

**ĐẠI HỌC QUỐC GIA HÀ NỘI  
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**



Hoàng Phi Hùng

**NEARBY EXPLORATION - HỆ THỐNG ĐÈ  
XUẤT VÀ LẬP LỊCH TRẢI NGHIỆM DỊCH VỤ  
ĐỊA PHƯƠNG CHO KHÁCH DU LỊCH**

**KHÓA LUẬN TỐT NGHIỆP ĐẠI HỌC  
Ngành: Khoa học Máy tính**

**HÀ NỘI – 2025**

ĐẠI HỌC QUỐC GIA HÀ NỘI  
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

Hoàng Phi Hùng

NEARBY EXPLORATION - HỆ THỐNG ĐỀ  
XUẤT VÀ LẬP LỊCH TRẢI NGHIỆM DỊCH VỤ  
ĐỊA PHƯƠNG CHO KHÁCH DU LỊCH

KHÓA LUẬN TỐT NGHIỆP ĐẠI HỌC

Ngành: Khoa học Máy tính

Cán bộ hướng dẫn: TS. Lê Khánh Trình

Cán bộ đồng hướng dẫn: TS. Đặng Trần Bình

HÀ NỘI – 2025

VIETNAM NATIONAL UNIVERSITY, HANOI  
UNIVERSITY OF ENGINEERING AND TECHNOLOGY

Hoang Phi Hung

**NEARBY EXPLORATION - A  
RECOMMENDATION AND SCHEDULING  
SYSTEM FOR LOCAL SERVICE  
EXPERIENCES FOR TOURISTS**

BACHELOR'S THESIS  
Major: Computer Science

Supervisor: Dr. Le Khanh Trinh

Co-Supervisor: Dr. Dang Tran Binh

HA NOI – 2025

## LỜI CẢM ƠN

Trước tiên, em xin gửi lời cảm ơn chân thành và sâu sắc đến tập thể các thầy cô tại Trường Đại học Công nghệ, đặc biệt là các giảng viên Khoa Công nghệ Thông tin, những người đã tận tâm giảng dạy và truyền đạt cho em những kiến thức chuyên môn cùng kinh nghiệm thực tiễn quý báu trong suốt hành trình học tập tại trường. Sự tận tình và nhiệt huyết của các thầy cô đã đặt nền móng vững chắc, giúp em tự tin bước tiếp trên con đường sau này.

Em cũng xin bày tỏ lòng biết ơn sâu sắc đến Thầy giáo, TS. Lê Khánh Trình và TS. Đặng Trần Bình, Khoa Công nghệ Thông tin – Trường Đại học Công nghệ – ĐHQGHN, người đã trực tiếp hướng dẫn và đồng hành cùng em trong quá trình thực hiện khóa luận này. Sự tận tình chỉ bảo, những ý kiến đóng góp giá trị cùng sự động viên, khích lệ không ngừng từ Thầy đã tiếp thêm động lực để em vượt qua khó khăn và hoàn thành tốt khóa luận. Những bài học và lời khuyên quý báu từ Thầy không chỉ giúp em hoàn thiện khóa luận mà còn là hành trang quý giá cho chặng đường phía trước.

Bên cạnh đó, em xin gửi lời cảm ơn chân thành đến gia đình và bạn bè, những người luôn là nguồn động viên tinh thần to lớn. Sự yêu thương và ủng hộ từ gia đình cùng những lời động viên kịp thời từ bạn bè đã giúp em vượt qua những thử thách trong suốt quá trình học tập và thực hiện khóa luận.

Cuối cùng, em xin cảm ơn tất cả mọi người đã hỗ trợ em trong khóa luận. Mặc dù đã cố gắng hết sức, tuy nhiên khóa luận này khó tránh khỏi những thiếu sót. Em rất mong nhận được những ý kiến đóng góp quý báu của thầy cô để khóa luận được hoàn thiện hơn. Em xin chân thành cảm ơn !

## LỜI CAM ĐOAN

Em là Hoàng Phi Hùng, sinh viên lớp QH-2021-CS1 Khóa 66 ngành Khoa học máy tính trường Đại học Công Nghệ - Đại học Quốc gia Hà Nội. Em xin cam đoan khoá luận "Khám Phá Xung Quanh" là công trình nghiên cứu do bản thân em thực hiện. Các nội dung nghiên cứu, kết quả trong khóa luận là trung thực.

Các thông tin sử dụng trong khóa luận là có cơ sở và không có nội dung nào sao chép từ các tài liệu mà không ghi rõ trích dẫn tham khảo. Em xin chịu trách nhiệm về lời cam đoan này.

Hà Nội, ngày ... tháng 5 năm 2025

Sinh viên

Hoàng Phi Hùng

## TÓM TẮT

Đề tài Khám Phá Xung Quanh mục đích là phát triển một hệ thống tìm kiếm dịch vụ địa phương thông minh, giúp người dùng có thể dễ dàng tìm kiếm và khám phá các dịch vụ ăn uống trong khu vực xung quanh vị trí của họ. Một trong những điểm nổi bật của hệ thống là cho phép người dùng có thể tìm kiếm món ăn bằng hình ảnh, hệ thống tích hợp LLaMA - mô hình ngôn ngữ lớn về AI của Meta, nó hoạt động dưới dạng mã nguồn mở để đề xuất món ăn dựa trên sở thích cũng như thời tiết hiện tại nơi người dùng.

Hệ thống Khám Phá Xung quanh không chỉ đơn thuần là một công cụ tìm kiếm mà còn là một nền tảng hỗ trợ người dùng trong việc khám phá và trải nghiệm các dịch vụ địa phương, tạo ra sự kết nối giữa người tiêu dùng và các nhà cung cấp dịch vụ. Mang lại những lợi ích to lớn về phát triển dịch vụ ăn uống, du lịch cho đất nước.

**Từ khóa:** *Tìm kiếm món ăn bằng hình ảnh, Khám phá nhà tự động, Đề xuất món ăn tự động, Khám phá dịch vụ địa phương, Dịch vụ ăn uống, Hệ thống gợi ý, Nhận diện món ăn từ ảnh, Nhà hàng gần đây, Địa điểm ăn uống nổi bật, v.v.*

## ABSTRACT

The NearBy Exploration project aims to develop an intelligent local service search system, enabling users to seamlessly discover and explore dining services in their surrounding area. A standout feature of the system is its capability to allow users to search for dishes using images. It integrates LLaMA, Meta's open-source large-scale AI language model, to provide dish recommendations tailored to user preferences and current local weather conditions.

NearBy Exploration is not merely a search tool but a platform designed to enhance users' discovery and engagement with local services, fostering connections between consumers and service providers. The system delivers substantial benefits for the growth of dining and tourism services in the country.

**Keywords:** *Image-based dish search, Automated restaurant discovery, Automated dish recommendation, Local service exploration, Dining services, Recommendation system, Dish recognition from images, Nearby restaurants, Popular dining locations, etc.*

# Mục lục

Lời cảm ơn.....	iv
Lời cam đoan .....	v
Tóm tắt .....	vi
Abstract .....	vii
<b>Chương 1. Đặt vấn đề .....</b>	<b>1</b>
1.1. Giới thiệu bài toán .....	1
1.2. Hiện trạng và mục tiêu khoá luận.....	1
1.3. Bố cục khóa luận .....	2
<b>Chương 2. Kiến thức nền tảng .....</b>	<b>4</b>
2.1. Phát triển backend với Java Spring.....	4
2.1.1. Giới thiệu chung .....	4
2.1.2. Đặc điểm nổi bật .....	4
2.2. Phát triển giao diện với ReactJS .....	5
2.2.1. Giới thiệu chung .....	5
2.2.2. Đặc điểm nổi bật .....	5
2.3. Phát triển API backend với FastAPI.....	6
2.3.1. Giới thiệu chung .....	6
2.3.2. Đặc điểm nổi bật .....	6
2.4. Quản lý và truy vấn dữ liệu với MySQL .....	6
2.4.1. Giới thiệu chung .....	7
2.4.2. Đặc điểm nổi bật .....	7
2.5. Nhận diện và xử lý ảnh món ăn .....	7
2.5.1. Giới thiệu chung .....	7
2.5.2. Công nghệ sử dụng .....	8
2.6. Triển khai hệ thống với quy trình CI/CD .....	8
2.6.1. Giới thiệu chung .....	8
2.6.2. Đặc điểm nổi bật .....	9
<b>Chương 3. Phân tích và thiết kế hệ thống .....</b>	<b>10</b>

3.1. Đặc tả yêu cầu .....	10
3.1.1. Yêu cầu chức năng.....	10
3.1.2. Yêu cầu phi chức năng.....	11
3.2. Biểu đồ tổng quan ca sử dụng hệ thống.....	14
3.3. Mô tả ca sử dụng .....	14
3.3.1. Ca sử dụng đăng ký tài khoản.....	15
3.3.2. Ca sử dụng khám phá tự động .....	19
3.3.3. Ca sử dụng tìm kiếm nhà hàng bằng hình ảnh.....	22
3.3.4. Ca sử dụng tạo nhà hàng .....	24
3.3.5. Ca sử dụng tạo món ăn .....	25
3.3.6. Ca sử dụng lưu nhà hàng .....	27
3.4. Thiết kế cơ sở dữ liệu.....	30
<b>Chương 4. Xây dựng và triển khai hệ thống .....</b>	<b>33</b>
4.1. Công nghệ sử dụng.....	33
4.1.1. Khái quát tổng thể.....	33
4.1.2. Công nghệ phía Frontend .....	33
4.1.3. Công nghệ phía Backend Java Spring .....	33
4.1.4. Công nghệ phía Backend FastAPI .....	34
4.1.5. Công nghệ tích hợp bên thứ ba .....	34
4.1.6. Công nghệ triển khai .....	35
4.1.7. Mô hình mạng nơ-ron tích chập nhận diện ảnh .....	35
4.1.8. Kiến trúc hệ thống.....	36
4.1.9. Triển khai hệ thống.....	36
4.2. Các chức năng chính của hệ thống.....	38
4.2.1. Trang chủ.....	39
4.2.2. Cập nhật hồ sơ cá nhân .....	39
4.2.3. Quản lý nhà hàng .....	40
4.2.4. Quản lý món ăn .....	40
4.2.5. Tìm kiếm món ăn bằng hình ảnh .....	41
4.2.6. Khám phá món ăn, nhà hàng tự động.....	42

4.2.7. Xem thông tin từng nhà hàng .....	44
4.2.8. Lưu nhà hàng vào mục yêu thích .....	44
4.2.9. Xem danh sách nhà hàng xung quanh.....	45
<b>Chương 5. Kiểm thử cho hệ thống .....</b>	<b>46</b>
5.1. Kiểm thử các xử lý logic phía máy chủ .....	46
5.1.1. Kiểm thử API .....	46
5.1.2. Kiểm thử đơn vị .....	49
<b>Chương 6. Kết luận .....</b>	<b>52</b>

# Danh sách hình vẽ

3.1. Biểu đồ ca sử dụng tổng quát của hệ thống. ....	14
3.2. Biểu đồ hoạt động ca sử dụng đăng ký tài khoản. ....	17
3.3. Biểu đồ tuần tự ca sử dụng đăng ký tài khoản. ....	18
3.4. Biểu đồ hoạt động ca sử dụng khám phá tự động.....	20
3.5. Biểu đồ tuần tự ca sử dụng khám phá tự động. ....	21
3.6. Biểu đồ hoạt động ca sử dụng tìm kiếm nhà hàng bằng hình ảnh.....	23
3.7. Biểu đồ tuần tự ca sử dụng tìm kiếm nhà hàng bằng hình ảnh. ....	23
3.8. Biểu đồ hoạt động ca sử dụng tạo nhà hàng.....	25
3.9. Biểu đồ tuần tự ca sử dụng tạo nhà hàng.....	25
3.10. Biểu đồ hoạt động ca sử dụng tạo món ăn. ....	27
3.11. Biểu đồ tuần tự ca sử dụng tạo món ăn.....	27
3.12. Biểu đồ hoạt động ca sử dụng lưu nhà hàng. ....	29
3.13. Biểu đồ tuần tự ca sử dụng lưu nhà hàng. ....	30
3.14. Sơ đồ thực thể quan hệ dữ liệu. ....	31
4.1. So sánh độ chính xác giữa các mô hình nhận diện món ăn. ....	35
4.2. Kiến trúc hệ thống.....	36
4.3. Giao diện trang chủ. ....	39
4.4. Giao diện hồ sơ cá nhân.....	40
4.5. Giao diện quản lý nhà hàng.....	40
4.6. Giao diện quản lý món ăn từng nhà hàng.....	41
4.7. Giao diện tìm kiếm nhà hàng bằng hình ảnh.....	42
4.8. Giao diện khám phá tự động.....	42
4.9. Giao diện hệ thống đề xuất các món ăn và thời tiết.....	43
4.10. Giao diện hệ thống đề xuất các món ăn dựa trên sở thích. ....	43
4.11. Giao diện hệ thống đề xuất các nhà hàng. ....	44
4.12. Giao diện chi tiết nhà hàng. ....	44
4.13. Giao diện các nhà hàng yêu thích của người dùng.....	45
4.14. Giao diện các nhà hàng xung quanh người dùng.....	45

5.1. Các API được kiểm thử với Postman.....	49
5.2. Kiểm thử đơn vị lớp dịch vụ. ....	50
5.3. Kiểm thử hiệu suất nhận diện món ăn của mô hình ResNet152V2.....	51

## Danh sách bảng

3.1. Mô tả yêu cầu phi chức năng hiệu suất.	11
3.2. Mô tả yêu cầu phi chức năng bảo mật.	12
3.3. Mô tả yêu cầu phi chức năng bảo trì.	12
3.4. Mô tả yêu cầu phi chức năng khả năng sử dụng.	12
3.5. Mô tả yêu cầu phi chức năng nhận diện ảnh.	13
3.6. Mô tả yêu cầu phi chức năng xử lý lỗi, phản hồi người dùng.	13
3.7. Mô tả chi tiết ca sử dụng đăng ký tài khoản.	16
3.8. Mô tả chi tiết ca sử dụng khám phá nhà hàng tự động.	19
3.9. Mô tả chi tiết ca sử dụng tìm kiếm nhà hàng bằng hình ảnh.	22
3.10. Mô tả chi tiết ca sử dụng tạo nhà hàng.	24
3.11. Mô tả chi tiết ca sử dụng tạo món ăn.	26
3.12. Mô tả chi tiết ca sử dụng lưu nhà hàng.	28

## Các từ viết tắt

Từ viết tắt	Từ đầy đủ tiếng Anh	Nghĩa tiếng Việt
API	Application Programming Interface	Giao diện lập trình ứng dụng
GPS	Global Positioning System	Hệ thống định vị toàn cầu
AI	Artificial Intelligence	Trí tuệ nhân tạo
MySQL	My Structured Query Language	Ngôn ngữ truy vấn có cấu trúc (CSDL)
DL	Deep Learning	Học sâu
OpenCV	Open Source Computer Vision	Thư viện mã nguồn mở xử lý ảnh
JWT	JSON Web Token	Chuỗi mã hóa định danh người dùng
SOLID	Single responsibility, Open/closed, Liskov substitution, Interface segregation, Dependency inversion	Nguyên tắc thiết kế hướng đối tượng
DRY	Don't Repeat Yourself	Nguyên tắc lập trình: đừng lặp lại chính mình
CI/CD	Continuous Integration / Continuous Deployment	Tích hợp liên tục / Triển khai liên tục
CSRF	Cross-Site Request Forgery	Giả mạo yêu cầu liên trang
XSS	Cross-Site Scripting	Tấn công kịch bản giữa các trang
ML	Machine Learning	Học máy
AI	Artificial Intelligence	Trí tuệ nhân tạo
HTTP	Hypertext Transfer Protocol	Giao thức truyền tải siêu văn bản
MVC	Model View Controller	Mô hình điều khiển - hiển thị - dữ liệu
ORM	Object-Relational Mapping	Ánh xạ đối tượng - quan hệ
CNN	Convolutional Neural Network	Mạng nơ-ron tích chập

Từ viết tắt	Từ đầy đủ tiếng Anh	Nghĩa tiếng Việt
LLaMA	Large Language Model Meta AI	Mô hình ngôn ngữ lớn của Meta
GIT	Global Information Tracker (phiên bản phổ biến)	Hệ thống quản lý phiên bản phân tán
FAQ	Frequently Asked Questions	Câu hỏi thường gặp
WEB	World Wide Web	Mạng lưới toàn cầu
UI	User Interface	Giao diện người dùng

# Chương 1.

## Đặt vấn đề

### 1.1. Giới thiệu bài toán

Trong nhịp sống hiện đại không ngừng chuyển động, nhu cầu khám phá và trải nghiệm các dịch vụ địa phương, đặc biệt là những địa điểm ẩm thực độc đáo, đã trở thành một phần không thể thiếu trong cuộc sống hàng ngày của người dân cũng như khách du lịch. Tuy nhiên, việc tìm kiếm thông tin về các quán ăn, nhà hàng phù hợp thường gặp phải nhiều thách thức: thông tin bị phân tán, thiếu cập nhật, hoặc không đáp ứng được nhu cầu cá nhân của từng người dùng. Hiểu được thực trạng đó, khóa luận phát triển hệ thống Khám Phá Xung Quanh với mục tiêu xây dựng một hệ thống tìm kiếm dịch vụ địa phương thông minh, cho phép người dùng dễ dàng khám phá, tiếp cận và lựa chọn những quán ăn, nhà hàng phù hợp nhất ngay tại khu vực xung quanh vị trí của họ, chỉ trong vài thao tác đơn giản.

Hơn cả một công cụ tìm kiếm thông thường, hệ thống hướng tới việc trở thành một nền tảng kết nối toàn diện, gắn kết chặt chẽ giữa người dùng và các nhà cung cấp dịch vụ, mang đến trải nghiệm tìm kiếm được cá nhân hóa, tối ưu và tiện lợi, thông qua việc cung cấp thông tin chi tiết, chính xác và luôn được cập nhật về các địa điểm ẩm thực, từ vị trí, menu món ăn, đánh giá cho đến các yếu tố bối cảnh như thời tiết hay sở thích cá nhân. Không chỉ giúp người dùng nhanh chóng đưa ra quyết định phù hợp với nhu cầu và sở thích của mình, hệ thống còn góp phần quan trọng vào sự phát triển bền vững của ngành ẩm thực và du lịch địa phương, mở ra cơ hội để các doanh nghiệp gia tăng khả năng tiếp cận khách hàng, quảng bá thương hiệu một cách hiệu quả và nâng cao sức cạnh tranh trên thị trường, từ đó tạo nên một hệ sinh thái dịch vụ địa phương bền vững và phát triển.

Với tầm nhìn chiến lược và những giá trị thiết thực mà hệ thống mang lại, Khám Phá Xung Quanh không chỉ dừng lại ở việc đáp ứng nhu cầu tìm kiếm của người dùng, mà còn trở thành một cầu nối quan trọng, góp phần thúc đẩy sự tăng trưởng kinh tế, lan tỏa vẻ đẹp văn hóa ẩm thực đặc sắc của đất nước, đồng thời tạo nên một cộng đồng kết nối bền vững giữa người dùng và các nhà cung cấp dịch vụ. Khóa luận không chỉ là một giải pháp công nghệ mà còn là một bước tiến nhằm nâng cao chất lượng trải nghiệm của người dùng, đồng thời hỗ trợ sự phát triển của các doanh nghiệp địa phương trong bối cảnh cạnh tranh ngày càng khốc liệt.

## 1.2. Hiện trạng và mục tiêu khoá luận

Trong bối cảnh nhu cầu trải nghiệm ẩm thực ngày càng phong phú, người dùng hiện nay thường tìm đến những nền tảng quen thuộc như Google Maps, Foody hay ShoppeFood để khám phá các địa điểm ăn uống xung quanh. Tuy nhiên, phần lớn các hệ thống này vẫn dừng lại ở mức hỗ trợ tìm kiếm thông qua từ khóa hoặc vị trí địa lý, trong khi lại chưa thực sự thấu hiểu được sở thích cá nhân, tâm trạng hay điều kiện sử dụng thực tế của người dùng. Đặc biệt, nhu cầu tìm kiếm món ăn thông qua hình ảnh, một nhu cầu hết sức tự nhiên khi người dùng bắt gặp một món ăn hấp dẫn nhưng không biết tên gọi, lại chưa được các nền tảng hiện nay khai thác hiệu quả. Bên cạnh đó, từ góc độ của những người cung cấp dịch vụ như chủ quán ăn hay nhà hàng, nhu cầu giới thiệu và quảng bá món ăn, dịch vụ của mình đến khách hàng tiềm năng cũng ngày một tăng cao. Họ mong muốn có một không gian giống như một mạng xã hội nhỏ, nơi có thể chia sẻ hình ảnh, cập nhật thực đơn, tương tác với khách hàng và tiếp cận đúng đối tượng người dùng có nhu cầu. Tuy nhiên, các nền tảng hiện nay lại chưa thực sự hỗ trợ tốt cho vai trò này, khiến việc quảng bá vẫn còn khá thủ công và thiếu hiệu quả.

Khắc phục những hạn chế đó, hệ thống được xây dựng như một giải pháp toàn diện, tiên phong trong việc kết hợp công nghệ nhận diện hình ảnh thông minh. Chỉ với một bức ảnh, người dùng có thể nhanh chóng tìm ra tên món ăn, địa điểm phục vụ gần nhất, thậm chí là những đánh giá, nhận xét từ cộng đồng. Không dừng lại ở đó, hệ thống còn kết hợp các công nghệ hiện đại như định vị GPS và ứng dụng mô hình ngôn ngữ lớn LLaMA để đề xuất món ăn, gợi ý dựa trên các tiêu chí linh hoạt như sở thích, điều kiện thời tiết, thời gian trong ngày và xu hướng thưởng thức ẩm thực tại từng khu vực. Nhờ đó, hệ thống không chỉ dừng lại ở vai trò một công cụ tìm kiếm thông thường, mà còn trở thành người bạn đồng hành tinh tế, giúp người dùng dễ dàng kết nối, khám phá và trải nghiệm trọn vẹn thế giới ẩm thực xung quanh mình. Hệ thống không chỉ đóng vai trò là một công cụ tìm kiếm thông minh cho người dùng, mà còn là cầu nối hiệu quả giữa khách hàng và người cung cấp dịch vụ.

Đây chính là hướng tiếp cận mới mẻ, mở ra tiềm năng xây dựng một nền tảng tìm kiếm và đề xuất dịch vụ địa phương thân thiện, thông minh và giàu tính tương tác.

## 1.3. Bố cục khóa luận

Các chương còn lại của khóa luận sẽ được trình bày như sau. Chương 2 trình bày cơ sở lý thuyết về các công nghệ đã áp dụng để xây dựng hệ thống. Chương

3 phân tích và mô tả yêu cầu của bài toán, đặc tả các ca sử dụng, xây dựng biểu đồ tuần tự, thiết kế cơ sở dữ liệu hệ thống. Chương 4 xây dựng và triển khai hệ thống, trình bày việc áp dụng công nghệ vào xây dựng triển khai hệ thống và một số hình ảnh về những chức năng đã được đề cập trong khóa luận. Chương 5 trình bày kết quả kiểm thử cho hệ thống. Chương 6 trình bày tổng quan kết quả những gì đã đạt được cùng với các định hướng phát triển tiếp theo.

# Chương 2.

## Kiến thức nền tảng

### 2.1. Phát triển backend với Java Spring

#### 2.1.1. Giới thiệu chung

Spring là một nền tảng mã nguồn mở mạnh mẽ dành cho phát triển ứng dụng Java, được thiết kế nhằm đơn giản hóa việc xây dựng các ứng dụng doanh nghiệp. Spring cung cấp một bộ công cụ toàn diện với kiến trúc linh hoạt, cho phép lập trình viên dễ dàng cấu hình, mở rộng và tích hợp nhiều thành phần. Một trong những module cốt lõi của Spring là Spring Boot, giúp rút ngắn thời gian phát triển bằng cách cung cấp các cấu hình mặc định, tự động hóa việc khởi tạo ứng dụng và đơn giản hóa quá trình triển khai. Bên cạnh đó, các module như Spring Data JPA, Spring Security, Spring Web, và Spring Validation giúp xử lý toàn diện các tác vụ như truy vấn dữ liệu, bảo mật, xây dựng API RESTful và kiểm tra dữ liệu đầu vào. Nhờ khả năng mở rộng, tổ chức theo mô-đun và hỗ trợ tốt nguyên lý lập trình hướng đối tượng, Spring hiện là lựa chọn phổ biến trong phát triển các hệ thống backend hiện đại, quy mô từ nhỏ đến lớn.

#### 2.1.2. Đặc điểm nổi bật

Ứng dụng backend trong hệ thống được phát triển bằng Java, sử dụng các thành phần trong hệ sinh thái Spring. Dưới đây là những đặc điểm nổi bật của Spring:

- Kiến trúc module linh hoạt: Cho phép lựa chọn và kết hợp các module phù hợp với nhu cầu thực tế của dự án.
- Spring Boot: Hỗ trợ tạo ứng dụng độc lập, cấu hình tối giản, dễ dàng triển khai và mở rộng.
- Spring Data JPA: Đơn giản hóa việc thao tác với cơ sở dữ liệu thông qua các repository và truy vấn tự động.
- Spring Security: Cung cấp giải pháp bảo mật mạnh mẽ, dễ tích hợp xác thực người dùng và phân quyền.
- Spring Web: Hỗ trợ xây dựng RESTful API một cách rõ ràng, chuẩn hóa và hiệu quả.
- Khả năng tích hợp tốt: Dễ dàng tích hợp với các công nghệ khác như cơ sở dữ liệu, hệ thống CI/CD, công cụ giám sát và các dịch vụ bên ngoài.

- Quản lý phụ thuộc bằng Spring IoC Container: Giúp tăng tính tái sử dụng và dễ kiểm thử.
- Cộng đồng lớn và tài liệu phong phú: Hỗ trợ mạnh mẽ từ cộng đồng lập trình Java trên toàn thế giới.

## 2.2. Phát triển giao diện với ReactJS

### 2.2.1. Giới thiệu chung

ReactJS là một thư viện JavaScript mã nguồn mở do Facebook phát triển, dùng để xây dựng giao diện người dùng cho các ứng dụng web hiện đại. Với triết lý xây dựng ứng dụng dưới dạng các component tái sử dụng, React giúp tổ chức mã nguồn rõ ràng, dễ bảo trì và dễ mở rộng. Thông qua cơ chế Virtual DOM, React tối ưu hiệu năng bằng cách giảm thiểu thao tác trực tiếp với DOM thật, chỉ cập nhật những phần thực sự thay đổi. Ngoài ra, React hỗ trợ mạnh mẽ việc quản lý trạng thái và xử lý tương tác phức tạp trên giao diện người dùng, rất phù hợp với các ứng dụng web có tính năng động cao như dashboard, hệ thống quản lý hay các nền tảng thương mại điện tử.

### 2.2.2. Đặc điểm nổi bật

Trong khoá luận này, ReactJS được sử dụng để xây dựng phần giao diện người dùng, kết nối linh hoạt với backend thông qua API. Các đặc điểm nổi bật của React bao gồm:

- Component based: Giao diện được chia thành các thành phần nhỏ, độc lập, giúp dễ tái sử dụng và bảo trì.
- Virtual DOM: Cải thiện hiệu năng bằng cách cập nhật có chọn lọc các phần tử giao diện thay vì làm việc trực tiếp với DOM thật.
- JSX (JavaScript XML): Cho phép viết HTML bên trong JavaScript, giúp tăng tính trực quan và dễ phát triển.
- Hooks: Hỗ trợ quản lý trạng thái và vòng đời component một cách hiệu quả trong các component dạng hàm.
- Tương tác API linh hoạt: Dễ dàng tích hợp với các API RESTful hoặc GraphQL từ backend.
- Hệ sinh thái phong phú: Cộng đồng lớn, hỗ trợ nhiều thư viện bổ trợ như React Router, Redux, Tailwind CSS, Antd, v.v.

- Tối ưu trải nghiệm người dùng: Giao diện mượt mà, phản hồi nhanh, phù hợp cho cả thiết bị máy tính và di động.

## 2.3. Phát triển API backend với FastAPI

### 2.3.1. Giới thiệu chung

FastAPI là một framework web hiện đại dành cho Python, được thiết kế để xây dựng các API RESTful nhanh, dễ bảo trì và hiệu năng cao. Với cú pháp ngắn gọn, hỗ trợ khai báo kiểu dữ liệu tường minh và tích hợp sẵn các công cụ như Swagger UI, FastAPI đang ngày càng được ưa chuộng trong các dự án phát triển web backend. Trong khóa luận này, FastAPI được sử dụng để xây dựng backend xử lý các chức năng AI như nhận diện món ăn qua ảnh, cung cấp các API giao tiếp giữa hệ thống frontend và các mô hình học sâu.

### 2.3.2. Đặc điểm nổi bật

Vìệc sử dụng FastAPI trong hệ thống mang lại nhiều lợi ích:

- Hiệu năng cao: FastAPI được xây dựng trên nền tảng Starlette và Pydantic, tối ưu hóa cho các tác vụ bất đồng bộ, giúp xử lý lượng lớn request đồng thời với độ trễ thấp.
- Khai báo rõ ràng, kiểm tra kiểu dữ liệu: FastAPI hỗ trợ typing của Python, giúp phát hiện lỗi ngay từ lúc viết code và dễ dàng kiểm tra dữ liệu đầu vào/đầu ra qua Pydantic.
- Tự động sinh tài liệu API: FastAPI tích hợp Swagger UI và Redoc để tự động tạo ra tài liệu API trực quan, hỗ trợ kiểm thử API nhanh chóng.
- Tích hợp mô hình AI dễ dàng: Cho phép tích hợp trực tiếp với các thư viện như TensorFlow, OpenCV hoặc PyTorch để triển khai mô hình học máy phục vụ xử lý ảnh, nhận dạng, v.v.
- Hỗ trợ tốt JSON và HTTP: FastAPI tối ưu cho các hệ thống RESTful, đặc biệt phù hợp với các ứng dụng frontend hiện đại sử dụng React, Vue hoặc mobile app.
- Triển khai linh hoạt: Có thể dễ dàng container hóa với Docker, triển khai trên các nền tảng cloud như Heroku, AWS hoặc server riêng.

## 2.4. Quản lý và truy vấn dữ liệu với MySQL

### 2.4.1. Giới thiệu chung

MySQL là hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) mã nguồn mở phổ biến, được sử dụng rộng rãi trong các ứng dụng web và doanh nghiệp. MySQL hỗ trợ ngôn ngữ truy vấn có cấu trúc (SQL), giúp lưu trữ, tổ chức và truy xuất dữ liệu một cách hiệu quả. Với khả năng xử lý nhanh, tính ổn định cao và khả năng mở rộng, MySQL phù hợp để quản lý dữ liệu của các hệ thống từ nhỏ đến lớn. Trong khoá luận này, MySQL được lựa chọn làm hệ quản trị cơ sở dữ liệu chính để lưu trữ thông tin người dùng, nhà hàng, món ăn, đánh giá và các dữ liệu liên quan đến quá trình tương tác người dùng.

### 2.4.2. Đặc điểm nổi bật

Việc sử dụng MySQL trong hệ thống mang lại nhiều lợi ích, bao gồm:

- Cấu trúc dữ liệu rõ ràng: Dữ liệu được lưu trữ dưới dạng bảng quan hệ, giúp tổ chức thông tin chặt chẽ và dễ truy vấn.
- Ngôn ngữ SQL mạnh mẽ: Hỗ trợ truy vấn linh hoạt, từ đơn giản đến phức tạp với các phép nối bảng, lọc, nhóm và sắp xếp dữ liệu.
- Tối ưu truy vấn: Cơ chế lập chỉ mục giúp tăng tốc độ truy vấn trên các bảng lớn.
- Tích hợp tốt với Spring Data JPA: Đã dàng ánh xạ dữ liệu từ Java sang bảng CSDL thông qua các thực thể entity, giúp tăng hiệu quả phát triển và bảo trì hệ thống.
- Độ tin cậy và bảo mật: Hỗ trợ các cơ chế phân quyền, sao lưu, và khôi phục dữ liệu, đảm bảo an toàn cho hệ thống.
- Khả năng mở rộng: Phù hợp với các hệ thống từ nhỏ đến quy mô lớn với nhu cầu truy cập đồng thời cao.

## 2.5. Nhận diện và xử lý ảnh món ăn

### 2.5.1. Giới thiệu chung

Nhận diện món ăn từ hình ảnh là một ứng dụng phổ biến của thị giác máy tính, giúp máy tính hiểu và phân loại nội dung hình ảnh thông qua các mô hình học sâu. Trong bối cảnh người dùng ngày càng có xu hướng tìm kiếm món ăn thông qua hình ảnh, công nghệ này trở nên đặc biệt hữu ích đối với các hệ thống gợi ý và tìm kiếm ẩm thực. Việc nhận diện chính xác các món ăn từ hình ảnh đòi hỏi

hệ thống phải xử lý tốt các đặc trưng hình ảnh như màu sắc, hình dạng, kết cấu, vốn có thể thay đổi lớn do góc chụp, ánh sáng, hoặc cách bày trí.

### 2.5.2. Công nghệ sử dụng

Nhận diện hình ảnh món ăn thường sử dụng mạng nơ-ron tích chập (Convolutional Neural Networks – CNN), một loại mô hình học sâu chuyên biệt cho xử lý ảnh. Các mô hình CNN có khả năng tự động học các đặc trưng phân biệt từ dữ liệu huấn luyện, giúp phân loại hình ảnh một cách chính xác.

Một số mô hình CNN nổi bật thường được dùng trong bài toán phân loại món ăn gồm:

- ResNet: Được biết đến với khả năng huấn luyện hiệu quả các mạng rất sâu, ResNet giải quyết vấn đề suy giảm độ chính xác nhờ cơ chế kết nối tắt (skip connections).
- InceptionResNet: Kết hợp giữa kiến trúc Inception và ResNet, mô hình này vừa khai thác được đặc trưng đa tỉ lệ vừa giữ được độ sâu của mạng, phù hợp với dữ liệu phức tạp như ảnh món ăn.
- MobileNet, EfficientNet: Là các mô hình nhẹ, hiệu quả, phù hợp triển khai trên các thiết bị có tài nguyên hạn chế như điện thoại hoặc web.

Ngoài ra, các thư viện như TensorFlow và Keras thường được sử dụng để xây dựng, huấn luyện và triển khai mô hình học sâu. Thư viện OpenCV cũng đóng vai trò quan trọng trong khâu tiền xử lý hình ảnh, bao gồm cắt, đổi kích thước, chuẩn hóa hoặc tăng cường dữ liệu (augmentation) nhằm cải thiện khả năng học của mô hình.

## 2.6. Triển khai hệ thống với quy trình CI/CD

### 2.6.1. Giới thiệu chung

CI/CD (Continuous Integration/Continuous Deployment) là một phương pháp phát triển phần mềm hiện đại giúp tự động hóa quy trình kiểm thử, xây dựng và triển khai ứng dụng. Với CI/CD, mã nguồn sau khi được cập nhật sẽ tự động được kiểm tra, đóng gói và triển khai mà không cần thao tác thủ công. Phương pháp này giúp giảm thiểu lỗi trong quá trình phát triển, tăng tốc độ đưa sản phẩm đến tay người dùng, và duy trì sự ổn định của hệ thống. Trong khoa luận này, hệ thống được triển khai với sự hỗ trợ của GitHub Actions kết hợp Docker, cho phép đóng gói cả backend và frontend và database thành các container độc lập, sẵn sàng triển khai lên server hoặc nền tảng cloud.

## 2.6.2. Đặc điểm nổi bật

Việc áp dụng CI/CD với Docker trong hệ thống mang lại nhiều lợi ích:

- Tự động hóa quy trình: Các bước kiểm thử, build Docker image, và triển khai đều được thực hiện tự động qua GitHub Actions.
- Tái sử dụng và nhất quán: Docker đảm bảo môi trường chạy ứng dụng nhất quán giữa các máy phát triển, môi trường staging và production.
- Tích hợp trực tiếp với GitHub: Mỗi lần có thay đổi được đẩy lên repository, workflow CI/CD sẽ tự động kích hoạt.
- Triển khai dễ dàng: Docker image sau khi build có thể được đẩy lên Docker Hub hoặc container registry, sẵn sàng cho triển khai tại bất kỳ máy chủ nào hỗ trợ Docker.
- Tách biệt frontend và backend: Hai phần có thể được đóng gói và triển khai độc lập, giúp dễ dàng kiểm thử và cập nhật từng thành phần.
- Hỗ trợ mở rộng: Có thể dễ dàng tích hợp thêm các bước kiểm thử đơn vị, kiểm tra mã nguồn, hoặc triển khai lên các nền tảng như AWS, GCP, hoặc máy chủ nội bộ.

# Chương 3.

## Phân tích và thiết kế hệ thống

### 3.1. Đặc tả yêu cầu

#### 3.1.1. Yêu cầu chức năng

Hệ thống được thiết kế để hỗ trợ khách du lịch và người dùng tìm kiếm, khám phá, và lập lịch trải nghiệm các dịch vụ địa phương, tập trung chủ yếu vào lĩnh vực ẩm thực. Hệ thống cung cấp các tính năng thông minh như tìm kiếm bằng hình ảnh, đề xuất cá nhân hóa dựa trên sở thích, vị trí, thời tiết, và tích hợp công nghệ AI để nâng cao trải nghiệm người dùng.

##### 3.1.1.1. Đăng nhập và quản lý tài khoản

- Đăng nhập/Đăng ký: Người dùng có thể đăng nhập hoặc đăng ký tài khoản bằng email, số điện thoại.
- Cập nhật hồ sơ cá nhân: Người dùng có thể cập nhật thông tin cá nhân (tên, ảnh đại diện, sở thích ăn uống, lịch sử tìm kiếm).
- Quản lý mật khẩu: Hỗ trợ thay đổi mật khẩu và khôi phục tài khoản qua email.
- Bảo mật: Sử dụng JWT Authentication để xác thực và bảo vệ thông tin người dùng.

##### 3.1.1.2. Tìm kiếm và khám phá

- Tìm kiếm theo tên: Người dùng có thể tìm kiếm nhà hàng bằng văn bản.
- Tìm kiếm theo vị trí: Người dùng có thể tìm kiếm nhà hàng theo địa điểm đã lọc trước.
- Tìm kiếm bằng hình ảnh: Người dùng có thể tải lên hình ảnh của món ăn để tìm ra tên món ăn và địa điểm phục vụ gần nhất.
- Tìm kiếm nhà hàng xung quanh: Tính năng để khám phá nhà hàng và món ăn trong khu vực mà không cần tìm kiếm cụ thể.

##### 3.1.1.3. Đề xuất cá nhân hóa

- Dựa trên sở thích: Đề xuất nhà hàng, món ăn dựa trên sở thích ăn uống của người dùng.
- Dựa trên vị trí: Gợi ý địa điểm dựa trên vị trí hiện tại của người dùng.

- Dựa trên thời tiết: Đề xuất món ăn hoặc nhà hàng phù hợp với điều kiện thời tiết hiện tại.
- Biểu đạt tâm trạng ăn uống: Người dùng có thể biểu đạt tâm trạng hoặc sở thích ăn uống hiện tại để nhận gợi ý phù hợp.

#### **3.1.1.4. Tích hợp tính năng chat với AI**

Trả lời câu hỏi tự nhiên: Cho phép người dùng hỏi về thông tin ẩm thực bằng ngôn ngữ tự nhiên và nhận câu trả lời từ AI.

#### **3.1.1.5. Quản lý hồ sơ cho nhà cung cấp dịch vụ**

Tạo và quản lý hồ sơ nhà hàng: Nhà hàng có thể tạo và cập nhật thông tin về nhà hàng, thực đơn, địa điểm, v.v.

### **3.1.2. Yêu cầu phi chức năng**

Hệ thống được thiết kế để hỗ trợ người dùng tìm kiếm và khám phá các dịch vụ ẩm thực địa phương một cách thông minh và tiện lợi. Để đảm bảo hệ thống hoạt động hiệu quả, an toàn, và mang lại trải nghiệm tốt, các yêu cầu phi chức năng dưới đây được xác định, bao gồm các khía cạnh như hiệu suất, khả năng mở rộng, bảo mật, khả năng sử dụng, độ tin cậy, và khả năng bảo trì.

#### **3.1.2.1. Hiệu suất**

Mô tả chi tiết yêu cầu phi chức năng hiệu suất được thể hiện ở bảng 3.1 dưới đây.

Bảng 3.1. Mô tả yêu cầu phi chức năng hiệu suất.

<b>Yêu cầu</b>	<b>Giải pháp/mục tiêu</b>
Thời gian phản hồi tìm kiếm	$\leq 3$ giây cho truy vấn tìm kiếm nhà hàng/món ăn
Thời gian nhận diện hình ảnh	$\leq 10$ giây để xử lý hình ảnh và gợi ý địa điểm
Thời gian phản hồi tác vụ cơ bản	$\leq 2$ giây cho đăng nhập, đăng ký, cập nhật hồ sơ
Hiệu suất định vị địa lý	$\leq 3$ giây để xác định vị trí và gợi ý nhà hàng gần nhất

#### **3.1.2.2. Bảo mật**

Mô tả chi tiết yêu cầu phi chức năng bảo mật được thể hiện ở bảng 3.2 dưới đây.

Bảng 3.2. Mô tả yêu cầu phi chức năng bảo mật.

Yêu cầu	Giải pháp/mục tiêu
Mã hóa dữ liệu	Mã hóa AES-256 cho dữ liệu lưu trữ, TLS 1.3+ cho truyền tải
Xác thực và ủy quyền	JWT với thuật toán HS256/RSA256, kèm OAuth2.0 cho bên thứ ba
Bảo vệ lỗ hổng web	Triển khai WAF, kiểm tra đầu vào (input validation), CSRF tokens

### 3.1.2.3. Khả năng bảo trì

Mô tả chi tiết yêu cầu phi chức năng bảo trì được thể hiện ở bảng 3.3 dưới đây.

Bảng 3.3. Mô tả yêu cầu phi chức năng bảo trì.

Yêu cầu	Giải pháp/mục tiêu
Chất lượng mã nguồn	Tuân thủ SOLID, coverage unit test $\geq 90\%$ , Git với quy trình GitFlow
Tài liệu hóa	Swagger cho API, Markdown cho hướng dẫn dev, Wiki người dùng
CI/CD	Docker container hóa, Jenkins/GitHub Actions tự động hóa deploy

### 3.1.2.4. Khả năng sử dụng

Mô tả chi tiết yêu cầu phi chức năng khả năng sử dụng được thể hiện ở bảng 3.4 dưới đây.

Bảng 3.4. Mô tả yêu cầu phi chức năng khả năng sử dụng.

Yêu cầu	Giải pháp/mục tiêu
Giao diện trực quan	Đạt điểm SUS $\geq 85\%$ , hỗ trợ đa ngôn ngữ (tiếng Việt/Anh)
Accessibility	Tuân thủ WCAG 2.1 AA, tương thích trình đọc màn hình

### 3.1.2.5. Mô hình nhận diện ảnh

Mô tả chi tiết yêu cầu phi chức năng nhận diện ảnh được thể hiện ở bảng 3.5 dưới đây.

Bảng 3.5. Mô tả yêu cầu phi chức năng nhận diện ảnh.

Yêu cầu	Giải pháp/mục tiêu
Độ chính xác	Precision $\geq 80\%$ trên tập test (F1-score) với dataset đa dạng
Hiệu suất inference	Tối ưu hóa model bằng TensorRT/TFLite để chạy trên mobile

### 3.1.2.6. Xử lý lỗi và phản hồi người dùng

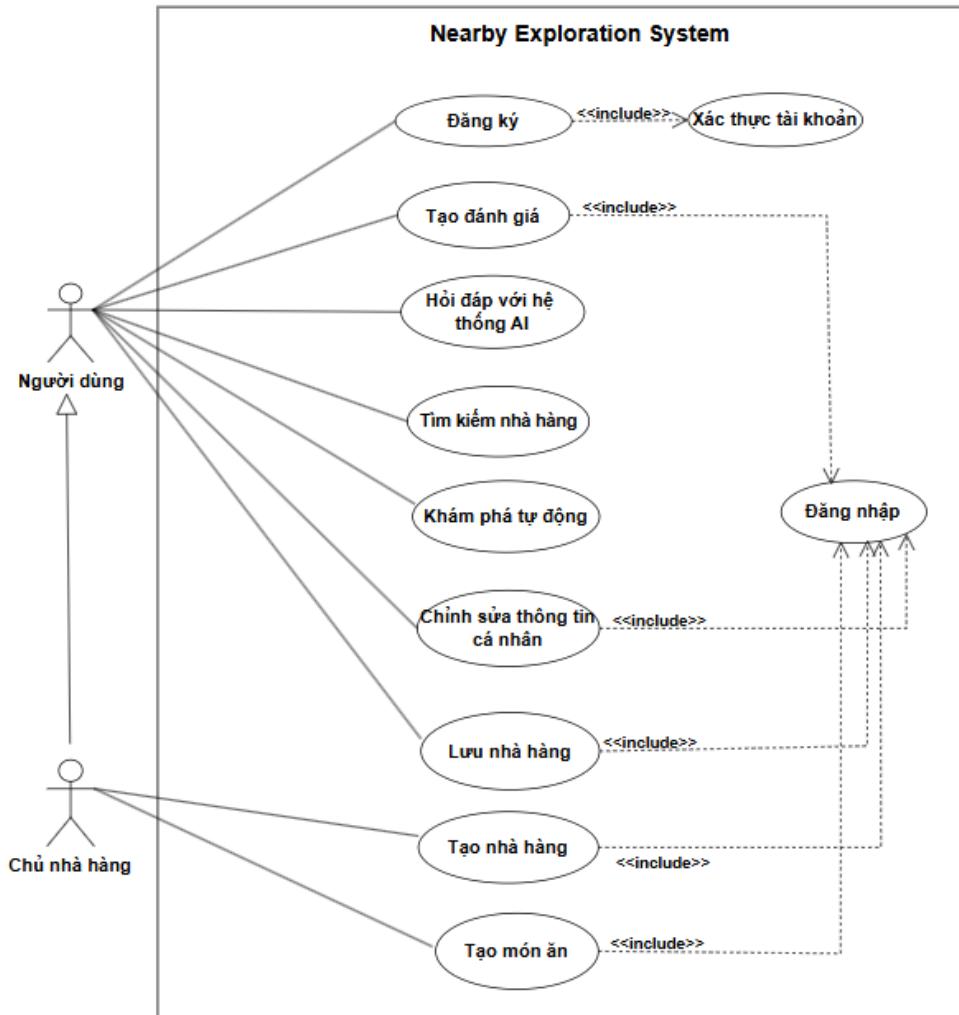
Mô tả chi tiết yêu cầu phi chức năng xử lý lỗi, phản hồi người dùng được thể hiện ở bảng 3.6 dưới đây.

Bảng 3.6. Mô tả yêu cầu phi chức năng xử lý lỗi, phản hồi người dùng.

Yêu cầu	Giải pháp/mục tiêu
Thông báo lỗi	Mã lỗi HTTP tiêu chuẩn + message human-readable (ví dụ: "Không tìm thấy nhà hàng trong bán kính 5km")
Kênh phản hồi	Tích hợp chatbot/Zalo OA + form phản hồi trong app

### 3.2. Biểu đồ tổng quan ca sử dụng hệ thống

Biểu đồ ca sử dụng được trình bày trong hình 3.1 mô tả tổng quan về các mối quan hệ giữa các ca sử dụng trong hệ thống. Các tác nhân chính của hệ thống gồm có: Người dùng và Chủ nhà hàng.



Hình 3.1. Biểu đồ ca sử dụng tổng quát của hệ thống.

### 3.3. Mô tả ca sử dụng

Dưới đây là mô tả chi tiết các ca sử dụng chính của hệ thống, dựa trên các tác nhân người dùng và chủ nhà hàng và các chức năng chính được xác định trong biểu đồ. Mỗi ca sử dụng được mô tả theo cấu trúc thống nhất, bao gồm tác nhân, điều kiện tiên quyết, luồng xử lý chính, luồng xử lý thay thế và điều kiện sau khi

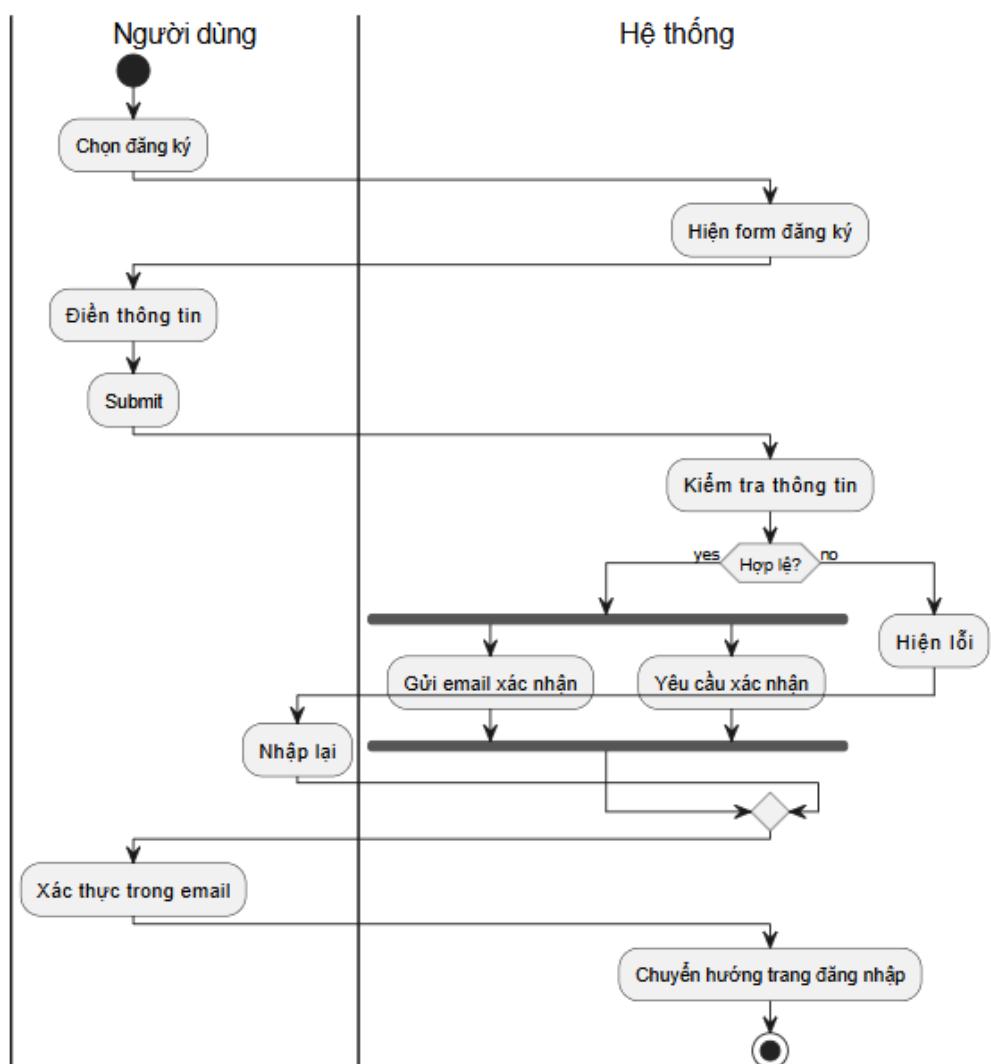
thực hiện. Việc phân tích các ca sử dụng giúp đảm bảo tính đầy đủ của các tính năng và cung cấp cái nhìn trực quan về cách thức vận hành của ứng dụng từ góc độ người dùng.

### **3.3.1. Ca sử dụng đăng ký tài khoản**

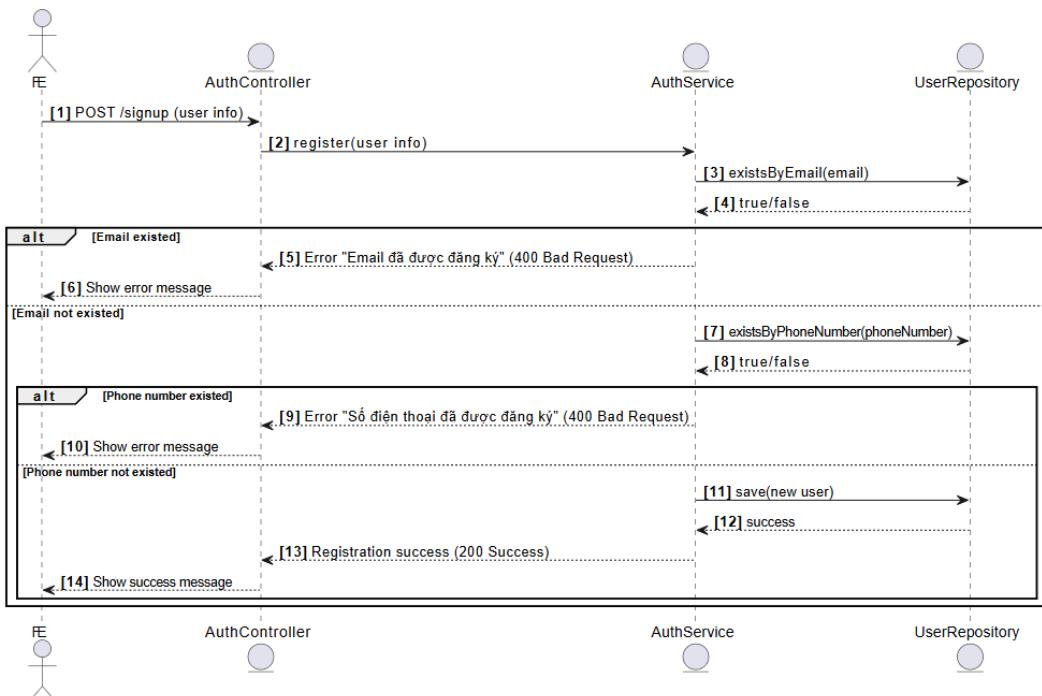
Mô tả chi tiết cho ca sử dụng đăng ký tài khoản được thể hiện ở bảng 3.7 dưới đây. Tiếp theo là hình 3.2 về biểu đồ hoạt động và hình 3.3 về biểu đồ tuần tự của ca sử dụng này.

Bảng 3.7. Mô tả chi tiết ca sử dụng đăng ký tài khoản.

<b>Mô tả</b>	Cho phép người dùng tạo một tài khoản mới trên hệ thống để truy cập các tính năng cá nhân hóa, lưu trữ thông tin, quản lý hồ sơ cá nhân.
<b>Luồng cơ bản</b>	<ol style="list-style-type: none"> <li>1. Người dùng truy cập trang đăng ký tài khoản.</li> <li>2. Hệ thống hiển thị giao diện đăng ký</li> <li>3. Người dùng nhập các thông tin: Email, số điện thoại, mật khẩu, và xác nhận mật khẩu.</li> <li>4. Người dùng nhấn nút Đăng ký để gửi thông tin.</li> <li>5. Hệ thống kiểm tra thông tin: <ul style="list-style-type: none"> <li>- Nếu thông tin hợp lệ, hệ thống gửi email xác thực chứa liên kết kích hoạt tài khoản.</li> <li>- Nếu thông tin không hợp lệ, chuyển sang luồng thay thế.</li> </ul> </li> <li>6. Người dùng nhấp vào liên kết trong email để xác thực tài khoản.</li> <li>7. Hệ thống kích hoạt tài khoản và thông báo: "Đăng ký thành công."</li> </ol>
<b>Luồng thay thế</b>	<ul style="list-style-type: none"> <li>- Nếu email đã được đăng ký trước đó, hệ thống thông báo: "Email đã được đăng ký."</li> <li>- Nếu số điện thoại đã được đăng ký trước đó, hệ thống thông báo: "Số điện thoại đã được đăng ký."</li> <li>- Nếu mật khẩu nhập lại không trùng nhau, hệ thống thông báo: "Mật khẩu chưa khớp."</li> </ul>
<b>Tiền kiện</b>	<ul style="list-style-type: none"> <li>- Hệ thống đã hiển thị thành công trên trình duyệt.</li> <li>- Email và số điện thoại của người dùng chưa được đăng ký trên hệ thống.</li> </ul>
<b>Hậu kiện</b>	<ul style="list-style-type: none"> <li>- Tài khoản được tạo thành công sau khi người dùng xác thực email, và người dùng có thể sử dụng tài khoản để đăng nhập.</li> <li>- Một hồ sơ người dùng được tạo với các thông tin cơ bản và có thể được chỉnh sửa sau này.</li> </ul>
<b>Yêu cầu phi chức năng</b>	<ul style="list-style-type: none"> <li>- Hệ thống phải thông báo cho người dùng khi đang gửi email xác thực.</li> <li>- Thời gian phản hồi của hệ thống khi kiểm tra thông tin đăng ký không vượt quá 3 giây.</li> <li>- Thông tin đăng ký (email, mật khẩu) phải được mã hóa trước khi truyền qua mạng, sử dụng giao thức HTTPS.</li> </ul>



Hình 3.2. Biểu đồ hoạt động ca sử dụng đăng ký tài khoản.



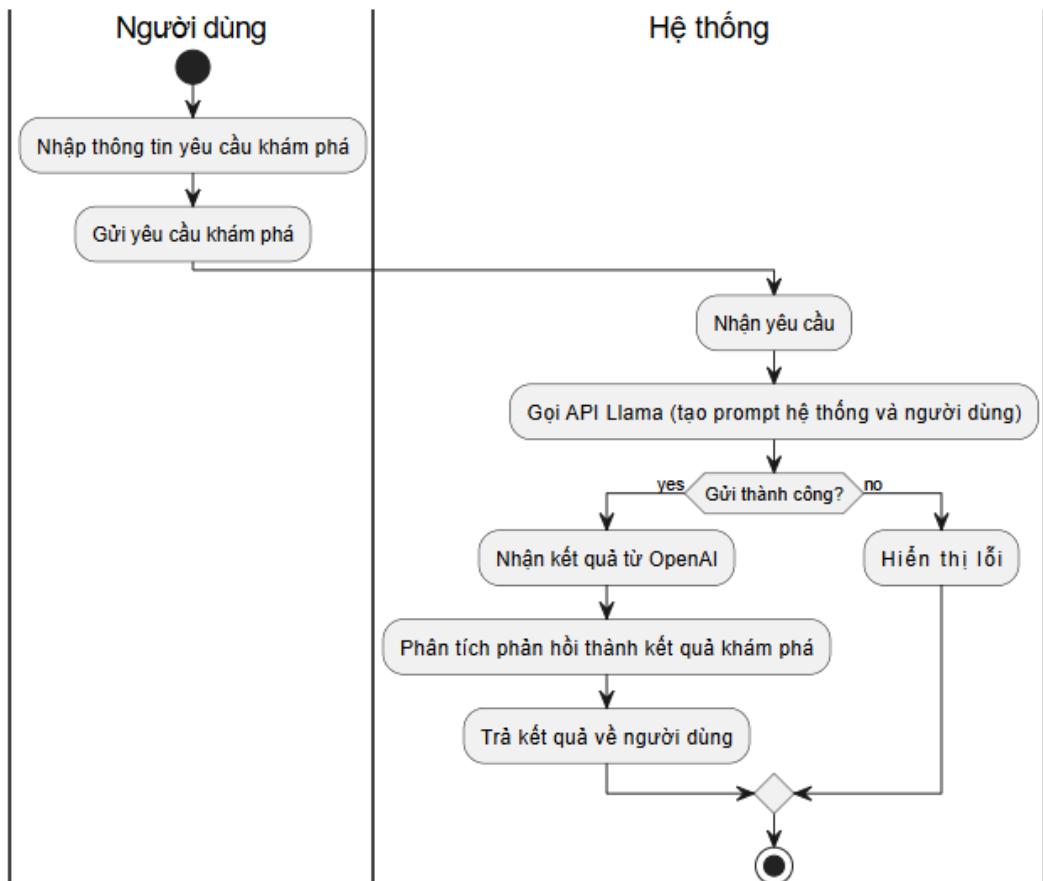
Hình 3.3. Biểu đồ tuần tự ca sử dụng đăng ký tài khoản.

### 3.3.2. Ca sử dụng khám phá tự động

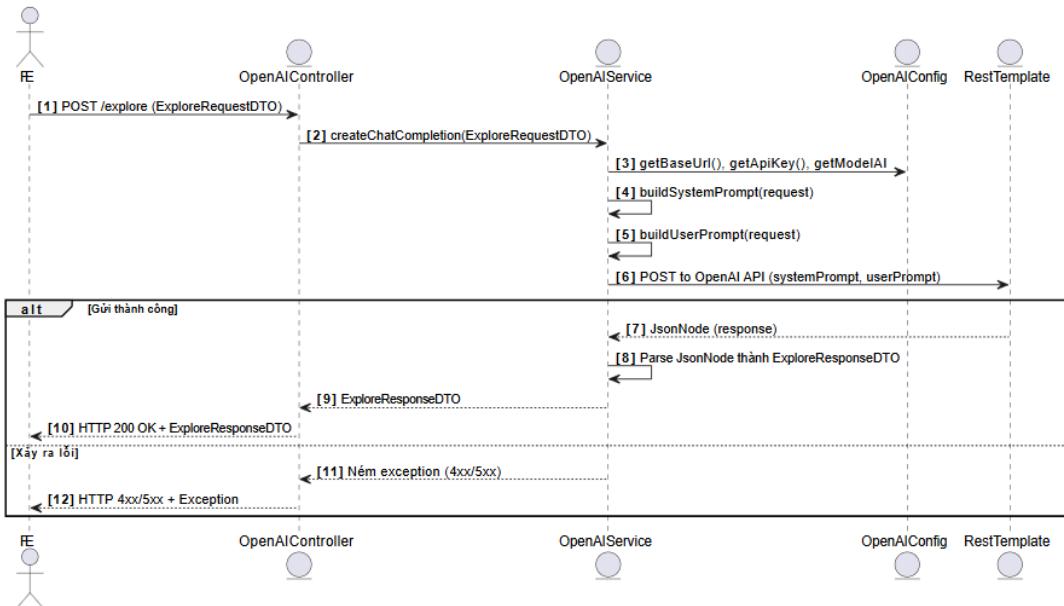
Mô tả chi tiết cho ca sử dụng khám phá tự động được thể hiện ở bảng 3.8 dưới đây. Tiếp theo là hình 3.4 về biểu đồ hoạt động và hình 3.5 về biểu đồ tuần tự của ca sử dụng này.

Bảng 3.8. Mô tả chi tiết ca sử dụng khám phá nhà hàng tự động.

<b>Mô tả</b>	Người dùng nhận gợi ý nhà hàng và món ăn dựa trên sở thích, thời tiết, và vị trí.
<b>Luồng cơ bản</b>	<ol style="list-style-type: none"> <li>1. Người dùng truy cập trang chủ và nhấn nút Khám phá tự động.</li> <li>2. Hệ thống thu thập sở thích, thời tiết và vị trí.</li> <li>3. Hệ thống phân tích: <ul style="list-style-type: none"> <li>- Sở thích (món Á, chay).</li> <li>- Thời tiết (món nóng nếu lạnh, món mát nếu nóng).</li> <li>- Vị trí (nhà hàng gần bán kính 15km).</li> </ul> </li> <li>4. Hệ thống trả về gợi ý: món ăn, nhà hàng .</li> <li>5. Người dùng xem và chọn chi tiết.</li> </ol>
<b>Luồng thay thế</b>	<ul style="list-style-type: none"> <li>- Nếu không có nhà hàng phù hợp, thông báo: "Không tìm thấy nhà hàng."</li> <li>- Nếu lỗi API thời tiết, gợi ý dựa trên sở thích/vị trí, thông báo: "Không lấy được dữ liệu thời tiết."</li> </ul>
<b>Tiền điều kiện</b>	<ul style="list-style-type: none"> <li>- Hệ thống hiển thị thành công, có dữ liệu nhà hàng, thời tiết.</li> <li>- Người dùng cấp quyền vị trí.</li> </ul>
<b>Hậu điều kiện</b>	<ul style="list-style-type: none"> <li>- Hệ thống trả danh sách gợi ý món ăn, nhà hàng.</li> <li>- Người dùng lưu hoặc xem chi tiết nhà hàng/món ăn.</li> </ul>
<b>Yêu cầu phi chức năng</b>	<ul style="list-style-type: none"> <li>- Phản hồi trong 2 giây.</li> <li>- Độ chính xác gợi ý 80%.</li> <li>- Bảo mật dữ liệu vị trí, sở thích.</li> </ul>



Hình 3.4. Biểu đồ hoạt động ca sử dụng khám phá tự động.



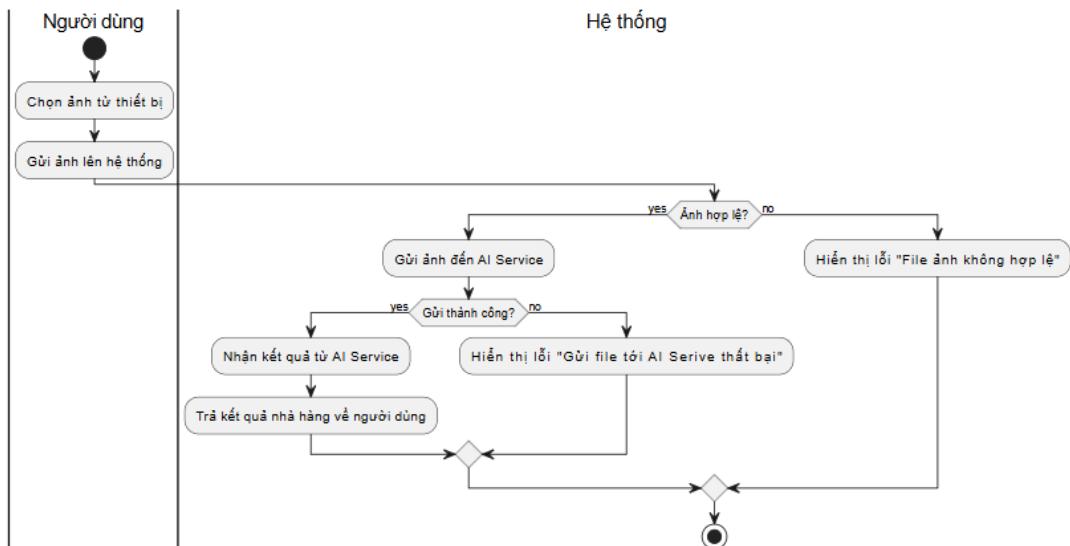
Hình 3.5. Biểu đồ tuần tự ca sử dụng khám phá tự động.

### 3.3.3. Ca sử dụng tìm kiếm nhà hàng bằng hình ảnh

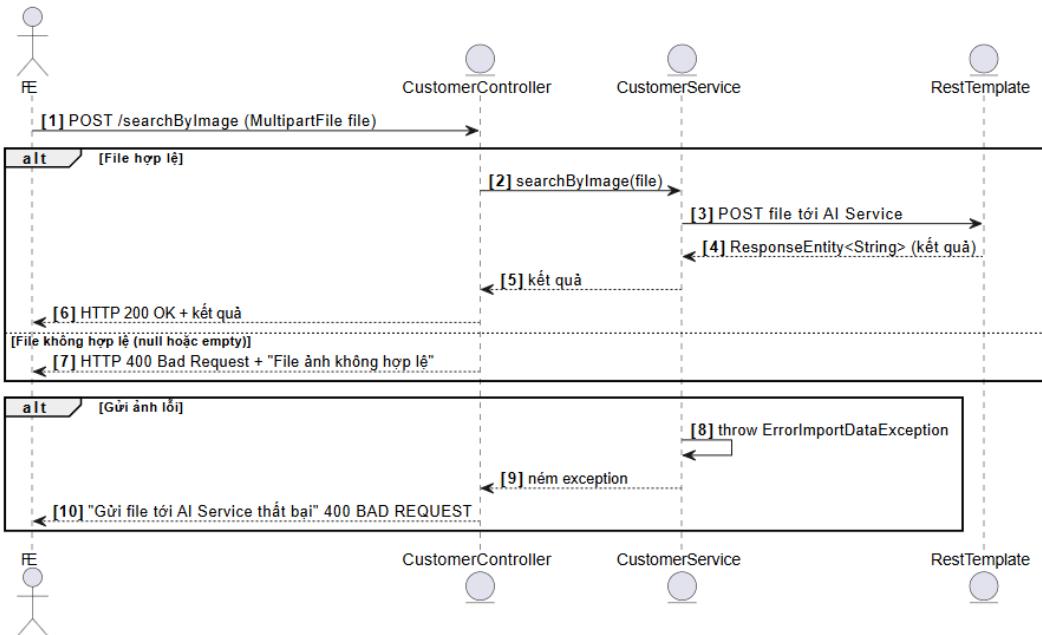
Mô tả chi tiết cho ca sử dụng tìm kiếm nhà hàng bằng hình ảnh được thể hiện ở bảng 3.9 dưới đây. Tiếp theo là hình 3.6 về biểu đồ hoạt động và hình 3.7 về biểu đồ tuần tự của ca sử dụng này.

Bảng 3.9. Mô tả chi tiết ca sử dụng tìm kiếm nhà hàng bằng hình ảnh.

<b>Mô tả</b>	Tính năng cho phép người dùng tìm kiếm nhà hàng bằng hình ảnh món ăn.
<b>Luồng cơ bản</b>	<ol style="list-style-type: none"> <li>Người dùng truy cập bất kỳ trang nào trên hệ thống.</li> <li>Thanh tìm kiếm ở header hiển thị với nút máy ảnh.</li> <li>Người dùng nhấn nút máy ảnh, chọn tải/chụp hình món ăn.</li> <li>Người dùng nhấn nút Tìm kiếm.</li> <li>Hệ thống phân tích hình ảnh, trả về danh sách nhà hàng phù hợp: tên, địa chỉ, đánh giá.</li> <li>Người dùng chọn nhà hàng để xem chi tiết.</li> </ol>
<b>Luồng thay thế</b>	<ul style="list-style-type: none"> <li>Nếu hình ảnh không rõ hoặc không chứa món ăn, hệ thống thông báo: "Hình ảnh không thể phân tích. Vui lòng tải hình ảnh khác."</li> <li>Nếu không tìm thấy nhà hàng phù hợp, hệ thống thông báo: "Không tìm thấy nhà hàng phù hợp với hình ảnh. Thử hình ảnh khác hoặc tìm kiếm bằng từ khóa."</li> <li>Nếu lỗi server hoặc không phân tích được hình ảnh, hệ thống thông báo: "Đã có lỗi xảy ra. Vui lòng thử lại sau."</li> <li>Nếu người dùng không chọn hình ảnh và thoát, hệ thống trở về trạng thái ban đầu.</li> </ul>
<b>Tiền điều kiện</b>	<ul style="list-style-type: none"> <li>Trang hiển thị thành công với thanh tìm kiếm ở header.</li> <li>Hệ thống có dữ liệu nhà hàng và khả năng phân tích hình ảnh.</li> <li>Thiết bị hỗ trợ tải/chụp hình ảnh.</li> </ul>
<b>Hậu điều kiện</b>	<ul style="list-style-type: none"> <li>Hệ thống trả danh sách nhà hàng phù hợp (nếu thành công).</li> <li>Người dùng xem chi tiết nhà hàng hoặc thử lại nếu thất bại.</li> </ul>
<b>Yêu cầu phi chức năng</b>	<ul style="list-style-type: none"> <li>Phản hồi trong 5 giây.</li> <li>Thanh tìm kiếm hiển thị rõ, dễ dùng trên mọi trang.</li> <li>Hỗ trợ định dạng JPEG, PNG, tệp tối đa 5MB.</li> </ul>



Hình 3.6. Biểu đồ hoạt động ca sử dụng tìm kiếm nhà hàng bằng hình ảnh.



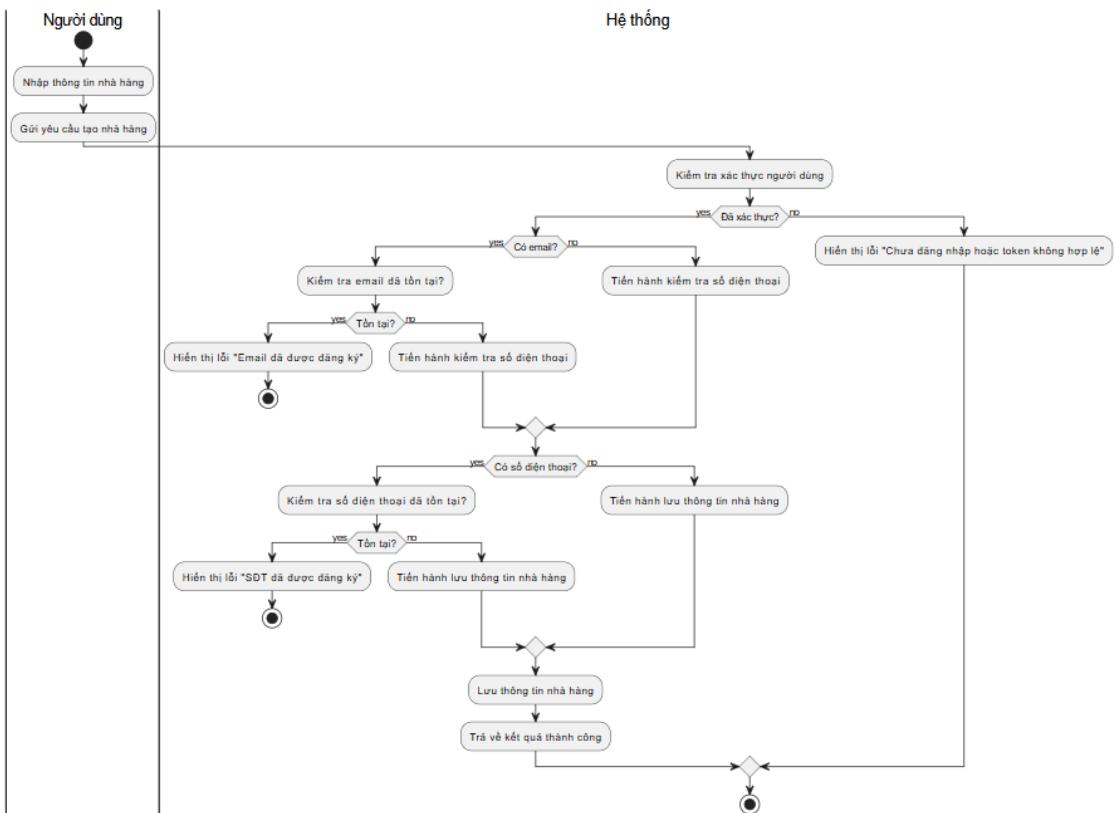
Hình 3.7. Biểu đồ tuần tự ca sử dụng tìm kiếm nhà hàng bằng hình ảnh.

### 3.3.4. Ca sử dụng tạo nhà hàng

Mô tả chi tiết cho ca sử dụng tạo nhà hàng được thể hiện ở bảng 3.10 dưới đây. Tiếp theo là hình 3.8 về biểu đồ hoạt động và hình 3.9 về biểu đồ tuần tự của ca sử dụng này.

Bảng 3.10. Mô tả chi tiết ca sử dụng tạo nhà hàng.

<b>Mô tả</b>	Cho phép người dùng tạo một nhà hàng trên hệ thống.
<b>Luồng cơ bản</b>	<ol style="list-style-type: none"> <li>Người dùng truy cập giao diện tạo nhà hàng.</li> <li>Hệ thống hiển thị biểu mẫu yêu cầu nhập thông tin nhà hàng (tên, địa chỉ, mô tả, v.v.).</li> <li>Người dùng điền đầy đủ thông tin và nhấn nút "Tạo nhà hàng".</li> <li>Hệ thống kiểm tra tính hợp lệ của dữ liệu đầu vào.</li> <li>Nếu hợp lệ, hệ thống lưu thông tin vào cơ sở dữ liệu và hiển thị thông báo tạo thành công.</li> </ol>
<b>Luồng thay thế</b>	<ul style="list-style-type: none"> <li>Nếu người dùng nhập thiếu hoặc sai định dạng dữ liệu, hệ thống hiển thị thông báo lỗi tương ứng để người dùng sửa lại.</li> <li>Nếu xảy ra lỗi từ phía máy chủ, hệ thống thông báo lỗi và không lưu thông tin nhà hàng.</li> </ul>
<b>Tiền điều kiện</b>	<ul style="list-style-type: none"> <li>Người dùng đã đăng nhập vào hệ thống và có quyền tạo mới nhà hàng.</li> <li>Giao diện tạo nhà hàng đã được tải thành công.</li> </ul>
<b>Hậu điều kiện</b>	<ul style="list-style-type: none"> <li>Một nhà hàng mới được lưu trong cơ sở dữ liệu và liên kết với người dùng tạo nó.</li> <li>Người dùng có thể chỉnh sửa thông tin nhà hàng hoặc thêm thực đơn sau đó.</li> </ul>
<b>Yêu cầu phi chức năng</b>	<ul style="list-style-type: none"> <li>Hệ thống cần phản hồi nhanh (dưới 2 giây) sau khi nhấn nút "Thêm nhà hàng".</li> <li>Nếu tạo thành công, cần hiển thị thông báo xác nhận rõ ràng cho người dùng.</li> </ul>



Hình 3.8. Biểu đồ hoạt động ca sử dụng tạo nhà hàng.



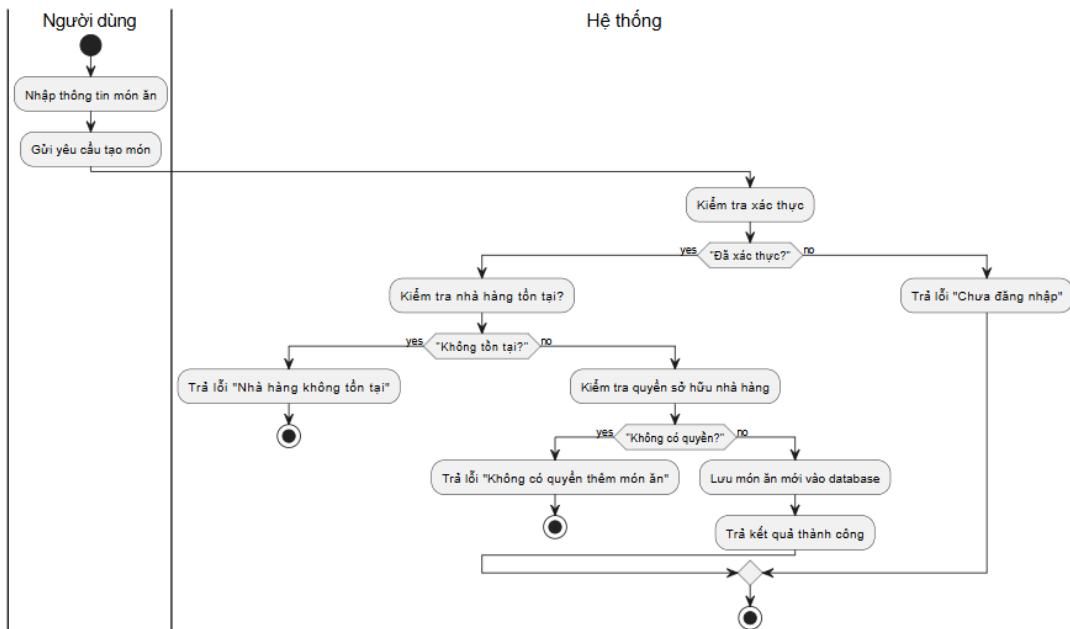
Hình 3.9. Biểu đồ tuần tự ca sử dụng tạo nhà hàng.

### 3.3.5. Ca sử dụng tạo món ăn

Mô tả chi tiết cho ca sử dụng tạo món ăn được thể hiện ở bảng 3.11 dưới đây. Tiếp theo là hình 3.10 về biểu đồ hoạt động và hình 3.11 về biểu đồ tuần tự của ca sử dụng này.

Bảng 3.11. Mô tả chi tiết ca sử dụng tạo món ăn.

<b>Mô tả</b>	Cho phép chủ nhà hàng tạo một món ăn mới trên hệ thống để bổ sung vào món ăn nhà hàng.
<b>Luồng cơ bản</b>	<ol style="list-style-type: none"> <li>Chủ nhà hàng đăng nhập vào hệ thống và truy cập trang quản lý món ăn của nhà hàng.</li> <li>Hệ thống hiển thị giao diện quản lý món ăn, và giao diện “Thêm/sửa món ăn”.</li> <li>Chủ nhà hàng nhập các thông tin: tên món ăn, mô tả, giá</li> <li>Chủ nhà hàng nhấn nút “Thêm món ăn” để tạo món ăn.</li> <li>Hệ thống kiểm tra thông tin: <ul style="list-style-type: none"> <li>Nếu thông tin hợp lệ, món ăn được tạo và hiển thị trong menu nhà hàng.</li> <li>Nếu thông tin không hợp lệ, chuyển sang luồng thay thế.</li> </ul> </li> </ol>
<b>Luồng thay thế</b>	<ul style="list-style-type: none"> <li>Nếu tên món ăn đã tồn tại trong menu của nhà hàng, hệ thống thông báo: “Món ăn đã tồn tại. Vui lòng chọn tên khác.”</li> <li>Nếu có lỗi tại máy chủ hệ thống thông báo: “Đã có lỗi xảy ra. Vui lòng thử lại sau.”</li> </ul>
<b>Tiền điều kiện</b>	<ul style="list-style-type: none"> <li>Chủ nhà hàng đã đăng nhập vào hệ thống với tài khoản có quyền quản lý nhà hàng.</li> <li>Hệ thống đã hiển thị thành công trên trình duyệt.</li> <li>Nhà hàng đã được tạo trên hệ thống (có thông tin cơ bản như tên, địa chỉ).</li> </ul>
<b>Hậu điều kiện</b>	<ul style="list-style-type: none"> <li>Món ăn được tạo thành công và hiển thị trong menu của nhà hàng trên hệ thống.</li> <li>Khách hàng có thể xem thông tin món ăn (tên, mô tả, giá, hình ảnh) khi truy cập nhà hàng.</li> </ul>
<b>Yêu cầu phi chức năng</b>	<ul style="list-style-type: none"> <li>Thời gian phản hồi của hệ thống khi lưu món ăn không vượt quá 2 giây.</li> <li>Dữ liệu món ăn (bao gồm hình ảnh) phải được lưu trữ an toàn, đảm bảo không bị truy cập trái phép.</li> </ul>



Hình 3.10. Biểu đồ hoạt động ca sử dụng tạo món ăn.



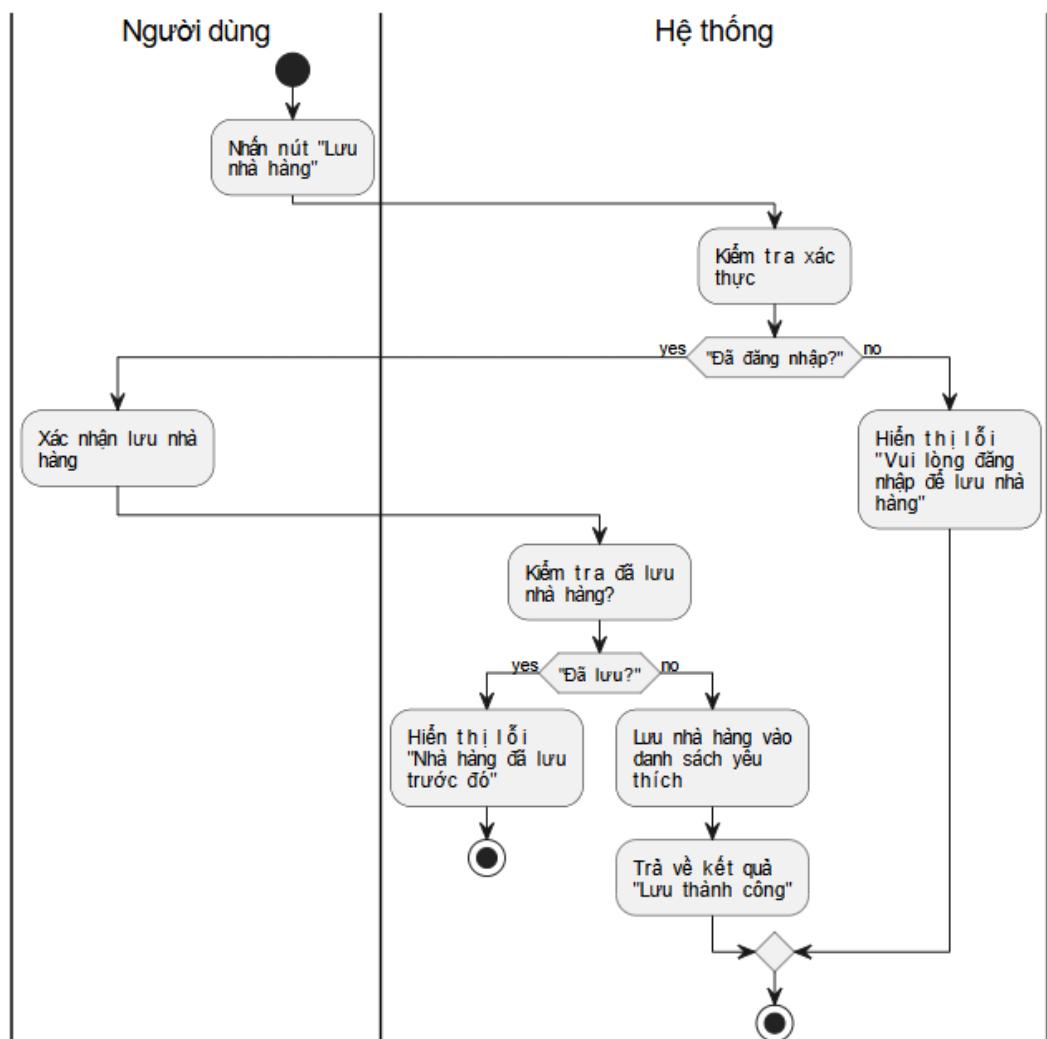
Hình 3.11. Biểu đồ tuần tự ca sử dụng tạo món ăn.

### 3.3.6. Ca sử dụng lưu nhà hàng

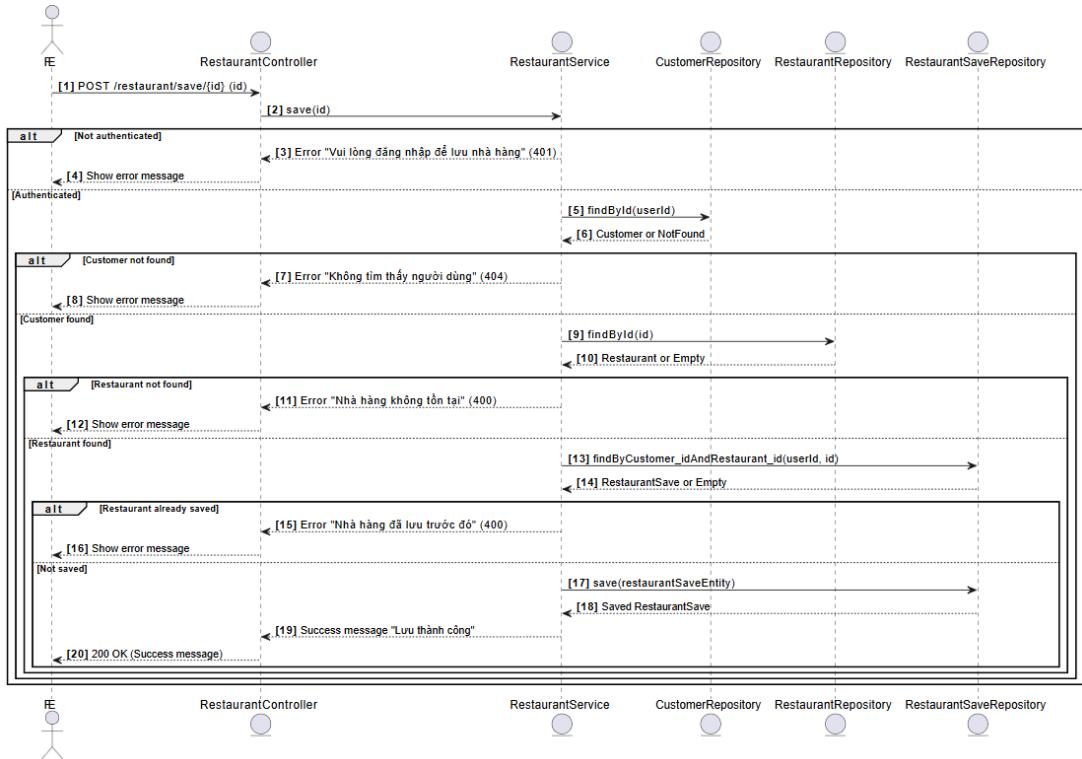
Mô tả chi tiết cho ca sử dụng lưu nhà hàng được thể hiện ở bảng 3.12 dưới đây. Tiếp theo là hình 3.12 về biểu đồ hoạt động và hình 3.13 về biểu đồ tuần tự của ca sử dụng này.

Bảng 3.12. Mô tả chi tiết ca sử dụng lưu nhà hàng.

<b>Mô tả</b>	Cho phép người dùng lưu nhà hàng trên hệ thống để dễ dàng truy cập và theo dõi sau này.
<b>Luồng cơ bản</b>	<ol style="list-style-type: none"> <li>Người dùng đăng nhập vào hệ thống và truy cập trang danh sách nhà hàng hoặc trang chi tiết của một nhà hàng.</li> <li>Hệ thống hiển thị thông tin nhà hàng, bao gồm nút "Lưu vào yêu thích".</li> <li>Người dùng nhấn nút "Lưu vào yêu thích".</li> <li>Hệ thống kiểm tra: <ul style="list-style-type: none"> <li>Nếu nhà hàng chưa được lưu, hệ thống thêm nhà hàng vào danh sách yêu thích của người dùng và hiển thị thông báo: "Đã lưu nhà hàng vào danh sách yêu thích."</li> <li>Nếu nhà hàng đã được lưu trước đó, chuyển sang luồng thay thế.</li> </ul> </li> </ol>
<b>Luồng thay thế</b>	<ul style="list-style-type: none"> <li>Nếu nhà hàng đã có trong danh sách yêu thích, hệ thống thông báo: "Nhà hàng đã lưu trước đó."</li> <li>Nếu có lỗi tại máy chủ hệ thống thông báo: "Đã có lỗi xảy ra. Vui lòng thử lại sau."</li> <li>Nếu người dùng chưa đăng nhập, hệ thống hiển thị thông báo: "Bạn chưa đăng nhập."</li> </ul>
<b>Tiền điều kiện</b>	<ul style="list-style-type: none"> <li>Người dùng đã đăng nhập vào hệ thống với tài khoản cá nhân.</li> <li>Trang web đã hiển thị thành công trên trình duyệt.</li> <li>Nhà hàng đã tồn tại trên hệ thống và có thông tin cơ bản.</li> </ul>
<b>Hậu điều kiện</b>	<ul style="list-style-type: none"> <li>Nhà hàng được thêm vào danh sách yêu thích của người dùng trên hệ thống.</li> <li>Người dùng có thể xem danh sách nhà hàng yêu thích trong hồ sơ cá nhân hoặc mục "Đã lưu".</li> </ul>
<b>Yêu cầu phi chức năng</b>	<ul style="list-style-type: none"> <li>Hệ thống phải phản hồi thao tác lưu nhà hàng trong vòng 1 giây để đảm bảo trải nghiệm người dùng mượt mà.</li> <li>Hệ thống phải đảm bảo quyền riêng tư: danh sách yêu thích của người dùng không được hiển thị công khai cho người dùng khác.</li> </ul>



Hình 3.12. Biểu đồ hoạt động ca sử dụng lưu nhà hàng.



Hình 3.13. Biểu đồ tuần tự ca sử dụng lưu nhà hàng.

### 3.4. Thiết kế cơ sở dữ liệu

Để hỗ trợ các chức năng của hệ thống, cơ sở dữ liệu được thiết kế nhằm lưu trữ và quản lý thông tin một cách hiệu quả. Cơ sở dữ liệu bao gồm các bảng chính như: bảng Customer để lưu thông tin tài khoản người dùng, bảng Restaurant để quản lý thông tin nhà hàng, bảng Review để lưu trữ các đánh giá và xếp hạng từ người dùng. Các bảng này được liên kết với nhau thông qua các khóa ngoại để đảm bảo tính toàn vẹn dữ liệu và hỗ trợ truy vấn nhanh chóng. Sơ đồ thực thể quan được trình bày chi tiết trong hình 3.14.



Hình 3.14. Sơ đồ thực thể dữ liệu.

Hệ thống quản lý nhà hàng được xây dựng trên cơ sở dữ liệu bao gồm các bảng chính: customer, restaurant, category\_food, reviews và restaurant\_save.

Bảng customer lưu thông tin người dùng như họ tên, tuổi, giới tính, địa chỉ, email, số điện thoại, mật khẩu và vai trò. Đây là bảng trung tâm đại diện cho người dùng hệ thống.

Bảng restaurant chứa thông tin chi tiết về nhà hàng, bao gồm tên, địa chỉ, vị trí địa lý (kinh độ và vĩ độ), số lượng ảnh, điểm đánh giá trung bình, tổng số đánh giá, ảnh đại diện, cùng các thông tin phân cấp như quận/huyện và thành phố. Bảng này có khóa ngoại user\_id liên kết đến id của bảng customer, thể hiện mối quan hệ N:1, một người dùng có thể tạo nhiều nhà hàng.

Bảng category\_food lưu thông tin về các món ăn thuộc từng nhà hàng, bao gồm tên món, mô tả, giá, hình ảnh và khóa ngoại restaurant\_id liên kết đến bảng restaurant. Mỗi quan hệ là N:1, một nhà hàng có thể có nhiều món ăn.

Bảng reviews lưu trữ đánh giá từ người dùng, bao gồm bình luận, điểm số, thời gian tạo và các khóa ngoại: restaurant\_id, customer\_id và category\_food\_id. Những khóa ngoại này lần lượt liên kết đến bảng restaurant, customer và category\_food, thể hiện ba mối quan hệ N:1, một người dùng có thể đánh giá nhiều nhà hàng, mỗi đánh giá thuộc một món ăn cụ thể trong một nhà hàng.

Bảng restaurant\_save lưu thông tin về các nhà hàng mà người dùng đã lưu lại. Bảng này có hai khóa ngoại: restaurant\_id và customer\_id, liên kết đến hai bảng tương ứng. Mỗi quan hệ này là N:N, do một người dùng có thể lưu nhiều nhà hàng và một nhà hàng có thể được nhiều người dùng lưu.

Tổng thể, sơ đồ thực thể quan hệ dữ liệu thể hiện rõ ràng các mối quan hệ giữa thực thể trong hệ thống, đảm bảo tính liên kết chặt chẽ và hỗ trợ hiệu quả cho các chức năng tìm kiếm, đánh giá và lưu trữ nhà hàng.

# Chương 4.

## Xây dựng và triển khai hệ thống

### 4.1. Công nghệ sử dụng

#### 4.1.1. Khái quát tổng thể

Hệ thống Khám Phá Xung Quanh được thiết kế dựa trên kiến trúc MVC với mô hình phân tầng (layered architecture), đảm bảo sự tách biệt rõ ràng giữa các thành phần và hỗ trợ khả năng bảo trì, mở rộng hiệu quả. Phía backend được xây dựng bằng Java Spring Boot, tổ chức thành ba lớp chính gồm Controller, Service và Repository, giúp tối ưu hóa việc xử lý logic nghiệp vụ và truy xuất dữ liệu. Phía client (View) tận dụng ReactJS để phát triển giao diện người dùng tương tác thời gian thực, mang đến trải nghiệm mượt mà và trực quan. Phía dữ liệu (Model) sử dụng MySQL làm hệ quản trị cơ sở dữ liệu quan hệ, đảm bảo lưu trữ và truy vấn dữ liệu ổn định. Hệ thống tích hợp Web Service và RESTful API để kết nối liền mạch giữa các tầng. Ngoài ra, một server AI backend riêng biệt được triển khai với FastAPI, tận dụng các mô hình pre-trained sau đó được fine-tune trên dữ liệu món ăn chính xác hơn cùng TensorFlow và OpenCV, cho phép nhận diện món ăn qua hình ảnh và trả về nhãn chính xác. Hệ thống cũng tích hợp các API bên ngoài như OpenStreetMap (định vị), Open-Meteo (thời tiết), và LLaMA API để cá nhân hóa gợi ý theo bối cảnh người dùng, từ thời tiết, vị trí đến sở thích cá nhân. Việc triển khai được thực hiện trên nền tảng đám mây, sử dụng Docker để đóng gói ứng dụng và CI/CD pipeline để tự động hóa quy trình, đảm bảo hiệu suất và tính ổn định.

#### 4.1.2. Công nghệ phía Frontend

- ReactJS: Xây dựng giao diện tương tác thời gian thực, hỗ trợ tái sử dụng thành phần, tối ưu hiệu suất.
- RESTful API Client: Giao tiếp với backend qua các API chuẩn hóa, đảm bảo trao đổi dữ liệu an toàn.

#### 4.1.3. Công nghệ phía Backend Java Spring

Hệ thống backend chính được chia thành 3 lớp:

- Controller: Xử lý các yêu cầu HTTP từ client, điều phối luồng xử lý, và gọi các service tương ứng.

- Service: Chứa logic nghiệp vụ chính, kết nối Controller với Repository, xử lý các tác vụ như tìm kiếm nhà hàng, khám phá tự động, v.v.
- Repository: Tương tác với MySQL, cung cấp các phương thức truy vấn dữ liệu (nhà hàng, món ăn, thông tin người dùng).

Các công nghệ hỗ trợ:

- Java Spring Boot: Framework chính để xây dựng backend, cung cấp các tính năng như Spring Web để phát triển các dịch vụ RESTful, Spring Boot Starter để quản lý phụ thuộc và cấu hình tự động, đảm bảo triển khai nhanh chóng và hiệu quả.
- Spring Data JPA (Hibernate): Tích hợp với MySQL để ánh xạ đối tượng-thực thể (ORM), hỗ trợ các truy vấn phức tạp, quản lý giao dịch và tối ưu hóa hiệu suất truy xuất dữ liệu (nhà hàng, món ăn, thông tin người dùng).
- Spring Security (JWT Authentication): Cung cấp cơ chế bảo mật mạnh mẽ, sử dụng JSON Web Token (JWT) để xác thực và phân quyền người dùng, bảo vệ các endpoint API và đảm bảo an toàn thông tin cá nhân.
- LLaMA API: Tích hợp API của LLaMA để cá nhân hóa gợi ý, phân tích bối cảnh (thời tiết, thời gian, vị trí, sở thích người dùng), tạo ra các gợi ý món ăn và nhà hàng phù hợp với từng cá nhân.
- Spring RestTemplate: Hỗ trợ gọi API đến server AI (FastAPI) để nhận diện món ăn qua hình ảnh, đảm bảo giao tiếp giữa các hệ thống nhanh chóng và đáng tin cậy.

#### **4.1.4. Công nghệ phía Backend FastAPI**

- FastAPI: Xây dựng Server AI backend, cung cấp API để nhận diện món ăn qua hình ảnh.
- TensorFlow và OpenCV: Tận dụng các mô hình pretrained sau đó fine-tuned trên tập dữ liệu món ăn chính xác để xử lý ảnh, trả về nhãn món ăn (ví dụ: “phở”, “bún bò”).
- Quy trình : Khi người dùng tìm kiếm bằng hình ảnh, tầng service của backend spring gọi API đến server AI FastAPI. Server AI FastAPI phân tích hình ảnh qua endpoint (/predict/url/’), trả về nhãn món ăn, sau đó tầng Service sử dụng nhãn này để truy vấn dữ liệu nhà hàng từ MySQL.

#### **4.1.5. Công nghệ tích hợp bên thứ ba**

- Open Street Map: Hỗ trợ định vị, xác định vị trí người dùng và nhà hàng.

- Open Meteo: Cung cấp dữ liệu thời tiết hiện tại để gợi ý món ăn phù hợp.

#### 4.1.6. Công nghệ triển khai

- Docker: Containerization để đóng gói cả backend chính và server AI, đảm bảo tính nhất quán giữa các môi trường.
- CI/CD Pipeline: Tự động hóa quy trình kiểm thử và triển khai, tăng hiệu quả vận hành.

#### 4.1.7. Mô hình mạng nơ-ron tích chập nhận diện ảnh

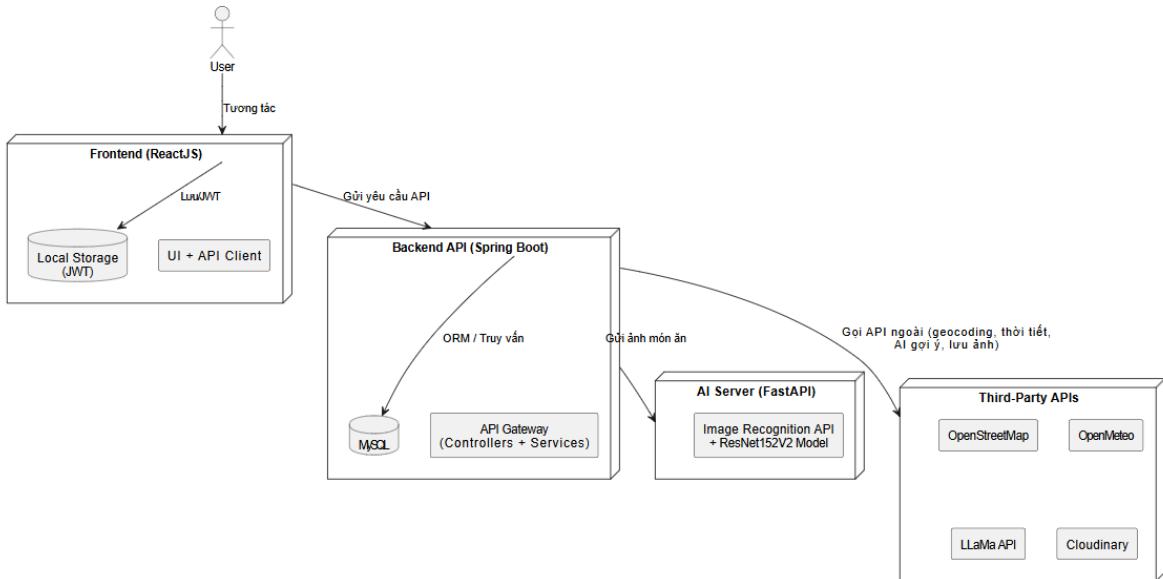
Hệ thống sử dụng mạng nơ-ron tích chập để nhận diện món ăn qua hình ảnh, được triển khai trên server AI backend FastAPI. Hai mô hình pre-trained là ResNet152V2 và InceptionResNetV2 đã được thử nghiệm, cả hai đều được huấn luyện trước trên tập dữ liệu ImageNet. Các mô hình này sau đó được fine-tune trên tập dữ liệu món ăn tự xây dựng, bao gồm các món ăn phổ biến tại Việt Nam như phở, bún bò, bánh mì, v.v. Quá trình fine-tune được thực hiện với TensorFlow, trong đó các tầng cuối của mỗi mô hình được điều chỉnh để phân loại chính xác các món ăn trong tập dữ liệu. OpenCV được sử dụng để tiền xử lý hình ảnh (resize, chuẩn hóa, tăng cường dữ liệu) trước khi đưa vào mô hình, đảm bảo chất lượng đầu vào tối ưu. Hình 4.1 cho thấy sau khi so sánh hiệu suất, mô hình ResNet152V2 có độ chính xác cao hơn so với InceptionResNetV2. Do đó, ResNet152V2 được chọn để triển khai chính thức. Kết quả đầu ra là nhãn món ăn (ví dụ: “phở”, “bánh chưng”, “bún chả”, v.v.), được trả về qua API của FastAPI, đáp ứng tốt yêu cầu nhận diện món ăn trong hệ thống.

	<b>Accuracy</b>	<b>Top 3 Accuracy</b>	<b>Top 5 Accuracy</b>
<b>ResNet152V2</b>	0.838	0.916	0.958
<b>InceptionResNetV2</b>	0.783	0.891	0.943

Hình 4.1. So sánh độ chính xác giữa các mô hình nhận diện món ăn.

#### 4.1.8. Kiến trúc hệ thống

Kiến trúc của hệ thống hoạt động theo như hình 4.2.



Hình 4.2. Kiến trúc hệ thống.

Kiến trúc của hệ thống được mô tả trong hình 4.2. Sơ đồ này thể hiện các thành phần chính của hệ thống, cách chúng tương tác với nhau và vai trò của từng thành phần trong việc xử lý các yêu cầu từ người dùng.

Hệ thống bao gồm ba lớp chính: Frontend, Backend và AI Server, tương tác qua các API RESTful. Frontend (ReactJS) là giao diện người dùng, nơi người dùng tương tác với hệ thống thông qua các UI Components. Các yêu cầu được gửi đến Backend thông qua API Client (axios), và dữ liệu được lưu trữ trong local storage.

Backend (Spring Boot) xử lý các yêu cầu từ frontend, bao gồm việc xác thực người dùng, quản lý nhà hàng và món ăn, và tích hợp với các API bên ngoài (như OpenStreetMap, OpenMeteo, và LLaMa).

AI Server (FastAPI) chứa mô hình học sâu CNN (ResNet152V2) để nhận diện ảnh món ăn từ người dùng. Cơ sở dữ liệu MySQL lưu trữ các dữ liệu quan trọng về người dùng, nhà hàng và món ăn. Với JWT được sử dụng để xác thực và bảo mật các API, đảm bảo chỉ những người dùng đã xác thực mới có quyền truy cập vào các tài nguyên nhạy cảm của hệ thống.

#### **4.1.9. Triển khai hệ thống**

Hệ thống Khám Phá Xung Quanh được triển khai dựa trên kiến trúc phân tán với ba lớp chính: Frontend, Backend, và AI Server. Mỗi lớp có chức năng riêng và giao tiếp với nhau thông qua các API.

##### **4.1.9.1. Frontend (ReactJS)**

- Frontend được phát triển bằng ReactJS, sử dụng các thành phần UI để tương tác với người dùng.
- Mọi yêu cầu của người dùng như đăng nhập, quản lý nhà hàng, và quản lý món ăn đều được gửi đến Backend qua API Client (Axios).
- Sau khi nhận được phản hồi từ Backend, giao diện người dùng sẽ được cập nhật.
- Các thông tin nhạy cảm như token xác thực được lưu trữ trong Local Storage của trình duyệt để giữ trạng thái đăng nhập của người dùng.

##### **4.1.9.2. Backend (Spring Boot)**

- Backend được triển khai trên nền tảng Spring Boot, với các dịch vụ RESTful API cho phép các tác vụ như đăng nhập, quản lý nhà hàng, món ăn, và lấy thông tin người dùng.
- Mỗi dịch vụ (AuthService, RestaurantService, DishService, WeatherService, CustomerService, v.v.) tương ứng với một Controller và giao tiếp với các Repository để thực hiện các thao tác trên cơ sở dữ liệu MySQL.
- AuthService đảm nhiệm xử lý logic đăng nhập, đăng ký, xác thực người dùng, v.v.
- CustomerService đảm nhiệm xử lý logic liên quan đến người dùng như cập nhật thông tin, thay đổi mật khẩu, đánh giá, tìm kiếm bằng hình ảnh, v.v.
- RestaurantService đảm nhiệm xử lý logic liên quan đến nhà hàng như đăng ký nhà hàng, chỉnh sửa nhà hàng, lấy danh sách nhà hàng, lưu nhà hàng yêu thích, v.v.
- DishService đảm nhiệm xử lý logic liên quan đến món ăn như tạo món ăn, chỉnh sửa món ăn, lấy danh sách món ăn, v.v.
- WeatherService đảm nhiệm xử lý logic liên quan đến thời tiết, cung cấp thông tin thời tiết từ OpenMeteo.

#### **4.1.9.3. Backend AI Server (FastAPI)**

- AI Server được triển khai sử dụng FastAPI và chứa một mô hình học sâu CNN ResNet152V2 để nhận diện món ăn.
- Khi người dùng tải lên ảnh món ăn, API nhận diện sẽ gọi mô hình CNN để phân tích và trả về kết quả nhận diện.
- Mô hình này giúp hệ thống nhận diện chính xác các món ăn từ hình ảnh.

#### **4.1.9.4. Cơ sở dữ liệu (MySQL)**

- Cơ sở dữ liệu MySQL lưu trữ tất cả thông tin quan trọng của hệ thống, bao gồm dữ liệu người dùng, nhà hàng, và món ăn.
- Các Repository trong Backend sẽ giao tiếp trực tiếp với MySQL để truy vấn và thao tác dữ liệu.

#### **4.1.9.5. Các API bên ngoài**

Hệ thống tích hợp với các API bên ngoài để nâng cao khả năng cung cấp dịch vụ.

- OpenStreetMap cung cấp dịch vụ geocoding giúp hệ thống chuyển đổi tọa độ địa lý sang thông tin địa chỉ.
- OpenMeteo cung cấp dữ liệu thời tiết.
- LLaMA được sử dụng để gợi ý các món ăn dựa trên ngữ cảnh.
- Cloudinary được sử dụng để lưu trữ và quản lý hình ảnh món ăn.

#### **4.1.9.6. Triển khai và bảo mật**

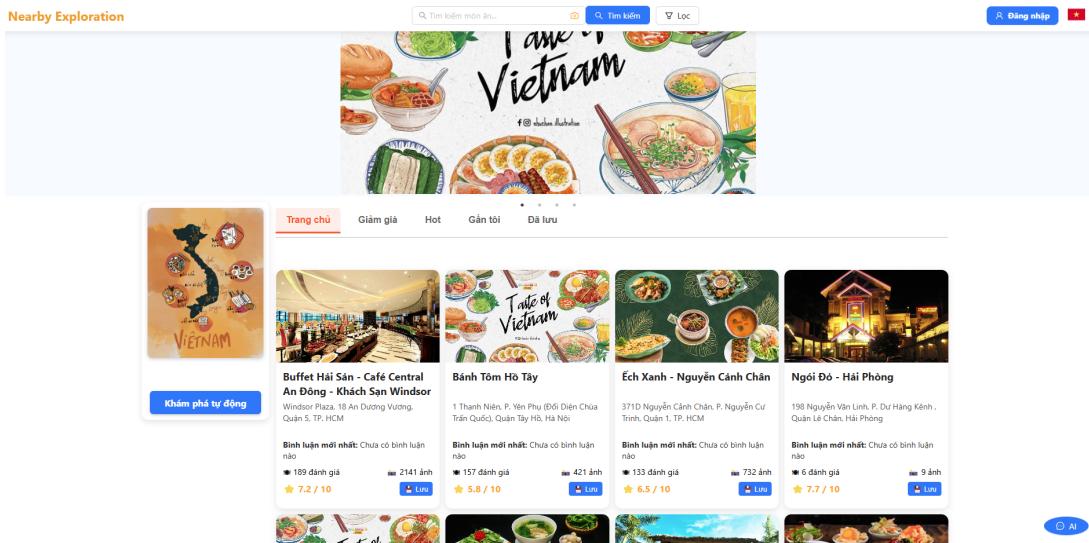
- Hệ thống sử dụng JWT để xác thực người dùng và bảo vệ các API nhạy cảm, đảm bảo rằng chỉ những người dùng đã đăng nhập mới có thể thực hiện các thao tác yêu cầu quyền truy cập.
- Các token này được lưu trữ trong Local Storage trên frontend và gửi theo mỗi yêu cầu API để xác thực.

## **4.2. Các chức năng chính của hệ thống**

Hệ thống Khám Phá Xung Quanh được xây dựng với mục tiêu mang đến một hành trình khám phá các dịch vụ địa phương đầy tiện ích, đặc biệt trong lĩnh vực ẩm thực, đáp ứng nhu cầu đa dạng của cả người dùng bản địa lẫn khách du lịch. Nhờ thiết kế tinh tế kết hợp với sự tích hợp của các công nghệ hiện đại, hệ thống đem lại những tính năng nổi bật sau đây:

#### 4.2.1. Trang chủ

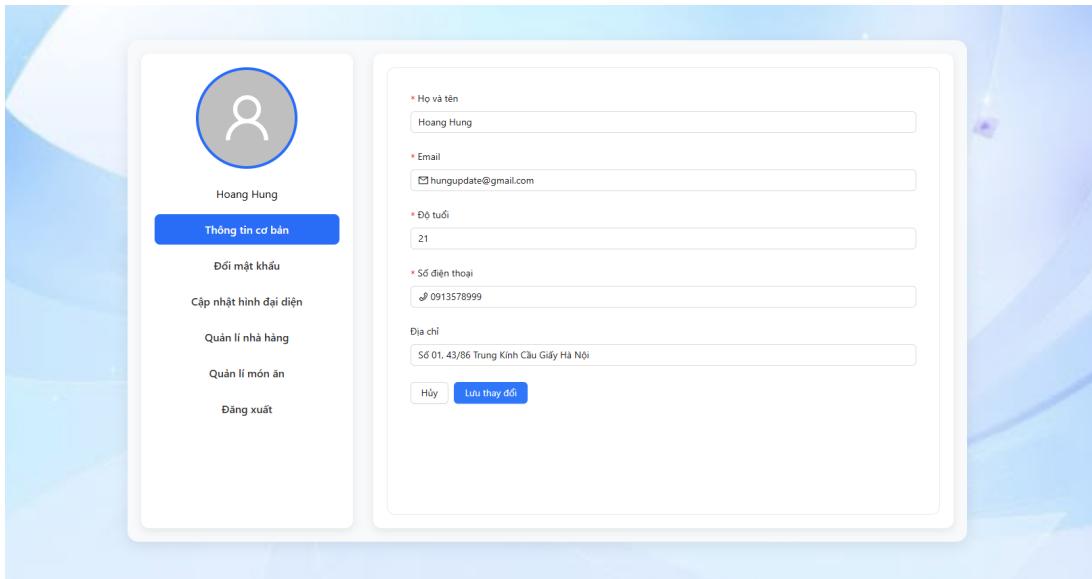
Giao diện trang chủ của hệ thống Khám Phá Xung Quanh ở hình 4.3 được thiết kế trực quan và thân thiện, hiển thị các tính năng chính như tìm kiếm món ăn, khám phá tự động và danh sách nhà hàng gần đây. Người dùng có thể dễ dàng truy cập các chức năng thông qua các nút điều hướng nổi bật, cùng với hình ảnh minh họa sinh động về các món ăn và nhà hàng địa phương.



Hình 4.3. Giao diện trang chủ.

#### 4.2.2. Cập nhật hồ sơ cá nhân

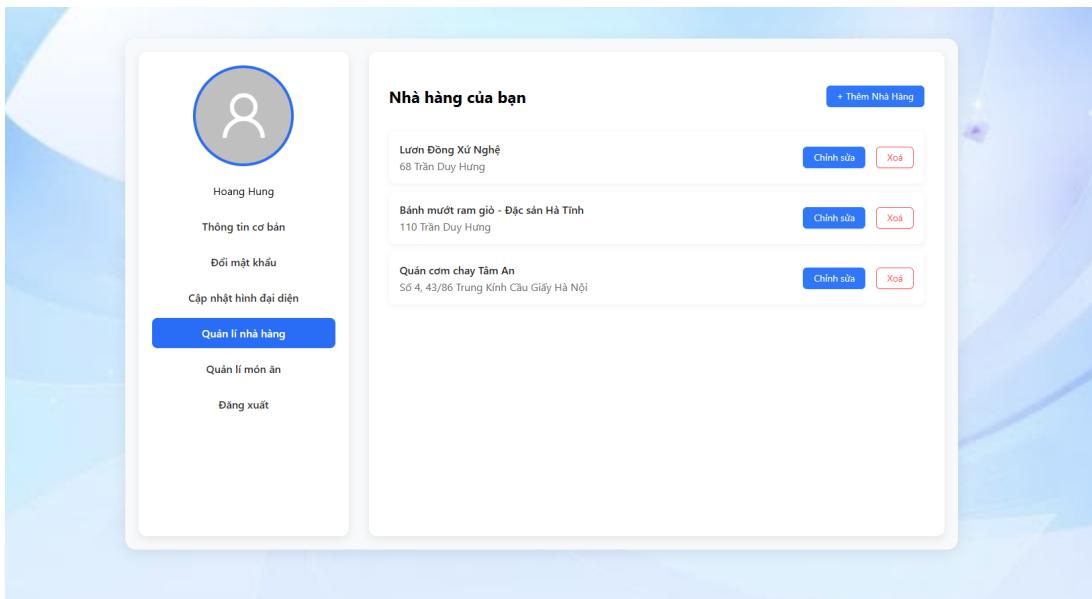
Giao diện cập nhật hồ sơ cá nhân ở hình 4.4 cho phép người dùng chỉnh sửa thông tin cá nhân như tên, số điện thoại, email, địa chỉ, v.v. Thiết kế đơn giản với các trường nhập liệu rõ ràng, kèm theo nút lưu thông tin, giúp người dùng dễ dàng quản lý và cá nhân hóa trải nghiệm tìm kiếm của mình trên hệ thống.



Hình 4.4. Giao diện hồ sơ cá nhân.

#### 4.2.3. Quản lý nhà hàng

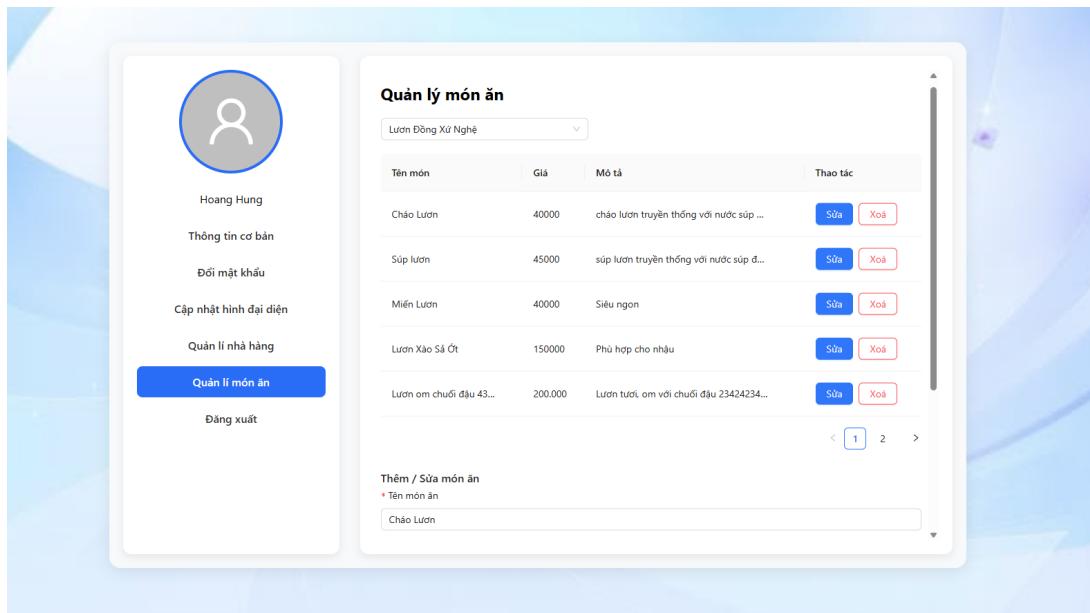
Giao diện quản lý nhà hàng ở hình 4.5 cho phép người dùng thêm/chỉnh sửa thông tin nhà hàng như tên, số điện thoại, email, địa chỉ, v.v. Thiết kế đơn giản với các trường nhập liệu rõ ràng, kèm theo nút lưu thông tin, giúp người dùng dễ dàng quản lý và cá nhân hóa trải nghiệm tìm kiếm của mình trên hệ thống.



Hình 4.5. Giao diện quản lý nhà hàng.

#### 4.2.4. Quản lý món ăn

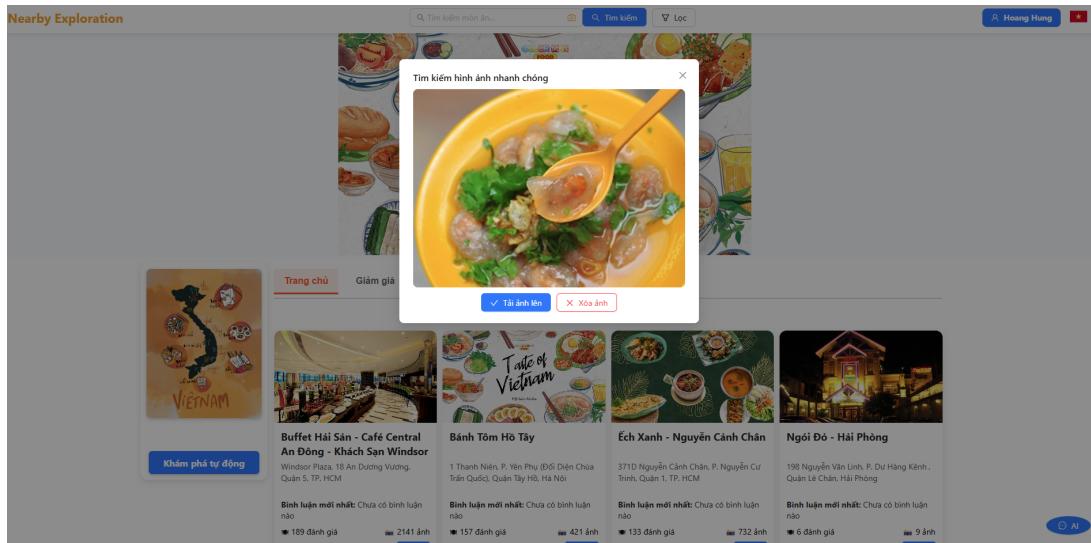
Giao diện quản lý món ăn ở hình 4.6 cho phép người dùng thêm/chỉnh sửa thông tin món ăn của từng nhà hàng như tên, mô tả, giá cả, v.v. Thiết kế đơn giản với các trường nhập liệu rõ ràng, kèm theo nút lưu thông tin, giúp người dùng dễ dàng quản lý và cá nhân hóa trải nghiệm tìm kiếm của mình trên hệ thống.



Hình 4.6. Giao diện quản lý món ăn từng nhà hàng.

#### 4.2.5. Tìm kiếm món ăn bằng hình ảnh

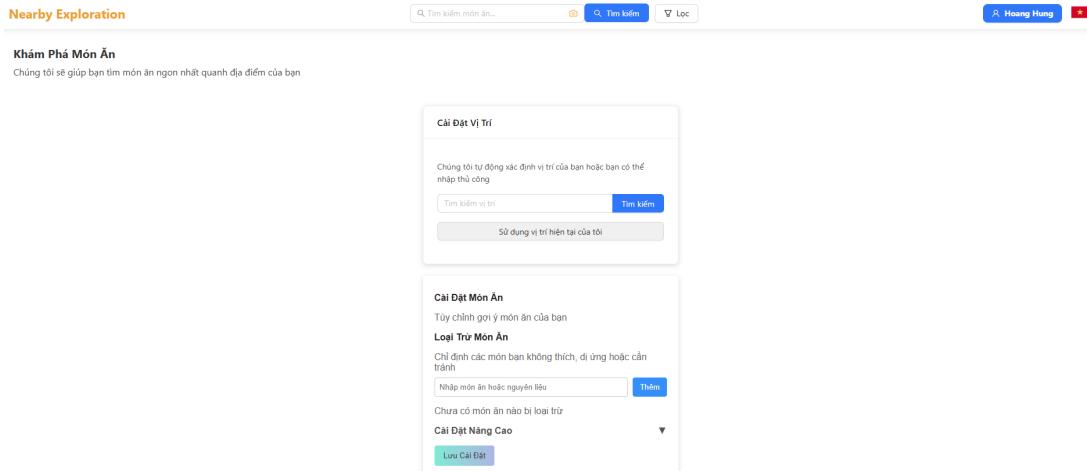
Giao diện tìm kiếm món ăn bằng hình ảnh ở hình 4.7 hiển thị khu vực tải ảnh lên hoặc chụp ảnh trực tiếp. Sau khi hệ thống nhận diện, hệ thống gợi ý danh sách các nhà hàng gần nhất có phục vụ món ăn đó, với thông tin cơ bản như tên nhà hàng, khoảng cách và đánh giá.



Hình 4.7. Giao diện tìm kiếm nhà hàng bằng hình ảnh.

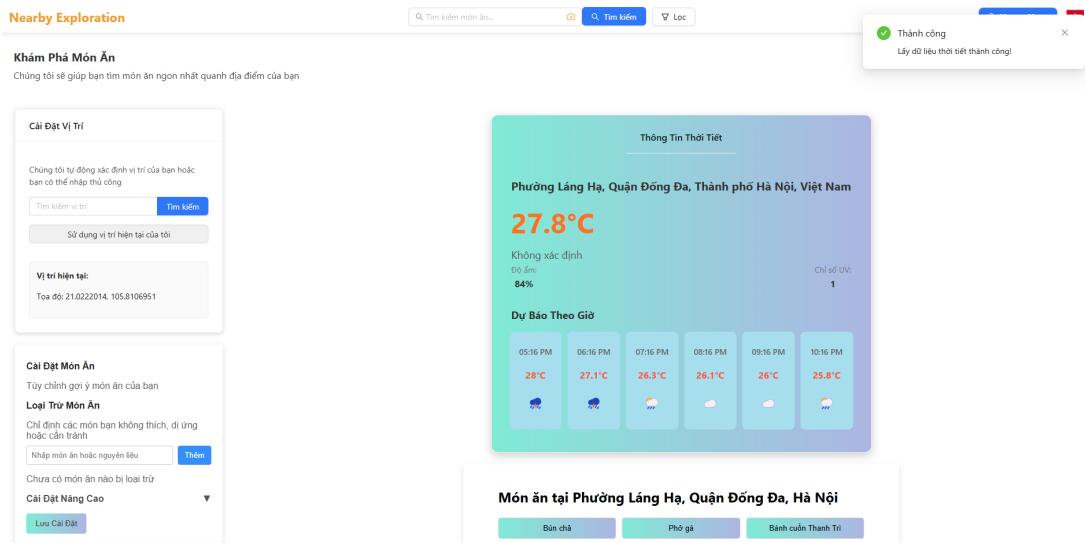
#### 4.2.6. Khám phá món ăn, nhà hàng tự động

Giao diện khám phá món ăn nhà hàng tự động ở hình 4.8 dựa trên các tiêu chí như địa điểm, sở thích cá nhân. Thiết kế đơn giản với các trường nhập liệu rõ ràng, giúp người dùng dễ dàng trải nghiệm tính năng này trên hệ thống.



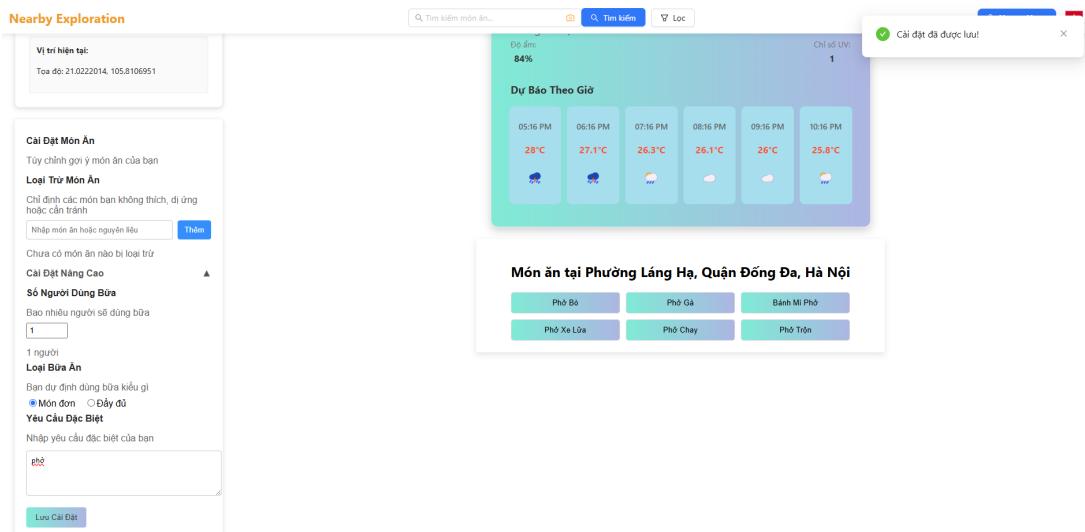
Hình 4.8. Giao diện khám phá tự động.

Giao diện gợi ý khám phá món ăn và dự báo thời tiết sau khi người dùng chọn vị trí khám phá ở hình 4.9 với giao diện trực quan thân thiện người dùng.



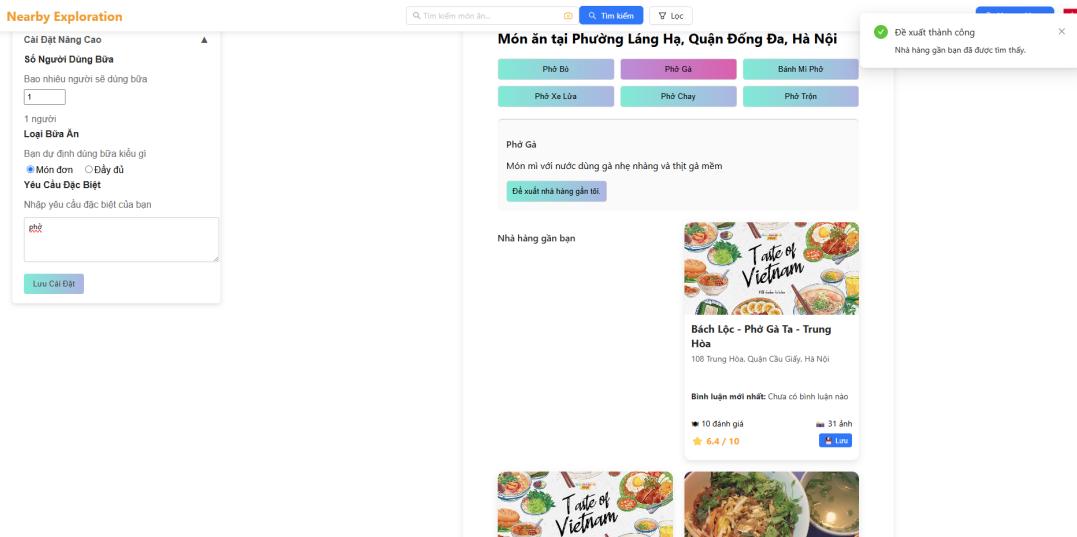
Hình 4.9. Giao diện hệ thống đề xuất các món ăn và thời tiết.

Giao diện hiển thị các món ăn mà hệ thống đề xuất ở hình 4.10 dựa trên các sở thích tiêu chí của người dùng, vị trí và thời tiết hiện tại. Thiết kế đơn giản và thân thiện với người dùng giúp tăng tính trải nghiệm.



Hình 4.10. Giao diện hệ thống đề xuất các món ăn dựa trên sở thích.

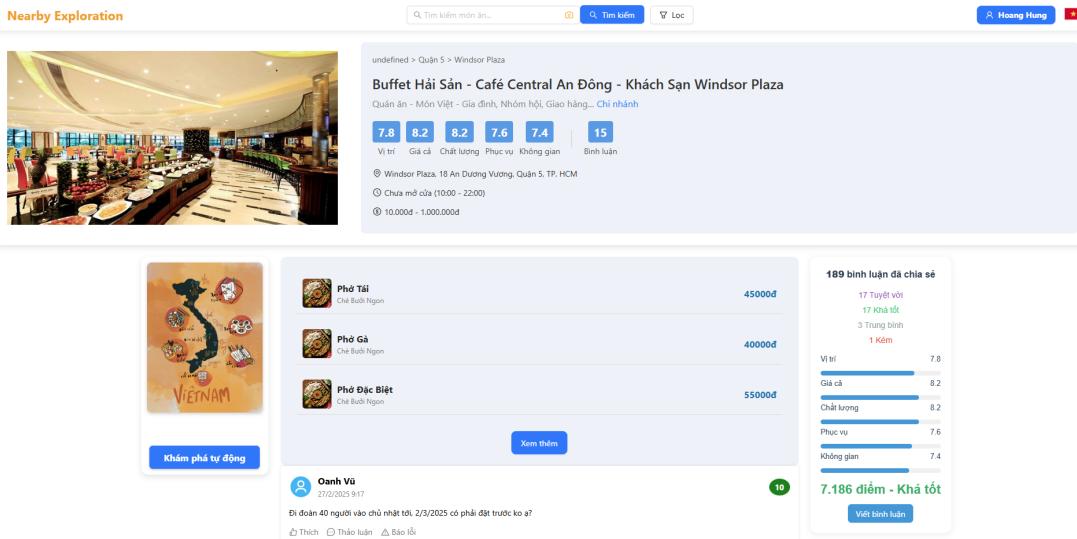
Giao diện hiển thị các nhà hàng mà hệ thống đề xuất nhà hàng dựa trên các món ăn mà người dùng muốn khám phá ở hình 4.11 với thiết kế đơn giản và thân thiện với người dùng, giúp tăng tính trải nghiệm tính năng khám phá tự động.



Hình 4.11. Giao diện hệ thống đề xuất các nhà hàng.

#### 4.2.7. Xem thông tin từng nhà hàng

Giao diện chi tiết nhà hàng ở hình 4.12 cung cấp thông tin đầy đủ về nhà hàng, bao gồm tên, địa chỉ, số điện thoại, giờ mở cửa, menu, hình ảnh món ăn và đánh giá từ người dùng khác. Người dùng cũng có thể đánh giá về chất lượng nhà hàng cũng như chọn lưu vào mục yêu thích.

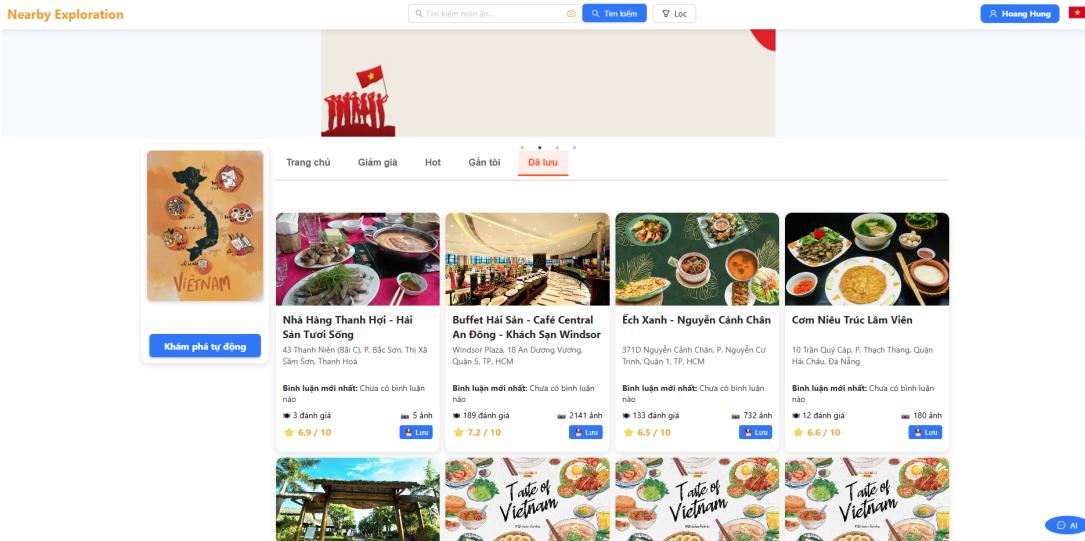


Hình 4.12. Giao diện chi tiết nhà hàng.

#### 4.2.8. Lưu nhà hàng vào mục yêu thích

Giao diện hiển thị danh sách các nhà hàng mà người dùng đã lưu vào mục yêu thích ở hình 4.13 với thông tin cơ bản như tên nhà hàng, địa chỉ, loại món ăn

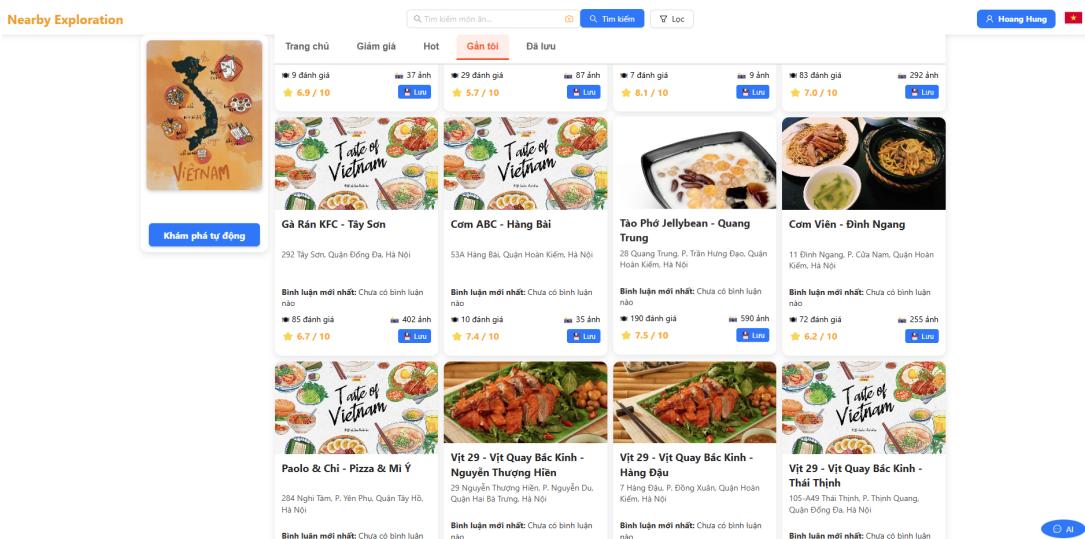
và nút điều hướng để xem chi tiết. Thiết kế dạng danh sách giúp người dùng dễ dàng quản lý và truy cập nhanh các nhà hàng ưa thích.



Hình 4.13. Giao diện các nhà hàng yêu thích của người dùng.

#### 4.2.9. Xem danh sách nhà hàng xung quanh

Giao diện ở hình 4.14 hiển thị danh sách các nhà hàng xung quanh vị trí hiện tại của người dùng, sắp xếp theo khoảng cách từ gần đến xa. Mỗi nhà hàng được hiển thị với hình ảnh đại diện, tên, khoảng cách, loại món ăn và đánh giá trung bình, giúp người dùng dễ dàng lựa chọn địa điểm phù hợp.



Hình 4.14. Giao diện các nhà hàng xung quanh người dùng.

# Chương 5.

## Kiểm thử cho hệ thống

### 5.1. Kiểm thử các xử lý logic phía máy chủ

Để đảm bảo các chức năng của hệ thống hoạt động chính xác, backend được kiểm thử với 2 phương pháp chính: kiểm thử đơn vị cho các hàm và module xử lý nghiệp vụ, và kiểm thử tích hợp API để xác minh sự tương tác giữa các thành phần như Http Request, Controller, Service, Repository, v.v. đảm bảo dữ liệu trả về chính xác.

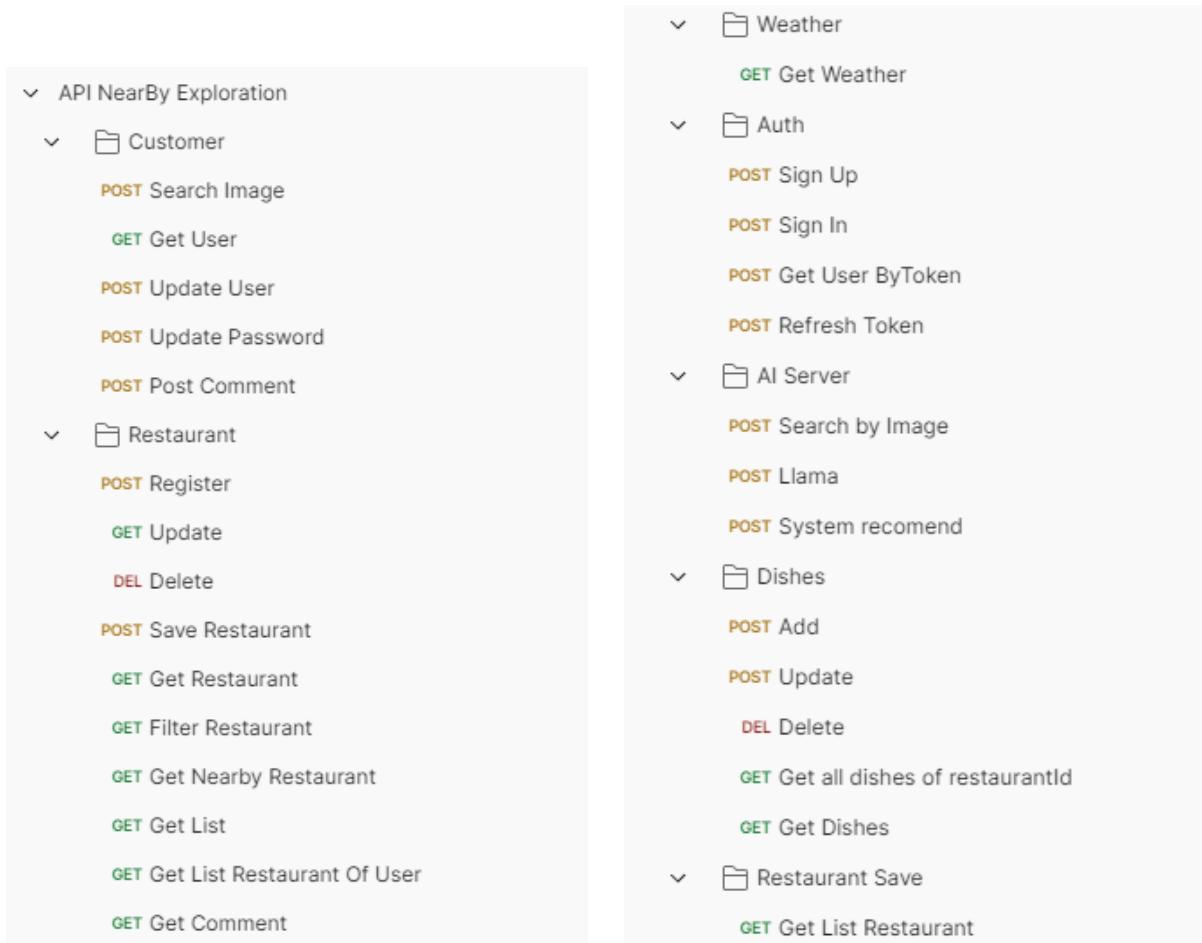
#### 5.1.1. Kiểm thử API

Chi tiết các ca kiểm thử được mô tả ở bảng dưới đây, và hình 5.1 mô tả các API được kiểm thử với Postman.

STT	API	Cấu hình thử	Kết quả kỳ vọng	Tình trạng
1	Tìm kiếm nhà hàng	Tìm kiếm thành công với từ khóa hợp lệ.	Hệ thống trả về mã 200 và danh sách kết quả được xếp hạng.	Đạt
		Tìm kiếm với từ khóa không có kết quả.	Hệ thống trả về mã 200 và danh sách kết quả rỗng.	Đạt
		Tìm kiếm nhà hàng bằng hình ảnh món ăn hợp lệ.	Hệ thống trả về mã 200 và danh sách các nhà hàng hợp lệ.	Đạt
		Tìm kiếm nhà hàng bằng hình ảnh không hợp lệ.	Hệ thống trả về mã lỗi 400 và kèm theo mã lỗi "Gửi ảnh tới AI Service thất bại".	Đạt
		Tìm kiếm nhà hàng dựa trên tiêu chí bộ lọc địa điểm.	Hệ thống trả về mã 200 và danh sách các nhà hàng dựa trên tiêu chí.	Đạt

2	Tìm kiếm nhà hàng	Tìm kiếm thành công với từ khóa hợp lệ.	Hệ thống trả về mã 200 và danh sách kết quả được xếp hạng.	Đạt
		Tìm kiếm với từ khóa không có kết quả.	Hệ thống trả về mã 200 và danh sách kết quả rỗng.	Đạt
		Tìm kiếm nhà hàng bằng hình ảnh món ăn hợp lệ.	Hệ thống trả về mã 200 và danh sách các nhà hàng hợp lệ.	Đạt
		Tìm kiếm nhà hàng bằng hình ảnh không hợp lệ.	Hệ thống trả về mã lỗi 400 và kèm theo mã lỗi "Gửi ảnh tới AI Service thất bại".	Đạt
		Tìm kiếm nhà hàng dựa trên tiêu chí bộ lọc địa điểm.	Hệ thống trả về mã 200 và danh sách các nhà hàng dựa trên tiêu chí.	Đạt
3	Khám phá tự động	Hệ thống trả về danh sách nhà hàng gần vị trí người dùng khi họ cho phép hệ thống truy cập vị trí hiện tại.	Hệ thống trả về mã 200 và danh sách nhà hàng bán kính 15km.	Đạt
		Hệ thống không trả về danh sách nhà hàng gần vị trí người dùng khi họ không cho phép hệ thống truy cập vị trí hiện tại.	Hệ thống trả về mã 404 và danh sách nhà hàng rỗng	Đạt
		Người dùng khám phá tự động nhà hàng và hệ thống lấy thành công dữ liệu từ người dùng và api bên thứ ba.	Hệ thống trả về mã 200 và danh sách 6 món ăn đề xuất và nhiều nhà hàng liên quan đến từng món ăn.	Đạt

		Người dùng khám phá tự động nhà hàng và hệ thống lấy không thành công dữ liệu từ api bên thứ ba.	Hệ thống trả về mã lỗi 500 và thông báo "Hệ thống đang bị lỗi"	Đạt
4	Đăng nhập thành công	Người dùng truy cập trang chỉnh sửa thông tin cá nhân	Hệ thống cho phép người dùng truy cập và chỉnh sửa thông tin cá nhân	Đạt
		Người dùng truy cập trang chỉnh sửa quản lý nhà hàng (điều kiện người dùng không có quyền quản lý nhà hàng)	Hệ thống không cho phép người dùng truy cập trang chỉnh sửa thông tin nhà hàng và trả về mã lỗi 404 "người dùng không đủ quyền"	Đạt
		Người dùng truy cập trang chỉnh sửa quản lý nhà hàng (điều kiện người dùng có quyền quản lý nhà hàng)	Hệ thống cho phép người dùng truy cập trang chỉnh sửa thông tin nhà hàng	Đạt
5	Đăng nhập thất bại	Người dùng truy cập trang chỉnh sửa thông tin cá nhân	Hệ thống không cho phép người dùng truy cập trang chỉnh sửa thông tin cá nhân và trả về mã lỗi 404 "người dùng chưa đăng nhập"	Đạt
		Người dùng truy cập trang các nhà hàng yêu thích đã lưu	Hệ thống trả về mã lỗi 404 "Người dùng chưa đăng nhập"	Đạt



Hình 5.1. Các API được kiểm thử với Postman.

## 5.1.2. Kiểm thử đơn vị

### 5.1.2.1. Phạm vi kiểm thử

Phạm vi kiểm thử đơn vị của hệ thống gồm kiểm thử các hàm xử lý backend Java Spring và kiểm thử độ chính xác của mô hình ResNet152V2 trong việc nhận diện một số món ăn Việt Nam thông qua đánh giá các chỉ số phân loại trên tập dữ liệu kiểm thử.

### 5.1.2.2. Mục tiêu kiểm thử

Mục tiêu của kiểm thử đơn vị là đảm bảo rằng các hàm xử lý hoạt động đúng như mong đợi trong phạm vi độc lập, không phụ thuộc vào các thành phần khác của hệ thống, đánh giá hiệu suất của mô hình ResNet152V2 trong việc nhận diện món ăn Việt Nam thông qua các chỉ số phân loại. Các kiểm thử đơn vị giúp phát hiện sớm lỗi logic trong quá trình phát triển, đồng thời hỗ trợ việc bảo trì mã nguồn về sau.

### 5.1.2.3. Công cụ và môi trường kiểm thử

Backend Java Spring: Sử dụng JUnit 5 kết hợp với Mockito để kiểm thử các lớp dịch vụ, controller và repository. Mô hình ResNet152V2 được đánh giá bằng scikit-learn trên tập dữ liệu kiểm thử để tính toán các chỉ số phân loại.

### 5.1.2.4. Các thành phần được kiểm thử

Các lớp service của Spring Boot như AuthService, CustomerServiceTest, RestaurantService, DishesService, v.v. bao gồm các phương thức xử lý chính như xác thực người dùng, đăng ký nhà hàng, thêm món ăn, lưu nhà hàng yêu thích, v.v. Các lớp controller đảm bảo trả về đúng mã trạng thái và phản hồi phù hợp. Mô hình ResNet152V2 được kiểm thử về khả năng nhận diện món ăn Việt Nam, tập trung vào độ chính xác và tính ổn định khi xử lý các đầu vào đa dạng.

### 5.1.2.5. Kết quả kiểm thử

Dưới đây là hình 5.2 với tất cả các lớp xử lý logic backend đều được bao phủ bởi kiểm thử đơn vị.

Current scope: all classes | com.example.foodtourbackend.service.serviceImpl

Coverage Summary for Package: com.example.foodtourbackend.service.serviceImpl

Package	Class, %	Method, %	Branch, %	Line, %
com.example.foodtourbackend.service.serviceImpl	100% (9/9)	98.1% (53/54)	85.5% (142/166)	99.2% (360/363)
Class ▾	Class, %	Method, %	Branch, %	Line, %
AuthService	100% (1/1)	100% (5/5)	87.5% (14/16)	100% (41/41)
CustomerService	100% (2/2)	88.9% (8/9)	85.2% (46/54)	97.8% (91/93)
DishesService	100% (1/1)	100% (6/6)	86.4% (19/22)	100% (48/48)
RestaurantSaveService	100% (1/1)	100% (2/2)	100% (4/4)	100% (9/9)
RestaurantService	100% (1/1)	100% (11/11)	84.3% (59/70)	100% (103/103)
UserDetails	100% (1/1)	100% (9/9)		100% (9/9)
UserDetailsService	100% (1/1)	100% (2/2)		100% (8/8)
WeatherService	100% (1/1)	100% (10/10)		98.1% (51/52)

Hình 5.2. Kiểm thử đơn vị lớp dịch vụ.

Hình 5.2 thể hiện kết quả kiểm thử đơn vị cho các lớp xử lý logic trong hệ thống backend, cụ thể là các lớp thuộc tầng service với nhiệm vụ xử lý logic cho hệ thống. Hệ thống đã tiến hành thực thi kiểm thử đơn vị cho toàn bộ 9 lớp trong tầng này, nhằm đảm bảo các chức năng cốt lõi được kiểm tra đầy đủ và hoạt động đúng như mong đợi.

Kết quả kiểm thử cho thấy:

- Tất cả các lớp đều đạt tỷ lệ bao phủ 100% ở cấp độ lớp.

- Tỷ lệ bao phủ phương thức đạt 98.1%, tức 53/54 phương thức đã được kiểm thử.
- Tỷ lệ bao phủ nhánh điều kiện là 85.5%, thể hiện phần lớn các rẽ nhánh trong mã nguồn đã được kiểm tra.
- Tỷ lệ bao phủ dòng lệnh đạt 99.2%, với 360/363 dòng mã đã được thực thi qua các ca kiểm thử.

Dưới đây là hình 5.3 trình bày độ chính xác của mô hình ResNet152V2 khi nhận diện từng loại món ăn thông qua các chỉ số Precision, Recall và F1-score.

Món ăn	Precision	Recall	F1-score
Bánh chưng	0.912	0.882	0.896
Bánh mì	0.881	0.842	0.861
Gỏi cuốn	0.791	0.750	0.770
Cơm tấm	0.831	0.768	0.798
Phở bò	0.792	0.745	0.768

Hình 5.3. Kiểm thử hiệu suất nhận diện món ăn của mô hình ResNet152V2.

Kết quả kiểm thử cho thấy: Bánh chưng và bánh mì đạt chỉ số cao nhất với F1-score lần lượt là 0.896 và 0.861, nhờ đặc trưng hình ảnh rõ ràng, dễ phân biệt với các món khác. Các món như gỏi cuốn, cơm tấm có hiệu quả phân loại ở mức khá, với F1-score dao động từ 0.770 đến 0.798. Phở bò có F1-score thấp nhất (0.768) do hình ảnh dễ bị nhầm lẫn với nhiều món nước khác như phở gà, bún bò Huế, mì quảng, hủ tiêu, v.v. dẫn đến việc mô hình gặp khó khăn trong việc phân biệt chính xác.

Trong quá trình kiểm thử, hệ thống phát hiện một số vấn đề như việc xử lý ngoại lệ chưa đầy đủ, hoặc một số nhánh điều kiện chưa được bao phủ. Sau khi phát hiện, hệ thống đã rà soát mã nguồn, bổ sung thêm các trường hợp kiểm thử phù hợp, đồng thời chỉnh sửa lại logic xử lý để đảm bảo tính đúng đắn và độ tin cậy của hệ thống.

Việc thực hiện kiểm thử đơn vị không chỉ giúp hệ thống phát hiện lỗi từ sớm mà còn nâng cao chất lượng phần mềm, góp phần đảm bảo hệ thống hoạt động ổn định và dễ dàng bảo trì trong tương lai.

# Chương 6.

## Kết luận

Trong bối cảnh công nghệ di động và trí tuệ nhân tạo ngày càng phát triển, nhu cầu khám phá và tìm kiếm các dịch vụ địa phương đặc biệt trong lĩnh vực ẩm thực ngày càng trở nên thiết yếu và thu hút sự quan tâm của người dùng, bên cạnh đó các doanh nghiệp rất cần một trang mạng xã hội chuyên nghiệp để quảng bá thương hiệu của mình. Chính vì thế, khóa luận Khám Phá Xung Quanh đã được ra đời nhằm mục đích xây dựng một nền tảng hỗ trợ người dùng quảng bá thương hiệu, tìm kiếm và khám phá các nhà hàng, quán ăn gần vị trí hiện tại, đồng thời áp dụng các kỹ thuật trí tuệ nhân tạo để mang lại trải nghiệm thông minh và cá nhân hóa hơn.

Trong quá trình thực hiện, hệ thống được triển khai với kiến trúc rõ ràng giữa frontend và backend, sử dụng RESTful API để đảm bảo khả năng mở rộng, tích hợp và bảo trì. Đặc biệt, hệ thống có tích hợp mô hình học sâu phục vụ nhận diện hình ảnh món ăn, tích hợp trí tuệ nhân tạo phục vụ cho người dùng hỏi đáp thông tin thuận tiện hơn cũng như đề xuất địa điểm dựa trên nhiều yếu tố như vị trí địa lý, thời tiết, thời gian và sở thích cá nhân của người dùng. Ngoài ra, người dùng có thể lưu lại các nhà hàng yêu thích, giúp hệ thống dần học hỏi hành vi và nâng cao chất lượng gợi ý trong những lần sử dụng tiếp theo.

Tuy nhiên, hệ thống vẫn còn tồn tại một số hạn chế. Mô hình nhận diện món ăn hiện tại chủ yếu được huấn luyện trên tập dữ liệu có quy mô còn hạn chế, điều này phần nào ảnh hưởng đến độ chính xác khi triển khai trong thực tế với đa dạng hình ảnh từ môi trường Việt Nam. Hơn nữa, cơ chế trả lời tin nhắn người dùng và đề xuất địa điểm hiện vẫn chủ yếu dựa trên các mô hình ngôn ngữ tổng quát như LLaMA, chưa được tùy biến sâu cho ngữ cảnh và nhu cầu đặc thù của người dùng Việt Nam nói chung và từng địa phương nói riêng.

Từ những kết quả đã đạt được, sau đây là các đề xuất hướng phát triển tiếp theo cho hệ thống: Thu thập và mở rộng tập dữ liệu hình ảnh món ăn, đặc biệt là các món ăn đặc trưng của từng vùng miền tại Việt Nam, sau đó đào tạo mô hình nhận diện ảnh nhằm cải thiện độ chính xác cao của mô hình nhận diện. Phát triển hệ thống gợi ý nhà hàng thông minh hơn mà không phụ thuộc vào các mã nguồn mở, có thể khai thác sâu hơn các yếu tố như thời tiết, khẩu vị, thời gian trong ngày, hành vi và lịch sử của người dùng. Thu thập mở rộng dữ liệu về hành vi người dùng, dữ liệu chi tiết từng địa phương (nhà hàng, địa điểm, món ăn đặc sản, v.v.), đào tạo mô hình ngôn ngữ lớn để tính năng trả lời người dùng đưa ra được kết quả phù hợp và chính xác nhất. Tích hợp cơ chế học liên tục từ phản hồi

thực tế của người dùng để nâng cao hiệu quả cá nhân hóa đề xuất. Nâng cao trải nghiệm người dùng thông qua tối ưu hóa giao diện, cải thiện hiệu suất hệ thống đáp ứng cho hàng triệu người dùng. Phát triển một số tính năng nâng cao tương tác thực giữa người dùng với nhau, người dùng chia sẻ các bài viết blog về trải nghiệm dịch vụ khám phá của họ. Trong tương lai hệ thống hướng tới như là một mạng xã hội phát triển về ăn uống du lịch cho người dùng Việt và khách du lịch khi tới Việt Nam.

Tóm lại, thông qua quá trình nghiên cứu và thực hiện, khóa luận đã xây dựng được một hệ thống có tính ứng dụng thực tiễn cao, góp phần kết nối công nghệ trí tuệ nhân tạo với nhu cầu khám phá dịch vụ địa phương của người dùng. Hy vọng rằng với nền tảng kỹ thuật hiện có, hệ thống Khám Phá Xung Quanh có thể tiếp tục được phát triển và hoàn thiện, trở thành một nền tảng hữu ích, mang lại giá trị thiết thực cho người dùng trong tương lai.

## Tài liệu tham khảo

- [1] Postman Inc. (2025). *Postman - The Collaboration Platform for API Development*. [online] Available at: <https://www.postman.com/> [Accessed 10 Feb. 2025].
- [2] Spring Team. (2025). *Spring Framework - Official Documentation*. [online] Available at: <https://spring.io/projects/spring-framework> [Accessed 10 Feb. 2025].
- [3] Spring Team. (2025). *Spring Security - Powerful and Customizable Authentication and Access-Control Framework*. [online] Available at: <https://spring.io/projects/spring-security> [Accessed 10 Feb. 2025].
- [4] Spring Team. (2025). *Spring Data JPA - Simplifying Database Access in Spring Applications*. [online] Available at: <https://spring.io/projects/spring-data-jpa> [Accessed 10 Feb. 2025].
- [5] Hibernate ORM Team. (2025). *Hibernate ORM - The Object Relational Mapping Framework*. [online] Available at: <https://hibernate.org/orm/> [Accessed 10 Feb. 2025].
- [6] Oracle Corporation. (2025). *MySQL - The World's Most Popular Open Source Database*. [online] Available at: <https://www.mysql.com/> [Accessed 10 Feb. 2025].
- [7] Python Software Foundation. (2025). *Python Official Website*. [online] Available at: <https://www.python.org/> [Accessed 10 Feb. 2025].
- [8] Meta Platforms, Inc. (2025). *React – A JavaScript Library for Building User Interfaces*. [online] Available at: <https://react.dev/> [Accessed 10 Feb. 2025].
- [9] Martin, R.C. (2000). *SOLID Principles of Object-Oriented Design*. [online] Available at: <https://objectmentor.com/> [Accessed 10 Feb. 2025].
- [10] Meta AI. (2025). *Llama - Open Foundation and Fine-Tuned Language Models*. [online] Available at: <https://ai.meta.com/llama/> [Accessed 10 Feb. 2025].
- [11] Mateo Weather. (2025). *Mateo Weather - Smart Weather Forecasting App*. [online] Available at: <https://www.mateoweather.com/> [Accessed 10 Feb. 2025].
- [12] OpenStreetMap Contributors. (2025). *OpenStreetMap - Free Wiki World Map*. [online] Available at: <https://www.openstreetmap.org/> [Accessed 10 Feb. 2025].
- [13] TensorFlow. (n.d.). *TensorFlow Official Website*. [online] Available at: <https://www.tensorflow.org/>

- //www.tensorflow.org/?hl=vi [Accessed 10 Feb. 2025].
- [14] TensorFlow. (n.d.). *ResNet152V2 - Keras Applications*. [online] Available at: [https://www.tensorflow.org/api\\_docs/python/tf/keras/applications/ResNet152V2](https://www.tensorflow.org/api_docs/python/tf/keras/applications/ResNet152V2) [Accessed 10 Feb. 2025].
- [15] Docker Inc. (n.d.). *Docker Documentation*. [online] Available at: <http://docs.docker.com/> [Accessed 10 Feb. 2025].
- [16] Hugging Face. (n.d.). *Inception ResNet V2 - Timm Model Documentation*. [online] Available at: <https://huggingface.co/docs/timm/models/inception-resnet-v2> [Accessed 10 Feb. 2025].