full-stack theo kiến trúc RESTful API.

LỜI NÓI ĐẦU

Trong thời đại công nghệ số phát triển mạnh mẽ, các hoạt động kinh doanh trong lĩnh vực dịch vụ – đặc biệt là ngành ẩm thực – đang từng bước chuyển mình theo hướng tự động hóa và tối ưu trải nghiệm khách hàng. Người dùng ngày nay có xu hướng ưu tiên các nền tảng trực tuyến giúp đặt bàn, đặt món nhanh chóng, tiện lợi và minh bạch. Nhận thấy nhu cầu thực tế đó, đề tài “Xây dựng website đặt bàn, đặt món cho nhà hàng thực dưỡng Hoa Ban sử dụng Vue.js, Spring Boot và PostgreSQL” được lựa chọn nhằm mang đến một giải pháp hiện đại, hiệu quả và phù hợp với định hướng phát triển của nhà hàng.

Với đặc thù phục vụ ẩm thực theo phong cách thực dưỡng, Nhà hàng Hoa Ban không chỉ chú trọng chất lượng món ăn mà còn hướng đến việc nâng cao trải nghiệm khách hàng trong toàn bộ quy trình – từ xem thực đơn, đặt món, đặt bàn đến thanh toán và đánh giá dịch vụ. Việc ứng dụng công nghệ thông tin, cụ thể là nền tảng web, giúp tối ưu quy trình vận hành, giảm sai sót, rút ngắn thời gian phục vụ và tăng mức độ hài lòng của khách hàng.

Dự án được xây dựng dựa trên ba công nghệ chủ đạo: Vue.js cho giao diện người dùng trực quan và phản hồi nhanh; Spring Boot cho backend mạnh mẽ, bảo mật và dễ mở rộng; PostgreSQL cho hệ quản trị cơ sở dữ liệu ổn định, đảm bảo tính toàn vẹn và hiệu suất cao. Việc kết hợp các công nghệ này không chỉ giúp hệ thống vận hành hiệu quả mà còn mang lại nhiều lợi thế trong phát triển, bảo trì và nâng cấp về sau.

Thông qua quá trình thực hiện đề tài, em đã có cơ hội vận dụng các kiến thức được học vào một sản phẩm thực tiễn, từ phân tích yêu cầu, thiết kế hệ thống, xây dựng API, thiết kế giao diện cho đến kiểm thử và triển khai. Hệ thống hoàn thiện với các chức năng như quản lý thực đơn, giờ hàng, đặt món trực tuyến, thanh toán VNPay QR, quản lý đơn hàng, tích điểm và voucher, đánh giá nhà hàng, cùng với trang quản trị đầy đủ đã góp phần mang lại một trải nghiệm tiện lợi và chuyên nghiệp cho người dùng.

Em xin gửi lời cảm ơn chân thành đến các thầy cô Trường Đại học Kiến trúc Đà Nẵng đã tận tình giảng dạy và trang bị cho em nền tảng kiến thức vững chắc trong suốt thời gian học tập. Em đặc biệt biết ơn cô Lý Quỳnh Trân, người đã định hướng và hỗ trợ em trong quá trình thực hiện đồ án, luôn đồng hành, góp ý và tạo điều kiện thuận lợi để em hoàn thành đề tài. Em cũng xin trân trọng cảm ơn gia đình và bạn bè đã luôn động viên, khích lệ và là chỗ dựa tinh thần quan trọng giúp em nỗ lực vượt qua khó khăn, hoàn thành tốt quá trình thực hiện dự án này.

CAM ĐOAN

Tôi, Nguyễn Quang Minh, sinh viên lớp 21CT3, mã số sinh viên 2151220144, sinh viên khóa 21, Khoa Công nghệ Thông tin, Trường Đại học Kiến trúc Đà Nẵng, xin cam đoan rằng đê tài:

“Xây dựng website đặt bàn, đặt món cho nhà hàng thực dưỡng Hoa Ban sử dụng Vue.js, Spring Boot và PostgreSQL”

là kết quả nghiên cứu và thực hiện của riêng tôi dưới sự hướng dẫn của cô Lý Quỳnh Trần. Các số liệu, dữ liệu, hình ảnh và thông tin được sử dụng trong đồ án đều có nguồn gốc rõ ràng, chính xác và trung thực. Tôi hoàn toàn chịu trách nhiệm trước Nhà trường về tính trung thực của toàn bộ nội dung trong đồ án này.

Sinh viên thực hiện

MỤC LỤC

TÓM TẮT	I
LỜI NÓI ĐẦU.....	III
CAM ĐOAN.....	IV
MỤC LỤC	V
DANH MỤC CÁC BẢNG.....	VIII
DANH MỤC CÁC HÌNH	IX
DANH MỤC KÝ HIỆU, CHỮ VIẾT TẮT	XI
MỞ ĐẦU.....	XIV
CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI.....	1
1.1. Tên đề tài	1
1.2. Giới thiệu đề tài	1
1.2.1. Bối cảnh và lý do chọn đề tài	1
1.2.2. Mục tiêu và phạm vi nghiên cứu	2
1.2.3. Đối tượng phục vụ	2
1.2.4. Giới thiệu về website	3
CHƯƠNG 2. CƠ SỞ LÝ THUYẾT VÀ CÔNG CỤ PHÁT TRIỂN.....	4
2.1. Cơ sở lý thuyết	4
2.1.1. Ngôn ngữ HTML	4
2.1.2. Ngôn ngữ CSS	5
2.1.3. Ngôn ngữ JavaScript	5
2.1.4. Ngôn ngữ lập trình Java	6
2.1.5. Cơ sở dữ liệu PostgreSQL	7
2.1.6. Spring Boot và Vue.js Framework	8
2.1.7. Mô hình ứng dụng SPA	14
2.1.8. Kiến trúc giao tiếp: RESTful API	15
2.1.9. JWT Authentication.....	16

2.2. Công cụ sử dụng	17
2.2.1. Visual Studio Code.....	17
2.2.2. IntelliJ IDEA	18
2.2.3. PgAdmin 4.....	20
2.2.4. GeminiAI	21
2.2.5. VNPay	22
2.3. Các bước cài đặt công cụ	22
2.3.1. Cài đặt Visual Studio Code trên Windows.....	22
2.3.2. Cài đặt IntelliJ IDEA	25
2.3.3. Cài đặt pgAdmin 4.....	26
2.3.4. Cài đặt Node.js và NPM	28
2.3.5. Cài đặt Java JDK	28
CHƯƠNG 3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG	29
3.1. Mô tả bài toán	29
3.2. Xác định Actor	30
3.2.1. Khách hàng.....	31
3.2.2. Admin	31
3.3. Xác định Usecase	32
3.4. Phân tích Usecase	34
3.4.1. Biểu đồ Usecase	34
3.4.2. Đặc tả usecase.....	37
3.5. Biểu đồ hoạt động	59
3.5.1. Biểu đồ hoạt động chức năng đặt món	59
3.5.2. Biểu đồ hoạt động chức năng đặt bàn	61
3.6. Biểu đồ tuần tự.....	63
3.6.1. Biểu đồ tuần tự đặt món	63
3.6.2. Biểu đồ tuần tự đặt bàn.....	67
3.7. Thiết kế cơ sở dữ liệu	72
3.7.1. Các bảng trong cơ sở dữ liệu	72

3.7.2. Biểu đồ quan hệ của các bảng	78
CHƯƠNG 4. TRIỂN KHAI HỆ THỐNG.....	79
4.1. Cấu hình triển khai	79
4.1.1. Backend – Spring Boot.....	79
4.1.2. Frontend – Vue.js	79
4.1.3. Cơ sở dữ liệu – PostgreSQL.....	79
4.2. Cấu trúc thư mục.....	80
4.2.1. Backend chính	80
4.2.2. Backend chatbot	81
4.2.3. Frontend.....	82
4.3. Kết quả đạt được	83
4.3.1. Giao diện người dùng	83
4.3.2. Giao diện người quản trị.....	87
KẾT LUẬN	93
TÀI LIỆU THAM KHẢO.....	95

DANH MỤC CÁC BẢNG

Bảng 3.1 Xác định Actor	30
Bảng 3.2 Xác định Usecase	32
Bảng 3.3 Đặc tả usecase đăng ký	37
Bảng 3.4 Đặc tả Usecase đăng nhập.....	39
Bảng 3.5 Đặc tả usecase đặt món/ đặt bàn	40
Bảng 3.6 Đặc tả usecase tích điểm và đổi voucher	42
Bảng 3.7 Đặc tả usecase đánh giá	43
Bảng 3.8 Đặc tả usecase lịch sử đơn hàng	44
Bảng 3.9 Đặc tả usecase Chatbot tư vấn món	46
Bảng 3.10 Đặc tả usecase quét mã QR đặt món.....	47
Bảng 3.11 Đặc tả usecase quản lý bàn và khu vực.....	49
Bảng 3.12 Đặc tả usecase quản lý món	50
Bảng 3.13 Đặc tả usecase quản lý combo	51
Bảng 3.14 Đặc tả usecase quản lý voucher	53
Bảng 3.15 Đặc tả usecase quản lý đánh giá.....	54
Bảng 3.16 Đặc tả usecase quản lý đặt món/ đặt bàn	57

DANH MỤC CÁC HÌNH

Hình 2-1: Logo PostgreSQL.....	8
Hình 2-2: Framework thư viện với các tài nguyên không thể thiếu đối với lập trình viên	9
Hình 2-3: Mô hình MVC	9
Hình 2-4: Logo Spring Boot.....	10
Hình 2-5: Logo Vue.js	12
Hình 2-6: Tổng quan về SPA	14
Hình 2-7: Sơ đồ RESTful API.....	15
Hình 2-8: Logo JWT	16
Hình 2-9: Logo Visual Studio Code	18
Hình 2-10: Logo intellij	19
Hình 2-11: Logo pgAdmin	20
Hình 2-12: Logo Gemini AI	21
Hình 2-13: Logo VNPay	22
Hình 3-1: Sơ đồ Usecase khách hàng.....	34
Hình 3-2: Sơ đồ Usecase Admin	35
Hình 3-3: Sơ đồ Usecase tổng quát	36
Hình 3-4: Đặc tả Usecase đăng ký	37
Hình 3-5: Đặc tả Usecase đăng nhập.....	38
Hình 3-6: Đặc tả Usecase đặt món/ đặt bàn	40
Hình 3-7: Đặc tả Usecase tích điểm và đổi voucher	41
Hình 3-8: Đặc tả Usecase đánh giá.....	43
Hình 3-9: Đặc tả Usecase lịch sử đơn hàng.....	44
Hình 3-10: Đặc tả Usecase Chatbot ai tư vấn món	45
Hình 3-11: Đặc tả Usecase quét mã QR đặt món.....	47
Hình 3-12: Đặc tả Usecase quản lý bàn và khu vực.....	48
Hình 3-13: Đặc tả Usecase quản lý món ăn	50
Hình 3-14: Đặc tả Usecase quản lý combo	51
Hình 3-15: Đặc tả Usecase quản lý voucher	53
Hình 3-16: Đặc tả Usecase quản lý đánh giá.....	54
Hình 3-17: Đặc tả Usecase quản lý đơn đặt món và đặt bàn.....	56
Hình 3-18: Biểu đồ hoạt động đặt món	59
Hình 3-19: Biểu đồ hoạt động đặt bàn.....	61
Hình 3-20: Biểu đồ tuần tự đặt món	64

Hình 3-21: Biểu đồ tuần tự đặt bàn	68
Hình 3-22: Bảng users	72
Hình 3-23: Bảng dishes	73
Hình 3-24: Bảng combos	74
Hình 3-25: Bảng orders	75
Hình 3-26: Bảng vouchers	76
Hình 3-27: Bảng reservations	76
Hình 3-28: Bảng restaurant_tables	77
Hình 3-29: Bảng reviews	77
Hình 3-30: Sơ đồ quan hệ ERD	78
Hình 4-1: Cấu trúc thư mục backend chính	80
Hình 4-2: Cấu trúc thư mục backend chatbot	81
Hình 4-3: Cấu trúc thư mục frontend	82
Hình 4-4: Giao diện trang chủ	83
Hình 4-5: Giao diện thực đơn	83
Hình 4-6: Giao diện combo món	84
Hình 4-7: Giao diện đặt bàn	84
Hình 4-8: Giao diện giỏ hàng	85
Hình 4-9: Giao diện thanh toán	85
Hình 4-10: Giao diện chatbot	86
Hình 4-11: Giao diện voucher	86
Hình 4-12: Giao diện đánh giá	87
Hình 4-13: Giao diện trang quản trị	87
Hình 4-14: Giao diện trang quản lý món ăn	88
Hình 4-15: Giao diện quản lý combo	88
Hình 4-16: Giao diện quản lý danh mục	88
Hình 4-17: Giao diện quản lý khu vực	89
Hình 4-18: Giao diện quản lý bàn	89
Hình 4-19: Giao diện quản lý voucher	90
Hình 4-20: Giao diện quản lý đơn hàng	90
Hình 4-21: Giao diện quản lý đặt bàn	91
Hình 4-22: Giao diện gán bàn cho đơn đặt bàn	91
Hình 4-23: Email thông báo thanh toán thành công	92
Hình 4-24: Email thông báo đặt bàn thành công	92

DANH MỤC KÝ HIỆU, CHỮ VIẾT TẮT

STT	Ký hiệu viết tắt	Chữ viết đầy đủ	Ghi chú
1	HTML	HyperText Markup Language	Ngôn ngữ đánh dấu tạo cấu trúc trang web
2	CSS	Cascading Style Sheets	Ngôn ngữ định kiểu cho giao diện web
3	JS	JavaScript	Ngôn ngữ lập trình phía client
4	SPA	Single Page Application	Ứng dụng một trang (Vue.js)
5	API	Application Programming Interface	Giao diện lập trình ứng dụng
6	REST	Representational State Transfer	Kiến trúc giao tiếp API
7	RESTful API	REST Application Programming Interface	Chuẩn API trong dự án
8	JSON	JavaScript Object Notation	Định dạng trao đổi dữ liệu
9	JWT	JSON Web Token	Token xác thực & phân quyền
10	MVC	Model – View – Controller	Mô hình kiến trúc phần mềm
11	MVVM	Model – View – ViewModel	Mô hình giao diện (Vue.js)
12	CRUD	Create – Read – Update – Delete	Tác vụ quản lý dữ liệu
13	ORM	Object–Relational Mapping	Spring JPA/Hibernate
14	SQL	Structured Query Language	Ngôn ngữ truy vấn dữ liệu

15	DB	Database	Cơ sở dữ liệu
16	RDBMS	Relational Database Management System	Hệ quản trị CSDL quan hệ
17	QR	Quick Response	Mã QR đặt món tại bàn
19	AI	Artificial Intelligence	Trí tuệ nhân tạo
20	NLP	Natural Language Processing	Xử lý ngôn ngữ tự nhiên
21	UI	User Interface	Giao diện người dùng
22	UX	User Experience	Trải nghiệm người dùng
23	VS Code	Visual Studio Code	Công cụ lập trình frontend
24	IDE	Integrated Development Environment	Môi trường phát triển tích hợp
26	DBMS	Database Management System	Hệ thống quản trị CSDL
27	URL	Uniform Resource Locator	Đường dẫn tài nguyên web
28	HTTP	HyperText Transfer Protocol	Giao thức truyền tải web
29	HTTPS	HyperText Transfer Protocol Secure	Giao thức web bảo mật
30	JDK	Java Development Kit	Bộ công cụ lập trình Java
31	JVM	Java Virtual Machine	Máy ảo Java
32	JRE	Java Runtime Environment	Môi trường chạy Java
33	DTO	Data Transfer Object	Đối tượng truyền dữ liệu
34	DAO	Data Access Object	Lớp truy cập dữ liệu
35	QR Pay	QR Payment	Thanh toán bằng mã QR
36	VNPay	Vietnam Payment	Cổng thanh toán QR

37	OTP	One Time Password	Mật khẩu xác thực một lần
38	SSL	Secure Sockets Layer	Bảo mật truyền dữ liệu
40	CPU	Central Processing Unit	Bộ xử lý trung tâm
41	RAM	Random Access Memory	Bộ nhớ truy cập ngẫu nhiên
42	CDN	Content Delivery Network	Mạng phân phối nội dung

MỞ ĐẦU

1. Mục đích thực hiện đề tài

Trong bối cảnh công nghệ số phát triển nhanh chóng, việc ứng dụng các nền tảng trực tuyến vào lĩnh vực dịch vụ ẩm thực đã trở thành xu hướng tất yếu. Các nhà hàng hiện đại không chỉ tập trung vào chất lượng món ăn mà còn chú trọng tối ưu trải nghiệm khách hàng, từ việc xem thực đơn, đặt bàn, đặt món cho đến thanh toán và đánh giá dịch vụ. Những hạn chế của phương thức quản lý truyền thống như thao tác thủ công, mất thời gian, dễ nhầm lẫn và khó mở rộng quy mô khiến nhu cầu xây dựng hệ thống đặt bàn – đặt món trực tuyến trở nên cấp thiết.

Xuất phát từ nhu cầu đó, đề tài “Xây dựng website đặt bàn, đặt món cho nhà hàng thực dưỡng Hoa Ban sử dụng Vue.js, Spring Boot và PostgreSQL” được thực hiện nhằm phát triển một nền tảng hiện đại, tiện lợi và an toàn, giúp khách hàng dễ dàng lựa chọn món ăn, đặt bàn và đặt món. Đồng thời, hệ thống hỗ trợ nhà hàng trong việc quản lý thực đơn, đơn hàng, khách hàng, bàn ăn, voucher và đánh giá, góp phần nâng cao hiệu quả vận hành, giảm tải công việc cho nhân viên và thúc đẩy quá trình chuyển đổi số trong hoạt động kinh doanh của nhà hàng.

2. Mục tiêu đề tài

Mục tiêu chính của đề tài là xây dựng một website hỗ trợ đặt bàn và đặt món trực tuyến cho nhà hàng thực dưỡng Hoa Ban, đáp ứng nhu cầu sử dụng trong bối cảnh ngành dịch vụ ẩm thực đang chuyển đổi mạnh mẽ sang mô hình số hóa. Cụ thể, đề tài hướng đến các mục tiêu sau:

- Xây dựng hệ thống quản lý thực đơn và danh mục món ăn đầy đủ, linh hoạt**

Phát triển website cho phép quản lý các món ăn và combo kèm theo thông tin chi tiết như tên món, mô tả, giá, hình ảnh, danh mục, tình trạng hoạt động. Hệ thống hỗ trợ phân loại món theo danh mục, tìm kiếm và sắp xếp món ăn theo nhiều tiêu chí để nâng cao trải nghiệm người dùng.

- Phát triển chức năng đặt món, đặt bàn và thanh toán trực tuyến**

Cho phép khách hàng xem thực đơn, thêm món vào giỏ hàng, điều chỉnh số lượng, ghi chú yêu cầu, chọn bàn hoặc quét mã QR để đặt món tại bàn. Hệ thống hỗ trợ thanh toán linh hoạt bằng tiền mặt hoặc VNPay QR Code, đảm bảo an toàn, nhanh chóng và tiện lợi.

- **Xây dựng hệ thống quản lý đơn hàng, bàn ăn và thông tin khách hàng**

Hệ thống cho phép quản trị viên xem chi tiết đơn hàng, cập nhật trạng thái đơn hàng và quản lý thông tin khách hàng, lịch sử đặt món. Đồng thời hỗ trợ quản lý giờ hàng, lịch sử giao dịch và thông tin hồ sơ của người dùng.

- **Tối ưu trải nghiệm người dùng với giao diện trực quan, thân thiện**

Website được thiết kế hiện đại, bố cục rõ ràng, dễ thao tác, tương thích tốt trên cả máy tính, máy tính bảng và điện thoại. Các hiệu ứng mượt mà và thông báo thời gian thực giúp cải thiện trải nghiệm sử dụng.

- **Hỗ trợ mở rộng và tích hợp các tính năng nâng cao trong tương lai**

Hệ thống được xây dựng với kiến trúc mở, dễ nâng cấp, cho phép tích hợp thêm các chức năng như chatbot tư vấn món ăn, AI gợi ý món theo khẩu vị, hệ thống loyalty nâng cao, thống kê doanh thu, phân tích hành vi khách hàng... nhằm nâng cao chất lượng dịch vụ và hiệu quả vận hành.

3. Đối tượng, phạm vi nghiên cứu

Đối tượng nghiên cứu

Đề tài tập trung nghiên cứu và xây dựng website đặt bàn – đặt món cho nhà hàng thực dưỡng Hoa Ban. Các đối tượng nghiên cứu chính bao gồm:

- **Công nghệ phát triển hệ thống**

Spring Boot (Backend)

- Nghiên cứu kiến trúc RESTful API.
- Tìm hiểu các thành phần của Spring Boot như Controller, Service, Repository, DTO.
- Sử dụng Spring Security và JWT cho xác thực và phân quyền.
- Ứng dụng JPA/Hibernate để thao tác dữ liệu hiệu quả.

Vue.js (Frontend)

- Nghiên cứu cách thức xây dựng giao diện SPA (Single Page Application).
- Tìm hiểu cơ chế component, routing, state management (Pinia/Vuex).
- Xây dựng giao diện hiện đại, thân thiện và responsive.

- **Công nghệ hỗ trợ xây dựng website**

- HTML / CSS / JavaScript

Dùng để xây dựng giao diện người dùng, xử lý tương tác và tối ưu trải nghiệm.

- SCSS / Tailwind

Hỗ trợ thiết kế giao diện nhanh, đảm bảo đồng bộ bố cục và tối ưu hiển thị trên nhiều thiết bị.

- PostgreSQL

Hệ quản trị cơ sở dữ liệu sử dụng để lưu trữ các thông tin: món ăn, combo, người dùng, giỏ hàng, đơn hàng, voucher, lịch sử đặt món.

- VNPay API

Tích hợp thanh toán qua QR Code giúp khách hàng thanh toán nhanh chóng và an toàn.

- **Công cụ và thư viện bổ sung**

- Gradle

Quản lý dependencies và build project Spring Boot.

Tệp build.gradle và công cụ ./gradlew giúp tự động hóa quá trình build, chạy và đóng gói ứng dụng.

- Postman / Swagger

Kiểm thử API trong quá trình phát triển backend.

- Chatbot tư vấn món (Microservice AI riêng biệt)

Hệ thống chatbot được triển khai dưới dạng một microservice độc lập, tách biệt hoàn toàn với backend chính của website. Microservice này được xây dựng bằng Spring Boot kết hợp Spring AI, sử dụng mô hình Gemini để xử lý ngôn ngữ tự nhiên, tạo phản hồi thông minh và phù hợp với ngữ cảnh đặt món của người dùng.

Microservice AI đảm nhiệm các chức năng:

- Gợi ý món ăn hoặc combo dựa trên sở thích và nhu cầu của khách.
- Trả lời câu hỏi về món ăn, giá cả, ...
- Tư vấn hỗ trợ khách trong quá trình đặt món trực tuyến.
- Hỗ trợ khách hàng đặt bàn.

- Giao tiếp với backend chính qua REST API, đảm bảo tính độc lập và dễ mở rộng.

Việc tách chatbot thành một AI microservice giúp:

- Giảm tải cho backend chính.
- Dễ dàng nâng cấp, thay đổi mô hình AI mà không ảnh hưởng hệ thống.
- Tăng tính linh hoạt, khả năng mở rộng và dễ bảo trì.
- Phù hợp với kiến trúc microservice hiện đại.
- QR Code cho từng bàn ăn
 - Khách có thể quét mã QR để xem thực đơn và đặt món trực tiếp tại bàn.
 - Tăng sự tiện lợi, giảm thời gian phục vụ.

Phạm vi nghiên cứu

Phạm vi nghiên cứu của đề tài tập trung vào việc xây dựng website hỗ trợ đặt bàn và đặt món trực tuyến, kết hợp quản lý món ăn, đơn hàng và người dùng cho nhà hàng thực dưỡng Hoa Ban. Cụ thể bao gồm:

1. Quản lý thực đơn và danh mục món ăn:

- Hệ thống cho phép quản lý danh mục món ăn, gồm các chức năng: thêm, sửa, xóa và cập nhật trạng thái hoạt động.
- Mỗi món ăn và combo đều có thông tin chi tiết như: tên món, mô tả, hình ảnh, giá, danh mục và các thuộc tính kèm theo.
- Hỗ trợ phân loại món theo danh mục, tìm kiếm và sắp xếp theo giá hoặc mức phổ biến.

2. Quản lý giỏ hàng, đặt món và thanh toán:

- Phát triển chức năng đặt món và quản lý giỏ hàng cho phép người dùng thêm món, điều chỉnh số lượng, ghi chú yêu cầu đặc biệt trước khi thanh toán.
- Hỗ trợ khách hàng đặt món trực tiếp tại bàn thông qua quét mã QR.
- Tích hợp phương thức thanh toán trực tuyến bằng VNPay QR Code hoặc thanh toán tiền mặt tại quầy.
- Hệ thống cho phép khách hàng xem lại lịch sử đặt món và chi tiết đơn hàng.

3. Quản lý người dùng và đơn hàng:

- Hỗ trợ người dùng đăng ký tài khoản, đăng nhập, cập nhật thông tin cá nhân và theo dõi lịch sử.
- Quản trị viên có thể xem chi tiết đơn hàng, cập nhật trạng thái đơn (Mở, Đã thanh toán, Đã hủy) và xử lý các yêu cầu phát sinh.
- Quản lý thông tin khách hàng, đơn hàng và hoạt động sử dụng voucher.

4. Giao diện người dùng

- Xây dựng giao diện trực quan, dễ sử dụng và tương thích với nhiều thiết bị (desktop, tablet, mobile).
- Áp dụng nguyên tắc UI/UX để tối ưu trải nghiệm người dùng, gồm hiệu ứng mượt mà, toast thông báo realtime và giao diện bố cục hợp lý.
- Sử dụng Vue.js làm nền tảng phát triển SPA (Single Page Application).

5. Bảo mật hệ thống

- Sử dụng JWT Authentication để đảm bảo an toàn thông tin người dùng và phân quyền truy cập.
- Đảm bảo an toàn dữ liệu khách hàng trong quá trình đăng nhập, đặt món và thanh toán.
- Áp dụng các biện pháp bảo mật API backend bằng Spring Security.

4. Phương pháp nghiên cứu

Phương pháp nghiên cứu được áp dụng trong quá trình thực hiện đề tài bao gồm:

Nghiên cứu tài liệu và phân tích yêu cầu hệ thống:

Tìm hiểu các tài liệu về mô hình đặt món – đặt bàn trực tuyến, nghiên cứu nhu cầu thực tế của nhà hàng Hoa Ban và phân tích yêu cầu chức năng/phi chức năng của hệ thống. Tham khảo các tài liệu về Vue.js, Spring Boot, PostgreSQL, chuẩn RESTful API và các mô hình bảo mật JWT.

Thiết kế hệ thống bằng các mô hình UML:

Xây dựng sơ đồ Use Case, Activity, Sequence và ERD để mô tả toàn bộ hoạt động của hệ thống. Thiết kế kiến trúc backend – frontend, thiết kế cơ sở dữ liệu và cấu trúc API rõ ràng, đảm bảo dễ phát triển và dễ mở rộng.

Tiến hành kiểm thử:

Tiến hành kiểm thử chức năng, kiểm thử giao diện, kiểm thử API và kiểm thử trải nghiệm người dùng (UX). Đánh giá hiệu năng, độ ổn định và mức độ thân thiện của hệ thống. Cập nhật và chỉnh sửa các lỗi phát sinh để đảm bảo chất lượng sản phẩm.

Lập trình và triển khai theo kiến trúc RESTful và SPA:

- Backend được phát triển bằng Spring Boot theo mô hình RESTful API.
- Frontend được xây dựng bằng Vue.js theo mô hình SPA (Single Page Application).
- Ứng dụng Spring Security – JWT để xác thực và phân quyền.
- Tích hợp VNPay QR Code phục vụ thanh toán trực tuyến.

Bố cục của đồ án

Đồ án tốt nghiệp được bố cục thành các phần chính như sau:

Chương 1: Giới thiệu đề tài

Chương 2: Cơ sở lý thuyết và công nghệ sử dụng

Chương 3: Phân tích và thiết kế hệ thống

Chương 4: Triển khai hệ thống và kết quả đạt được

Kết luận và hướng phát triển

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

1.1. Tên đề tài

Xây dựng website đặt bàn, đặt món cho nhà hàng thực dưỡng Hoa Ban sử dụng Vue.js, Spring Boot và PostgreSQL.

1.2. Giới thiệu đề tài

1.2.1. Bối cảnh và lý do chọn đề tài

Trong bối cảnh cuộc cách mạng công nghiệp 4.0 diễn ra mạnh mẽ, các doanh nghiệp trong lĩnh vực dịch vụ, đặc biệt là ngành ẩm thực, đang từng bước chuyển đổi từ mô hình quản lý truyền thống sang mô hình số hóa. Khách hàng ngày nay ưu tiên sự tiện lợi, nhanh chóng và minh bạch trong quá trình đặt bàn, gọi món và thanh toán. Những hệ thống quản lý thủ công như ghi chép bằng giấy, gọi món trực tiếp hoặc xử lý đơn bằng phương pháp truyền thống thường dẫn đến sai sót, mất nhiều thời gian và gây khó khăn trong quản lý khi lượng khách tăng cao.

Nhà hàng thực dưỡng Hoa Ban là mô hình ẩm thực hướng đến phong cách sống lành mạnh, chú trọng chất lượng và trải nghiệm của khách hàng. Tuy nhiên, việc chưa có một nền tảng trực tuyến hỗ trợ khách hàng xem thực đơn, đặt bàn, đặt món và thanh toán khiến quy trình phục vụ chưa được tối ưu, gây hạn chế trong việc tiếp cận khách hàng mới và nâng cao hiệu quả vận hành.

Xuất phát từ nhu cầu thực tế đó, đề tài “Xây dựng website đặt bàn, đặt món cho nhà hàng thực dưỡng Hoa Ban sử dụng Vue.js, Spring Boot và PostgreSQL” được lựa chọn với mong muốn xây dựng một hệ thống hiện đại, thân thiện và phù hợp với xu hướng chuyển đổi số trong ngành F&B. Việc ứng dụng công nghệ vào quy trình hoạt động của nhà hàng giúp:

- Tăng tính chuyên nghiệp trong phục vụ;
- Giảm tải khói lượng công việc cho nhân viên;
- Tránh sai sót khi ghi nhận đơn hàng;
- Nâng cao trải nghiệm khách hàng;
- Tạo lợi thế cạnh tranh so với mô hình truyền thống.

Sự kết hợp giữa nhu cầu thị trường, xu hướng tiêu dùng hiện đại và khả năng triển khai công nghệ thông qua Vue.js – Spring Boot – PostgreSQL chính là lý do em lựa chọn đề tài này.

1.2.2. Mục tiêu và phạm vi nghiên cứu

1.2.2.1. Mục tiêu

Xây dựng website đặt bàn và đặt món trực tuyến, cung cấp đầy đủ các tính năng hỗ trợ khách hàng trong việc xem thực đơn, lựa chọn món ăn, đặt món, đặt bàn và thanh toán. Hệ thống bao gồm các chức năng như quản lý thực đơn, quản lý giờ hàng, đặt món qua mã QR tại bàn, thanh toán trực tuyến bằng VNPay, quản lý đơn hàng, quản lý người dùng và tích hợp chatbot tư vấn món nhằm nâng cao trải nghiệm khách hàng.

1.2.2.2. Phạm vi

Dự án tập trung phát triển một nền tảng đặt bàn và đặt món trực tuyến với các chức năng chính sau:

- Quản lý danh mục món ăn và thông tin chi tiết món như giá, mô tả, hình ảnh, danh mục, trạng thái hoạt động, cùng các combo món kèm ưu đãi.
- Hỗ trợ đặt món và thanh toán, bao gồm đặt món trực tiếp trên website hoặc đặt món tại bàn thông qua quét mã QR, và thanh toán trực tuyến qua VNPay QR Code hoặc thanh toán tiền mặt.
- Quản lý đơn hàng và lịch sử đặt món, cho phép theo dõi trạng thái đơn hàng, xem lại chi tiết món đã đặt và kiểm soát quá trình xử lý đơn từ phía quản trị viên.
- Tích hợp Chatbot sử dụng GeminiAI để tư vấn món, gợi ý món ăn phù hợp với sở thích người dùng và hỗ trợ giải đáp thắc mắc trong quá trình đặt món.

1.2.3. Đối tượng phục vụ

Website hướng đến các đối tượng có nhu cầu xem thực đơn, đặt bàn và đặt món tại nhà hàng, cụ thể gồm:

- Khách hàng cá nhân muốn xem thực đơn, lựa chọn món ăn và đặt món trực tuyến một cách nhanh chóng, tiện lợi.
- Nhóm bạn hoặc gia đình có nhu cầu đặt bàn trước để đảm bảo chỗ ngồi khi đến nhà hàng, đặc biệt vào giờ cao điểm.
- Khách hàng dùng bữa tại chỗ muốn đặt món trực tiếp tại bàn thông qua quét mã QR để rút ngắn thời gian chờ và tối ưu trải nghiệm.
- Khách hàng thường xuyên muốn tích lũy điểm, sử dụng voucher và theo dõi lịch sử đặt món của mình.
- Quản trị viên và nhân viên nhà hàng có nhu cầu quản lý món ăn, combo, đơn hàng, thông tin khách hàng, voucher và đánh giá.

1.2.4. Giới thiệu về website

Website đặt bàn – đặt món được thiết kế nhằm mang lại một nền tảng hiện đại, hỗ trợ khách hàng xem thực đơn, lựa chọn món ăn và đặt món nhanh chóng, thuận tiện và thân thiện với người dùng. Sử dụng Vue.js cho frontend và Spring Boot cho backend, hệ thống đảm bảo tốc độ xử lý tốt, độ ổn định cao, tính bảo mật mạnh mẽ và khả năng mở rộng trong tương lai.

Các tính năng chính của hệ thống bao gồm:

- Quản lý thực đơn và món ăn:

Cho phép thêm, sửa, xóa và cập nhật thông tin món ăn và combo, bao gồm: giá, mô tả, hình ảnh, danh mục, tình trạng hoạt động và các thông tin kèm theo.

- Đặt món trực tuyến:

Hỗ trợ khách hàng xem thực đơn, chọn món, điều chỉnh số lượng, thêm ghi chú và gửi yêu cầu đặt món. Khách hàng có thể đặt món trực tiếp tại bàn thông qua quét mã QR.

- Phương thức thanh toán:

Tích hợp thanh toán trực tuyến qua VNPay QR Code hoặc thanh toán tiền mặt khi nhận món, giúp khách hàng linh hoạt và thuận tiện trong quá trình thanh toán.

- Quản lý đơn hàng:

Cho phép người dùng xem lại lịch sử đặt món, theo dõi trạng thái đơn hàng (Mở, Đã thanh toán, Đã hủy), đồng thời giúp quản trị viên xử lý và cập nhật đơn hàng theo thời gian thực.

- Tìm kiếm và lọc món ăn:

Hỗ trợ tìm kiếm món theo tên, loại món, giá hoặc combo để giúp khách hàng dễ dàng lựa chọn món phù hợp.

- Chatbot hỗ trợ khách hàng:

Ứng dụng GeminiAI để tư vấn món tự động, giải đáp thắc mắc, gợi ý món hoặc combo theo khẩu vị người dùng, đặt bàn theo yêu cầu của khách hàng, hỗ trợ khách hàng 24/7 với trải nghiệm tương tác tự nhiên.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT VÀ CÔNG CỤ PHÁT TRIỂN

2.1. Cơ sở lý thuyết

2.1.1. Ngôn ngữ HTML

2.1.1.1. Giới thiệu về HTML

HTML (HyperText Markup Language – Ngôn ngữ đánh dấu siêu văn bản) là ngôn ngữ đánh dấu tiêu chuẩn dùng để xây dựng cấu trúc của các trang web trên World Wide Web. HTML không phải là ngôn ngữ lập trình mà là ngôn ngữ mô tả nội dung, sử dụng các “thẻ” (tags) để tổ chức và hiển thị thông tin như văn bản, hình ảnh, bảng, biểu mẫu và các thành phần giao diện khác.

2.1.1.2. Ưu điểm và nhược điểm của HTML

Ưu điểm

- Dễ học, dễ sử dụng: Phù hợp với người mới bắt đầu lập trình web, cú pháp rõ ràng và đơn giản.
- Được hỗ trợ bởi tất cả trình duyệt: HTML là tiêu chuẩn chung được Chrome, Firefox, Safari, Edge và mọi trình duyệt khác hỗ trợ đầy đủ.
- Tính tương thích cao: Kết hợp với CSS và JavaScript để xây dựng giao diện và tạo tương tác linh hoạt.
- Nhẹ, tốc độ tải nhanh: Cấu trúc tối giản, giúp các trang web tải nhanh và hoạt động ổn định.

Nhược điểm

- Chỉ tạo ra nội dung tĩnh: HTML không xử lý logic hay dữ liệu động như các ngôn ngữ lập trình backend.
- Không hỗ trợ xử lý tác vụ phức tạp: Những nghiệp vụ như đặt món, tính tổng tiền, quản lý đơn hàng... phải kết hợp thêm JavaScript hoặc backend như Spring Boot.
- Cần phối hợp với CSS/Javascript để tạo giao diện đẹp và tương tác người dùng tốt hơn.

2.1.2. Ngôn ngữ CSS

2.1.2.1. Giới thiệu về CSS

CSS (Cascading Style Sheets) là ngôn ngữ dùng để mô tả cách trình bày và định dạng các phần tử HTML trên trang web. Nếu HTML chịu trách nhiệm tạo cấu trúc nội dung, thì CSS giúp định hình giao diện như bố cục, màu sắc, phông chữ, kích thước, khoảng cách và hiệu ứng thị giác.

2.1.2.2. Ưu và nhược điểm của CSS

Ưu điểm

- Tách biệt cấu trúc và kiểu dáng: CSS giúp tách riêng giao diện với nội dung HTML, giúp mã nguồn dễ quản lý và bảo trì.
- Tối ưu tốc độ tải trang: Các stylesheet được tải một lần và tái sử dụng cho nhiều trang, giảm dung lượng và tăng tốc độ tải.
- Dễ tái sử dụng và mở rộng: Một đoạn CSS có thể áp dụng cho nhiều phần tử và nhiều trang web.
- Hỗ trợ responsive: Cho phép thiết kế giao diện phù hợp với mọi kích thước màn hình thông qua media queries.
- Tương thích với các thư viện UI: Như Tailwind CSS, Bootstrap hoặc SCSS để tăng tốc quá trình thiết kế.

Nhược điểm

- Không hỗ trợ xử lý động: CSS chỉ quản lý giao diện, không thể thực hiện các tác vụ logic như tính toán hoặc xử lý dữ liệu.
- Có thể gặp lỗi khác nhau trên các trình duyệt: Một số thuộc tính CSS hiển thị không đồng nhất giữa các trình duyệt, đòi hỏi lập trình viên phải kiểm tra và tối ưu thêm.
- Quản lý khó trong dự án lớn: Nếu không tổ chức tốt, CSS có thể bị rối hoặc xung đột giữa các class.

2.1.3. Ngôn ngữ JavaScript

2.1.3.1. Giới thiệu về JavaScript

JavaScript là một trong những ngôn ngữ lập trình phổ biến nhất hiện nay, được phát triển vào năm 1995 nhằm bổ sung tính tương tác cho các trang web vốn chỉ có HTML

và CSS dạng tĩnh. JavaScript cho phép kiểm soát hành vi của trang web, xử lý sự kiện, tương tác với người dùng và cập nhật nội dung động mà không cần tải lại trang.

2.1.3.2. Ưu nhược điểm của JavaScript

Ưu điểm:

- Tạo chức năng động và tương tác mạnh mẽ: Cho phép xử lý sự kiện, hiệu ứng, hoạt động kéo-thả, cập nhật giao diện mà không cần tải lại trang.
- Hỗ trợ rộng rãi: Tất cả trình duyệt đều hỗ trợ JavaScript, giúp trang web hoạt động ổn định và nhất quán.
- Dễ tích hợp: Kết hợp linh hoạt với HTML, CSS và phù hợp với các framework hiện đại như Vue.js, React, Angular.
- Hiệu suất cao: Với các engine hiện đại như V8, JavaScript chạy nhanh, phù hợp với SPA và các website tương tác nhiều

Nhược điểm:

- Dễ gây lỗi nếu không quản lý tốt: Việc thao tác DOM hoặc viết mã không chuẩn có thể dẫn đến lỗi khó kiểm soát.
- Bảo mật cần lưu ý: JavaScript chạy phía client nên có thể bị khai thác nếu không xử lý hợp lý, ví dụ như XSS.
- Dễ gây lỗi nếu không quản lý tốt: Việc thao tác DOM hoặc viết mã không chuẩn có thể dẫn đến lỗi khó kiểm soát.

2.1.4. Ngôn ngữ lập trình Java

2.1.4.1. Giới thiệu về Java

Java là một trong những ngôn ngữ lập trình phổ biến và mạnh mẽ nhất trên thế giới, được phát triển bởi Sun Microsystems (nay thuộc Oracle). Java được thiết kế với triết lý Write Once, Run Anywhere (WORA), nghĩa là mã nguồn được biên dịch một lần và có thể chạy trên mọi nền tảng hỗ trợ Java Virtual Machine (JVM).

Java là ngôn ngữ hướng đối tượng (OOP) mạnh mẽ, có cú pháp rõ ràng, dễ bảo trì, đồng thời sở hữu hệ sinh thái thư viện phong phú, hỗ trợ tốt trong việc xây dựng các hệ thống lớn, ổn định và an toàn.

2.1.4.2. Tính chất của java

Java sở hữu một số tính chất nổi bật giúp nó trở thành lựa chọn lý tưởng để xây dựng hệ thống web:

- Hướng đối tượng (OOP): hỗ trợ các khái niệm lớp, đối tượng, kế thừa, đa hình, đóng gói, giúp tổ chức mã nguồn logic và dễ bảo trì.
- Quản lý bộ nhớ tự động: thông qua cơ chế Garbage Collector, giúp giảm rủi ro rò rỉ bộ nhớ.
- An toàn và bảo mật: hỗ trợ nhiều cơ chế bảo mật tích hợp như ClassLoader, Security Manager và các thư viện mã hóa.
- Đa luồng (Multithreading): giúp Java xử lý nhiều tác vụ đồng thời, phù hợp cho ứng dụng cần hiệu năng.

2.1.4.3. Java là ngôn ngữ đa nền tảng

Một trong những ưu điểm quan trọng của Java là khả năng chạy trên nhiều hệ điều hành khác nhau. Nhờ JVM, các chương trình Java có thể chạy trên:

- Windows
- Linux
- macOS
- Các hệ điều hành máy chủ chuyên dụng

2.1.4.4. Tập tin Java

Các tập tin Java thường có phần mở rộng “.java”, chứa mã nguồn được viết bằng ngôn ngữ Java. Mỗi tập tin Java thường chứa một lớp (class) hoặc giao diện (interface) tương ứng, và được biên dịch bởi trình biên dịch javac để tạo ra các tập tin “.class” chạy trên Java Virtual Machine (JVM).

2.1.5. Cơ sở dữ liệu PostgreSQL

2.1.5.1. Giới thiệu về cơ sở dữ liệu PostgreSQL

PostgreSQL là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) mã nguồn mở mạnh mẽ và tiên tiến, được phát triển từ năm 1986 tại Đại học California, Berkeley. PostgreSQL nổi tiếng với tính ổn định cao, khả năng mở rộng mạnh mẽ và tuân thủ nghiêm ngặt các tiêu chuẩn SQL.

Không chỉ hỗ trợ dữ liệu quan hệ truyền thống, PostgreSQL còn hỗ trợ dữ liệu phi quan hệ (NoSQL) như JSON, JSONB, HSTORE, giúp linh hoạt trong việc lưu trữ nhiều loại cấu trúc dữ liệu khác nhau.

PostgreSQL được sử dụng rộng rãi trong các hệ thống lớn, ứng dụng doanh nghiệp, tài chính, thương mại điện tử và các hệ thống đòi hỏi độ tin cậy cao.



Hình 2-1: Logo PostgreSQL

2.1.5.2. Ưu điểm và nhược điểm của PostgreSQL

Ưu điểm:

- Mã nguồn mở và miễn phí: Cho phép sử dụng tự do, đồng thời được cộng đồng phát triển và cập nhật liên tục.
- Tính ổn định và an toàn dữ liệu cao: Kiến trúc mạnh mẽ, hỗ trợ các cơ chế giao dịch ACID, WAL (Write-Ahead Logging), giúp đảm bảo an toàn dữ liệu.
- Hỗ trợ JSON/JSONB: Giúp PostgreSQL xử lý dữ liệu dạng phi cấu trúc giống như NoSQL Database.
- Khả năng mở rộng mạnh: Hỗ trợ các kiểu dữ liệu tùy chỉnh, hàm tùy chỉnh, trigger, extension (như PostGIS).
- Hiệu năng tốt: Tối ưu hóa truy vấn thông minh với hệ thống lập kế hoạch (Query Planner) mạnh mẽ.
- Tuân thủ chuẩn SQL cao: Hỗ trợ nhiều tính năng nâng cao như foreign keys, subquery, window functions,...

Nhược điểm:

- Cấu hình phức tạp hơn MySQL: Việc tối ưu hóa và quản trị PostgreSQL yêu cầu nhiều kiến thức kỹ thuật hơn.
- Tốc độ đọc trong một số trường hợp kém hơn MySQL: Đặc biệt đối với hệ thống đơn giản hoặc chỉ tập trung truy vấn đọc.
- Tốn tài nguyên hơn: PostgreSQL sử dụng nhiều CPU và RAM hơn trong các tác vụ phức tạp hoặc hệ thống lớn.

2.1.6. Spring Boot và Vue.js Framework

2.1.6.1. Framework là gì? Java Framework là gì?

Framework là một bộ khung phần mềm gồm các công cụ, thư viện và cấu trúc đã được xây dựng sẵn, hỗ trợ lập trình viên phát triển ứng dụng một cách nhanh chóng và hiệu

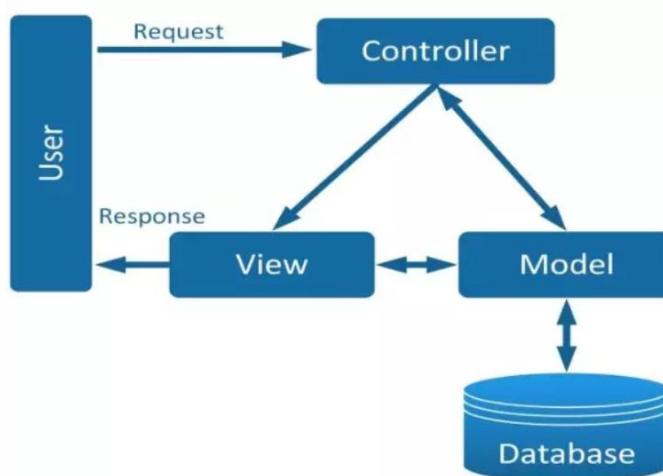
quả hơn. Framework cung cấp các thành phần cốt lõi như mô-đun xử lý, cấu trúc dự án, các thư viện tiện ích và các quy tắc chung, giúp lập trình viên tái sử dụng mã nguồn đã được kiểm thử, giảm thiểu lỗi và đảm bảo tính nhất quán trong quá trình phát triển ứng dụng.



Hình 2-2: Framework thư viện với các tài nguyên không thể thiếu đối với lập trình viên

Java Framework là tập hợp các công cụ và thư viện hỗ trợ lập trình ứng dụng bằng ngôn ngữ Java. Những framework này giúp đơn giản hóa quá trình xây dựng ứng dụng thông qua việc cung cấp các cơ chế quản lý vòng đời đối tượng, quản lý phụ thuộc, xử lý bảo mật, kết nối cơ sở dữ liệu và xây dựng API. Nhờ đó, lập trình viên Java có thể tập trung vào logic nghiệp vụ thay vì phải xử lý các tác vụ lặp lại hoặc mang tính kỹ thuật phức tạp.

2.1.6.2. Mô hình MVC là gì?



Hình 2-3: Mô hình MVC

MVC (Model-View-Controller) là một mẫu kiến trúc phần mềm phổ biến, được sử dụng rộng rãi trong phát triển ứng dụng web. Mục tiêu của mô hình MVC là tách biệt

rõ ràng giữa phần xử lý nghiệp vụ, phần điều khiển luồng dữ liệu và phần hiển thị giao diện người dùng. Điều này giúp hệ thống dễ mở rộng, dễ bảo trì và hạn chế sự phụ thuộc giữa các thành phần.

Ba thành phần ấy bao gồm:

- **Model:** là thành phần quản lý dữ liệu và các quy tắc nghiệp vụ của ứng dụng. Nó chịu trách nhiệm truy xuất, lưu trữ, cập nhật dữ liệu và mô tả cấu trúc dữ liệu. Model không phụ thuộc vào giao diện, đảm bảo dữ liệu và nghiệp vụ được xử lý một cách độc lập.
- **View:** chịu trách nhiệm hiển thị dữ liệu cho người dùng. Đây là phần giao diện mà người dùng tương tác trực tiếp. View chỉ hiển thị thông tin do Controller cung cấp và không xử lý logic nghiệp vụ.
- **Controller:** là thành phần trung gian, tiếp nhận yêu cầu từ người dùng, xử lý yêu cầu đó (có thể thông qua Model) và trả kết quả cho View. Controller điều phối luồng dữ liệu giữa Model và View.

Bằng cách này, thông tin nội hàm được xử lý tách biệt với phần thông tin xuất hiện trong giao diện người dùng. Bảo đảm các nguyên tắc nghề nghiệp của lập trình viên.

2.1.6.3. Spring Boot Framework là gì?

Spring Boot là một framework mã nguồn mở thuộc hệ sinh thái Spring, được thiết kế nhằm đơn giản hóa quá trình xây dựng các ứng dụng Java, đặc biệt là ứng dụng web và dịch vụ RESTful. Spring Boot mở rộng nền tảng Spring Framework truyền thống bằng cách cung cấp cấu hình tự động (auto-configuration), bộ thư viện khởi tạo (starter dependencies) và máy chủ nhúng (embedded server) giúp lập trình viên xây dựng ứng dụng nhanh chóng mà không cần cấu hình phức tạp.

Spring Boot hỗ trợ mạnh mẽ cho kiến trúc microservices, nhờ khả năng triển khai độc lập, dễ mở rộng và tích hợp tốt với các công nghệ bảo mật, cơ sở dữ liệu và các thư viện thuộc hệ sinh thái Java.



Hình 2-4: Logo Spring Boot

Những tính năng nổi bật của Spring Boot là:

- **Spring Initializr:** Công cụ tạo project nhanh, giúp cấu hình ban đầu đơn giản.
- **Starter Dependencies:** Cung cấp các gói phụ thuộc được cấu hình sẵn theo từng mục đích (web, security, data, mail...).
- **Auto Configuration:** Tự động cấu hình ứng dụng dựa trên dependencies có trong project.
- **Embedded Server:** Tích hợp sẵn Tomcat/Jetty/Undertow, không cần cài đặt server thủ công.
- **Spring Data JPA:** Hỗ trợ thao tác cơ sở dữ liệu dễ dàng với JPA/Hibernate.
- **Spring Security:** Cung cấp cơ chế bảo mật mạnh mẽ trong xác thực và phân quyền.
- **Actuator:** Giám sát ứng dụng, theo dõi sức khỏe, log, metrics.
- **CLI & Developer Tools:** Hỗ trợ reload nhanh và nâng cao tốc độ phát triển.

2.1.6.4. Ưu điểm của Spring Boot

- **Giảm cấu hình phức tạp:** Nhờ auto-configuration, ứng dụng có thể chạy ngay với cấu hình mặc định.
- **Phát triển nhanh:** Kết hợp Starter Dependencies và các công cụ hỗ trợ tăng tốc độ phát triển.
- **Dễ mở rộng:** Phù hợp với kiến trúc Microservices, hỗ trợ triển khai độc lập.
- **Bảo mật mạnh mẽ:** Tích hợp Spring Security và nhiều cơ chế mã hóa hiện đại.
- **Tích hợp mạnh với hệ sinh thái Java:** JPA, Hibernate, JDBC, Kafka, RabbitMQ, Redis...
- **Hỗ trợ RESTful API chuyên nghiệp:** Dễ dàng xây dựng các endpoint API JSON.
- **Cộng đồng lớn, tài liệu đầy đủ:** Giúp học và khắc phục lỗi nhanh chóng.

2.1.6.5. Nhược điểm Spring Boot

- **Đường cong học tập cao:** So với các framework như Laravel hay Node.js, Spring Boot yêu cầu kiến thức vững về Java và Spring Core.
- **Tốn tài nguyên:** Ứng dụng Spring Boot đòi hỏi sử dụng RAM và CPU nhiều hơn các framework nhẹ.

- **Tự động cấu hình đôi khi gây khó debug:** Vì Spring Boot tự xử lý nhiều phần, nên khi gặp lỗi liên quan dependency hoặc config sẽ khó xác định nguyên nhân.
- **Dự án lớn dễ phức tạp:** Nếu không tổ chức tốt, số lượng lớp (Controller, Service, Repository) dễ tăng quá nhiều.
- **Jar/War có kích thước lớn:** Do chứa nhiều thư viện tích hợp sẵn.

2.1.6.6. *Vue.js framework là gì?*

Vue.js là một framework JavaScript tiên bộ (progressive framework) dùng để xây dựng giao diện người dùng và ứng dụng web theo mô hình **MVVM (Model–View–ViewModel)**. Vue.js được đánh giá cao nhờ sự linh hoạt, dễ học, dễ tích hợp và hiệu năng tốt.

Khác với các framework phức tạp, Vue.js có thể được áp dụng dần dần vào các phần nhỏ của giao diện hoặc dùng để phát triển các ứng dụng đơn trang (**SPA – Single Page Application**) hoàn chỉnh. Nhờ cú pháp rõ ràng và khả năng tái sử dụng component, Vue.js trở thành lựa chọn phổ biến trong phát triển web hiện đại.



Hình 2-5: Logo Vue.js

2.1.6.7. *Đặc điểm nổi bật của Vue.js*

- **Dễ học:** Cú pháp thân thiện, rõ ràng, dễ tiếp cận với người mới bắt đầu.
- **Component-based:** Giao diện được chia thành các component độc lập, dễ tái sử dụng và bảo trì.
- **Reactivity (tính phản ứng):** Vue.js sử dụng hệ thống theo dõi thay đổi dữ liệu, giúp cập nhật giao diện theo thời gian thực một cách hiệu quả.
- **Virtual DOM:** Giảm tải cho trình duyệt bằng cách chỉ cập nhật những phần tử thay đổi thay vì render toàn bộ trang.
- **Hỗ trợ SPA:** Khi kết hợp với Vue Router và Pinia/Vuex, Vue.js chạy rất tốt với mô hình SPA.
- **Kích thước nhỏ gọn:** Nhẹ hơn nhiều framework khác như Angular hoặc React.

- **Dễ tích hợp:** Có thể nhúng vào bất kỳ dự án web nào, kể cả dự án đã có sẵn.

2.1.6.8. Các thành phần chính trong Vue.js

- **Template:** Định nghĩa giao diện bằng cú pháp HTML mở rộng.
- **Script:** Chứa logic xử lý, dữ liệu và phương thức của component.
- **Style:** Định nghĩa CSS cho component, có thể sử dụng scoped style để tránh trùng CSS.
- **Directives:** Các từ khóa đặc biệt như v-if, v-for, v-model giúp thao tác DOM hiệu quả.
- **Computed & Watch:** Hỗ trợ tính toán tự động hoặc theo dõi thay đổi dữ liệu.
- **Lifecycle Hooks:** Các hàm chạy theo vòng đời component như mounted(), created().

2.1.6.9. Ưu điểm của Vue.js

- Cú pháp đơn giản, dễ đọc, dễ mở rộng.
- Tài liệu hướng dẫn rõ ràng, dễ học.
- Hiệu năng cao nhờ Virtual DOM.
- Hệ sinh thái mạnh mẽ (Vue Router, Pinia/Vuex, Vite).
- Tính linh hoạt cao: dùng được cho cả dự án nhỏ và lớn.
- Dễ bảo trì nhờ thiết kế component.

2.1.6.10. Nhược điểm của Vue.js

- Ít phổ biến hơn React trong các dự án doanh nghiệp nước ngoài.
- Một số plugin không đầy đủ bằng hệ sinh thái của React/Angular.
- Với dự án quá lớn, việc quản lý state phức tạp yêu cầu sử dụng Pinia/Vuex để kiểm soát tốt hơn.
- Vẫn có sự khác biệt giữa các phiên bản (Vue 2 và Vue 3) gây khó khăn cho người mới.

2.1.7. Mô hình ứng dụng SPA

SPA (Single Page Application) là mô hình phát triển ứng dụng web trong đó toàn bộ trang web chỉ được tải một lần duy nhất. Sau khi tải lần đầu, mọi thao tác của người dùng như chuyển trang, xem thông tin, cập nhật dữ liệu... đều được thực hiện **mà không cần tải lại toàn bộ trang**, nhờ việc cập nhật động giao diện bằng JavaScript.

Trong SPA, frontend giao tiếp với backend thông qua **RESTful API**, nhận dữ liệu dưới dạng JSON và hiển thị lên giao diện.



Hình 2-6: Tổng quan về SPA

Đặc điểm của SPA

- **Tải trang một lần (One-time loading)** → tốc độ nhanh và trải nghiệm mượt mà.
- **Cập nhật nội dung động** mà không reload cả trang.
- **Frontend – Backend tách biệt** hoàn toàn.
- **Dùng JavaScript để render giao diện**, thường thông qua các framework như Vue.js, React, Angular.
- **Giao tiếp dữ liệu qua AJAX/Fetch API**.

2.1.7.1. Ưu điểm của SPA

- Tốc độ xử lý nhanh, chuyển trang mượt.
- Tạo trải nghiệm giống ứng dụng mobile.
- Giảm tải cho server (server chỉ trả JSON, không trả HTML).
- Dễ phát triển và mở rộng.

2.1.7.2. Nhược điểm của SPA

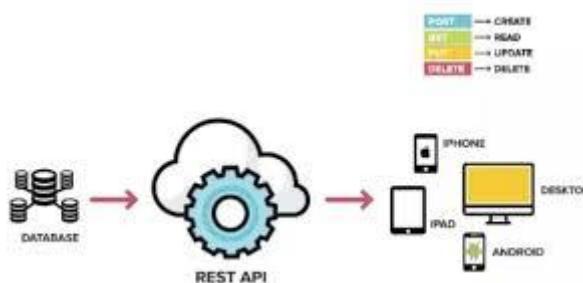
- SEO hạn chế hơn so với mô hình truyền thống.
- Thời gian tải đầu tiên có thể lâu hơn.

- Cần quản lý state phức tạp khi ứng dụng lớn.
- Phụ thuộc vào JavaScript: nếu JS bị lỗi → giao diện không hoạt động.

2.1.8. Kiến trúc giao tiếp: RESTful API

RESTful API (Representational State Transfer) là một phong cách kiến trúc được sử dụng để xây dựng các dịch vụ web, cho phép giao tiếp giữa client và server thông qua giao thức HTTP. Đây là chuẩn giao tiếp phổ biến nhất trong các hệ thống web hiện đại, đặc biệt trong mô hình tách biệt frontend – backend.

RESTful API định nghĩa cách các tài nguyên (resources) được truy xuất thông qua URL và các phương thức HTTP như GET, POST, PUT, DELETE.



Hình 2-7: Sơ đồ RESTful API

Đặc điểm của RESTful API

- **Stateless (Không lưu trạng thái)**: mỗi request phải mang đủ thông tin để server xử lý.
- **Sử dụng HTTP Methods**: GET, POST, PUT, PATCH, DELETE.
- **Dữ liệu trả về dạng JSON** → nhẹ, dễ xử lý.
- **URL mô tả tài nguyên** (không chứa động từ).
Ví dụ: /api/products
/api/orders/123
- **Client–Server tách biệt**: frontend chỉ gọi API để lấy dữ liệu.

Các phương thức HTTP và ý nghĩa :

- **GET** : Lấy dữ liệu
- **POST** : Tạo dữ liệu mới
- **PUT** : Cập nhật toàn bộ dữ liệu
- **PATCH** : Cập nhật một phần dữ liệu
- **DELETE** : Xóa dữ liệu

2.1.8.1. Ưu điểm của RESTful API

- Dễ sử dụng, dễ mở rộng.
- Tối ưu cho web và mobile.
- Hoạt động nhanh vì chỉ truyền JSON.
- Phù hợp với kiến trúc SPA và microservices.

2.1.8.2. Nhược điểm của RESTful API

- Không phù hợp ứng dụng realtime cao (cần WebSocket). Tối ưu cho web và mobile.
- Stateless → client phải tự quản lý phiên đăng nhập.

2.1.9. JWT Authentication

JWT (JSON Web Token) là một chuẩn mã hóa dạng chuỗi (token) dùng để xác thực và phân quyền trong các ứng dụng web hiện đại. JWT được sử dụng phổ biến trong mô hình tách biệt **Frontend – Backend**, đặc biệt khi giao tiếp thông qua **RESTful API**.

JWT chứa thông tin người dùng dưới dạng JSON và được ký (ký HMAC hoặc RSA) để đảm bảo tính toàn vẹn dữ liệu. Nhờ đó, server có thể xác thực người dùng mà **không cần lưu session**, giúp hệ thống nhẹ và dễ mở rộng.



Hình 2-8: Logo JWT

2.1.9.1. Cấu trúc của JWT

Một JWT gồm 3 phần được phân tách bằng dấu chấm: Header.Payload.Signature

- Header: Chứa thông tin thuật toán kí (HS256, RS256...) và loại token.
- Payload: Chứa nội dung (claims).
- Signature: Dùng thuật toán kí với secret key để đảm bảo token không bị thay đổi.

2.1.9.2. Cách hoạt động của JWT

1. Client gửi yêu cầu đăng nhập (username + password).

2. Server kiểm tra thông tin, nếu hợp lệ sẽ tạo token và gửi lại client.
3. Client lưu token
4. Mỗi request tiếp theo, client đính kèm token vào Header:
Authorization: Bearer <jwt-token>
5. Server xác minh token, nếu hợp lệ thì cho phép truy cập tài nguyên

2.1.9.3. Ưu điểm của JWT

- **Không lưu trạng thái (stateless)** → server nhẹ, dễ scale.
- **Nhanh**, chỉ cần kiểm tra chữ ký token.
- **Dễ tích hợp** với ứng dụng mobile, web SPA, microservices.
- **Hỗ trợ phân quyền** (role, permission).
- Tránh lỗi session timeout hoặc session lưu sai người dùng.

2.1.9.4. Nhược điểm của JWT

- Không thể thu hồi (revoke) đơn giản trừ khi dùng blacklist.
- Nếu lưu token không an toàn (localStorage) → có thể bị XSS đánh cắp.
- Token lớn hơn session ID → tốn băng thông hơn.
- Khi token bị lộ → hacker có thể truy cập đến khi token hết hạn.

2.2. Công cụ sử dụng

2.2.1. Visual Studio Code

2.2.1.1. Giới thiệu về Visual Studio Code

Visual Studio Code (VS Code) là một trình soạn thảo mã nguồn mở, miễn phí do Microsoft phát triển, hỗ trợ trên nhiều hệ điều hành như Windows, macOS và Linux. VS Code được đánh giá cao nhờ sự nhẹ, tốc độ xử lý nhanh, giao diện trực quan và khả năng tùy biến mạnh mẽ.

Với thiết kế hiện đại và linh hoạt, Visual Studio Code hỗ trợ lập trình viên làm việc hiệu quả thông qua các tính năng như tự động hoàn thành mã (IntelliSense), quản lý dự án, chạy và gỡ lỗi trực tiếp, tích hợp Git và hệ sinh thái extensions phong phú. Nhờ đó, VS Code trở thành một trong những công cụ lập trình phổ biến nhất trong cộng đồng phát triển phần mềm hiện nay.



Hình 2-9: Logo Visual Studio Code

2.2.1.2. Một số tính năng của Visual Studio Code

- Hỗ trợ đa ngôn ngữ: VS Code hỗ trợ hầu hết các ngôn ngữ lập trình phổ biến như HTML, CSS, JavaScript, TypeScript, Java, Python, PHP, C++, Go,... thông qua các extension, giúp lập trình viên làm việc dễ dàng trên nhiều nền tảng công nghệ.
- Tính năng gỡ lỗi tích hợp (Debugging): VS Code cung cấp giao diện gỡ lỗi trực quan ngay trong trình soạn thảo, cho phép đặt breakpoint, theo dõi biến, kiểm tra giá trị runtime và phân tích luồng thực thi của chương trình.
- Tích hợp Git mạnh mẽ: Trình biên tập hỗ trợ đầy đủ các thao tác Git như commit, pull, push, xem log, quản lý nhánh,... giúp lập trình viên quản lý mã nguồn hiệu quả mà không cần sử dụng thêm phần mềm bên ngoài.

- Kho Extensions phong phú: VS Code cho phép cài đặt hàng nghìn tiện ích mở rộng để hỗ trợ lập trình viên trong nhiều công việc khác nhau như hỗ trợ framework (Vue.js, React, Laravel...), format code, kiểm tra lỗi, tạo snippet hoặc hỗ trợ giao diện.
- Tích hợp Terminal: VS Code cung cấp terminal ngay bên trong ứng dụng, giúp lập trình viên chạy lệnh, build dự án hay quản lý công cụ phát triển thuận tiện mà không cần mở ứng dụng bên ngoài.

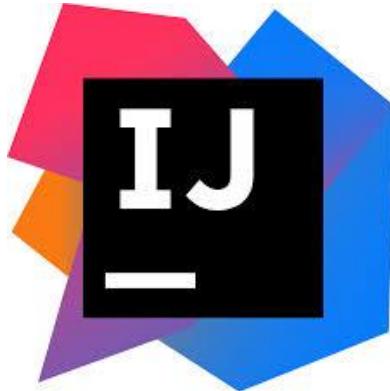
2.2.2. IntelliJ IDEA

2.2.2.1. Giới thiệu về IntelliJ IDEA

IntelliJ IDEA là một môi trường phát triển tích hợp (IDE) mạnh mẽ do JetBrains phát triển, được sử dụng rộng rãi trong việc xây dựng các ứng dụng Java, đặc biệt là những dự án sử dụng các framework như Spring Boot, Hibernate hay Maven/Gradle.

IntelliJ IDEA nổi bật nhờ khả năng phân tích mã thông minh, tự động gợi ý chính xác, tích hợp sâu với hệ sinh thái Java, giao diện hiện đại và các công cụ hỗ trợ phát

triển mạnh mẽ. Đây là IDE được nhiều lập trình viên Java chuyên nghiệp lựa chọn bởi sự ổn định, hiệu suất cao và trải nghiệm lập trình tối ưu.



Hình 2-10: Logo intelliJ

2.2.2.2. *Ưu điểm*

- Hỗ trợ mạnh mẽ cho Java và Spring Boot: IntelliJ IDEA được xem là IDE tốt nhất cho Java nhờ khả năng phân tích mã thông minh, hỗ trợ annotation, tự động gợi ý dependency và tích hợp sâu với các framework như Spring, Hibernate, JPA.
- Tính năng Code Insight vượt trội: Khả năng tự động hoàn thành mã thông minh, highlight lỗi theo thời gian thực và hỗ trợ refactor nâng cao giúp giảm thiểu sai sót và tăng tốc quá trình phát triển.
- Tích hợp nhiều công cụ phát triển trong một IDE: Hỗ trợ Git, Gradle, Maven, Docker, Database Tools, các công cụ kiểm thử và profiler — tất cả đều được tích hợp sẵn, giúp lập trình viên không cần cài nhiều phần mềm khác nhau.
- Debug và theo dõi luồng ứng dụng mạnh mẽ: Khả năng debug chi tiết, đặt breakpoint thông minh, theo dõi biến runtime và chạy ứng dụng Spring Boot với chỉ một cú nhấp chuột.
- Kho plugin phong phú: Thư viện plugin khổng lồ hỗ trợ nhiều công nghệ và framework, dễ dàng mở rộng tính năng theo nhu cầu dự án.
- Hiệu suất làm việc cao: Giao diện trực quan, thao tác mượt mà, nhiều phím tắt hỗ trợ lập trình viên làm việc nhanh và hiệu quả hơn.

2.2.2.3. *Nhược điểm*

- Bản Ultimate có phí: Phiên bản Ultimate hỗ trợ đầy đủ Spring Boot, JPA, Docker... nhưng chi phí khá cao đối với sinh viên.
- Tốn tài nguyên hệ thống: IntelliJ IDEA yêu cầu cấu hình máy mạnh hơn so với các editor nhẹ như VS Code vì IDE chạy nhiều dịch vụ nền, indexing và phân tích mã.

- Thời gian khởi động ban đầu lâu: Do phải tải và phân tích toàn bộ project, IntelliJ IDEA có thể mất vài giây đến hàng chục giây để khởi động.
- Đôi khi xảy ra lỗi indexing: IDE đôi khi phải “Indexing...” lại toàn bộ dự án, khiến lập trình viên phải chờ đợi vài phút tùy kích thước project.

2.2.3. PgAdmin 4

2.2.3.1. Giới thiệu về pgAdmin4

pgAdmin 4 là công cụ quản trị và phát triển cơ sở dữ liệu PostgreSQL phổ biến nhất hiện nay, được phát triển bởi PostgreSQL Global Development Group. Đây là ứng dụng giao diện đồ họa (GUI) hỗ trợ quản lý PostgreSQL dễ dàng bằng trực quan, giúp người dùng thao tác với cơ sở dữ liệu mà không cần viết quá nhiều câu lệnh SQL thủ công.

pgAdmin 4 cung cấp đầy đủ tính năng quản lý như tạo bảng, sửa bảng, truy vấn SQL, xem dữ liệu, theo dõi hoạt động của server, xuất – nhập dữ liệu và quản lý user. Công cụ được thiết kế đa nền tảng, hỗ trợ Windows, macOS, Linux và chạy ổn định trên cả trình duyệt web và desktop app.



Hình 2-11: Logo pgAdmin

2.2.3.2. Ưu điểm

- Giao diện trực quan, dễ sử dụng: pgAdmin 4 cung cấp giao diện đồ họa dạng cây giúp người dùng thao tác với bảng, schema và dữ liệu dễ dàng mà không cần dùng dòng lệnh.
- Hỗ trợ đầy đủ chức năng quản trị PostgreSQL: Cho phép tạo, sửa, xóa bảng; quản lý trigger, index, view và cấu hình quyền truy cập nhanh chóng.
- Trình chạy truy vấn SQL mạnh mẽ: Query Tool hỗ trợ viết, chạy truy vấn, highlight cú pháp và hiển thị kết quả rõ ràng.
- Giám sát hoạt động server hiệu quả: Dashboard tích hợp giúp theo dõi session, tiến trình và hiệu năng PostgreSQL theo thời gian thực.
- Miễn phí và đa nền tảng: Là công cụ chính thức của PostgreSQL, chạy trên Windows, macOS và Linux hoàn toàn miễn phí.

- Hỗ trợ import/export dữ liệu linh hoạt: Cho phép sao lưu và phục hồi dữ liệu dễ dàng thông qua giao diện trực quan.

2.2.3.3. Nhược điểm

- Hiệu năng không thật sự tối ưu: Ứng dụng có thể chạy chậm trên máy cấu hình yếu, đặc biệt khi làm việc với nhiều bảng hoặc cơ sở dữ liệu lớn.
- Một số thao tác chưa thuận tiện: Một vài tùy chọn nằm sâu trong menu, kém trực quan hơn so với các công cụ như DBeaver hoặc DataGrip.
- Dễ bị treo khi xử lý dữ liệu lớn: Import hoặc export file dung lượng cao đôi khi khiến ứng dụng bị lag hoặc đứng.
- Tiêu tốn tài nguyên hệ thống: Vì chạy trên nền web nội bộ nên pgAdmin sử dụng khá nhiều RAM và CPU khi mở nhiều tab.
- Trình chạy SQL đôi khi phản hồi chậm: Khi chạy truy vấn phức tạp hoặc bảng có nhiều liên kết, tốc độ phản hồi có thể bị giảm.

2.2.4. GeminiAI

2.2.4.1. Giới thiệu về GeminiAI

GeminiAI là nền tảng trí tuệ nhân tạo do Google phát triển, hỗ trợ tạo lập nội dung, xử lý ngôn ngữ tự nhiên (NLP), và xây dựng chatbot tự động tương tác với người dùng trên website. Dựa trên mô hình ngôn ngữ tiên tiến, GeminiAI có khả năng phân tích ngữ cảnh, hiểu câu hỏi và đưa ra câu trả lời tự nhiên, chính xác. Nhờ đó, nền tảng này trở thành công cụ mạnh mẽ giúp doanh nghiệp nâng cao trải nghiệm khách hàng, tối ưu hóa quy trình hỗ trợ và tiết kiệm chi phí vận hành.



Hình 2-12: Logo Gemini AI

2.2.4.2. Ưu điểm của GeminiAI

- Khả năng hiểu ngôn ngữ tự nhiên mạnh mẽ: GeminiAI sử dụng mô hình AI mới nhất, cho phép chatbot trả lời tự nhiên, chính xác và đa dạng tình huống.
- Tích hợp linh hoạt: Dễ dàng tích hợp vào website hoặc ứng dụng thông qua API.

- Tùy chỉnh linh hoạt: Hỗ trợ tùy chỉnh kịch bản hội thoại, dữ liệu huấn luyện và ngữ cảnh theo nhu cầu ứng dụng thực tế.
- Hỗ trợ đa ngôn ngữ: Có khả năng xử lý và giao tiếp bằng nhiều ngôn ngữ khác nhau, phù hợp với lượng khách du lịch đa dạng.

2.2.4.3. Nhược điểm của GeminiAI

- Phụ thuộc vào Internet: Để hoạt động hiệu quả, GeminiAI yêu cầu kết nối mạng ổn định.
- Chi phí nâng cao: Một số tính năng nâng cao hoặc mức sử dụng lớn có thể phát sinh chi phí tùy theo gói dịch vụ của Google.

2.2.5. VNPay

2.2.5.1. Giới thiệu về VNPay

VNPay là một trong những cổng thanh toán điện tử hàng đầu tại Việt Nam, cung cấp dịch vụ thanh toán trực tuyến thông qua nhiều phương thức như QR Pay, thanh toán thẻ nội địa (ATM), thẻ quốc tế (Visa/Mastercard/JCB), ví điện tử và Mobile Banking của hơn 40 ngân hàng. Trong đó, VNPay-QR là phương thức nổi bật nhất, cho phép người dùng thanh toán bằng cách quét mã QR trực tiếp trên ứng dụng ngân hàng mà không cần sử dụng tiền mặt hay nhập thông tin thẻ.

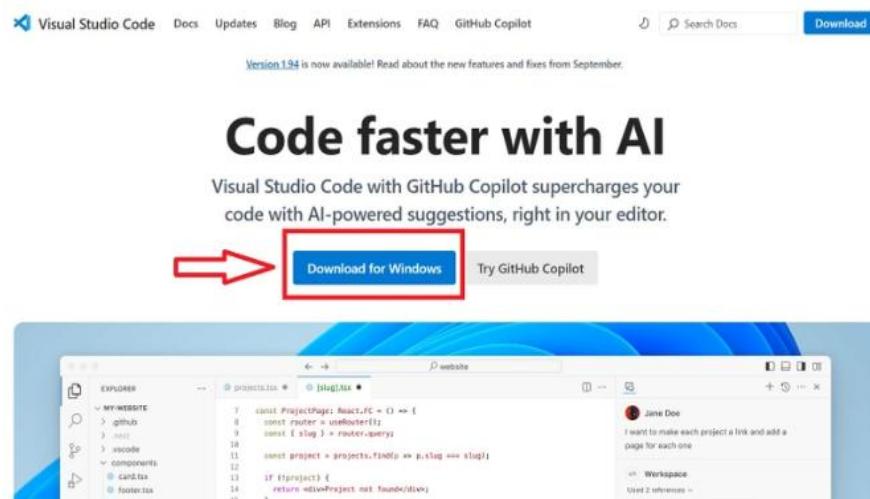


Hình 2-13: Logo VNPAY

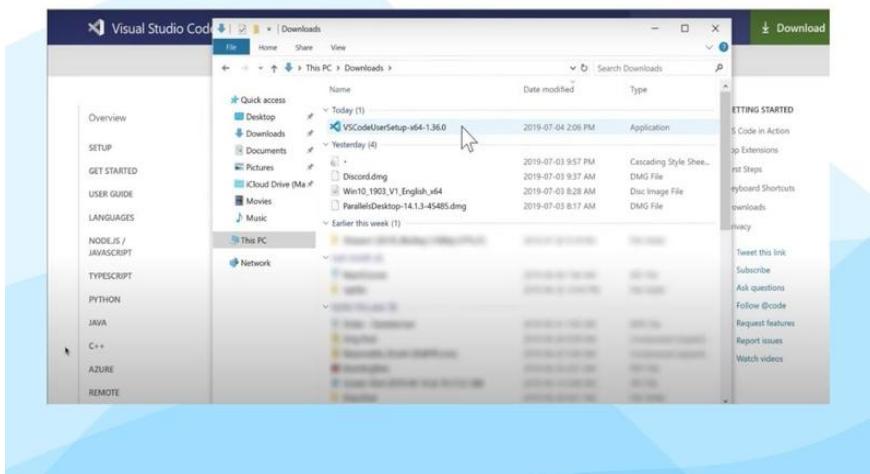
2.3. Các bước cài đặt công cụ

2.3.1. Cài đặt Visual Studio Code trên Windows

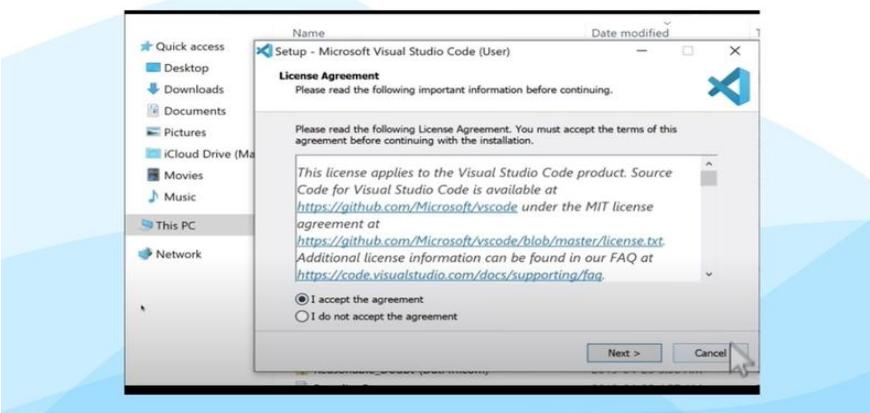
Bước 1: Truy cập vào [trang web chính thức của Visual Studio Code](#), sau đó bạn chọn hệ điều hành **windows** để bắt đầu tải về.



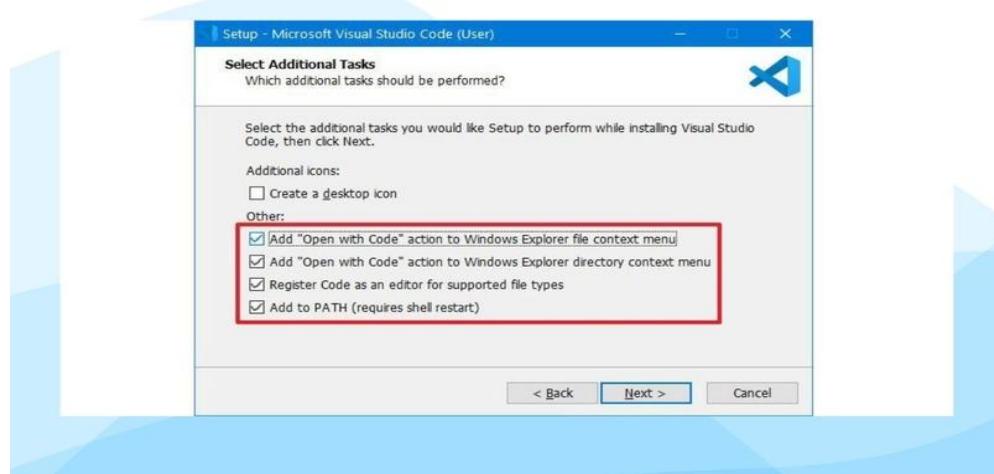
Bước 2: Sau khi tải xong, mở file cài đặt để khởi động quá trình cài đặt phần mềm.



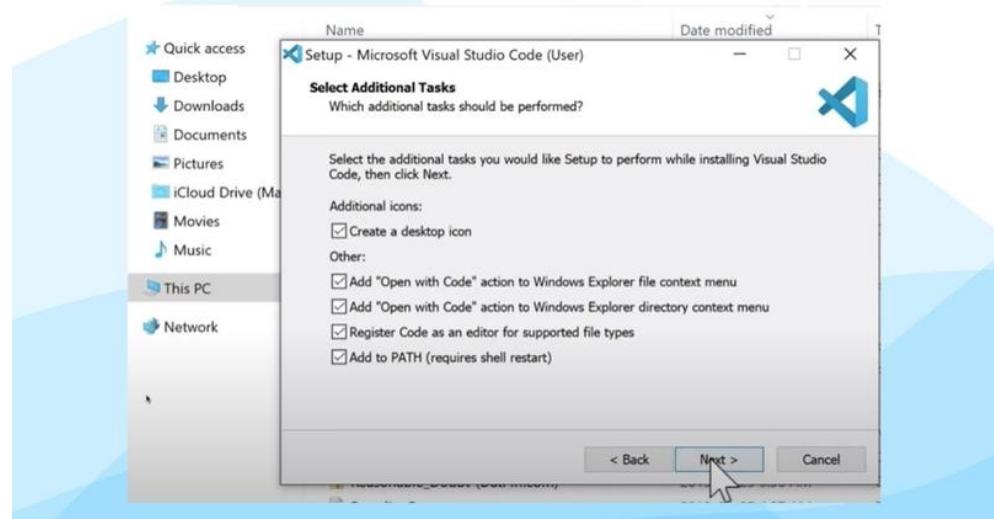
Bước 3: Trong cửa sổ cài đặt, bạn cần chọn I Agree the argeement để đồng ý với các điều khoản sử dụng của Visual Studio Code.



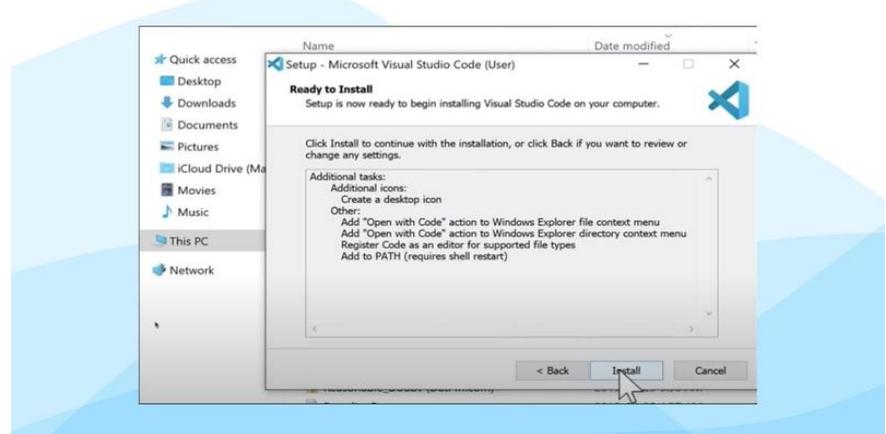
Bước 4: Tiếp theo, bạn sẽ thấy một số tùy chọn cài đặt khác. Hãy tích chọn các ô trong mục Other theo nhu cầu của bạn.



Bước 5: Nhấn Next để tiếp tục qua các bước cài đặt tiếp theo.



Bước 6: Đảm bảo rằng tùy chọn Add to PATH được kiểm tra, sau đó nhấn Install để bắt đầu quá trình cài đặt.

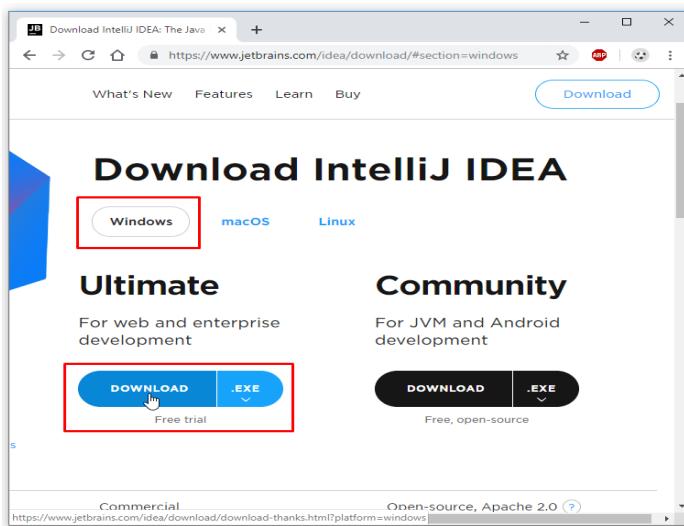


Bước 7: Sau khi cài đặt hoàn tất, nhấn Finish để kết thúc.

2.3.2. Cài đặt IntelliJ IDEA

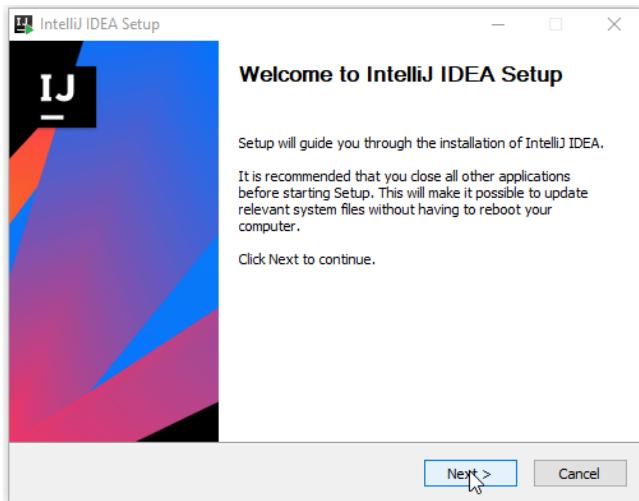
Bước 1: Truy cập vào trang web chính thức của IntelliJ IDEA tại địa chỉ <https://www.jetbrains.com/idea/download/>.

Bước 2: Tại trang download, bạn sẽ thấy các phiên bản của IntelliJ IDEA được liệt kê. Chọn phiên bản Ultimate để tải xuống bản Ultimate của IntelliJ IDEA..



Bước 3: Sau khi chọn phiên bản, bạn sẽ được đưa đến trang tải xuống. Nhấp vào liên kết để tải xuống file cài đặt (file .exe trên Windows hoặc file .dmg trên Mac).

Bước 4: Khi tải xuống hoàn tất, nhấp đúp vào file .exe (Windows) hoặc file .dmg (Mac) để bắt đầu quá trình cài đặt.



Bước 5: Tiến hành chọn folder để cài đặt IntelliJ IDEA trên máy tính của bạn.

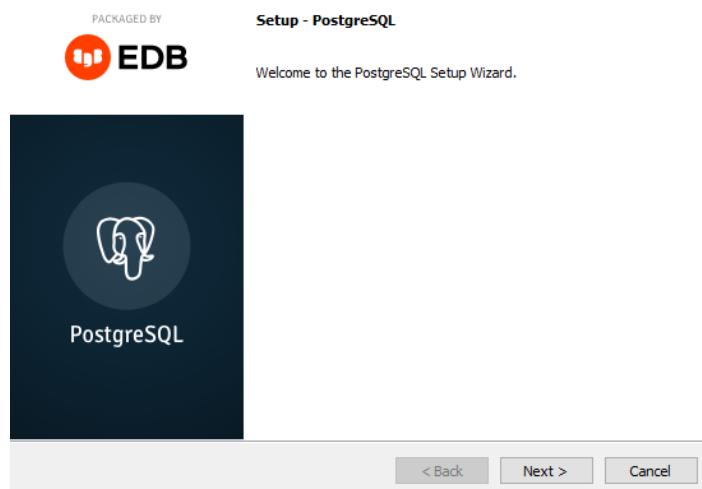
Bước 6: Hoàn thành quá trình cài đặt bằng cách tuân theo các hướng dẫn trên màn hình.

2.3.3. Cài đặt pgAdmin 4

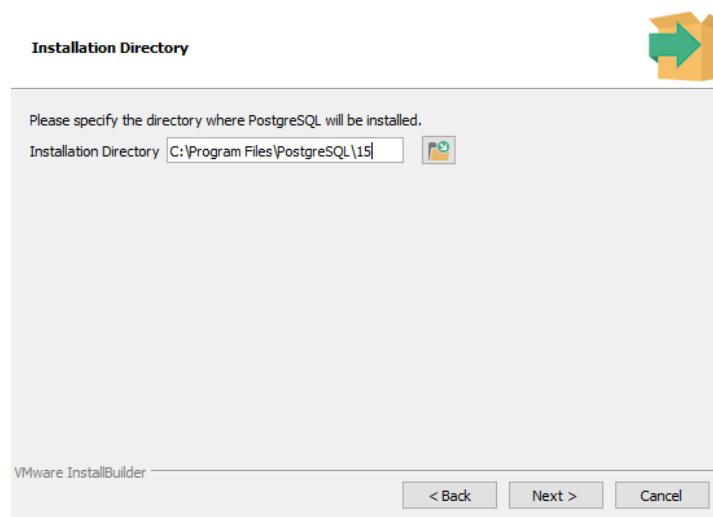
Bước 1: Truy cập <https://www.enterprisedb.com/downloads/postgres-postgresql-downloads> và tải xuống phiên bản mới nhất tương thích với hệ điều hành của bạn.



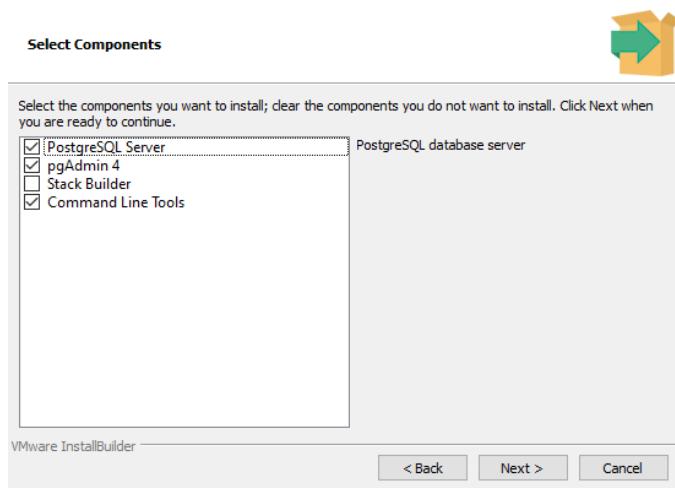
Bước 2: Khi quá trình tải xuống ta tiến hành cài đặt



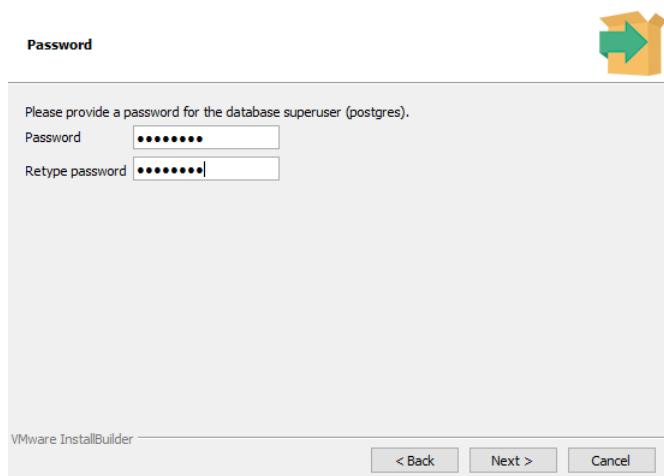
Bước 3: Chỉ định thư mục



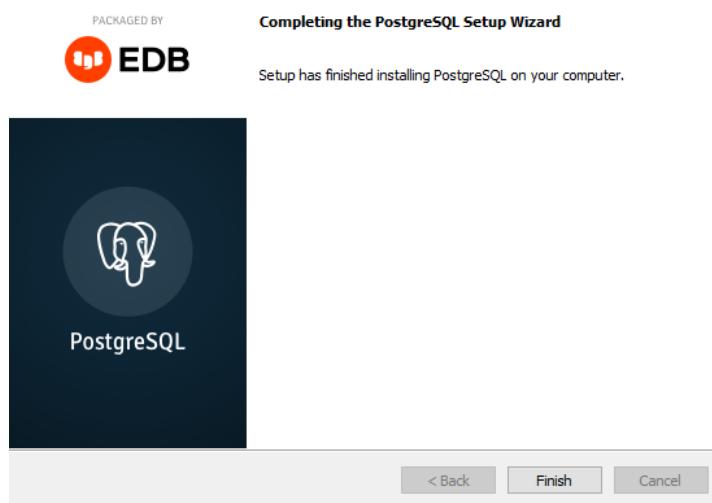
Bước 4: Chọn thành phần



Bước 5: Cài đặt mật khẩu và port



Bước 6: Hoàn thành quá trình cài đặt bằng cách tuân theo các hướng dẫn trên màn hình.



2.3.4. Cài đặt Node.js và NPM

Bước 1: Truy cập <https://nodejs.org>

Bước 2: Chọn bản LTS để đảm bảo ổn định.

Bước 3: Cài đặt và kiểm tra bằng lệnh node -v và npm -v

```
C:\Users\ADMIN>node -v  
v22.19.0
```

```
C:\Users\ADMIN>npm -v  
10.9.3
```

2.3.5. Cài đặt Java JDK

Bước 1: Truy cập trang: <https://adoptium.net>

Bước 2: Tải Temurin JDK 17 hoặc 21.

Bước 3: Cài đặt và kiểm tra bằng lệnh java -version và javac -version

```
C:\Users\ADMIN>java -version  
java version "24.0.1" 2025-04-15  
Java(TM) SE Runtime Environment (build 24.0.1+9-30)  
Java HotSpot(TM) 64-Bit Server VM (build 24.0.1+9-30, mixed mode, sharing)
```

```
C:\Users\ADMIN>javac -version  
javac 24.0.1
```

CHƯƠNG 3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

3.1. Mô tả bài toán

Dành cho khách hàng

Mục tiêu: Trang web được thiết kế nhằm tạo điều kiện để khách hàng tương tác với nhà hàng một cách thuận tiện hơn, giảm thời gian chờ đợi và nâng cao trải nghiệm khi đặt bàn – đặt món.

Chức năng:

- Xem thực đơn món ăn và combo:** Khách hàng có thể duyệt danh sách món ăn theo danh mục (Khai vị, Lẩu, Món nướng...), xem giá, hình ảnh, mô tả, và tình trạng còn bán.
- Đặt món và giỏ hàng:** Thêm món vào giỏ, điều chỉnh số lượng, gi chú yêu cầu riêng.
- Đặt bàn trước:** Khách hàng có thể đặt bàn, chọn khu vực và ngày giờ đến.
- Đặt món qua mã QR:** Cho phép khách hàng quét mã QR tại bàn để gọi trực tiếp món ăn.
- Thanh toán:** Hỗ trợ thanh toán bằng mã QR nhanh chóng và an toàn
- Quản lý thông tin cá nhân:** Khách hàng có thể cập nhật thông tin cá nhân, mật khẩu, voucher hoặc xem các đơn hàng đã đặt.
- Chatbot:** gợi ý hỗ trợ đặt bàn và trả lời câu hỏi về món ăn, giá cả...

Dành cho quản lý

Mục tiêu: Cung cấp hệ thống quản trị giúp nhân viên và quản lý dễ dàng vận hành hoạt động nhà hàng, bao gồm quản lý thực đơn, xử lý đơn hàng, theo dõi bàn ăn, quản lý khách hàng và thống kê.

Chức năng:

- Quản lý món ăn:** Thêm, sửa, xóa món ăn và phân loại theo danh mục.
- Quản lý Combo:** Tạo combo nhiều món, chỉnh sửa giá, mô tả và danh sách món.
- Quản lý bàn ăn và khu vực:** Thêm, sửa, xóa bàn và quản lý theo khu vực.
- Quản lý người dùng:** Cung cấp chức năng quản lý thông tin khách hàng, từ hồ sơ cá nhân đến lịch sử đặt tour, giúp phục vụ khách hàng nhanh chóng và hiệu quả.

5. **Quản lý đơn đặt bàn và đặt món:** Nhận đơn hàng và xử lý gián bàn/ giao món ăn cho khách hàng.
6. **Quản lý voucher:** Tạo voucher theo giảm giá trực tiếp hoặc phần trăm đơn hàng của khách hàng.
7. **Quản lý đánh giá:** Xem đánh giá của khách hàng và có thể ẩn đánh giá nếu cần

3.2. Xác định Actor

Bảng 3.1 Xác định Actor

Actor	Chức năng
Khách hàng	Đăng kí, đăng nhập, quên mật khẩu
	Xem danh sách món ăn và combo
	Tìm kiếm món ăn, thêm vào giỏ hàng và cập nhật số lượng
	Đặt món / đặt bàn
	Quét mã QR tại bàn để đặt món
	Đánh giá
	Chat với Ai để được tư vấn món ăn và đặt bàn
	Tích điểm và đổi voucher
	Xem lịch sử
Admin	Sử dụng voucher và thanh toán(VNPay hoặc tiền mặt)
	Đăng nhập
	Quản lý danh mục, món ăn và combo
	Quản lý bàn ăn và khu vực
	Quản lý người dùng
	Quản lý đơn đặt bàn và đặt món
	Quản lý voucher
	Quản lý đánh giá

3.2.1. Khách hàng

Khách hàng là những người có nhu cầu xem thực đơn, đặt bàn và đặt món trực tuyến tại Nhà hàng thực dưỡng Hoa Ban. Trong mô hình truyền thống, khách hàng thường phải đến trực tiếp nhà hàng hoặc gọi điện để đặt bàn, gây mất thời gian và dễ xảy ra nhầm lẫn, đặc biệt vào giờ cao điểm. Việc gọi món tại bàn cũng thường được thực hiện bằng cách ghi chép thủ công, dẫn đến sai sót hoặc chậm trễ.

Hệ thống website đặt bàn – đặt món giúp khách hàng tương tác dễ dàng hơn với nhà hàng thông qua các chức năng hiện đại:

- Xem thực đơn chi tiết gồm tên món, hình ảnh, mô tả và giá.
- Tìm kiếm và lọc món ăn nhanh chóng theo danh mục hoặc mức giá.
- Thêm món vào giỏ hàng, điều chỉnh số lượng, ghi chú yêu cầu đặc biệt (ít dầu, không cay...).
- Đặt bàn trước ngay trên website hoặc quét mã QR tại bàn để gọi món trực tiếp mà không cần nhân viên ghi nhận.
- Thanh toán linh hoạt bằng VNPay QR hoặc tiền mặt.
- Xem lại lịch sử đặt món, theo dõi trạng thái đơn hàng.
- Đánh giá dịch vụ, tạo lợi ích tham khảo cho khách hàng khác và cải thiện chất lượng phục vụ của nhà hàng.
- Sử dụng điểm tích lũy và đổi voucher để nhận ưu đãi.
- ChatBOT AI hỗ trợ đặt bàn, tư vấn món ăn theo khẩu vị và nhu cầu dinh dưỡng của khách hàng.

Nhờ các tính năng trên, hệ thống mang lại trải nghiệm đặt món nhanh chóng, tiện lợi, giảm thời gian chờ đợi và tạo sự hiện đại trong quy trình phục vụ tại Nhà hàng Hoa Ban.

3.2.2. Admin

Admin là người quản trị hệ thống, chịu trách nhiệm điều hành toàn bộ hoạt động trên nền tảng đặt bàn – đặt món của nhà hàng. Admin có quyền truy cập vào giao diện quản trị để quản lý dữ liệu, theo dõi đơn hàng và đảm bảo hệ thống hoạt động ổn định.

Các nhiệm vụ chính của Admin gồm:

- Quản lý danh mục và món ăn: thêm mới, chỉnh sửa, xóa món; cập nhật hình ảnh, giá và mô tả.
- Quản lý combo: tạo combo, thêm món vào combo, quản lý giá ưu đãi.

- Quản lý bàn ăn và khu vực: cập nhật trạng thái bàn, tạo mới bàn và phân khu vực.
- Quản lý người dùng: theo dõi danh sách khách hàng, khóa/mở tài khoản khi cần thiết.
- Quản lý đơn đặt bàn và đơn đặt món: xem chi tiết đơn, xác nhận xử lý, cập nhật trạng thái.
- Quản lý voucher: tạo voucher, thiết lập điều kiện sử dụng, vô hiệu hóa voucher khi cần.
- Quản lý đánh giá: xem đánh giá của khách hàng, xử lý các đánh giá không phù hợp.

Admin đóng vai trò quan trọng giúp nhà hàng vận hành trơn tru, đảm bảo tính chính xác của hệ thống, nâng cao hiệu quả phục vụ và mang đến trải nghiệm tốt nhất cho khách hàng.

3.3. Xác định Usecase

Bảng 3.2 Xác định Usecase

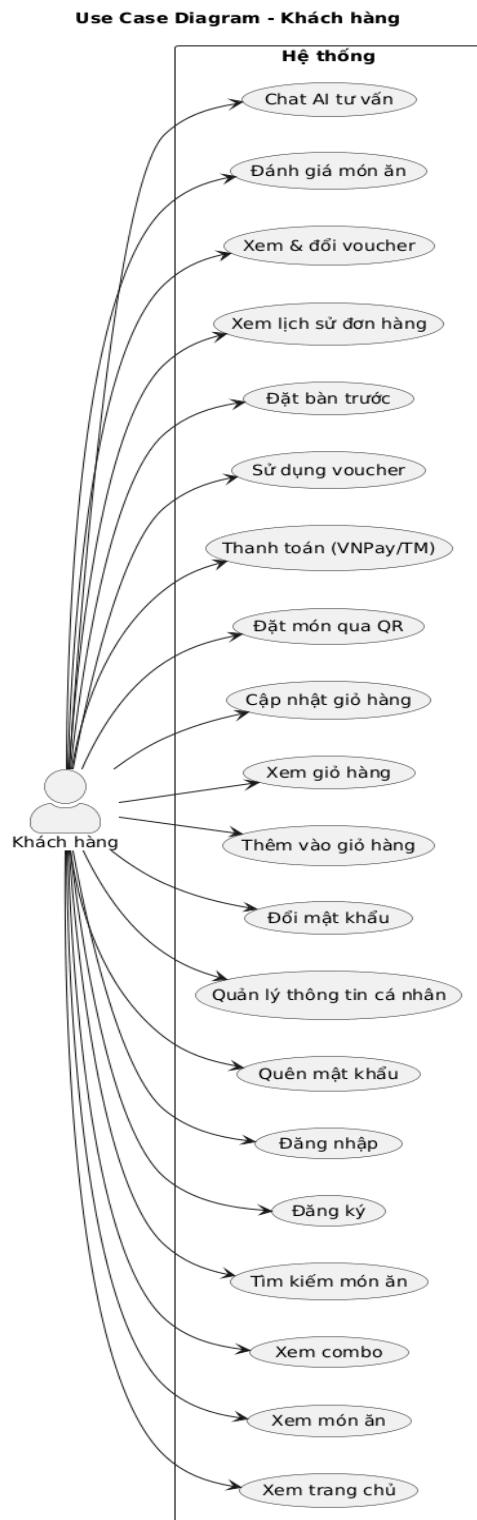
Actor	Usecase
Khách hàng	UC01: Xem trang chủ UC02: Xem danh sách món ăn UC03: Xem danh sách combo UC04: Tìm kiếm món ăn UC05: Đăng ký tài khoản UC06: Đăng nhập UC07: Quên mật khẩu UC08: Xem và cập nhật thông tin cá nhân UC09: Đổi mật khẩu UC10: Thêm món vào giỏ hàng UC11: Xem giỏ hàng UC12: Cập nhật giỏ hàng UC13: Quét mã QR tại bàn để đặt món

	UC14: Thanh toán đơn hàng(VNPAY/Tiền mặt) UC15: Sử dụng voucher UC16: Đặt bàn trước UC17: Xem lịch sử đơn hàng UC18: Xem danh sách voucher và dùng điểm đổi UC19: Đánh giá sau khi đặt đồ ăn UC20: Chat với AI để được tư vấn món / hỗ trợ đặt bàn
Admin	UC21: Đăng nhập UC22: Quản lý danh mục món ăn UC23: Quản lý món ăn UC24: Quản lý combo UC25: Quản lý bàn ăn và khu vực UC26: Quản lý người dùng UC27: Quản lý đơn đặt bàn và đặt món UC28: Quản lý voucher UC29: Quản lý đánh giá

3.4. Phân tích Usecase

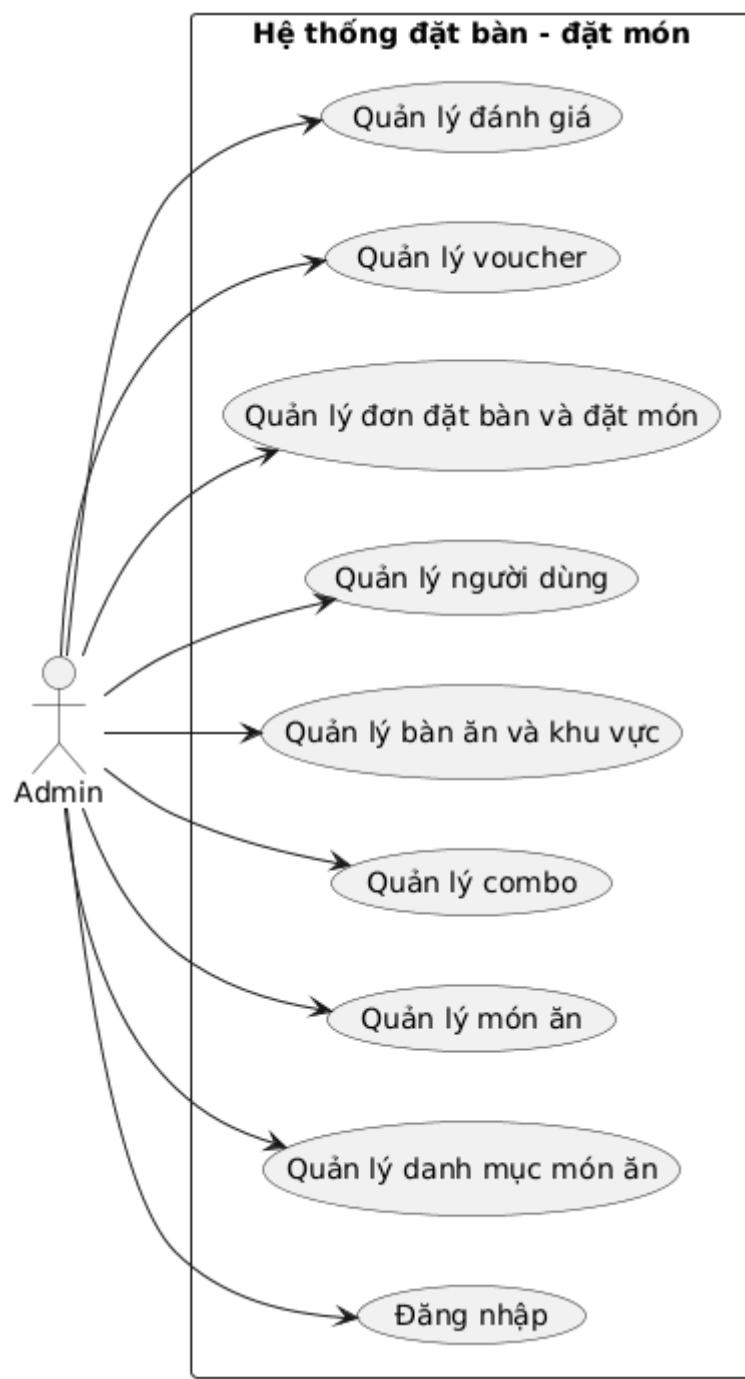
3.4.1. Biểu đồ Usecase

3.4.1.1. Usecase khách hàng



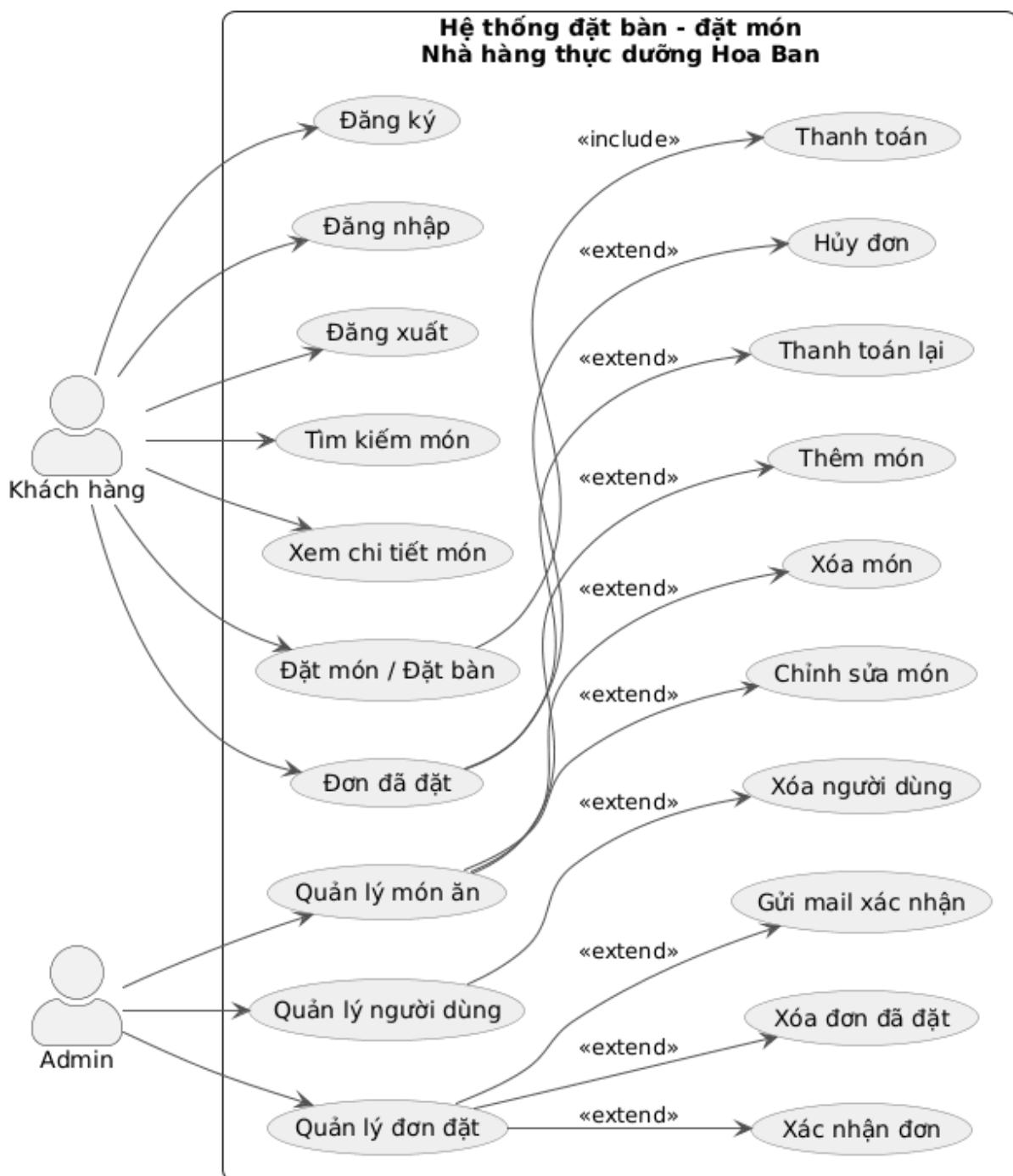
Hình 3-1: Sơ đồ Usecase khách hàng

3.4.1.2. Usecase Admin



Hình 3-2: Sơ đồ Usecase Admin

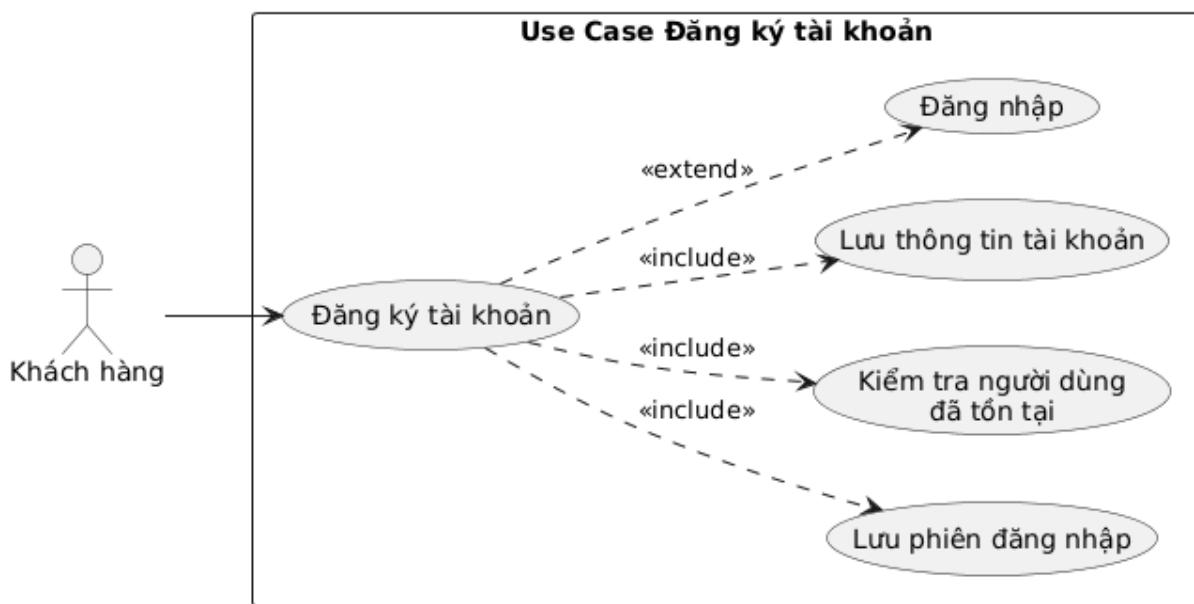
3.4.1.3. Usecase tổng quát



Hình 3-3: Sơ đồ Usecase tổng quát

3.4.2. Đặc tả usecase

3.4.2.1. Usecase đăng ký



Hình 3-4: Đặc tả Usecase đăng ký

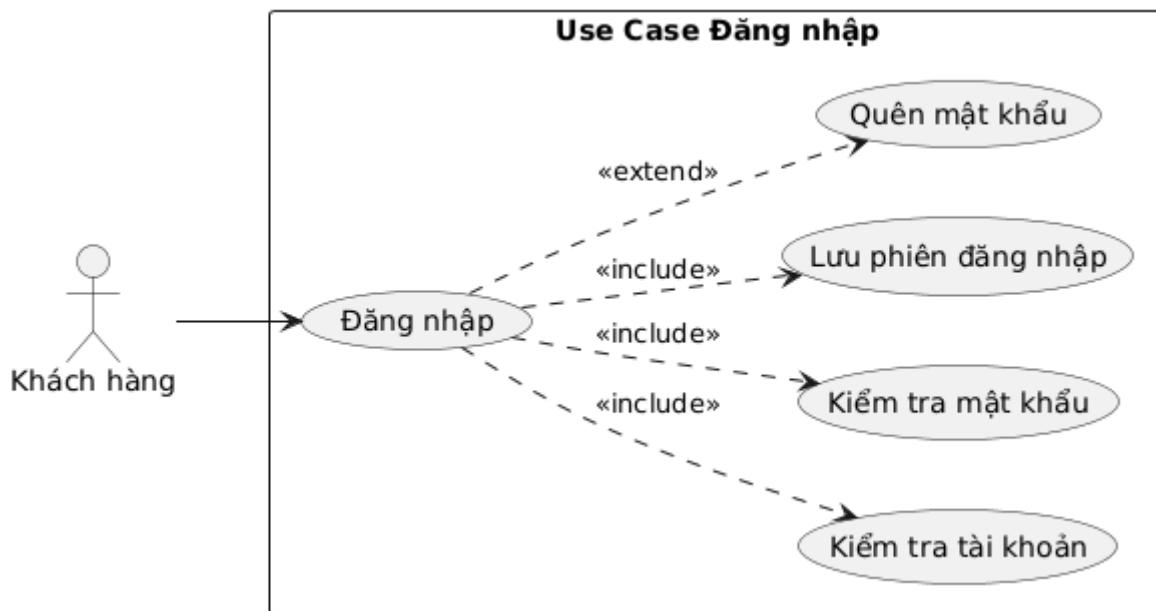
Đặc tả Usecase đăng ký

Bảng 3.3 Đặc tả usecase đăng ký

TT	Tên UC	Đăng ký tài khoản
1	Tác nhân	Khách hàng
2	Mô tả	Khách hàng tạo mới tài khoản để có thể đăng nhập và sử dụng các chức năng của hệ thống.
3	Điều kiện trước	Khách hàng chưa có tài khoản trong hệ thống.
4	Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Hệ thống hiển thị form đăng ký. 2. Khách hàng nhập thông tin lên form. 3. Khách hàng nhấp nút “đăng ký”. 4. Hệ thống tiến hành kiểm tra thông tin. 5. Tạo mới tài khoản với thông tin trên.

		6. Trở lại trang login.
5	Luồng sự kiện phụ	Nếu thông tin không hợp lệ hoặc tài khoản đã tồn tại, hệ thống hiển thị thông báo lỗi và yêu cầu khách hàng nhập lại.
6	Điều kiện sau	Nếu thành công thì chuyển hướng tới trang đăng nhập, cho phép thực hiện các tác vụ có yêu cầu phải đăng nhập.
7	Yêu cầu đặc biệt	Không có.

3.4.2.2. Usecase đăng nhập



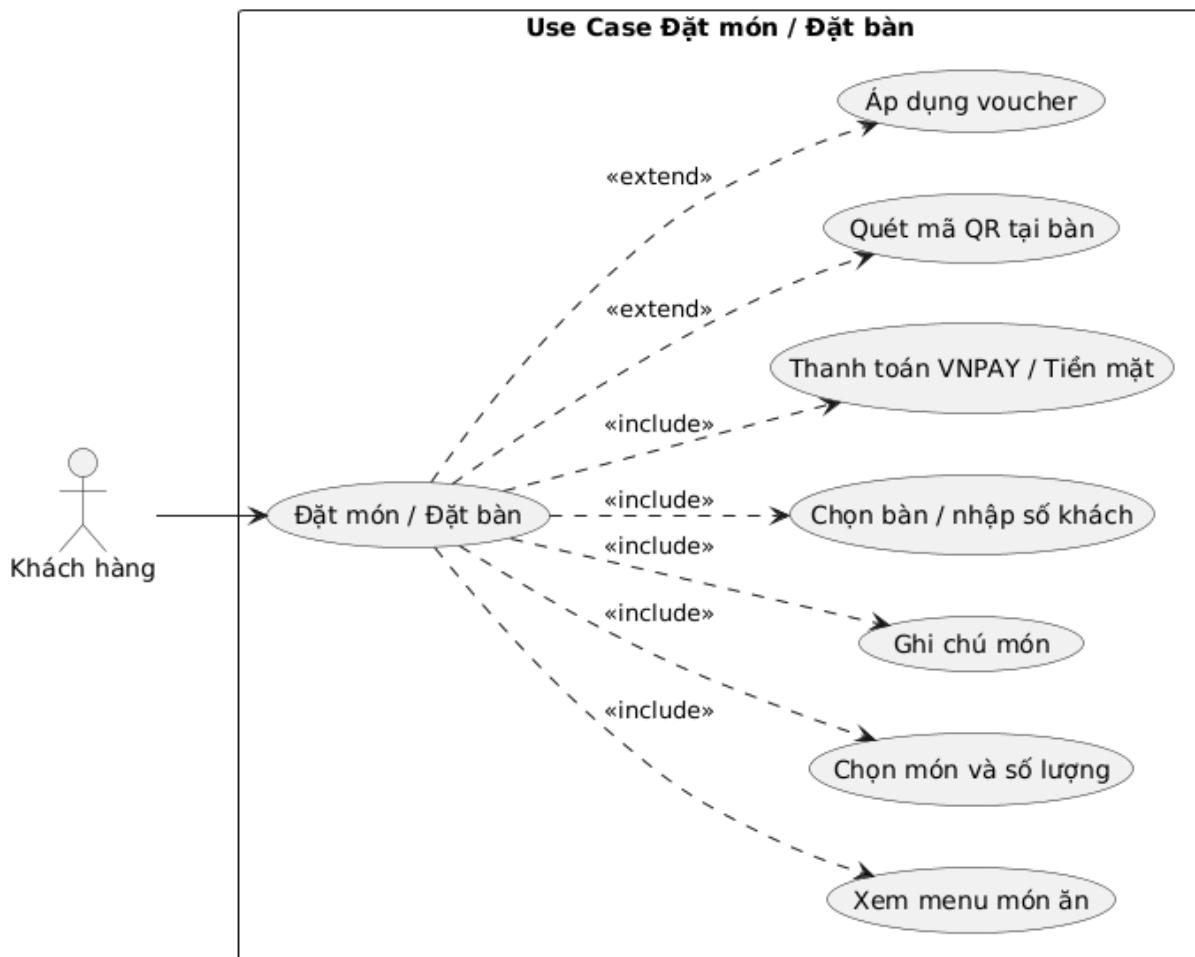
Hình 3-5: Đặc tả Usecase đăng nhập

Đặc tả Usecase đăng nhập

Bảng 3.4 Đặc tả Usecase đăng nhập

TT	Tên UC	Đăng nhập tài khoản
1	Tác nhân	User, Admin
2	Mô tả	Khách hàng nhập thông tin tài khoản (email/số điện thoại và mật khẩu) để truy cập vào hệ thống và sử dụng các chức năng dành cho người dùng đã xác thực.
3	Điều kiện trước	Người dùng phải có tài khoản của hệ thống được lưu trong database.
4	Luồng sự kiện chính	<ol style="list-style-type: none">Khách hàng truy cập trang đăng nhập.Hệ thống hiển thị form đăng nhập.Khách hàng nhập thông tin (email/sđt, mật khẩu).Khách hàng nhấp nút “Đăng nhập”.Hệ thống kiểm tra thông tin đăng nhập.Nếu thông tin hợp lệ, hệ thống tạo phiên đăng nhập và chuyển khách hàng về trang chủ / trang tài khoản.
5	Luồng sự kiện phụ	Nhập sai thông tin sẽ thông báo lỗi phần nhập sai và yêu cầu nhập lại hoặc nếu thiếu trường thì sẽ thông báo cho khách hàng nhập đầy đủ thông tin
6	Điều kiện sau	Nếu thành công thì chuyển hướng tới trang chủ, cho phép thực hiện các tác vụ có yêu cầu phải đăng nhập.
7	Yêu cầu đặc biệt	Không có.

3.4.2.3. Usecase đặt món/ đặt bàn



Hình 3-6: Đặc tả Usecase đặt món/ đặt bàn

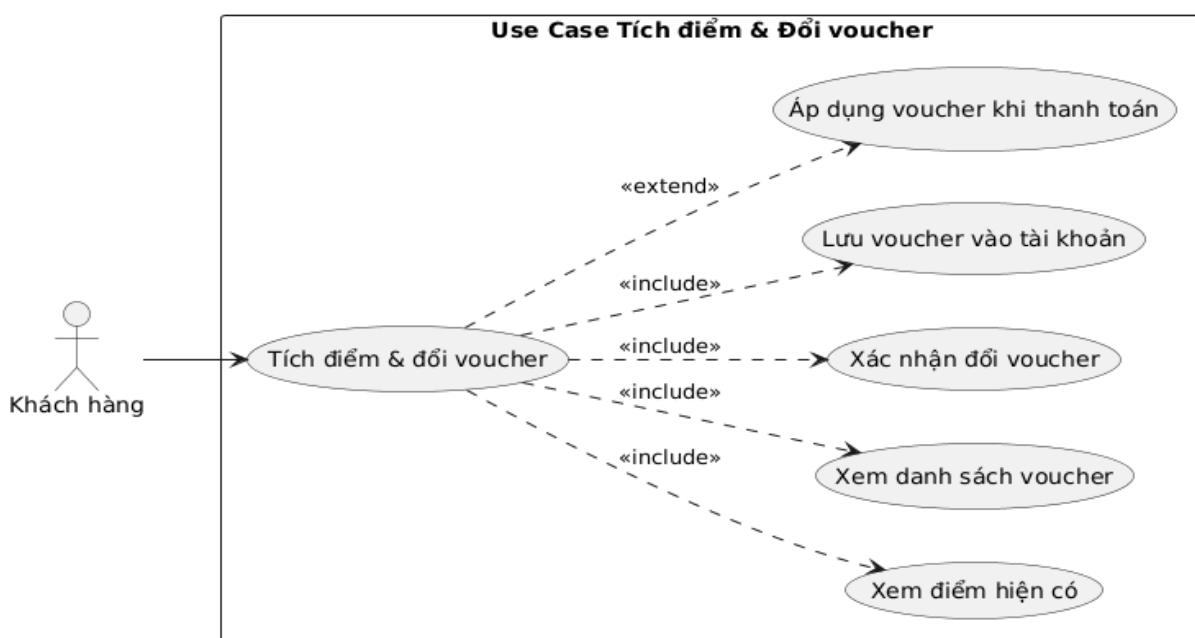
Đặc tả Usecase đặt món/ đặt bàn

Bảng 3.5 Đặc tả usecase đặt món/ đặt bàn

TT	Tên UC	Thêm sản phẩm- Admin
1	Tác nhân	Khách hàng
2	Mô tả	Khách hàng chọn món ăn, điều chỉnh số lượng, ghi chú và chọn bàn hoặc quét QR để đặt món trực tiếp tại bàn.

3	Điều kiện trước	Khách hàng đã đăng nhập vào hệ thống
4	Luồng sự kiện chính	<ol style="list-style-type: none"> Khách hàng xem menu, chọn món và thêm vào giỏ hàng Khách hàng nhập thời gian, khu vực và số lượng khách Khách hàng xác nhận đơn và gửi yêu cầu
5	Luồng sự kiện phụ	<ul style="list-style-type: none"> Khách quét QR tại bàn để đặt món nhanh. Khách dùng voucher giảm giá. Hệ thống báo lỗi nếu số lượng vượt tồn kho hoặc bàn đã được đặt.
6	Điều kiện sau	Admin xử lý và gửi email thông báo đến khách hàng
7	Yêu cầu đặc biệt	Tính toán tổng tiền theo voucher, phí dịch vụ, thuế.

3.4.2.4. Usecase tích điểm và đổi voucher

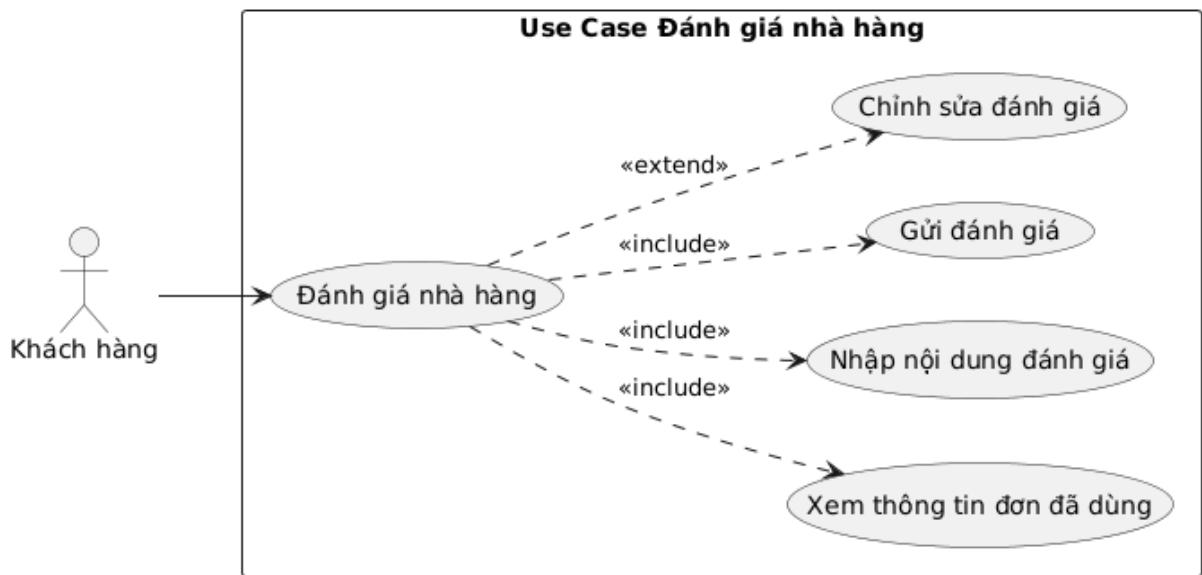


Hình 3-7: ĐẶC TẢ USECASE TÍCH ĐIỂM VÀ ĐỔI VOUCHER

Bảng 3.6 Đặc tả usecase tích điểm và đổi voucher

TT	Tên UC	Tích điểm và đổi voucher
1	Tác nhân	Khách hàng.
2	Mô tả	Khách hàng xem điểm tích lũy, chọn voucher cần đổi và sử dụng voucher trong các lần thanh toán.
3	Điều kiện trước	Khách hàng đã đăng nhập.
4	Luồng sự kiện chính	<ol style="list-style-type: none">Khách hàng xem điểm hiện có.Hệ thống hiển thị danh sách voucher có thể đổi.Khách hàng chọn voucher.Hệ thống kiểm tra điểm.Hệ thống trừ điểm và lưu voucher vào ví của khách hàng.
5	Luồng sự kiện phụ	Nếu không đủ điểm sẽ hiện thông báo.
6	Điều kiện sau	Voucher được thêm vào ví và khách hàng có thể sử dụng.
7	Yêu cầu đặc biệt	Không có

3.4.2.5. Usecase đánh giá



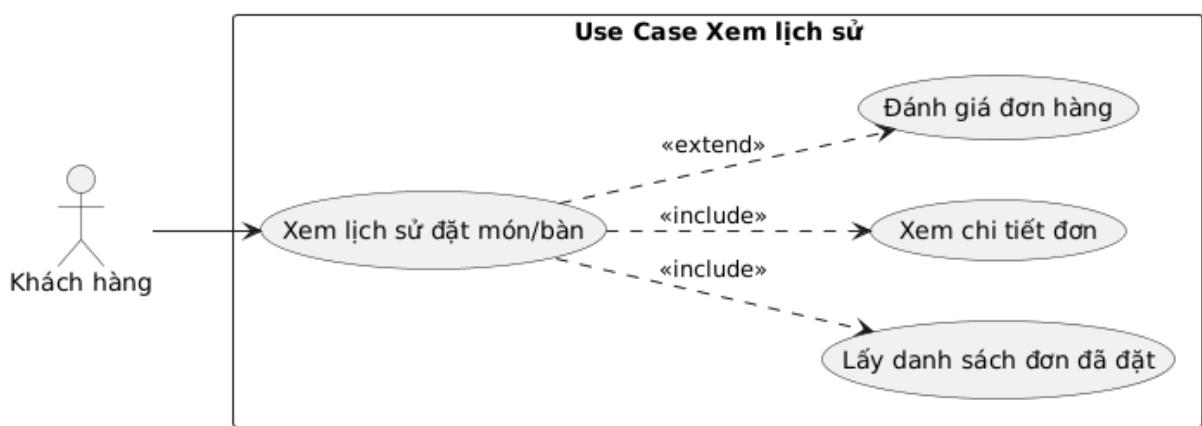
Hình 3-8: Đặc tả Usecase đánh giá

Bảng 3.7 Đặc tả usecase đánh giá

TT	Tên UC	Đánh giá
1	Tác nhân	Khách hàng.
2	Mô tả	Đánh giá nhà hàng sau khi sử dụng dịch vụ.
3	Điều kiện trước	Khách hàng đã đặt món trước đó.
4	Luồng sự kiện chính	<ul style="list-style-type: none"> 1. Khách hàng vào lịch sử đơn hàng. 2. Chọn đơn hàng và đánh giá. 3. Nhập nội dung và số sao hài lòng. 4. Chọn gửi đánh giá

5	Luồng sự kiện phụ	Hiển thị thông báo lỗi khi chưa đánh giá được.
6	Điều kiện sau	Đánh giá của khách hàng sẽ được cập nhật và hiển thị ở trang chủ.
7	Yêu cầu đặc biệt	Không có

3.4.2.6. Use case lịch sử đơn hàng



Hình 3-9: Đặc tả Usecase lịch sử đơn hàng

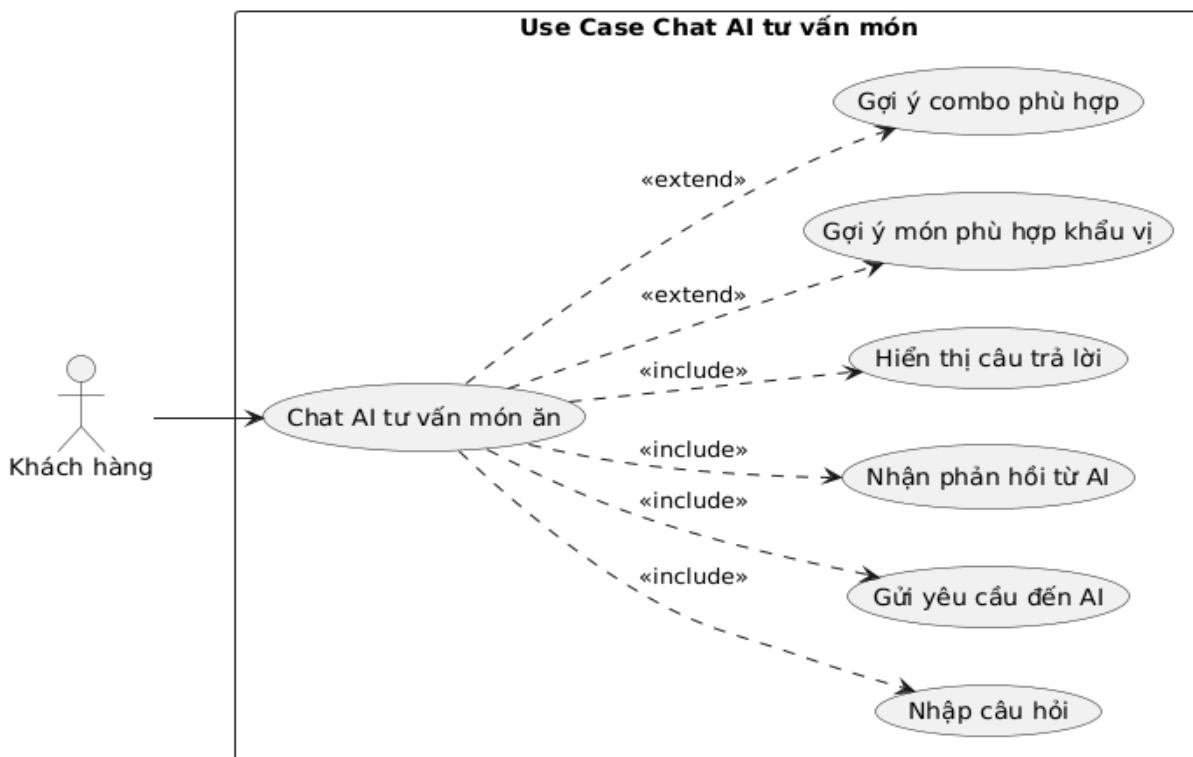
Đặc tả usecase lịch sử đơn hàng

Bảng 3.8 Đặc tả usecase lịch sử đơn hàng

TT	Tên UC	Lịch sử đơn hàng
1	Tác nhân	Khách hàng.
2	Mô tả	Khách hàng xem danh sách các đơn đã đặt (đơn đặt bàn, đặt món), kiểm tra chi tiết và trạng thái từng đơn.
3	Điều kiện trước	Khách hàng đã đăng nhập thành công.

4	Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Hệ thống hiển thị danh sách các đơn đã đặt. 2. Khách hàng chọn 1 đơn để xem chi tiết. 3. Hệ thống hiển thị đầy đủ thông tin đơn: món, giá, thời gian, trạng thái, hình thức thanh toán.
5	Luồng sự kiện phụ	<ul style="list-style-type: none"> • Nếu không có lịch sử → thông báo “Bạn chưa có đơn nào”. • Khách có thể chọn “Đánh giá” đơn đã hoàn thành.
6	Điều kiện sau	Khách hàng đã xem được thông tin chi tiết về đơn đã đặt.
7	Yêu cầu đặc biệt	Không có.

3.4.2.7. Usecase Chatbot ai tư vấn món

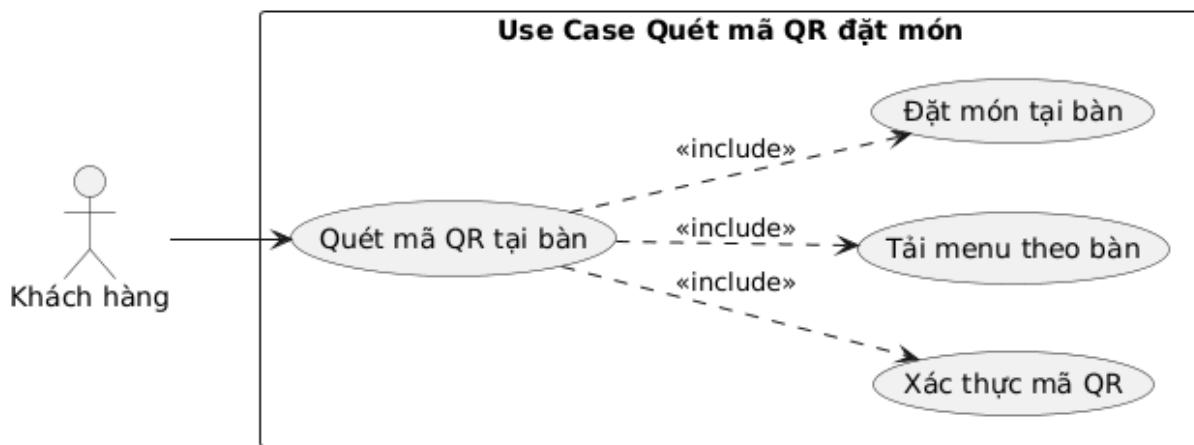


Hình 3-10: Đặc tả Usecase Chatbot ai tư vấn món

Bảng 3.9 Đặc tả usecase Chatbot tư vấn món

TT	Tên UC	Chatbot tư vấn món
1	Tác nhân	User.
2	Mô tả	Khách hàng đặt câu hỏi và nhận tư vấn tự động từ AI dựa trên dữ liệu món ăn, combo và khẩu vị.
3	Điều kiện trước	Khách hàng truy cập website, chatbot hoạt động.
4	Luồng sự kiện chính	<ol style="list-style-type: none">Khách nhập câu hỏi vào khung chat.Hệ thống gửi yêu cầu sang AI microservice.AI phân tích câu hỏi và dữ liệu món ăn.AI trả về câu trả lời.Website hiển thị nội dung phản hồi cho khách hàng.
5	Luồng sự kiện phụ	Nếu AI không hiểu câu hỏi → trả lời “Xin vui lòng mô tả rõ hơn...”.
6	Điều kiện sau	Khách hàng nhận được phản hồi AI.
7	Yêu cầu đặc biệt	Không có.

3.4.2.8. Usecase quét mã QR đặt món



Hình 3-11: Đặc tả Usecase quét mã QR đặt món

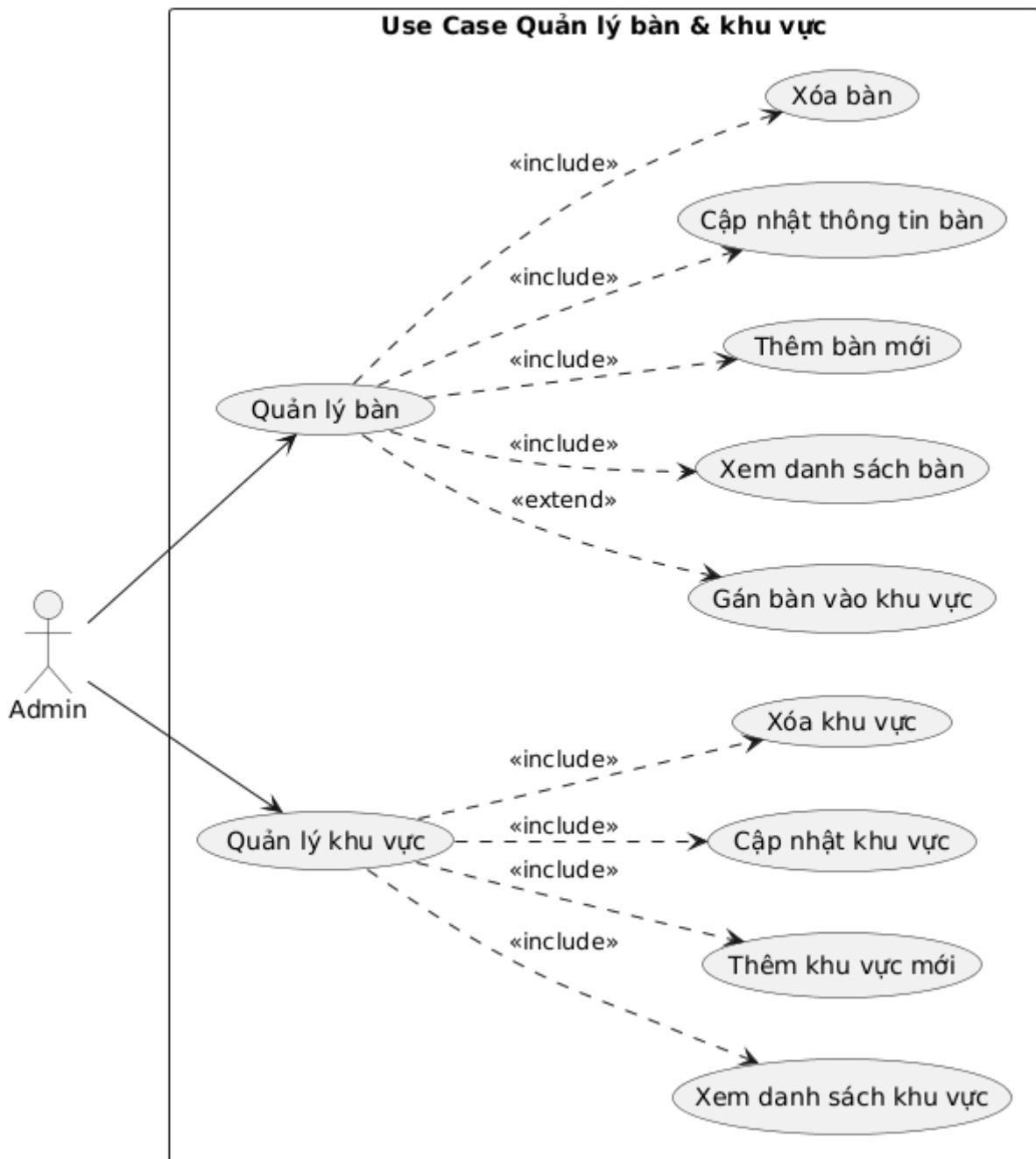
Đặc tả usecase quét mã QR đặt món

Bảng 3.10 Đặc tả usecase quét mã QR đặt món

TT	Tên UC	Quét mã QR đặt món
1	Tác nhân	Khách hàng tại nhà hàng.
2	Mô tả	Khách hàng dùng thiết bị di động để quét mã QR gắn trên từng bàn ăn để truy cập trang đặt món tương ứng với bàn đó.
3	Điều kiện trước	<ul style="list-style-type: none"> Bàn có mã QR hợp lệ. Hệ thống đã tạo QR cho từng bàn. Thiết bị khách có camera và internet.
4	Luồng sự kiện chính	<ol style="list-style-type: none"> Khách mở camera hoặc ứng dụng quét QR. Khách quét mã QR trên bàn. Hệ thống xác thực QR và nhận diện bàn. Hệ thống hiển thị menu tương ứng. Khách hàng chọn món và thêm vào giỏ. Gửi yêu cầu đặt món.
5	Luồng sự kiện phụ	<ul style="list-style-type: none"> QR không hợp lệ → thông báo lỗi.

		<ul style="list-style-type: none"> Mật mạng → yêu cầu thủ lại.
6	Điều kiện sau	Khách hàng đặt món thành công tại đúng bàn.
7	Yêu cầu đặc biệt	QR phải là duy nhất theo từng bàn.

3.4.2.9. Usecase quản lý bàn và khu vực



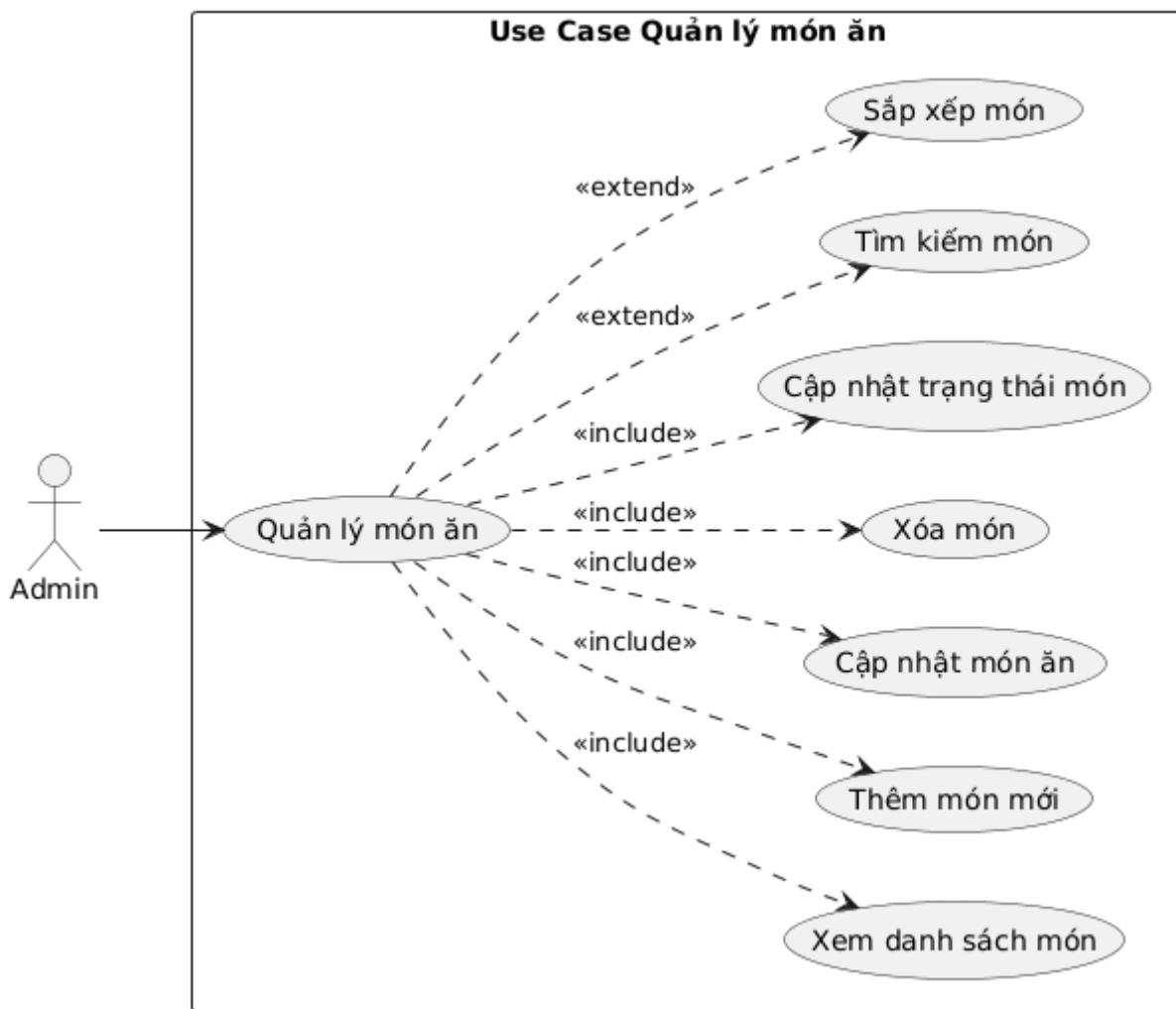
Hình 3-12: Đặc tả Usecase quản lý bàn và khu vực

Đặc tả usecase quản lý bàn và khu vực

Bảng 3.11 Đặc tả usecase quản lý bàn và khu vực

TT	Tên UC	Quản lý bàn và khu vực
1	Tác nhân	Admin.
2	Mô tả	Admin quản lý toàn bộ bàn ăn và khu vực trong nhà hàng: tạo mới, chỉnh sửa, xóa, sắp xếp và cập nhật trạng thái.
3	Điều kiện trước	Admin đã đăng nhập vào hệ thống quản trị.
4	Luồng sự kiện chính	<p>Quản lý bàn:</p> <ol style="list-style-type: none">1. Xem danh sách bàn.2. Thêm bàn mới (số bàn, QR, khu vực).3. Cập nhật thông tin bàn.4. Xóa bàn. <p>Quản lý khu vực:</p> <ol style="list-style-type: none">5. Xem danh sách khu vực.6. Thêm khu vực mới.7. Cập nhật tên khu vực.8. Xóa khu vực.
5	Luồng sự kiện phụ	Hiện thông báo lỗi khi thao tác quản lý.
6	Điều kiện sau	Dữ liệu bàn & khu vực được cập nhật đầy đủ trong hệ thống.
7	Yêu cầu đặc biệt	Mỗi khu vực và bàn đều có tên duy nhất.

3.4.2.10. Usecase quản lý món



Hình 3-13: Đặc tả Usecase quản lý món ăn

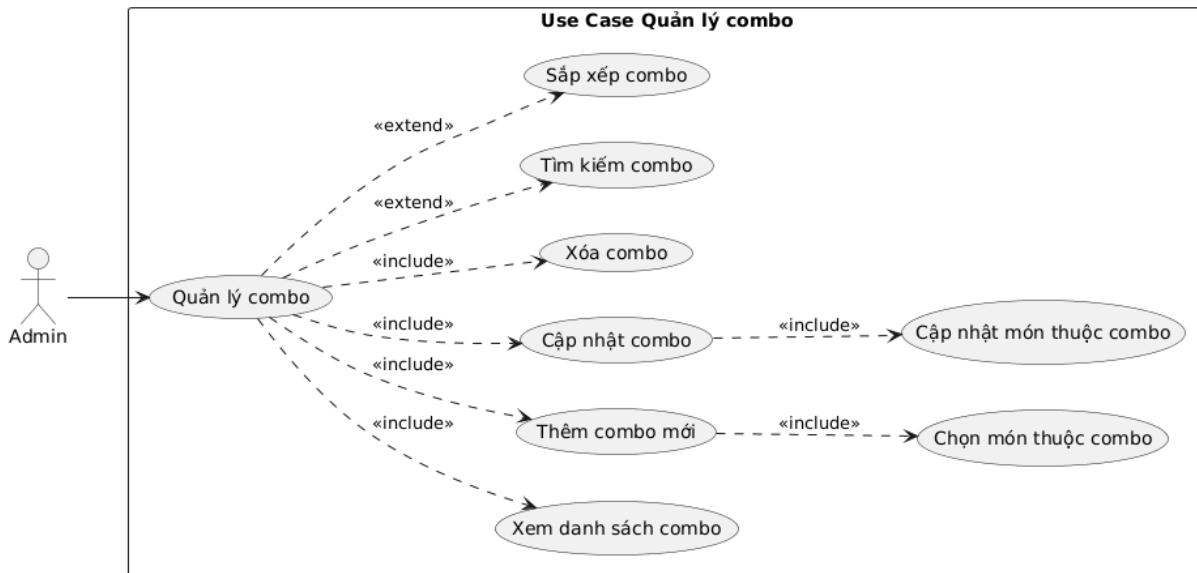
Đặc tả usecase quản lý món

Bảng 3.12 Đặc tả usecase quản lý món

TT	Tên UC	Quản lý món
1	Tác nhân	Admin.
2	Mô tả	Khách hàng dùng thiết bị di động để quét mã QR gắn trên từng bàn ăn để truy cập trang đặt món tương ứng với bàn đó.
3	Điều kiện trước	Admin đã đăng nhập hệ thống quản trị.

4	Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Admin xem danh sách món ăn. 2. Nhấn thêm món mới và nhập thông tin. 3. Chính sửa thông tin món khi cần. 4. Xóa món nếu không còn kinh doanh. 5. Cập nhật trạng thái (hiển thị / ngừng kinh doanh).
5	Luồng sự kiện phụ	<ul style="list-style-type: none"> • Tìm kiếm món theo tên hoặc danh mục (extend). • Sắp xếp món theo giá, ngày tạo, hoặc trạng thái (extend). • Xóa món ăn chỉ khi không thuộc combo.
6	Điều kiện sau	Danh sách món ăn được cập nhật hoàn chỉnh trong hệ thống.
7	Yêu cầu đặc biệt	Hình ảnh món phải được upload hợp lệ; Giá món không thể âm.

3.4.2.11. Usecase quản lý combo



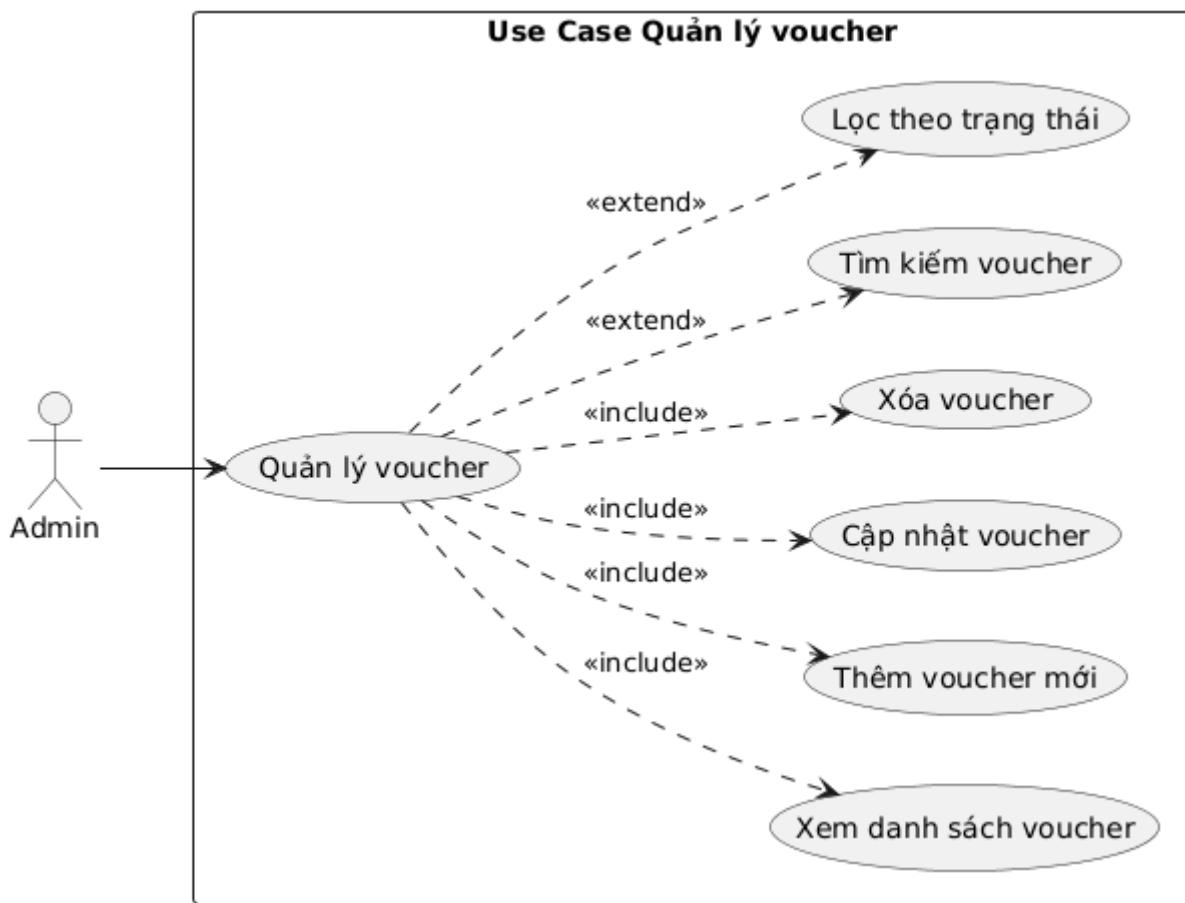
Hình 3-14: ĐẶC TẢ USECASE QUẢN LÝ COMBO

ĐẶC TẢ USECASE QUẢN LÝ COMBO

Bảng 3.13 ĐẶC TẢ USECASE QUẢN LÝ COMBO

TT	Tên UC	Quản lý combo
1	Tác nhân	Admin.
2	Mô tả	Admin quản lý các combo khuyến mãi gồm nhiều món ăn: tạo combo mới, chỉnh sửa thông tin combo, cập nhật món trong combo và xóa combo khi cần thiết.
3	Điều kiện trước	Admin đã đăng nhập hệ thống quản trị.
4	Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Xem danh sách combo. 2. Thêm combo mới (tên, mô tả, giá, hình ảnh). 3. Chọn các món cầu thành combo. 4. Cập nhật combo (sửa thông tin + thay đổi món thuộc combo). 5. Xóa combo nếu không còn áp dụng.
5	Luồng sự kiện phụ	<ul style="list-style-type: none"> • Tìm kiếm combo theo tên. • Sắp xếp combo theo giá, ngày tạo, trạng thái.
6	Điều kiện sau	Danh sách combo được cập nhật đầy đủ.
7	Yêu cầu đặc biệt	Combo phải có ít nhất một món ăn; Giá combo không được thấp hơn 0.

3.4.2.12. Usecase quản lý voucher



Hình 3-15: ĐẶC TẢ USECASE QUẢN LÝ VOUCHER

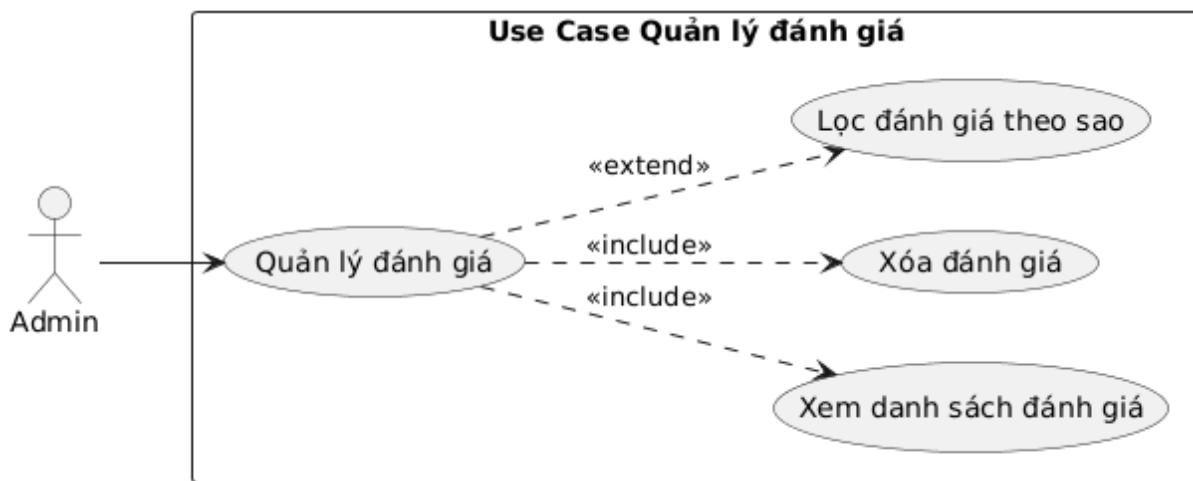
ĐẶC TẢ USECASE QUẢN LÝ VOUCHER

Bảng 3.14 ĐẶC TẢ USECASE QUẢN LÝ VOUCHER

TT	TÊN UC	QUẢN LÝ VOUCHER
1	Tác nhân	Admin.
2	Mô tả	Admin quản lý toàn bộ voucher giảm giá: tạo, chỉnh sửa, xóa, tìm kiếm và lọc theo trạng thái.
3	Điều kiện trước	Admin đã đăng nhập hệ thống quản trị.

4	Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Admin xem danh sách voucher. 2. Tạo voucher mới (theo % hoặc số tiền cố định). 3. Cập nhật thông tin voucher: giá trị, ngày hết hạn, điều kiện áp dụng. 4. Xóa voucher nếu không còn sử dụng.
5	Luồng sự kiện phụ	<ul style="list-style-type: none"> • Tìm kiếm voucher theo mã hoặc giá trị. • Lọc voucher theo trạng thái.
6	Điều kiện sau	Danh sách voucher được cập nhật đầy đủ và chính xác.
7	Yêu cầu đặc biệt	Đảm bảo mã voucher là duy nhất.

3.4.2.13. Usecase quản lý đánh giá



Hình 3-16: ĐẶC TẢ USECASE QUẢN LÝ ĐÁNH GIÁ

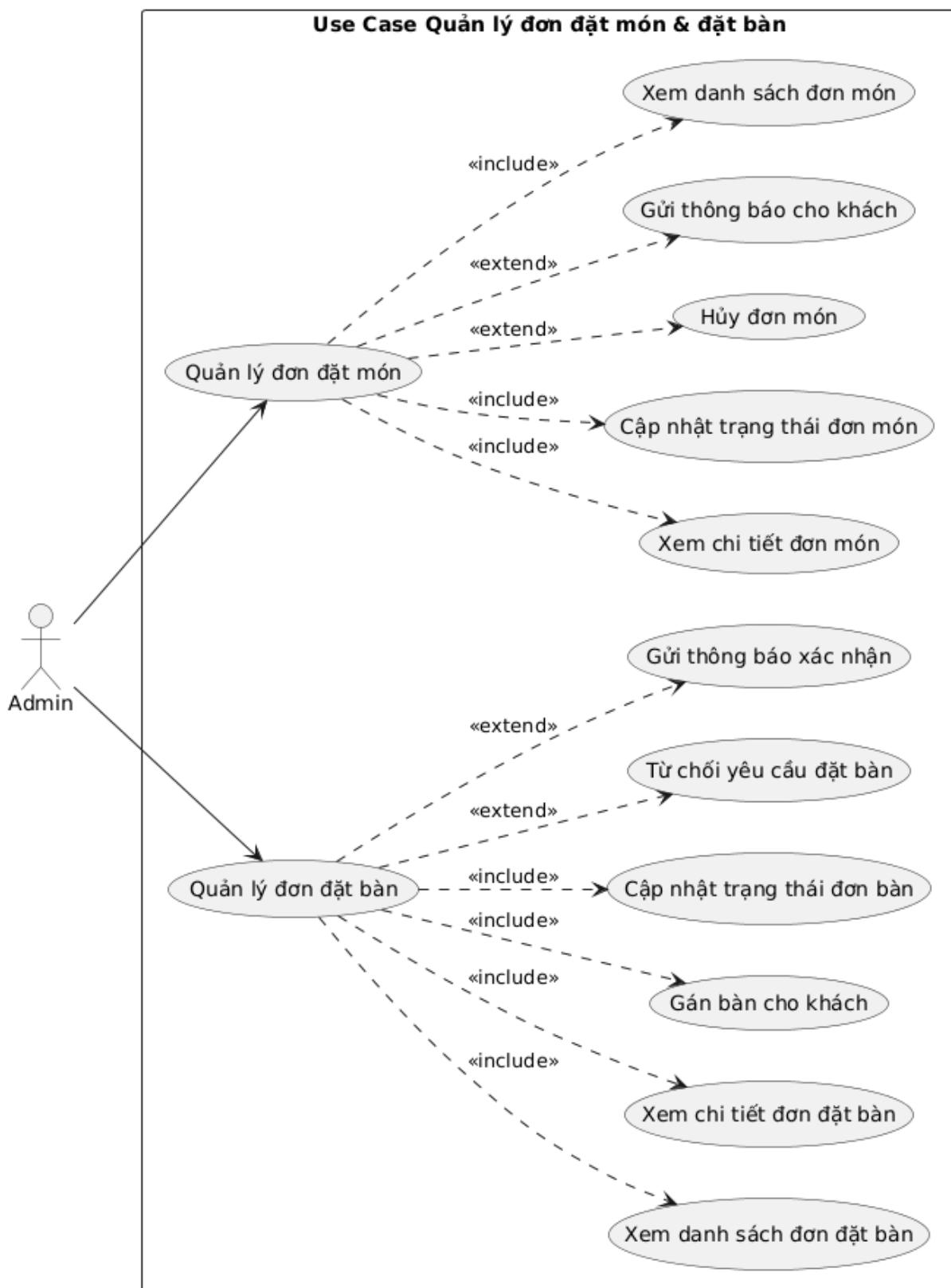
ĐẶC TẢ USECASE QUẢN LÝ ĐÁNH GIÁ

Bảng 3.15 ĐẶC TẢ USECASE QUẢN LÝ ĐÁNH GIÁ

TT	Tên UC	Quản lý đánh giá
1	Tác nhân	Admin.

2	Mô tả	Admin có thể xem, kiểm duyệt và xóa các đánh giá của khách hàng nếu không phù hợp (spam, nội dung không đúng sự thật, ngôn từ vi phạm).
3	Điều kiện trước	Admin đã đăng nhập hệ thống quản trị.
4	Luồng sự kiện chính	<ol style="list-style-type: none">5. Admin xem danh sách đánh giá.6. Xóa đánh giá không phù hợp.
5	Luồng sự kiện phụ	Lọc đánh giá theo sao.
6	Điều kiện sau	Dữ liệu đánh giá được kiểm duyệt, đảm bảo nội dung phù hợp và minh bạch.
7	Yêu cầu đặc biệt	Không chỉnh sửa nội dung đánh giá của khách hàng.

3.4.2.14. Usecase quản lý đơn đặt món/ đặt bàn



Hình 3-17: ĐẶC TẢ USECASE QUẢN LÝ ĐƠN ĐẶT MÓN VÀ ĐẶT BÀN

Đặc tả usecase quản lý đặt món/ đặt bàn

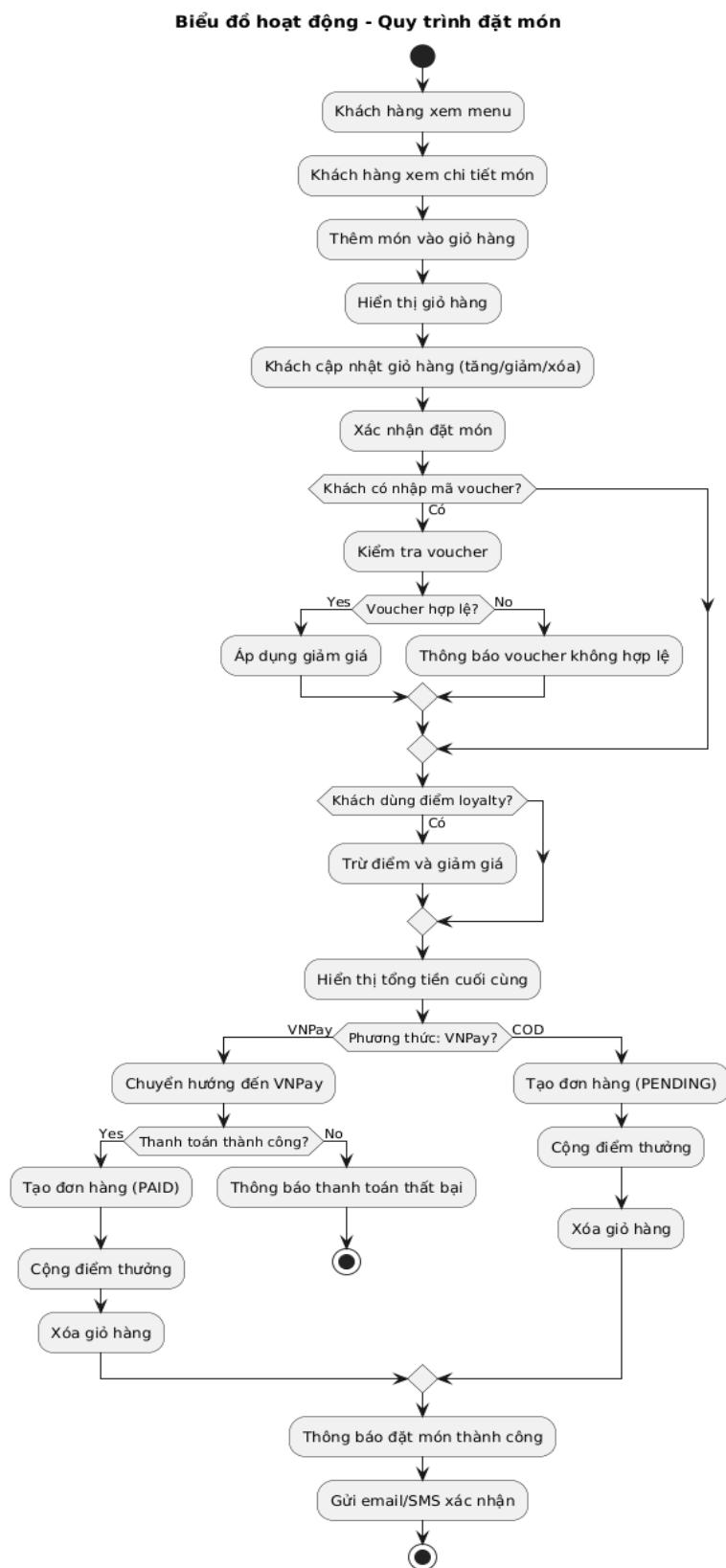
Bảng 3.16 Đặc tả usecase quản lý đặt món/ đặt bàn

TT	Tên UC	Quản lý đặt món/ đặt bàn
1	Tác nhân	Admin.
2	Mô tả	Admin quản lý toàn bộ đơn đặt món và đơn đặt bàn của khách: theo dõi, duyệt, gán bàn, cập nhật trạng thái và gửi thông báo cho khách hàng.
3	Điều kiện trước	Admin đã đăng nhập hệ thống quản trị.
4	Luồng sự kiện chính	<p>Đơn đặt món:</p> <ol style="list-style-type: none"> 1. Xem danh sách đơn món. 2. Xem chi tiết từng đơn. 3. Cập nhật trạng thái (đang xử lý → hoàn tất). <p>Đơn đặt bàn:</p> <ol style="list-style-type: none"> 4. Xem danh sách yêu cầu đặt bàn. 5. Xem thông tin chi tiết. 6. Gán bàn cho khách theo thời gian yêu cầu. 7. Cập nhật trạng thái (đã xác nhận, đã đến, đã hủy).
5	Luồng sự kiện phụ	<ul style="list-style-type: none"> • Hủy đơn nếu khách hủy hoặc nhà hàng không thể phục vụ (extend). • Gửi thông báo xác nhận / từ chối qua email hoặc hệ thống (extend). • Từ chối yêu cầu đặt bàn khi quá giờ hoặc không còn bàn trống.
6	Điều kiện sau	Danh sách đơn món và đơn bàn được cập nhật đầy đủ và chính xác.

7	Yêu cầu đặc biệt	Hệ thống phải đồng bộ với trạng thái bàn; Không cho phép gán bàn đã được đặt trước. Giá món không thể âm.
---	------------------	---

3.5. Biểu đồ hoạt động

3.5.1. Biểu đồ hoạt động chức năng đặt món

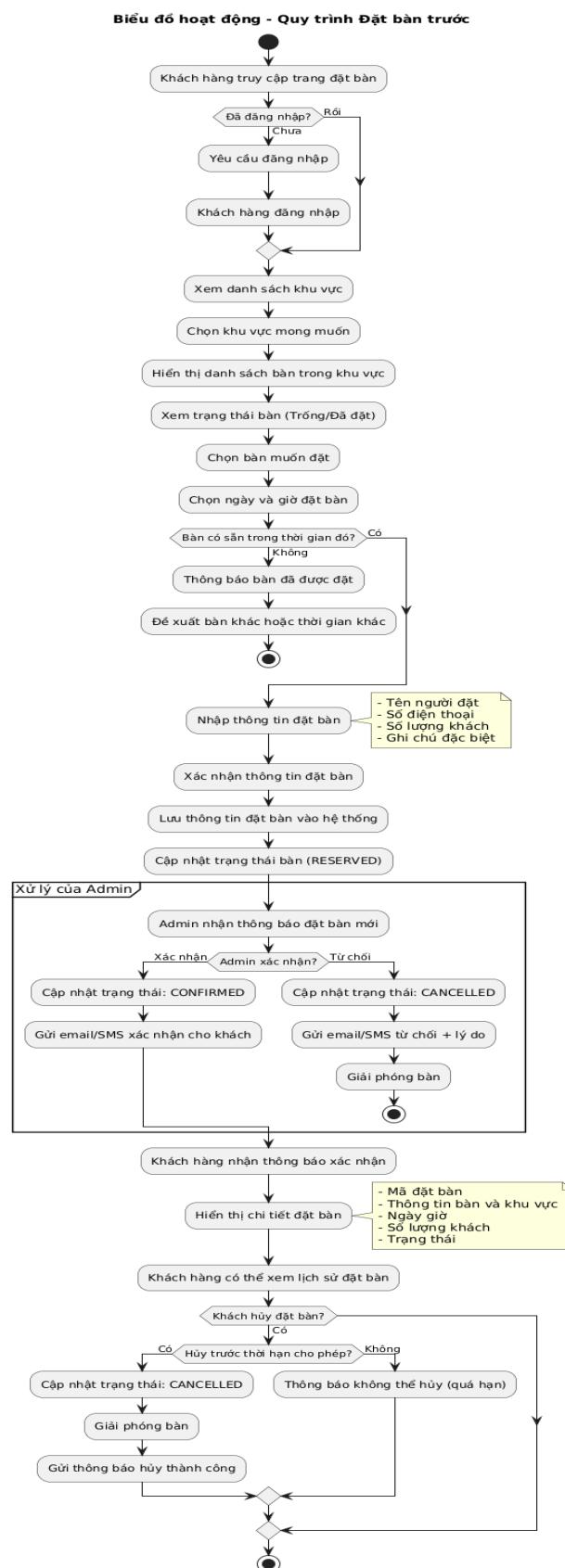


Hình 3-18: Biểu đồ hoạt động đặt món

Biểu đồ này mô tả quy trình khi khách hàng đặt món trong hệ thống quản lý nhà hàng. Toàn bộ luồng xử lý được mô tả như sau:

- Quy trình bắt đầu khi khách hàng truy cập vào trang menu để xem danh sách món ăn và combo có sẵn. Khách hàng xem chi tiết món ăn bao gồm hình ảnh, mô tả, giá cả và đánh giá từ khách hàng khác.
- Sau khi lựa chọn món, khách hàng thêm món vào giỏ hàng với số lượng mong muốn. Hệ thống hiển thị giỏ hàng với danh sách các món đã chọn và tổng tiền tạm tính. Khách hàng có thể cập nhật giỏ hàng bằng cách tăng/giảm số lượng hoặc xóa món không mong muốn.
- Khi khách hàng xác nhận đặt món, hệ thống kiểm tra trạng thái đăng nhập. Nếu chưa đăng nhập, quy trình yêu cầu khách hàng đăng nhập vào hệ thống. Nếu đã đăng nhập, khách hàng tiếp tục đến bước thanh toán.
- Khách hàng nhập thông tin giao hàng gồm tên người nhận, số điện thoại, địa chỉ và ghi chú (nếu có). Sau đó, khách hàng có thể áp dụng mã voucher (nếu có). Hệ thống kiểm tra voucher: nếu voucher hợp lệ thì áp dụng giảm giá, nếu không hợp lệ thì thông báo lỗi và yêu cầu nhập lại.
- Tiếp theo, khách hàng có thể chọn sử dụng điểm loyalty để giảm giá đơn hàng. Hệ thống trừ điểm và áp dụng giảm giá tương ứng.
- Hệ thống hiển thị tổng tiền cuối cùng sau khi đã áp dụng các ưu đãi. Khách hàng chọn phương thức thanh toán: VNPay hoặc thanh toán tiền mặt (COD).
- Nếu chọn VNPay: Hệ thống chuyển hướng đến cổng thanh toán VNPay. Nếu thanh toán thành công, hệ thống tạo đơn hàng với trạng thái PAID, cộng điểm thưởng cho khách hàng và xóa giỏ hàng. Nếu thanh toán thất bại, hệ thống thông báo lỗi và quy trình kết thúc.
- Nếu chọn COD: Hệ thống tạo đơn hàng với trạng thái PENDING (chờ thanh toán khi nhận hàng), cộng điểm thưởng và xóa giỏ hàng.
- Sau khi tạo đơn hàng thành công, hệ thống thông báo đặt món thành công và gửi email/SMS xác nhận chi tiết đơn hàng đến khách hàng. Khách hàng có thể xem chi tiết đơn hàng và theo dõi trạng thái giao hàng trong mục lịch sử đơn hàng.

3.5.2. Biểu đồ hoạt động chức năng đặt bàn



Hình 3-19: Biểu đồ hoạt động đặt bàn

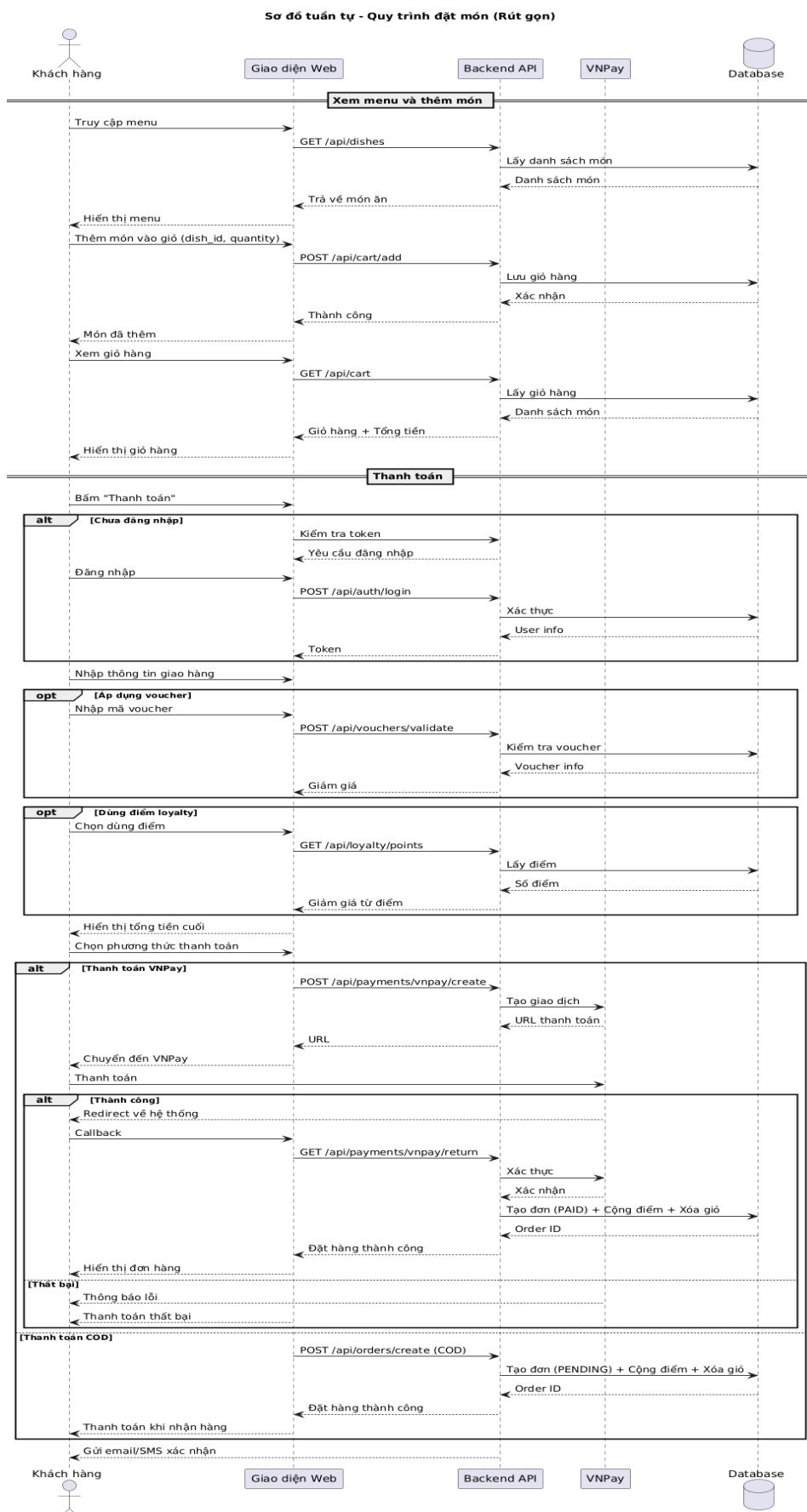
Biểu đồ này mô tả quy trình khi khách hàng đặt bàn trước trong hệ thống quản lý nhà hàng. Toàn bộ luồng xử lý được mô tả như sau:

- Quy trình bắt đầu khi khách hàng truy cập vào trang đặt bàn để tìm và đặt chỗ trước khi đến nhà hàng. Hệ thống kiểm tra trạng thái đăng nhập: nếu khách hàng chưa đăng nhập, hệ thống yêu cầu đăng nhập trước khi tiếp tục. Nếu đã đăng nhập, khách hàng có thể tiến hành đặt bàn.
- Khách hàng xem danh sách khu vực có sẵn trong nhà hàng (VIP, tầng 1, tầng 2, ngoài trời, v.v.) và chọn khu vực mong muốn. Hệ thống hiển thị danh sách bàn trong khu vực đó kèm theo trạng thái bàn (Trống hoặc Đã đặt).
- Khách hàng chọn bàn muốn đặt và chọn ngày giờ đặt bàn. Hệ thống kiểm tra tình trạng bàn trong thời gian đó: nếu bàn đã được đặt, hệ thống thông báo lỗi và đề xuất bàn khác hoặc thời gian khác, sau đó quy trình kết thúc. Nếu bàn còn trống, khách hàng tiếp tục bước tiếp theo.
- Khách hàng nhập thông tin đặt bàn gồm tên người đặt, số điện thoại, số lượng khách và ghi chú đặc biệt (nếu có). Sau khi kiểm tra lại thông tin, khách hàng xác nhận đặt bàn.
- Hệ thống lưu thông tin đặt bàn vào cơ sở dữ liệu và cập nhật trạng thái bàn thành RESERVED (Đã đặt trước). Đồng thời, hệ thống gửi thông báo đến Admin về yêu cầu đặt bàn mới.
- Admin nhận thông báo và xem xét yêu cầu đặt bàn. Admin có thể xác nhận hoặc từ chối đặt bàn:
 - Nếu Admin xác nhận: Hệ thống cập nhật trạng thái đặt bàn thành CONFIRMED (Đã xác nhận) và gửi email/SMS xác nhận đến khách hàng với chi tiết đặt bàn gồm mã đặt bàn, thông tin bàn và khu vực, ngày giờ, số lượng khách và trạng thái.
 - Nếu Admin từ chối: Hệ thống cập nhật trạng thái đặt bàn thành CANCELLED (Đã hủy), giải phóng bàn và gửi email/SMS thông báo từ chối kèm lý do đến khách hàng. Quy trình kết thúc.
- Sau khi nhận được xác nhận, khách hàng có thể xem lịch sử đặt bàn trong tài khoản cá nhân để theo dõi trạng thái và chi tiết đặt bàn.
- Khách hàng có quyền hủy đặt bàn nếu có sự thay đổi kế hoạch. Hệ thống kiểm tra thời gian hủy: nếu khách hàng hủy trước thời hạn cho phép (ví dụ: trước 2 giờ), hệ thống cập nhật trạng thái thành CANCELLED, giải phóng bàn để khách khác có thể đặt và gửi thông báo hủy thành công. Nếu khách hàng hủy quá thời hạn, hệ thống thông báo không thể hủy và quy trình kết thúc.

3.6. Biểu đồ tuần tự

3.6.1. Biểu đồ tuần tự đặt món

Xây dựng website đặt bàn, đặt món cho nhà hàng thực dưỡng Hoa Ban sử dụng Vue.js, Spring Boot và PostgreSQL



Hình 3-20: Biểu đồ tuần tự đặt món

Khách hàng muốn thực hiện chức năng đặt món thì thực hiện như sau:

- (1) Khách hàng truy cập trang menu trên giao diện Vue.js.
- (2) Hệ thống frontend gửi yêu cầu lấy danh sách món ăn đến Backend (GET /api/dishes).
- (3) Backend nhận yêu cầu và truy vấn Database để lấy danh sách món ăn.
- (4) Database trả về danh sách món ăn kèm thông tin (tên, giá, hình ảnh, mô tả).
- (5) Backend gửi dữ liệu món ăn về Frontend.
- (6) Frontend hiển thị menu món ăn cho khách hàng.
- (7) Khách hàng chọn món và nhấn nút "Thêm vào giỏ hàng" (kèm dish_id, quantity).
- (8) Frontend gửi API thêm món đến Backend (POST /api/cart/add) kèm theo dữ liệu: dish_id, quantity, user_id (nếu đã đăng nhập).
- (9) Backend xử lý yêu cầu và lưu món vào giỏ hàng trong Database.
- (10) Database trả về xác nhận thêm món thành công.
- (11) Backend phản hồi lại Frontend "Thêm món thành công".
- (12) Frontend hiển thị thông báo và cập nhật số lượng giỏ hàng.
- (13) Khách hàng nhấn "Xem giỏ hàng".
- (14) Frontend gửi GET /api/cart đến Backend.
- (15) Backend truy vấn Database để lấy danh sách món trong giỏ.
- (16) Database trả về danh sách món và tính tổng tiền.
- (17) Backend trả dữ liệu giỏ hàng về Frontend.
- (18) Khách hàng xem giỏ hàng và nhấn "Thanh toán".
- (19) Frontend kiểm tra trạng thái đăng nhập. Nếu chưa đăng nhập, yêu cầu khách hàng đăng nhập.
- (20) Khách hàng nhập email, password và nhấn "Đăng nhập".
- (21) Frontend gửi API đăng nhập đến Backend (POST /api/auth/login).
- (22) Backend xác thực thông tin với Database.
- (23) Database trả về thông tin user nếu hợp lệ.
- (24) Backend trả về token và thông tin user cho Frontend.
- (25) Frontend lưu token và hiển thị trang thanh toán.
- (26) Khách hàng nhập thông tin giao hàng (tên, số điện thoại, địa chỉ, ghi chú).
- (27) Khách hàng nhập mã voucher (nếu có) và nhấn "Áp dụng".
- (28) Frontend gửi API validate voucher đến Backend (POST /api/vouchers/validate) kèm mã voucher.
- (29) Backend kiểm tra voucher trong Database.
- (30) Database trả về thông tin voucher (giảm giá %, số tiền, điều kiện).
- (31) Backend tính toán giảm giá và trả về Frontend.
- (32) Frontend hiển thị giá sau khi giảm.

(33) Khách hàng chọn phương thức thanh toán (VNPay hoặc COD).

(34) Nếu chọn VNPay:

- Frontend gửi API tạo thanh toán đến Backend
(POST /api/payments/vnpay/create) kèm tổng tiền, thông tin đơn hàng.
- Backend gửi yêu cầu tạo giao dịch đến VNPay.
- VNPay trả về URL thanh toán.
- Backend gửi URL về Frontend.
- Frontend chuyển hướng khách hàng đến trang VNPay.
- Khách hàng nhập thông tin thẻ/tài khoản và xác nhận thanh toán.
- VNPay xử lý thanh toán và chuyển hướng về hệ thống (callback URL).
- Frontend gửi API xác nhận thanh toán đến Backend
(GET/api/payments/vnpay/return) kèm params từ VNPay.
- Backend xác thực chữ ký với VNPay.
- VNPay xác nhận giao dịch thành công.

(35) Backend tạo đơn hàng trong Database với trạng thái PAID (nếu VNPay) hoặc PENDING (nếu COD).

(36) Database trả về order_id sau khi lưu thành công.

(37) Backend cộng điểm loyalty cho khách hàng và lưu vào Database.

(38) Backend xóa giỏ hàng của khách trong Database.

(39) Backend gửi email/SMS xác nhận đơn hàng cho khách hàng.

(40) Backend phản hồi lại Frontend "Đặt hàng thành công" kèm order_id.

(41) Frontend hiển thị trang "Đặt hàng thành công" với thông tin đơn hàng.

(42) Khách hàng nhấn "Xem chi tiết đơn hàng".

(43) Frontend gửi GET /api/orders/{id} đến Backend.

(44) Backend truy vấn Database để lấy thông tin chi tiết đơn hàng.

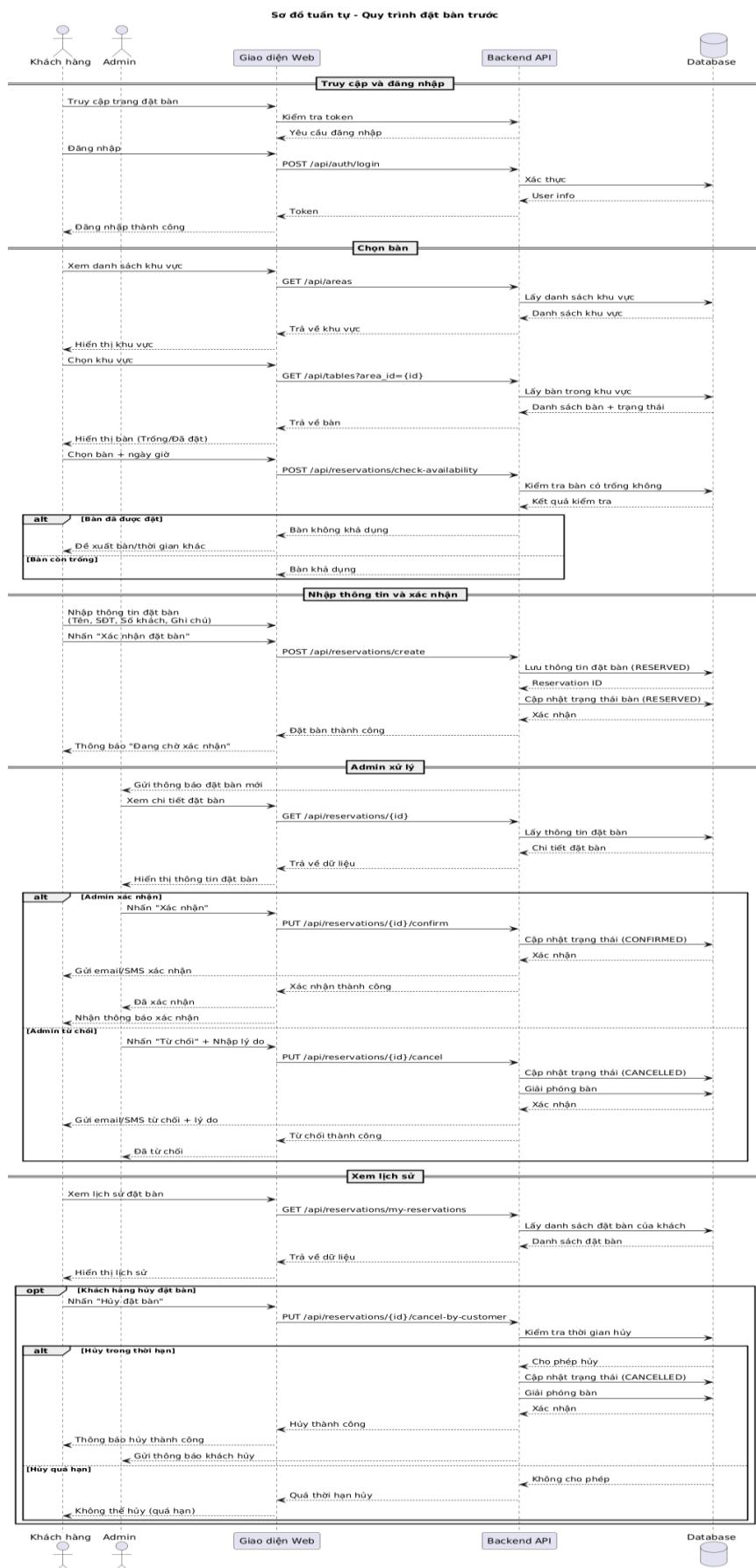
(45) Database trả dữ liệu đơn hàng (món ăn, giá, trạng thái, địa chỉ giao hàng).

(46) Backend trả dữ liệu về Frontend.

(47) Khách hàng xem toàn bộ thông tin đơn hàng và trạng thái giao hàng.

3.6.2. Biểu đồ tuần tự đặt bàn

Xây dựng website đặt bàn, đặt món cho nhà hàng thực dưỡng Hoa Ban sử dụng Vue.js, Spring Boot và PostgreSQL



Hình 3-21: Biểu đồ tuần tự đặt bàn

Khách hàng muốn thực hiện chức năng đặt bàn trước thì thực hiện như sau:

- (1) Khách hàng truy cập trang đặt bàn trên giao diện Vue.js.
- (2) Hệ thống frontend gửi yêu cầu kiểm tra token đến Backend.
- (3) Backend kiểm tra và phát hiện khách chưa đăng nhập.
- (4) Frontend hiển thị yêu cầu đăng nhập.
- (5) Khách hàng nhập email và password, sau đó nhấn "Đăng nhập".
- (6) Frontend gửi API đăng nhập đến Backend
(POST /api/auth/login) kèm email và password.
- (7) Backend xác thực thông tin với Database.
- (8) Database trả về thông tin user nếu hợp lệ.
- (9) Backend trả về token và thông tin user cho Frontend.
- (10) Frontend lưu token và chuyển đến trang đặt bàn.
- (11) Khách hàng xem danh sách khu vực.
- (12) Frontend gửi GET /api/areas đến Backend.
- (13) Backend truy vấn Database để lấy danh sách khu vực (VIP, tầng 1, tầng 2, ngoài trời).
- (14) Database trả về danh sách khu vực.
- (15) Backend gửi dữ liệu về Frontend.
- (16) Frontend hiển thị danh sách khu vực cho khách hàng.
- (17) Khách hàng chọn khu vực mong muốn.
- (18) Frontend gửi GET /api/tables?area_id={id} đến Backend.
- (19) Backend truy vấn Database để lấy danh sách bàn trong khu vực đó kèm trạng thái.
- (20) Database trả về danh sách bàn và trạng thái (Trống/Đã đặt).
- (21) Backend gửi dữ liệu về Frontend.
- (22) Frontend hiển thị danh sách bàn với trạng thái cho khách hàng.
- (23) Khách hàng chọn bàn và chọn ngày giờ đặt bàn.
- (24) Frontend gửi API kiểm tra tình trạng bàn đến Backend
(POST /api/reservations/check-availability) kèm table_id, date, time.
- (25) Backend kiểm tra trong Database xem bàn có bị đặt trong khung giờ đó không.
- (26) Database trả về kết quả kiểm tra.
- (27) Nếu bàn đã được đặt: Backend trả về "Bàn không khả dụng", Frontend hiển thị thông báo và đề xuất bàn khác hoặc thời gian khác. Quy trình dừng lại.
- (28) Nếu bàn còn trống: Backend trả về "Bàn khả dụng", khách hàng tiếp tục bước tiếp theo.
- (29) Khách hàng nhập thông tin đặt bàn (tên người đặt, số điện thoại, số lượng khách, ghi chú đặc biệt).

- (30) Khách hàng nhấn nút "Xác nhận đặt bàn".
- (31) Frontend gửi API tạo đặt bàn đến Backend (POST /api/reservations/create) kèm theo dữ liệu: table_id, date, time, customer_name, phone, guest_count, notes.
- (32) Backend xử lý yêu cầu và lưu thông tin đặt bàn vào Database với trạng thái RESERVED.
- (33) Database trả về reservation_id sau khi lưu thành công.
- (34) Backend cập nhật trạng thái bàn thành RESERVED trong Database.
- (35) Database xác nhận cập nhật thành công.
- (36) Backend phản hồi lại Frontend "Đặt bàn thành công, đang chờ xác nhận".
- (37) Frontend hiển thị thông báo "Đặt bàn thành công, vui lòng chờ xác nhận từ nhà hàng".
- (38) Backend lưu thông tin đặt bàn thành công và trả về kết quả cho khách hàng. Admin sẽ thấy đặt bàn mới khi truy cập trang quản lý đặt bàn hoặc Dashboard.
- (39) Admin nhận thông báo và nhấn xem chi tiết đặt bàn.
- (40) Frontend gửi GET /api/reservations/{id} đến Backend.
- (41) Backend truy vấn Database để lấy thông tin chi tiết đặt bàn.
- (42) Database trả dữ liệu đặt bàn (thông tin khách, bàn, khu vực, ngày giờ, số khách, ghi chú).
- (43) Backend trả dữ liệu về Frontend.
- (44) Admin xem toàn bộ thông tin đặt bàn.
- (45) Nếu Admin xác nhận:
- Admin nhấn nút "Xác nhận".
- Frontend gửi PUT /api/reservations/{id}/confirm đến Backend.
- Backend cập nhật trạng thái đặt bàn thành CONFIRMED trong Database.
- Database xác nhận cập nhật.
- Backend gửi email/SMS xác nhận đến khách hàng.
- Backend phản hồi Frontend "Xác nhận thành công".
- Frontend hiển thị trạng thái "Đã xác nhận" cho Admin.
- Khách hàng nhận thông báo xác nhận qua email/SMS.
- (46) Nếu Admin từ chối:
- Admin nhấn nút "Từ chối" và nhập lý do từ chối.
 - Frontend gửi PUT /api/reservations/{id}/cancel đến Backend kèm lý do.
 - Backend cập nhật trạng thái đặt bàn thành CANCELLED trong Database.
 - Backend giải phóng bàn (cập nhật trạng thái bàn thành AVAILABLE).
 - Database xác nhận cập nhật.

– Backend gửi email/SMS từ chối kèm lý do đến khách hàng.

– Backend phản hồi Frontend "Từ chối thành công".

– Frontend hiển thị trạng thái "Đã từ chối" cho Admin.

(47) Khách hàng có thể xem lịch sử đặt bàn.

(48) Frontend gửi GET /api/reservations/my-reservations đến Backend.

(49) Backend truy vấn Database để lấy danh sách đặt bàn của khách hàng.

(50) Database trả về danh sách đặt bàn.

(51) Backend trả dữ liệu về Frontend.

(52) Frontend hiển thị lịch sử đặt bàn cho khách hàng.

(53) Nếu khách hàng muốn hủy đặt bàn:

Khách hàng nhấn nút "Hủy đặt bàn".

Frontend gửi PUT /api/reservations/{id}/cancel-by-customer đến Backend.

Backend kiểm tra thời gian hủy trong Database (ví dụ: phải hủy trước 2 giờ).

(54) Nếu hủy trong thời hạn cho phép:

Database xác nhận cho phép hủy.

Backend cập nhật trạng thái đặt bàn thành CANCELLED.

Backend giải phóng bàn.

Database xác nhận cập nhật.

Backend phản hồi Frontend "Hủy thành công".

Frontend hiển thị thông báo "Đã hủy đặt bàn thành công".

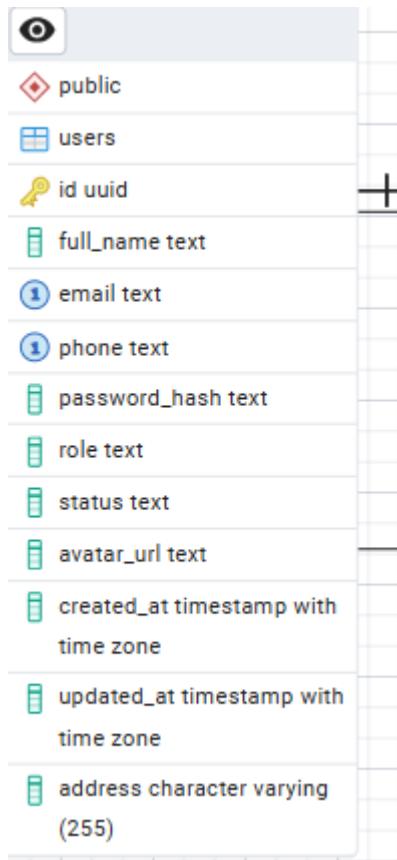
Backend gửi thông báo đến Admin về việc khách hủy đặt bàn.

3.7. Thiết kế cơ sở dữ liệu

3.7.1. Các bảng trong cơ sở dữ liệu

3.7.1.1. Bảng users

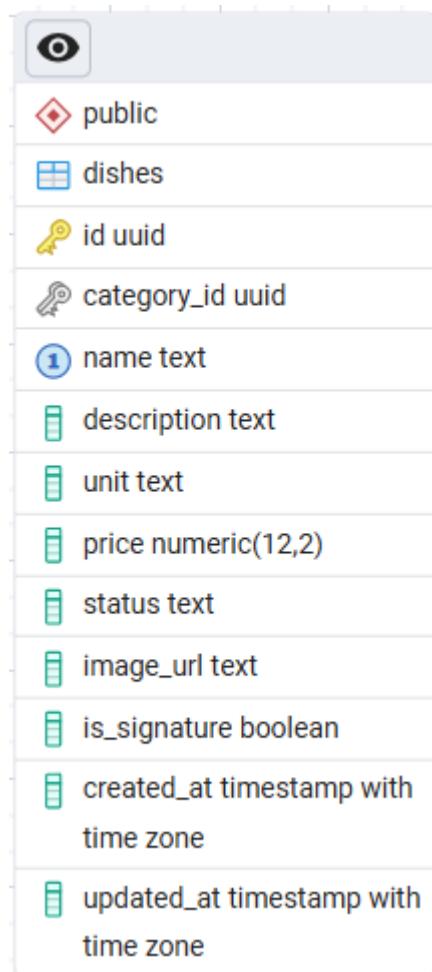
Bảng users dùng để lưu trữ thông tin tài khoản của người dùng trong hệ thống, bao gồm cả khách hàng và admin. Đây là bảng quan trọng phục vụ cho quá trình xác thực (authentication), phân quyền (authorization) và quản lý hồ sơ người dùng.



Hình 3-22: Bảng users

3.7.1.2. Bảng dishes

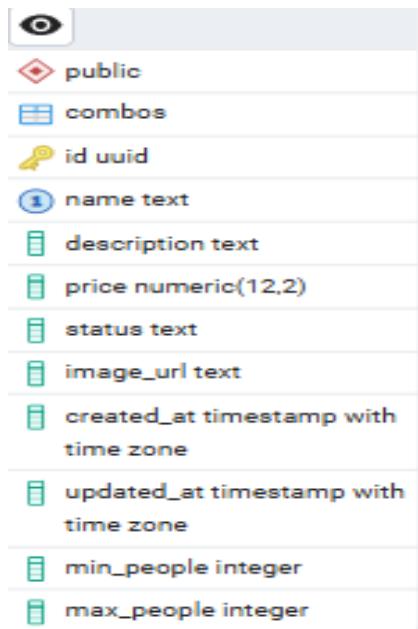
Bảng dishes dùng để lưu trữ thông tin các món ăn được phục vụ tại Nhà hàng thực dưỡng Hoa Ban. Đây là bảng quan trọng trong hệ thống vì dữ liệu của nó được sử dụng cho nhiều chức năng như hiển thị menu món ăn, đặt món, quản lý món, tạo combo,...



Hình 3-23: Bảng dishes

3.7.1.3. Bảng combos

Bảng combos dùng để lưu trữ thông tin về các combo món ăn của Nhà hàng Hoa Ban. Mỗi combo bao gồm nhiều món ăn khác nhau, được xây dựng nhằm tối ưu chi phí và phù hợp cho nhóm khách từ 2 đến nhiều người.



Hình 3-24: Bảng combos

3.7.1.4. Bảng orders

Bảng orders dùng để lưu thông tin các đơn đặt bàn và đặt món của khách hàng trong hệ thống.

Mỗi đơn sẽ gắn với người dùng hoặc bàn.

The screenshot shows a database schema diagram for a PostgreSQL database. At the top is a navigation bar with icons for 'public', 'orders', 'id uuid', 'user_id uuid', 'table_id uuid', 'types text', 'status text', 'subtotal numeric(12,2)', 'discount numeric(12,2)', 'tax numeric(12,2)', 'total numeric(12,2)', 'note text', 'created_at timestamp with time zone', 'updated_at timestamp with time zone', and 'applied_user_voucher_id uu id'. The 'orders' table is highlighted in blue.

public	◆
orders	订单
id uuid	主键
user_id uuid	用户ID
table_id uuid	餐桌ID
types text	类型
status text	状态
subtotal numeric(12,2)	小计
discount numeric(12,2)	折扣
tax numeric(12,2)	税金
total numeric(12,2)	总计
note text	备注
created_at timestamp with time zone	创建时间
updated_at timestamp with time zone	更新时间
applied_user_voucher_id uu id	已应用用户优惠券ID

Hình 3-25: Bảng orders

3.7.1.5. Bảng vouchers

Bảng vouchers dùng để lưu trữ thông tin các loại voucher khuyến mãi của nhà hàng, bao gồm voucher giảm giá, voucher đổi bằng điểm tích lũy.

The screenshot shows a PostgreSQL database schema diagram. The main table is 'vouchers' located in the 'public' schema. It contains the following columns:

- id uuid (primary key)
- code text
- name text
- type text
- value numeric(12,2)
- min_order numeric(12,2)
- max_discount numeric(12,2)
- status text
- start_at timestamp with time zone
- end_at timestamp with time zone
- created_at timestamp with time zone
- point_cost bigint

Hình 3-26: Bảng vouchers

3.7.1.6. Bảng reservations

Bảng reservations dùng để Lưu thông tin các yêu cầu đặt bàn của khách hàng tại Nhà hàng thực dưỡng Hoa Ban.

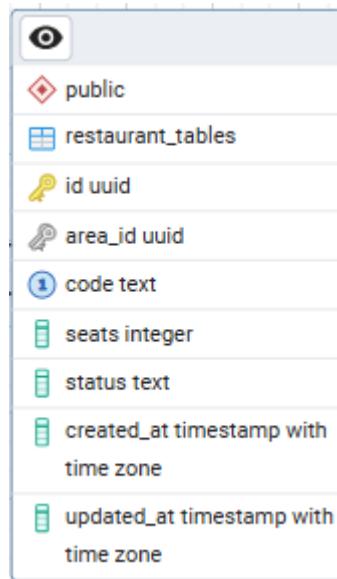
The screenshot shows a PostgreSQL database schema diagram. The main table is 'reservations' located in the 'public' schema. It contains the following columns:

- id uuid (primary key)
- user_id uuid
- start_time timestamp with time zone
- end_time timestamp with time zone
- party_size integer
- status text
- note text
- created_at timestamp with time zone
- updated_at timestamp with time zone
- cancel_reason text
- canceled_by text

Hình 3-27: Bảng reservations

3.7.1.7. Bảng restaurant_tables

Bảng restaurant_tables dùng để lưu thông tin các bàn ăn trong nhà hàng, bao gồm mã bàn, khu vực, số ghế, trạng thái và thời gian tạo/cập nhật.

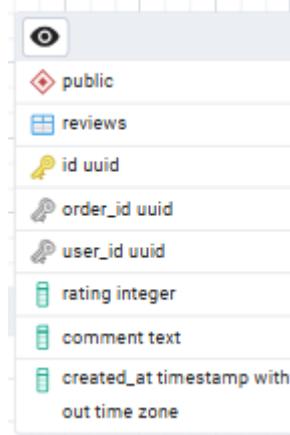


Hình 3-28: Bảng restaurant_tables

3.7.1.1. Bảng reviews

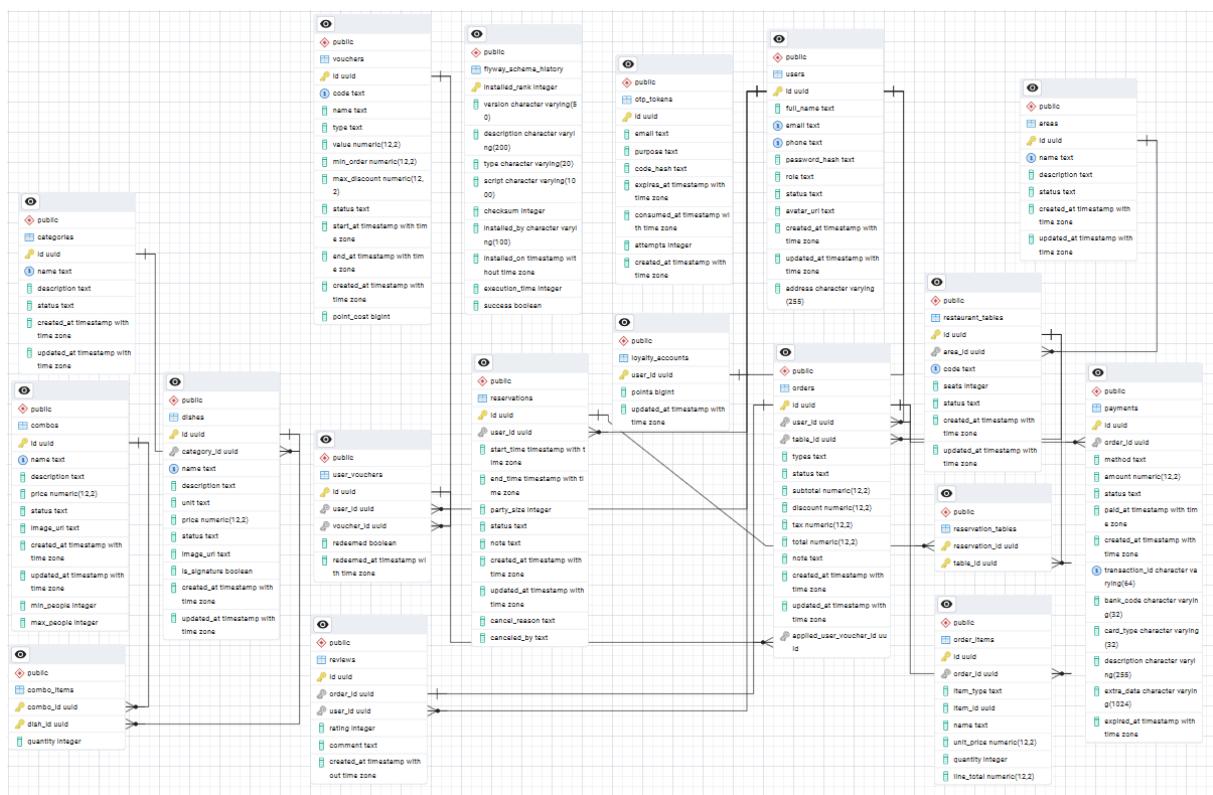
Bảng reviews dùng để Lưu trữ đánh giá của khách hàng sau khi đặt bàn hoặc đặt món tại Nhà hàng thực dưỡng Hoa Ban.

Mỗi đánh giá sẽ gắn với một đơn hàng và một người dùng.



Hình 3-29: Bảng reviews

3.7.2. Biểu đồ quan hệ của các bảng



Hình 3-30: Sơ đồ quan hệ ERD

CHƯƠNG 4. TRIỂN KHAI HỆ THỐNG

4.1. Cấu hình triển khai

4.1.1. Backend – Spring Boot

- Framework: Spring Boot 3.5.3
- Ngôn ngữ: Java (JDK 21)
- Quản lý phụ thuộc: Gredlew
- Các module chính:
- Spring Web
- Spring Security + JWT
- Spring Data JPA
- PostgreSQL Driver
- Spring Validation
- Spring AI (ChatBOT tư vấn món ăn)

4.1.2. Frontend – Vue.js

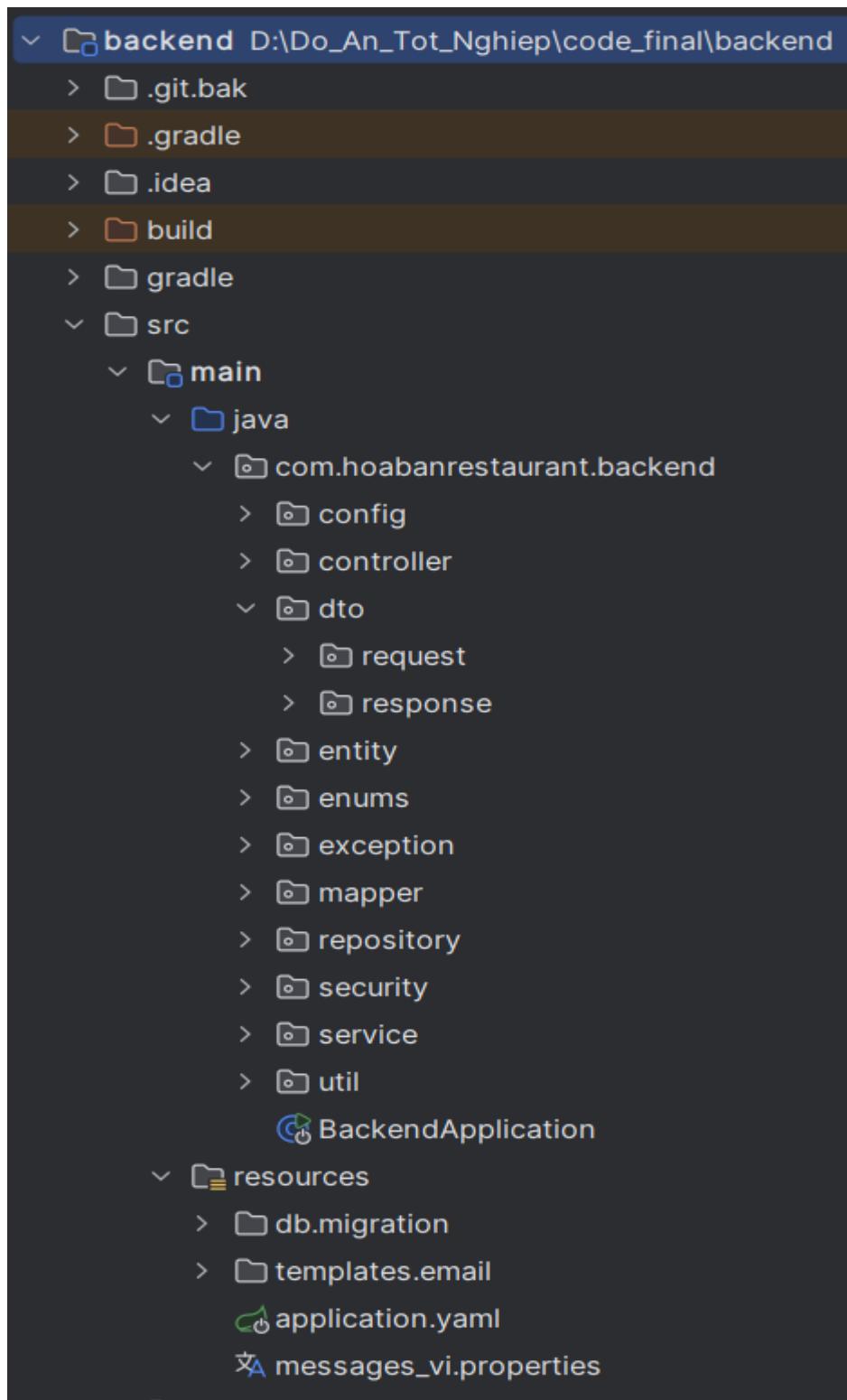
- Vue.js phiên bản: Vue 3 (Composition API)
- Công cụ build: Vite
- Router: Vue Router 4
- State management: Pinia
- UI Framework: Vuetify / TailwindCSS / Element Plus

4.1.3. Cơ sở dữ liệu – PostgreSQL

- Quản lý dữ liệu toàn hệ thống
- Thiết kế theo mô hình quan hệ (ERD)
- Dùng pgAdmin để quản trị

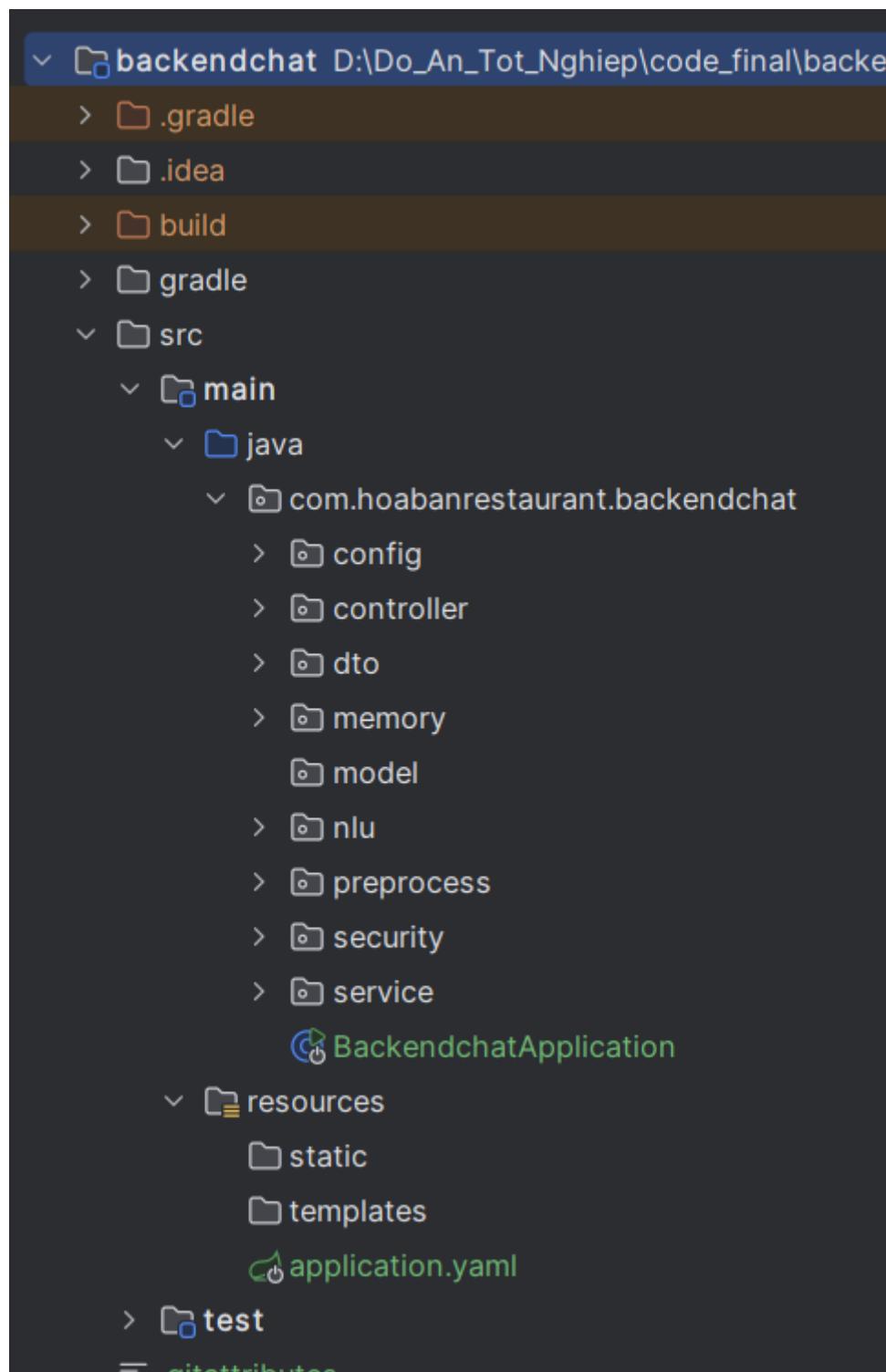
4.2. Cấu trúc thư mục

4.2.1. Backend chính



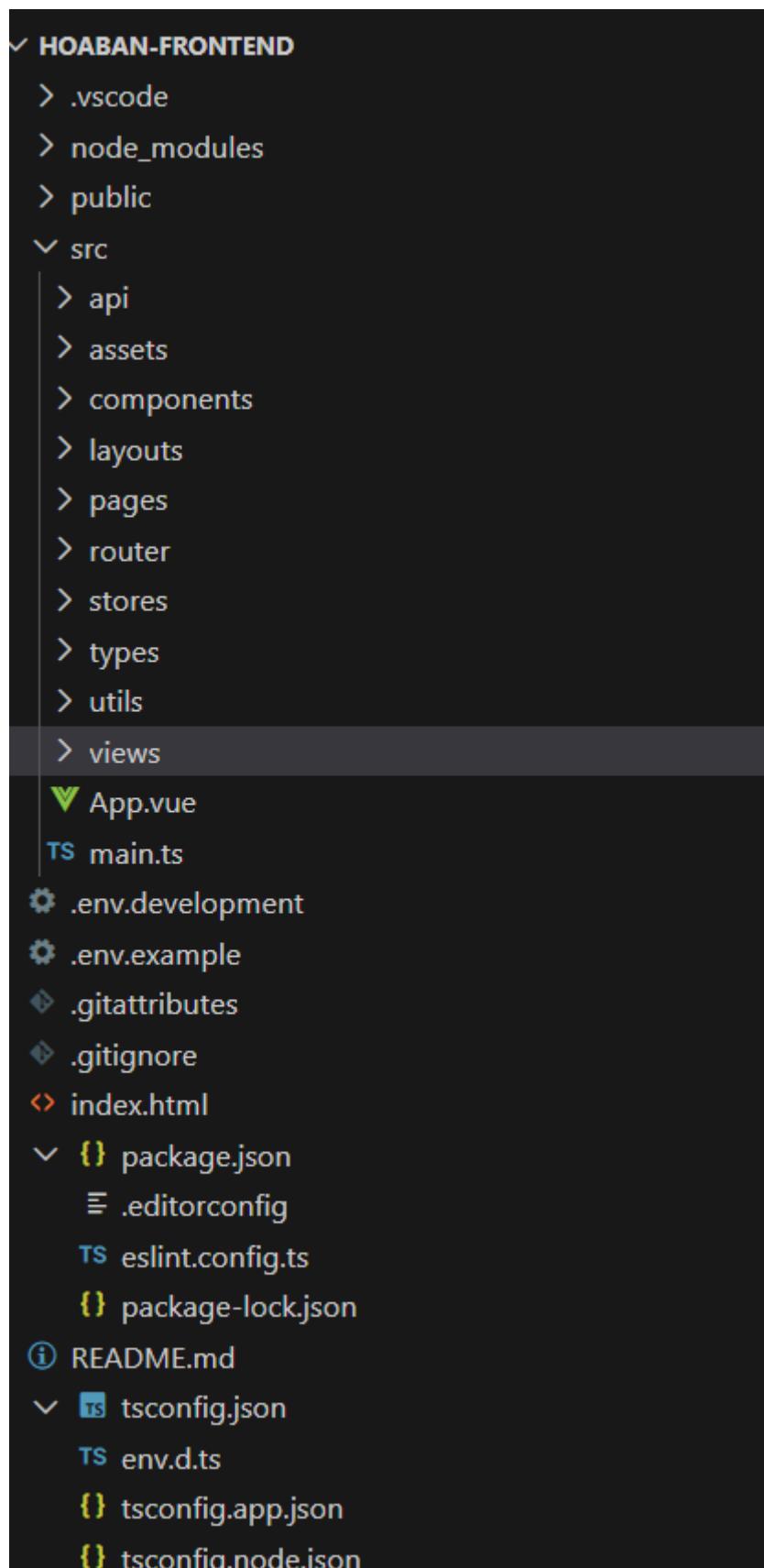
Hình 4-1: Cấu trúc thư mục backend chính

4.2.2. Backend chatbot



Hình 4-2: Cấu trúc thư mục backend chatbot

4.2.3. Frontend



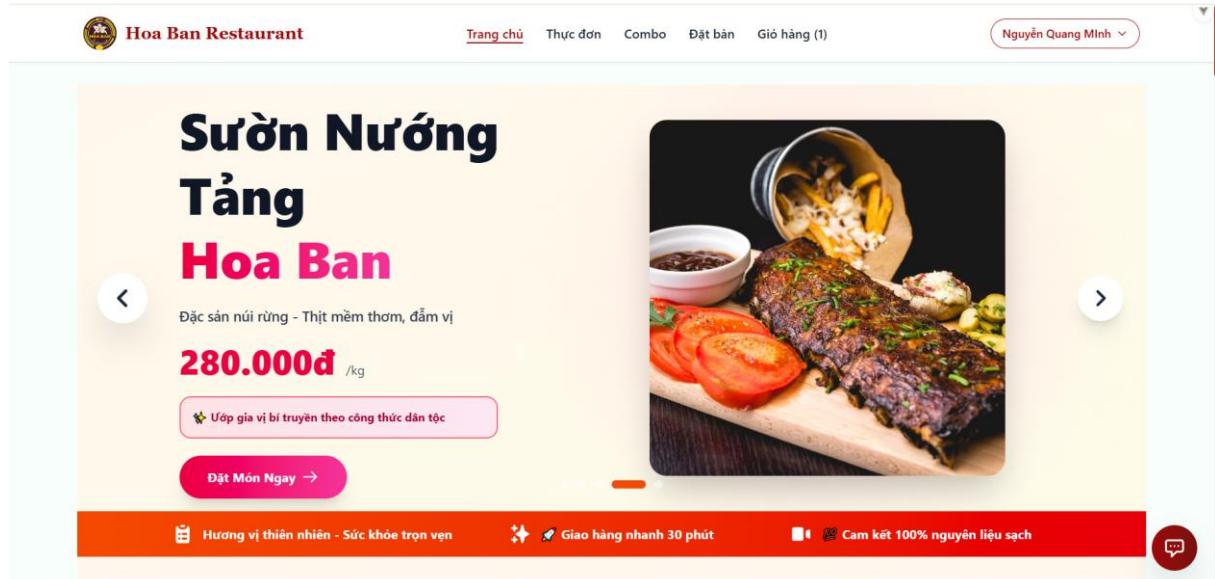
Hình 4-3: Cấu trúc thư mục frontend

4.3. Kết quả đạt được

4.3.1. Giao diện người dùng

4.3.1.1. Giao diện trang chủ cho user

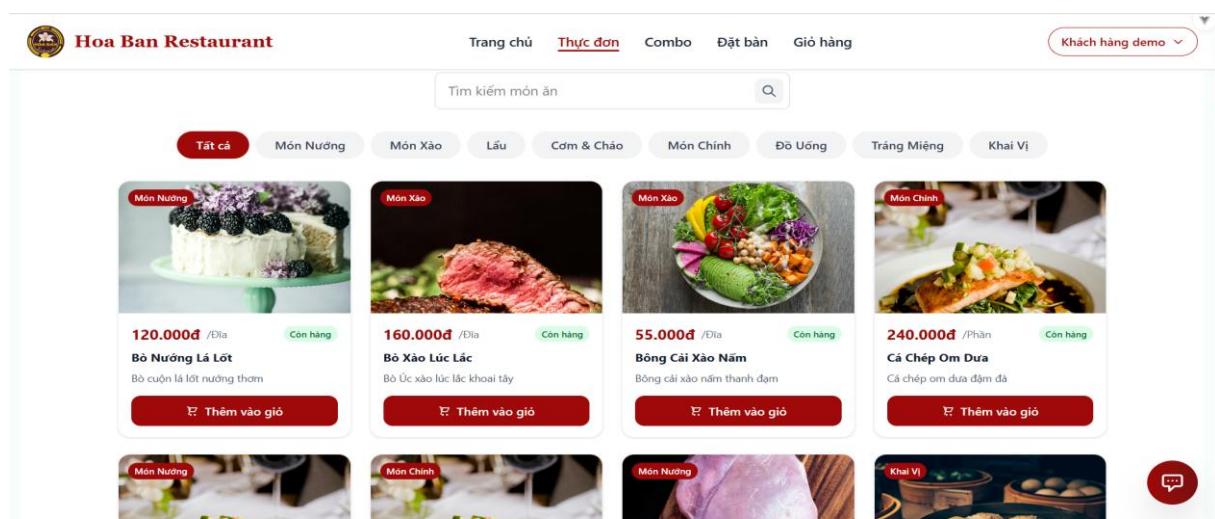
khách hàng sẽ nhìn thấy hình ảnh nổi bật về không gian của Nhà hàng thực dưỡng Hoa Ban kèm theo khẩu hiệu giới thiệu về phong cách ẩm thực và giá trị của nhà hàng.



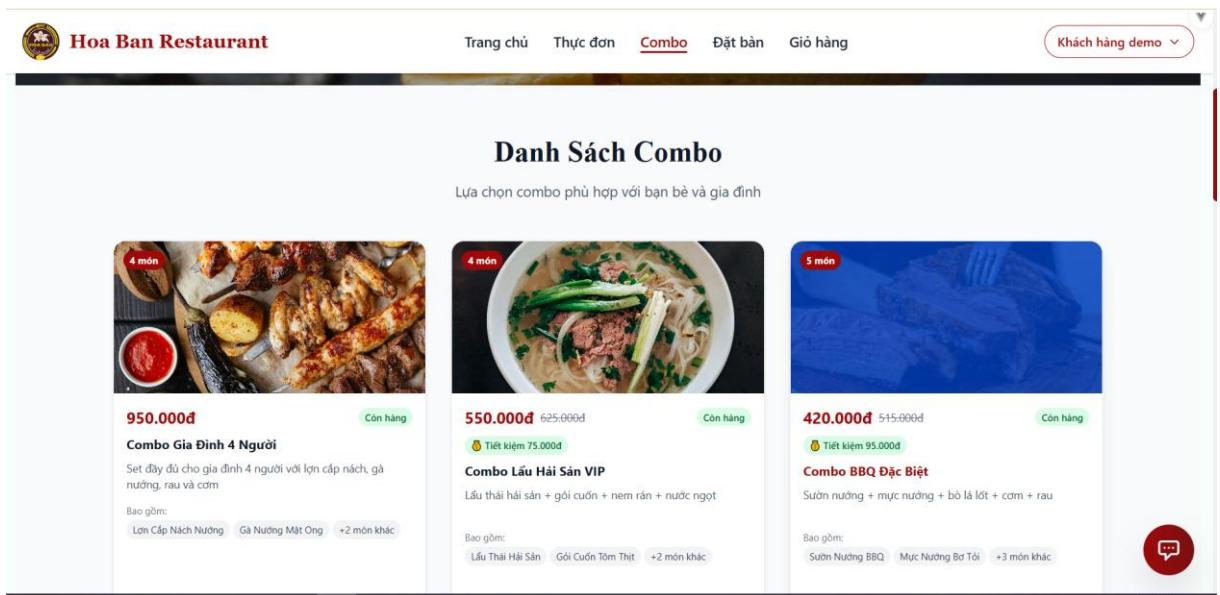
Hình 4-4: Giao diện trang chủ

4.3.1.2. Xem danh sách món ăn và combo

Người dùng có thể xem chi tiết từng món ăn hoặc combo với các thông tin giá cả, hình ảnh...



Hình 4-5: Giao diện thực đơn



Hình 4-6: Giao diện combo món

4.3.1.3. Chức năng đặt bàn

Hiển thị form đặt bàn để khách hàng gửi yêu cầu.

Thông tin đặt bàn	Đặt bàn
Tên khách hàng * <input type="text" value="Khách hàng demo"/> <small>Tự động điền</small>	Số điện thoại * <input type="text" value="Nhập số điện thoại của bạn"/>
Ngày đặt * <input type="text" value="dd/mm/yyyy"/>	Giờ đặt * <input type="text" value="Chọn giờ"/>
Số lượng khách * <input type="text" value="2 người"/>	Khu vực <input type="text" value="Chọn khu vực"/>
Yêu cầu đặc biệt <small>Sinh nhật, kỷ niệm, món ăn chay, dị ứng thực phẩm...</small>	
<input type="button" value="Đặt bàn ngay"/>	

Hình 4-7: Giao diện đặt bàn

4.3.1.4. Chức năng giỏ hàng

Chức năng giúp khách hàng thêm những món yêu thích vào giỏ hàng để đặt món.

Hình 4-8: Giao diện giỏ hàng

4.3.1.5. Chức năng thanh toán trực tuyến

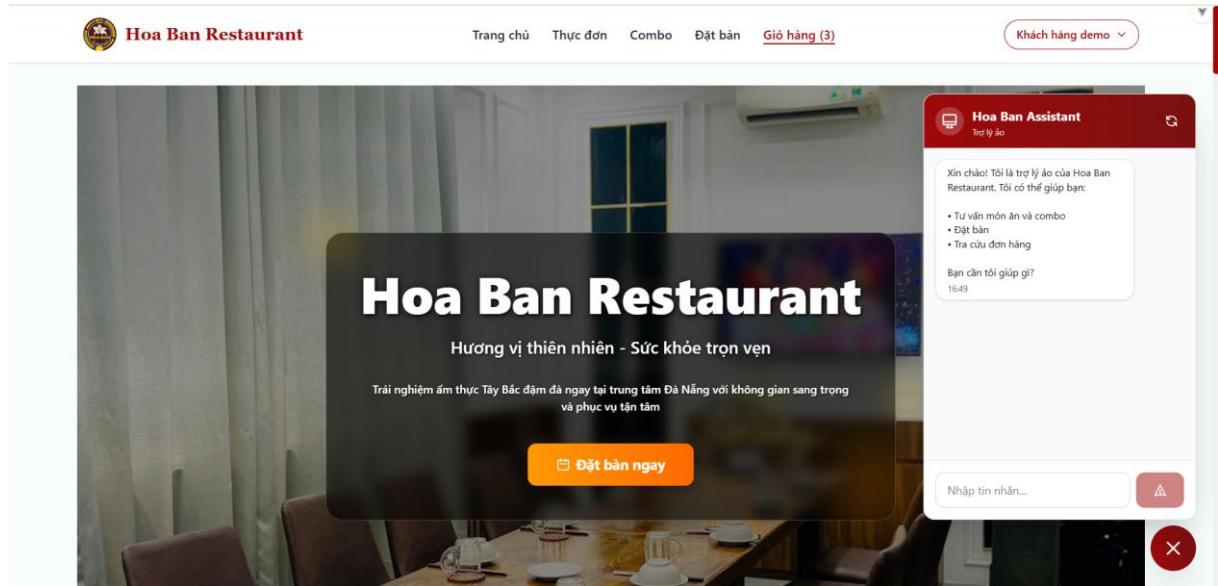
Người dùng có thể lựa chọn giữa hai phương thức thanh toán:

- Thanh toán trực tiếp:** Người dùng có thể chọn thanh toán khi nhận món.
- VNPay:** Nếu người dùng chọn thanh toán qua VNPay, họ sẽ được chuyển đến trang có mã qr chỉ cần quét và thanh toán.

Hình 4-9: Giao diện thanh toán

4.3.1.6. Chức năng Chatbot cho user

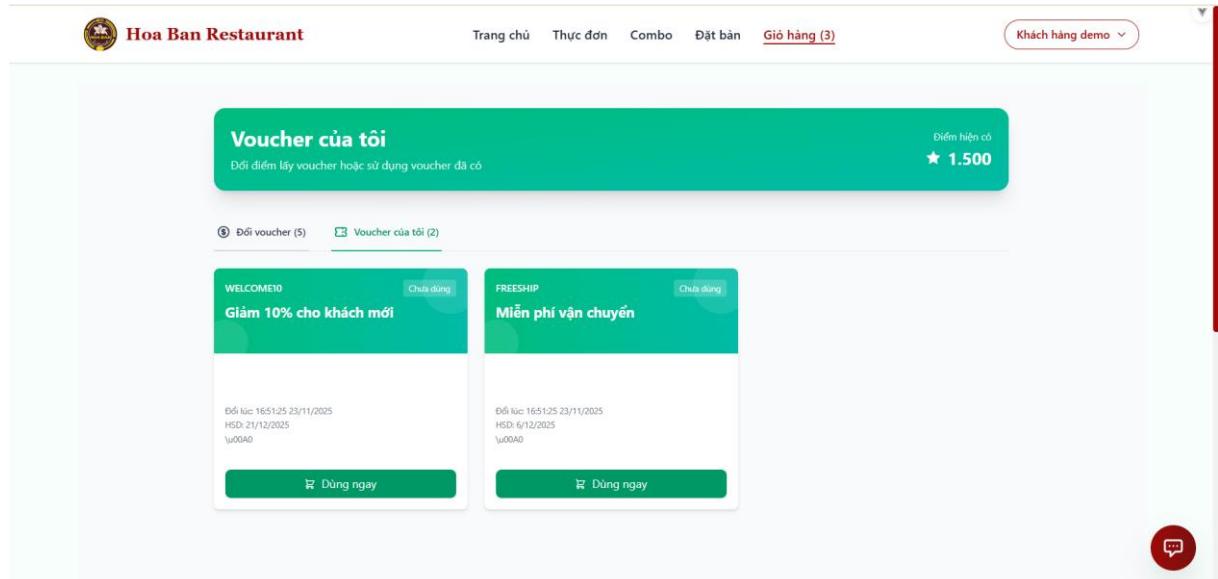
ChatBOT được tích hợp để hỗ trợ người dùng trả lời các câu hỏi thường gặp liên quan đến tư vấn món ăn. Tính năng này hoạt động 24/7, mang lại trải nghiệm tiện lợi và tức thời cho khách hàng.



Hình 4-10: Giao diện chatbot

4.3.1.7. Chức năng tích điểm và đổi voucher

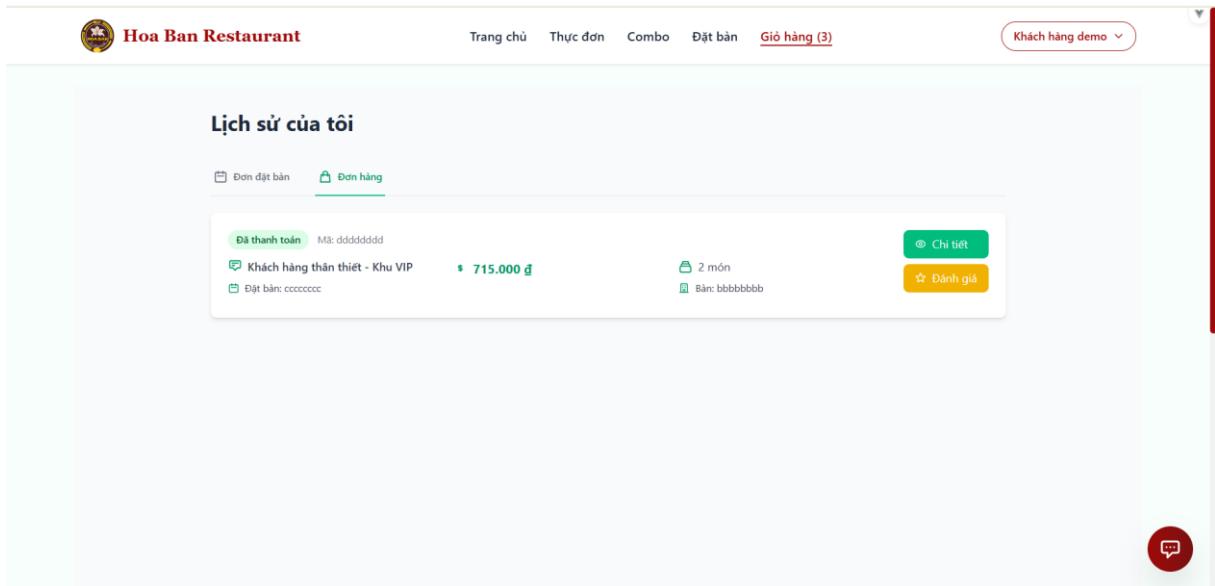
Chức năng này giúp khách hàng có thêm sự hài lòng khi sử dụng dịch vụ.



Hình 4-11: Giao diện voucher

4.3.1.8. Chức năng đánh giá

Chức năng giúp khách hàng chia sẻ cảm nhận tích cực khi trải nghiệm dịch vụ.

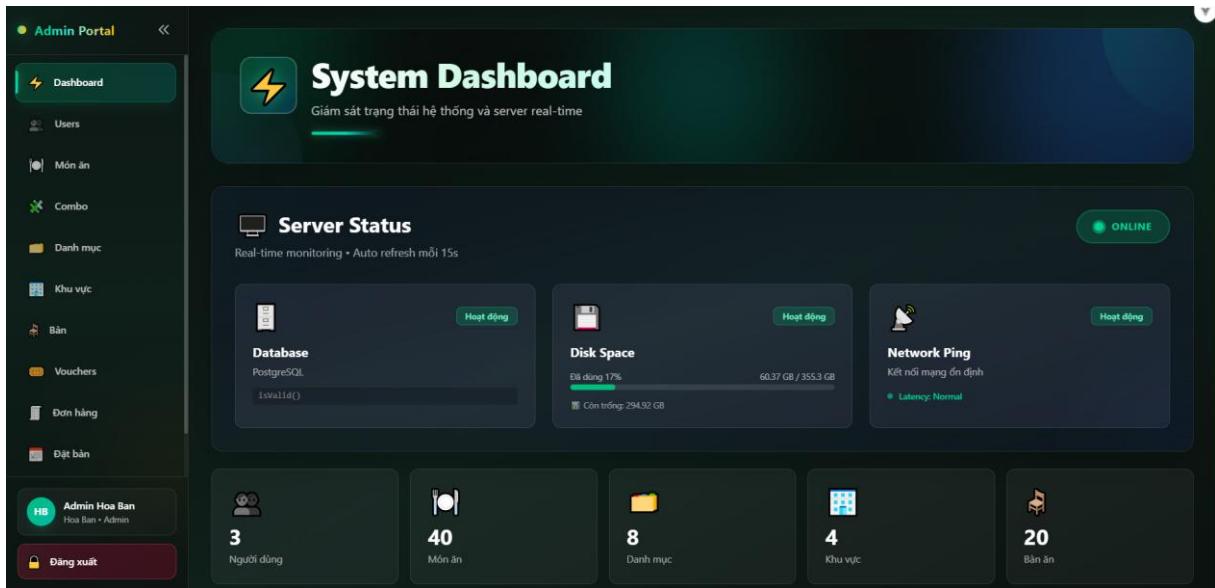


Hình 4-12: Giao diện đánh giá

4.3.2. Giao diện người quản trị

4.3.2.1. Trang dasdbord

Khi quản trị viên đăng nhập với tài khoản admin sẽ điều hướng đến trang admin và hiển thị tổng quát về server và thông kê số lượng.



Hình 4-13: Giao diện trang quản trị

4.3.2.2. Giao diện trang quản lý món ăn, combo và danh mục

Trang quản lý món ăn, combo và danh mục giúp admin chia rõ các tác vụ thêm, sửa và xóa các món ăn, combo và danh mục một cách trực quan.

Xây dựng website đặt bàn, đặt món cho nhà hàng thực dưỡng Hoa Ban sử dụng Vue.js, Spring Boot và PostgreSQL

The screenshot shows the 'Admin Portal' interface. On the left sidebar, under the 'Món ăn' (Food) section, there is a sub-menu for 'Combo'. The main content area is titled 'Tạo món ăn mới' (Create new dish). It includes fields for 'Chọn danh mục' (Select category), 'Tên món' (Dish name), 'Mô tả' (Description), 'Phần' (Portion), 'URL hình ảnh hoặc chọn từ máy' (Image URL or choose from machine), and 'Hoạt động' (Status). Below this is a 'Danh sách món' (List of dishes) table with columns: Hình ảnh (Image), Tên món (Dish name), Danh mục (Category), Đơn vị (Unit), Giá (Price), Trạng thái (Status), and Thao tác (Actions). Two dishes are listed: 'Bia Sài Gòn' (Danh mục: Đồ Uống, Giá: 25.000 đ, Trạng thái: Tạm dừng) and 'Bò Nướng Lá Lốt' (Danh mục: Món Nướng, Giá: 120.000 đ, Trạng thái: Hoạt động).

Hình 4-14: Giao diện trang quản lý món ăn

The screenshot shows the 'Admin Portal' interface. On the left sidebar, under the 'Combo' section, there is a sub-menu for 'Danh mục'. The main content area is titled 'Tạo combo' (Create combo). It includes fields for 'Tên combo' (Combo name), 'Chọn món ăn' (Select dish), 'Mô tả' (Description), 'URL ảnh hoặc chọn từ máy' (Image URL or choose from machine), and 'Chọn ảnh' (Choose image). Below this is a 'Danh sách danh mục' (List of categories) table with columns: Tên danh mục (Category name), Mô tả (Description), and Hoạt động (Status). Three combos are listed: 'Combo BBQ Đặc Biệt' (Mô tả: Sườn nướng + mực nướng + bò lá lốt + cơm + rau, Giá: 420.000đ), 'Combo Chay Thành Tịnh' (Mô tả: Bột hủ chiên + rau muống xào + bông cải xào nấm + cơm trắng, Giá: 120.000đ), and 'Combo Couple Lãng Mạn' (Mô tả: Set cho 2 người: cá hồi nướng, salad, bò xào, cơm chiên vả nước, Giá: 480.000đ).

Hình 4-15: Giao diện quản lý combo

The screenshot shows the 'Admin Portal' interface. On the left sidebar, under the 'Danh mục' section, there is a sub-menu for 'Món ăn'. The main content area is titled 'Tạo danh mục mới' (Create new category). It includes fields for 'Tên danh mục' (Category name), 'Hoạt động' (Status), and 'Mô tả (tùy chọn)' (Description). Below this is a 'Danh sách danh mục' (List of categories) table with columns: Tên danh mục (Category name), Mô tả (Description), Trạng thái (Status), and Thao tác (Actions). Five categories are listed: 'Món Nướng' (Mô tả: Các món nướng đặc sản Tây Bắc), 'Món Xào' (Mô tả: Các món xào hấp dẫn), 'Lẩu' (Mô tả: Lẩu các loại cho gia đình và nhóm bạn), 'Cơm & Cháo' (Mô tả: Cơm và cháo dinh dưỡng), and 'Món Chính' (Mô tả: Các món chính đặc sản).

Hình 4-16: Giao diện quản lý danh mục

4.3.2.3. Các chức năng quản lý bàn và khu vực

Trang quản lý bàn và khu vực giúp admin quản lý khu vực và bàn thực tế của nhà hàng.

Tên	Mô tả	Trạng thái	Thao tác
Khu VIP	Khu vực VIP sang trọng, phục vụ khách đặc biệt	Hoạt động	Sửa Xoá
Nhà Bạt	Khu nhà bạt mát mẻ, thoáng đãng	Hoạt động	Sửa Xoá
Nhà Gỗ	Khu nhà gỗ truyền thống lily Bắc	Hoạt động	Sửa Xoá
Sân Vườn	Khu vực sân vườn ngoài trời	Hoạt động	Sửa Xoá

Hình 4-17: Giao diện quản lý khu vực

MÃ BÀN	KHU VỰC	SỐ GHẾ	TRẠNG THÁI	THAO TÁC
NB01	Nhà Bạt	4 ghế	Có sẵn	QR Sửa Xoá
NB02	Nhà Bạt	6 ghế	Có sẵn	QR Sửa Xoá
NB03	Nhà Bạt	8 ghế	Có sẵn	QR Sửa Xoá
NB04	Nhà Bạt	4 ghế	Có sẵn	QR Sửa Xoá
NB05	Nhà Bạt	6 ghế	Có sẵn	QR Sửa Xoá

Hình 4-18: Giao diện quản lý bàn

4.3.2.4. Quản lý voucher

Trang quản lý voucher giúp admin thêm, sửa và xóa voucher một cách trực quan và dễ dàng.

Xây dựng website đặt bàn, đặt món cho nhà hàng thực dưỡng Hoa Ban sử dụng Vue.js, Spring Boot và PostgreSQL

Hình 4-19: Giao diện quản lý voucher

4.3.2.5. Quản lý đơn hàng

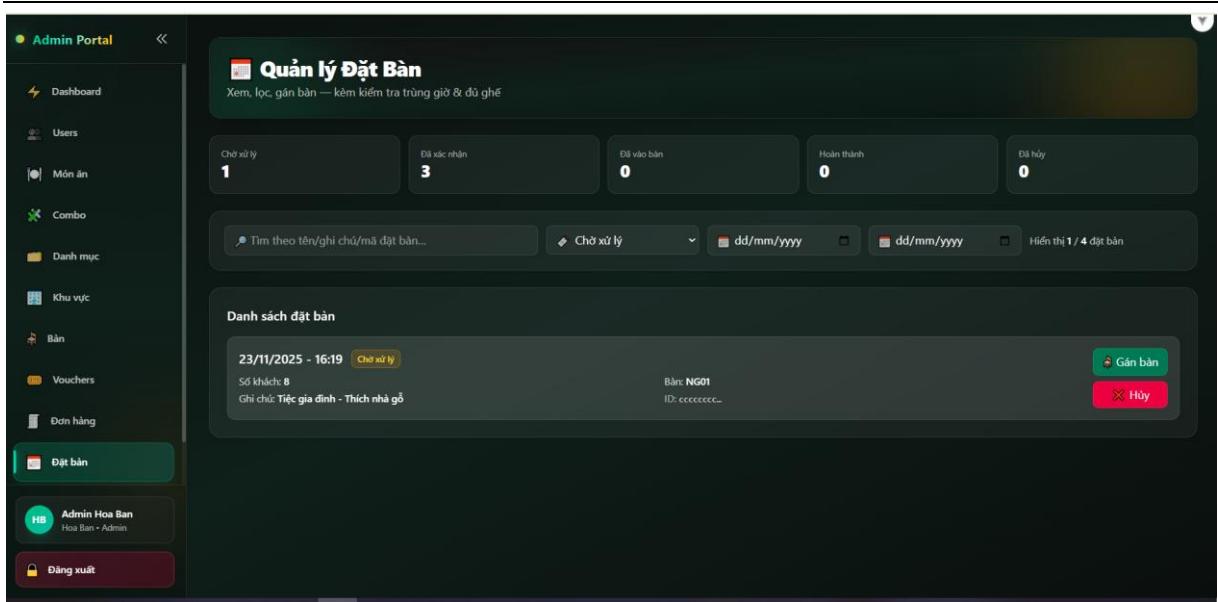
Danh sách đơn hàng cung cấp thông tin chi tiết về từng đơn. Admin có thể lọc, tìm kiếm đơn hàng theo trạng thái hỗ trợ thanh toán hoặc hủy đơn hàng.

Hình 4-20: Giao diện quản lý đơn hàng

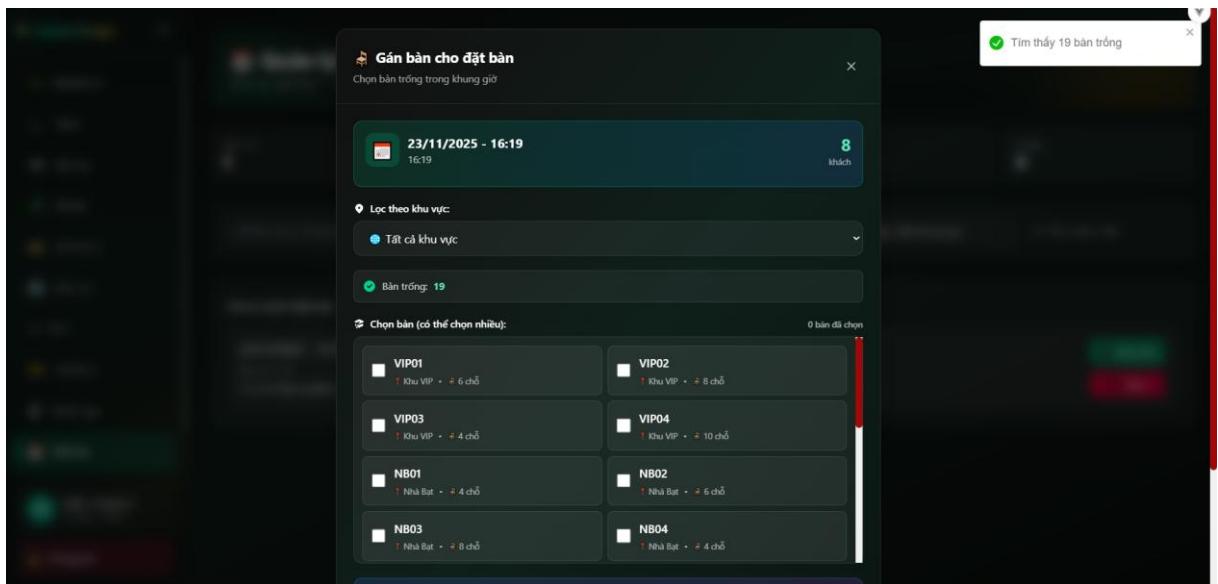
4.3.2.6. Trang quản đặt bàn

Trang quản lý đặt bàn giúp admin xem danh sách đơn trong ngày và hỗ trợ để xuất bàn trống để gán cho khách hàng.

Xây dựng website đặt bàn, đặt món cho nhà hàng thực dưỡng Hoa Ban sử dụng Vue.js, Spring Boot và PostgreSQL



Hình 4-21: Giao diện quản lý đặt bàn

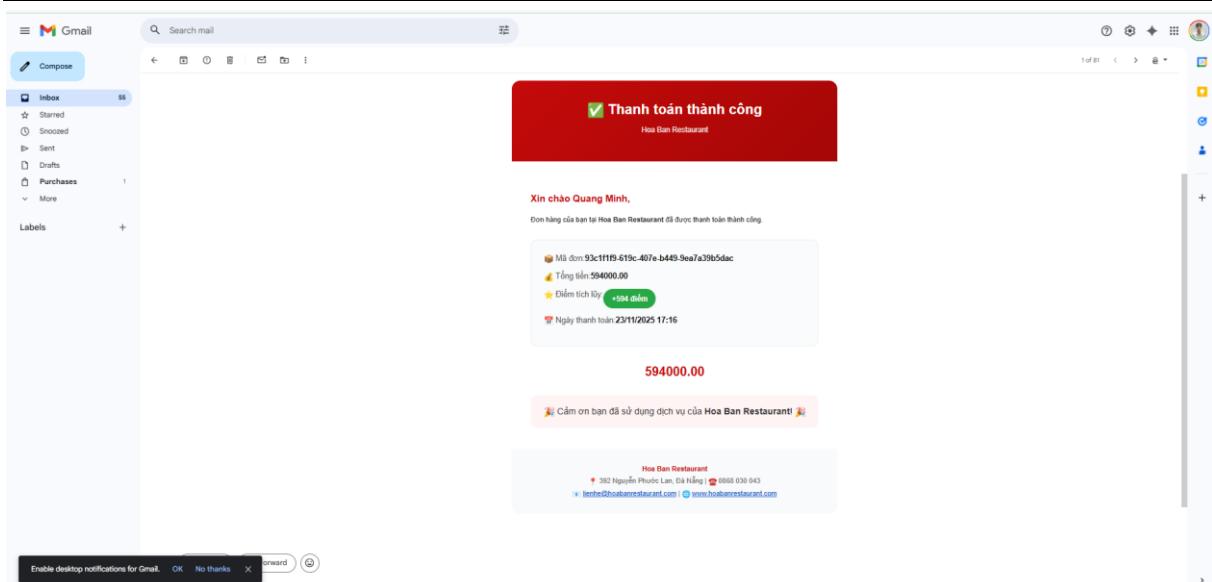


Hình 4-22: Giao diện gán bàn cho đơn đặt bàn

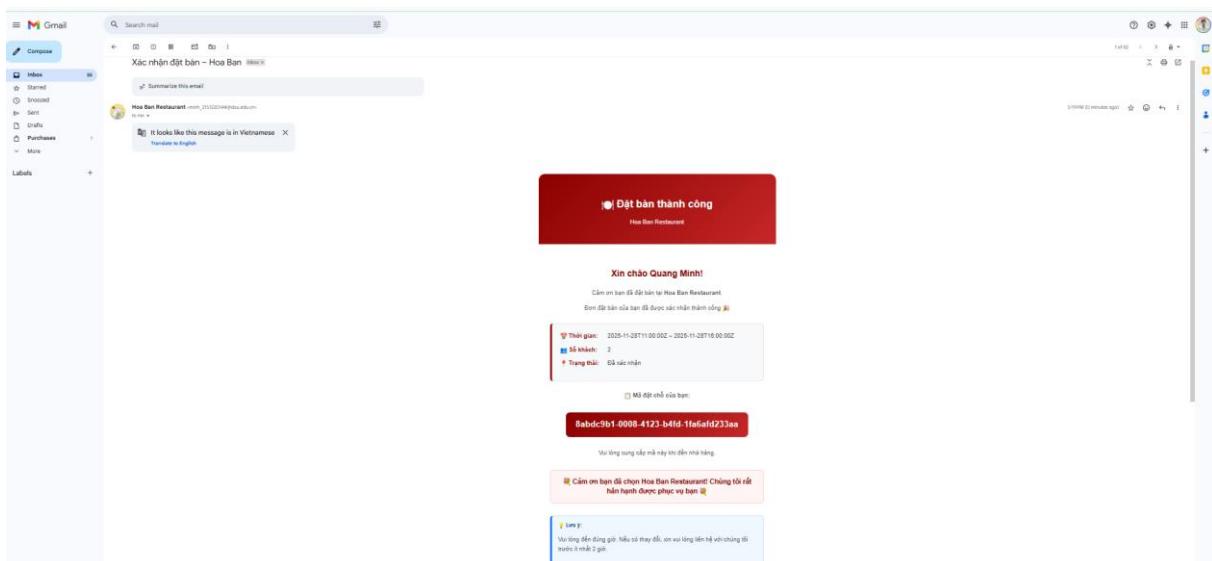
4.3.2.7. Chức năng gửi email cho khách hàng

Chức năng gửi email tự động khi xử lý đơn hàng hoặc đặt bàn của khách hàng giúp tăng độ chính chu của nhà hàng dành cho khách hàng.

Xây dựng website đặt bàn, đặt món cho nhà hàng thực dưỡng Hoa Ban sử dụng Vue.js, Spring Boot và PostgreSQL



Hình 4-23: Email thông báo thanh toán thành công



Hình 4-24: Email thông báo đặt bàn thành công

KẾT LUẬN

Đề tài “Xây dựng website đặt bàn – đặt món cho Nhà hàng thực dưỡng Hoa Ban bằng Vue.js, Spring Boot và PostgreSQL” đã được hoàn thành với những kết quả khả quan, đáp ứng đúng mục tiêu nghiên cứu và yêu cầu thực tiễn của hệ thống đặt món – đặt bàn trong thời đại số hóa.

1. Thành tựu đạt được:

- Xây dựng thành công hệ thống đặt bàn – đặt món trực tuyến: Hệ thống cho phép khách hàng xem thực đơn, combo, tìm kiếm món, đặt bàn, đặt món theo giờ hàng, quét mã QR tại bàn, thanh toán qua VNPay, sử dụng voucher, xem lịch sử đặt hàng và đánh giá chất lượng dịch vụ.
- Giao diện thân thiện – trải nghiệm người dùng tối ưu: Thiết kế bằng Vue.js kết hợp UI hiện đại giúp khách hàng thao tác dễ dàng. Giao diện Admin được xây dựng trực quan, hỗ trợ quản lý món ăn, combo, bàn ăn – khu vực, người dùng, voucher và các đơn đặt của khách.
- Tích hợp công nghệ hiện đại: Spring Boot cho backend giúp đảm bảo tính ổn định, bảo mật và mở rộng. PostgreSQL đảm bảo hiệu năng truy vấn tốt, lưu trữ dữ liệu an toàn. Thanh toán VNPay QR mang lại trải nghiệm thanh toán nhanh chóng và tiện lợi. ChatBOT AI tư vấn món ăn được tích hợp từ mô hình Gemini hỗ trợ khách hàng chọn món theo khẩu vị.
- Tính ứng dụng thực tế cao: Hệ thống hỗ trợ doanh nghiệp du lịch: Xây dựng nền tảng đặt tour trực tuyến hiệu quả, giảm tải các quy trình thủ công, gia tăng khả năng tiếp cận khách hàng trong kỷ nguyên số

2. Đóng góp và ý nghĩa:

- Đối với học thuật: Đề tài là minh chứng rõ ràng cho khả năng ứng dụng các công nghệ hiện đại như Spring Boot, Vue.js, PostgreSQL vào việc xây dựng một hệ thống đặt bàn – đặt món thực tế. Qua đó, đề tài góp phần làm rõ quy trình tổ chức và phát triển một hệ thống web hoàn chỉnh, từ xây dựng API RESTful, thiết kế giao diện SPA, đến tích hợp thanh toán VNPay và chatbot AI. Đồng thời, đề tài giúp cung cấp kiến thức về phân tích yêu cầu, thiết kế use case, mô hình dữ liệu và quy trình triển khai hệ thống trong thực tế.
- Đối với thực tiễn: Website có tiềm năng triển khai trong thực tế, hỗ trợ nhà hàng tối ưu quy trình vận hành từ nhận đơn – quản lý món – quản lý bàn – xử lý thanh toán theo quy trình số hóa, nâng cao mức độ hài lòng của khách hàng nhờ trải nghiệm đặt bàn – gọi món nhanh chóng, thuận tiện, hạn chế sai sót trong quá trình phục vụ. Nền tảng góp phần đưa mô hình ẩm thực thực dưỡng, dinh dưỡng

xanh đến gần hơn với cộng đồng thông qua số hóa, đồng thời hỗ trợ nhà hàng nâng cao hiệu quả kinh doanh, giảm tải nhân sự, tối ưu hóa thời gian phục vụ và nâng cao chất lượng dịch vụ.

3. Hạn chế và bài học kinh nghiệm:

- **Hạn chế:** Một số tính năng nâng cao như dashboard phân tích doanh thu, biểu đồ thống kê, AI phân tích hành vi khách hàng chưa được triển khai do giới hạn thời gian và nguồn lực. Ứng dụng di động chưa được phát triển, mới chỉ có phiên bản web responsive. Hệ thống chưa có phân quyền chi tiết theo nhiều vai trò (nhân viên, thu ngân, quản lý ca,...)
- **Bài học kinh nghiệm:** Qua quá trình thực hiện đề tài, nhóm hiểu rõ hơn quy trình làm việc theo MVC/MVVM và kiến trúc RESTful API – SPA, biết cách phân tích yêu cầu, lập use case, ERD, xây dựng flow hoạt động và triển khai theo vòng đời phát triển phần mềm. Đồng thời có kinh nghiệm thực tế trong việc triển khai hệ thống full-stack, xử lý lỗi backend – frontend – database, rèn luyện kỹ năng tối ưu API, thiết kế UI/UX và quản lý dự án khi làm việc với hệ thống lớn.

4. Hướng phát triển:

- Tích hợp Chatbot AI nâng cao: gợi ý món theo khẩu vị – calo – bệnh lý – sở thích cá nhân.
- Phát triển mobile app (Android/iOS) để tối ưu trải nghiệm người dùng.
- Bổ sung dashboard phân tích doanh thu, biểu đồ thống kê theo ngày/tháng/năm.
- Nâng cấp bảo mật JWT, phân quyền chi tiết cho các nhóm nhân sự: thu ngân, nhân viên phục vụ, quản lý.
- Tối ưu hiệu suất hệ thống, bổ sung bộ nhớ đệm Redis, triển khai Docker/CI-CD.

5. Tổng kết:

Đề tài đã được triển khai thành công, thể hiện sự cống gắng trong suốt quá trình phân tích – thiết kế – lập trình – kiểm thử – triển khai. Hệ thống đặt bàn – đặt món Nhà hàng Hoa Ban không chỉ mang lại giá trị thực tiễn cho hoạt động kinh doanh của nhà hàng, mà còn là minh chứng rõ ràng cho khả năng ứng dụng công nghệ vào thực tế. Sản phẩm góp phần hỗ trợ quá trình chuyên đổi số trong ngành ẩm thực, nâng cao chất lượng phục vụ và mang giá trị văn hóa thực dưỡng đến gần hơn với cộng đồng. Đề tài giúp em hoàn thiện kỹ năng chuyên môn, rèn luyện tư duy phân tích hệ thống và tích lũy kinh nghiệm quan trọng cho các dự án công nghệ trong tương lai.

TÀI LIỆU THAM KHẢO

[1] Spring Boot Documentation (2024). Spring Boot 3.x Reference Guide.

<https://spring.io/projects/spring-boot>

[2] Vue.js Documentation (2024). Vue.js Guide and API.

<https://vuejs.org>

[3] PostgreSQL Documentation (2024). PostgreSQL Official Documentation.

<https://www.postgresql.org/docs>

[4] VNPay. Hướng dẫn tích hợp cổng thanh toán VNPay QR.

<https://vnpay.vn>

[5] Gemini AI (2024). Google AI Gemini – Build AI Experiences.

<https://gemini.google.com>

[6] MDN Web Docs. JavaScript, HTML, CSS Documentation.

<https://developer.mozilla.org>