

ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA CÔNG NGHỆ THÔNG TIN

ĐỒ ÁN TỐT NGHIỆP

NGÀNH: CÔNG NGHỆ THÔNG TIN
CHUYÊN NGÀNH: CÔNG NGHỆ PHẦN MỀM

ĐỀ TÀI:

ỨNG DỤNG NHẬN DIỆN MÓN ĂN VIỆT NAM VÀ TÌM KIẾM QUÁN ĂN

Người hướng dẫn: TS. LÊ THỊ MỸ HẠNH
Sinh viên thực hiện: NGUYỄN DƯƠNG BẢO NHẬT
Số thẻ sinh viên: 102210268
Lớp: 21TCLC_DT4

Đà Nẵng, 06/2025

ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA CÔNG NGHỆ THÔNG TIN

ĐỒ ÁN TỐT NGHIỆP

NGÀNH: CÔNG NGHỆ THÔNG TIN
CHUYÊN NGÀNH: CÔNG NGHỆ PHẦN MỀM

ĐỀ TÀI:

ỨNG DỤNG NHẬN DIỆN MÓN ĂN VIỆT NAM VÀ TÌM KIẾM QUÁN ĂN

Người hướng dẫn: TS. LÊ THỊ MỸ HẠNH

Sinh viên thực hiện: NGUYỄN DƯƠNG BẢO NHẬT

Số thẻ sinh viên: 102210268

Lớp: 21TCLC_DT4

Đà Nẵng, 06/2025

TÓM TẮT

Tên đề tài: Ứng dụng nhận diện món ăn Việt Nam và tìm kiếm quán ăn

Sinh viên thực hiện: Nguyễn Dương Bảo Nhật

Số thẻ SV: 102210268 Lớp: 21TCLC_DT4

Tóm tắt nội dung đồ án:

Đồ án xây dựng một hệ thống ứng dụng di động hỗ trợ người dùng nhận diện và tìm hiểu về các món ăn Việt Nam thông qua công nghệ trí tuệ nhân tạo (AI). Ứng dụng cho phép người dùng chụp ảnh món ăn bằng camera trên thiết bị di động, sau đó sử dụng mô hình AI để tự động nhận diện tên món ăn. Dựa trên kết quả nhận diện, hệ thống cung cấp cho người dùng các thông tin chi tiết về món ăn như nguồn gốc, nguyên liệu, cách chế biến và các địa điểm gợi ý để thưởng thức món ăn đó.

Bên cạnh việc đáp ứng nhu cầu tìm hiểu, khám phá ẩm thực Việt Nam một cách nhanh chóng, tiện lợi, ứng dụng còn góp phần quảng bá văn hóa ẩm thực Việt tới cộng đồng và du khách. Hệ thống được thiết kế với giao diện thân thiện, thao tác đơn giản, phù hợp với nhiều đối tượng người dùng.

Quá trình thực hiện đồ án giúp em có cơ hội tiếp cận và vận dụng các kiến thức về trí tuệ nhân tạo, đặc biệt là lĩnh vực nhận diện hình ảnh, cũng như các kỹ năng phát triển ứng dụng di động. Đề tài không chỉ nâng cao hiểu biết về công nghệ AI và quy trình xây dựng hệ thống thực tế mà còn cho thấy tiềm năng ứng dụng AI vào các lĩnh vực đời sống, hướng tới việc tối ưu hóa trải nghiệm người dùng và nâng cao giá trị các sản phẩm công nghệ Việt Nam.

NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

Họ tên sinh viên: Nguyễn Dương Bảo Nhật

Số thẻ sinh viên: 102210268

Lớp: 21TCLC_DT4

Khoa: Công Nghệ Thông Tin

Ngành: Công nghệ phần mềm (Đặc thù - Hợp tác doanh nghiệp)

1. *Tên đề tài đồ án:* Ứng dụng nhận diện món ăn Việt Nam và tìm kiếm quán ăn
2. *Đề tài thuộc diện:* Có ký kết thỏa thuận sở hữu trí tuệ đối với kết quả thực hiện
3. *Các số liệu và dữ liệu ban đầu:* không có

4. *Nội dung các phần thuyết minh và tính toán:*

Nội dung thuyết minh bao gồm các phần:

Mở đầu: Giới thiệu tổng quan vấn đề, mục đích và phạm vi của đề tài, hướng tiếp cận và bố cục của đồ án

Chương 1. Cơ sở lý thuyết: Trình bày các lý thuyết, các công cụ và công nghệ đã sử dụng.

Chương 2. Phân tích và thiết kế: Chương này trình bày các biểu đồ, hướng tiếp cận và cách giải quyết các vấn đề.

Chương 3. Triển khai và đánh giá: mô tả chức năng vận hành hệ thống.

Kết luận: Trình bày các kết quả đạt được, chỉ ra các hạn chế và đề xuất hướng phát triển.

5. *Các bản vẽ, đồ thị (ghi rõ các loại và kích thước bản vẽ): không có*

6. *Họ tên người hướng dẫn:* TS. Lê Thị Mỹ Hạnh

7. *Ngày giao nhiệm vụ đồ án:*/...../2025

8. *Ngày hoàn thành đồ án:*/...../2025

Đà Nẵng, ngày tháng 06 năm 2025

Trưởng bộ môn

Người hướng dẫn

.....

LỜI NÓI ĐẦU

Trong bối cảnh trí tuệ nhân tạo ngày càng được ứng dụng rộng rãi, việc kết hợp công nghệ hiện đại vào văn hóa, du lịch, và đặc biệt là ẩm thực trở nên phổ biến và cần thiết. Với mục tiêu hỗ trợ người dùng, đặc biệt là du khách quốc tế, dễ dàng nhận diện các món ăn Việt Nam và tìm kiếm các địa điểm thưởng thức phù hợp, đề tài **“Ứng dụng nhận diện món ăn Việt Nam và tìm kiếm quán ăn”** đã được thực hiện. Ứng dụng công nghệ nhận diện hình ảnh kết hợp nền tảng bản đồ số không chỉ đáp ứng nhu cầu thực tiễn mà còn góp phần quảng bá hiệu quả văn hóa ẩm thực Việt Nam.

Qua đề tài này, em đã có cơ hội tổng hợp và ứng dụng các kiến thức về trí tuệ nhân tạo, lập trình ứng dụng di động và tích hợp API, đồng thời rèn luyện kỹ năng nghiên cứu và phát triển phần mềm, phù hợp với định hướng đào tạo của Trường Đại học Bách Khoa – Đại học Đà Nẵng.

Trong quá trình thực hiện đề án, em đã nhận được sự hỗ trợ, hướng dẫn tận tình từ quý Thầy, Cô khoa Công nghệ Thông tin, đặc biệt là cô TS. Lê Thị Mỹ Hạnh – người đã trực tiếp hướng dẫn, định hướng và luôn đồng hành cùng em. Em cũng xin cảm ơn gia đình và bạn bè, những người đã luôn động viên và giúp đỡ em hoàn thành đề án này.

Mặc dù đã nỗ lực hoàn thành tốt nhất trong khả năng của mình, nhưng chắc chắn không tránh khỏi những thiếu sót. Em rất mong nhận được những góp ý quý báu từ quý Thầy, Cô để em có thể hoàn thiện hơn trong tương lai, cũng như tích lũy thêm kinh nghiệm cho con đường nghề nghiệp sắp tới.

Một lần nữa em xin chân thành cảm ơn.

CAM ĐOAN

Tôi xin cam đoan:

1. Báo cáo đồ án tốt nghiệp với tên đề tài : “Ứng dụng nhận diện món ăn Việt Nam và tìm kiếm quán ăn” là công trình nghiên cứu của chính cá nhân tôi dưới sự hướng dẫn trực tiếp của giảng viên TS. Lê Thị Mỹ Hạnh.
2. Tôi đã tự đọc nghiên cứu, dịch tài liệu và tổng hợp các kiến thức đã làm nên báo cáo này và đảm bảo không sao chép ở bất cứ đâu.
3. Những lý thuyết trong luận văn đều được sử dụng tài liệu như tôi đã tham khảo ở phần tài liệu tham khảo đã có trong báo cáo.

Nếu có vi phạm, tôi xin chịu hoàn toàn trách nhiệm.

Sinh viên thực hiện

Nguyễn Dương Bảo Nhật

MỤC LỤC

TÓM TẮT -----	i
NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP -----	ii
LỜI NÓI ĐẦU -----	iii
CAM ĐOAN -----	iv
MỤC LỤC -----	v
DANH SÁCH CÁC BẢNG, HÌNH VẼ -----	vii
DANH SÁCH CÁC TỪ VIẾT TẮT -----	ix
MỞ ĐẦU -----	1
Chương 1: CƠ SỞ LÝ THUYẾT -----	4
1.1. Tổng quan về trí tuệ nhân tạo và nhận diện hình ảnh -----	4
1.1.1. Khái niệm trí tuệ nhân tạo (AI), học sâu (deep learning)-----	4
1.1.2. Nhận diện hình ảnh bằng AI-----	4
1.2. Mô hình EfficientNet B4 trong nhận diện hình ảnh -----	5
1.2.1. Tổng quan kiến trúc EfficientNet-----	5
1.2.2. Đặc điểm và ưu điểm của EfficientNet B4-----	7
1.3. Phát triển ứng dụng di động với Flutter -----	8
1.3.1. Giới thiệu về Flutter-----	8
1.3.2. Ưu điểm của Flutter-----	8
1.3.3. Cấu trúc dự án Flutter sử dụngGetX-----	8
1.4. Xây dựng backend với FastAPI -----	10
1.4.1. Giới thiệu về FastAPI-----	10
1.4.2. Ưu điểm của FastAPI-----	10
1.4.3. Ứng dụng của FastAPI trong hệ thống-----	10
1.5. Hệ quản trị cơ sở dữ liệu PostgreSQL -----	11
1.5.1. Giới thiệu về PostgreSQL-----	11
1.5.2. Ưu điểm của PostgreSQL-----	11
1.5.3. Ứng dụng của PostgreSQL trong hệ thống-----	12

1.6. Tích hợp các dịch vụ ngoài trong hệ thống	12
1.6.1. Google Maps API (Places, Directions)	12
1.6.2. Google Sign-in	12
1.6.3. Firebase Cloud Messaging (FCM)	13
1.6.4. Zoho Mail	13
1.6.5. Cloudinary	13
1.7. Kết chương	14
Chương 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG	15
2.1. Phân tích yêu cầu	15
2.1.1. Giới thiệu bài toán	15
2.1.2. Đặc tả yêu cầu chức năng	16
2.1.3. Đặc tả yêu cầu phi chức năng	18
2.2. Thiết kế hệ thống	19
2.2.1. Sơ đồ phân rã chức năng	19
2.2.2. Biểu đồ ca sử dụng	20
2.2.3. Biểu đồ hoạt động	26
2.2.4. Biểu đồ tuần tự	28
2.2.5. Thiết kế cơ sở dữ liệu	33
2.2.6. Xây dựng mô hình nhận diện món ăn	38
2.2.7. Kiến trúc hệ thống	43
Chương 3: TRIỂN KHAI VÀ KẾT QUẢ ĐẠT ĐƯỢC	45
3.1. Môi trường phát triển	45
3.2. Kết quả đạt được	46
3.3. Đánh giá kết quả	54
KẾT LUẬN	55
TÀI LIỆU THAM KHẢO	57

DANH SÁCH CÁC BẢNG, HÌNH VẼ

Bảng 2.1: Bảng Users	34
Bảng 2.2: Bảng Dishes	34
Bảng 2.3: Bảng Dish_translations	35
Bảng 2.4: Bảng Favorite_dishes.....	35
Bảng 2.5: Bảng Favorite_restaurants	35
Bảng 2.6: Bảng Friend_requests.....	36
Bảng 2.7: Bảng Friendships	36
Bảng 2.8: Bảng Recognition_history	36
Bảng 2.9: Bảng Shared_items	37
Bảng 2.10: Bảng Training_data.....	37
Bảng 2.11: Bảng User_fcm_tokens.....	38
Bảng 3.1: Môi trường và công nghệ triển khai hệ thống.....	45
Hình 1.1: So sánh các phương pháp mở rộng mạng nơ-ron: mở rộng độ rộng (width), độ sâu (depth), độ phân giải (resolution) và phương pháp Compound Scaling trong EfficientNet.	6
Hình 1.2: Sơ đồ kiến trúc tổng quát của EfficientNet-B0	6
Hình 1.3: Biểu đồ so sánh hiệu suất và kích thước mô hình EfficientNet và các kiến trúc CNN phổ biến khác.	7
Hình 2.1: Sơ đồ phân rã chức năng	19
Hình 2.2: Biểu đồ ca sử dụng của quản trị viên	20
Hình 2.3: Biểu đồ ca sử dụng màn hình quản lý người dùng.....	21
Hình 2.4: Biểu đồ ca sử dụng màn hình quản lý món ăn	21
Hình 2.5: Biểu đồ ca sử dụng của người dùng	22
Hình 2.6: Biểu đồ ca sử dụng của chức năng tương tác món ăn.....	23
Hình 2.7: Biểu đồ ca sử dụng của chức năng quản lý tài khoản	23
Hình 2.8: Biểu đồ ca sử dụng của chức năng Lịch sử và yêu thích	24
Hình 2.9: Biểu đồ ca sử dụng của chức năng tương tác quán ăn	25
Hình 2.10: Biểu đồ ca sử dụng của tính năng xã hội.....	25
Hình 2.11: Biểu đồ hoạt động chức năng đăng nhập	26
Hình 2.12: Biểu đồ hoạt động của chức năng nhận diện món ăn.....	27
Hình 2.13: Biểu đồ hoạt động chức năng tìm kiếm và xem chi tiết quán ăn	27
Hình 2.14: Biểu đồ hoạt động mô tả chức năng quản lý tài khoản	28

Hình 2.15: Biểu đồ tuần tự chức năng đăng nhập	29
Hình 2.16: Biểu đồ tuần tự chức năng nhận diện món ăn	30
Hình 2.17: Biểu đồ tuần tự chức năng tìm kiếm và xem chi tiết quán ăn	30
Hình 2.18: Biểu đồ tuần tự cho chức năng quản lý tài khoản gồm xem và cập nhật thông tin, đổi mật khẩu, đổi email	31
Hình 2.19: Biểu đồ tuần tự cho chức năng quản lý tài khoản gồm liên kết/hủy liên kết tài khoản Goolge và xoá tài khoản	32
Hình 2.20: Lược đồ ERD.....	33
Hình 2.21: Ma trận nhầm lẫn mô hình EfficientNet B4 – các lớp từ 0-19.....	41
Hình 2.22: Ma trận nhầm lẫn mô hình EfficientNet B4 – các lớp từ 20-39.....	41
Hình 2.23: Báo cáo các chỉ số precision, recall, f1-score theo từng lớp món ăn.	42
Hình 2.24: Sơ đồ kiến trúc hệ thống theo mô hình Client – Server	43
Hình 3.1: Giao diện màn hình chào mừng và đăng nhập của ứng dụng Vietfood Lens	46
Hình 3.2: Giao diện quy trình đăng ký của ứng dụng Vietfood Lens	47
Hình 3.3: Giao diện quá trình nhận diện món ăn	48
Hình 3.4: Giao diện xem thông tin món ăn và tìm kiếm quán ăn gần đó.....	49
Hình 3.5: Giao diện chức năng yêu thích và lịch sử nhận diện.....	50
Hình 3.6: Giao diện quản lý bạn bè.....	51
Hình 3.7: Giao diện quản lý tài khoản người dùng	52
Hình 3.8: Giao diện trang quản trị hệ thống.....	53
Hình 3.9: Giao diện trang quản lý người dùng.....	53

DANH SÁCH CÁC TỪ VIẾT TẮT

Từ viết tắt	Tên đầy đủ
CSDL	Cơ sở dữ liệu
CNTT	Công nghệ thông tin
DL	Dữ liệu
API	Application Programming Interface
AI	Artificial Intelligence (Trí tuệ nhân tạo)

MỞ ĐẦU

1. Lý do chọn đề tài

Ẩm thực Việt Nam nổi tiếng với sự phong phú và đa dạng, thu hút sự quan tâm của người dùng trong nước và du khách quốc tế. Tuy nhiên, việc nhận diện món ăn Việt Nam qua hình ảnh và tìm kiếm địa điểm thưởng thức phù hợp vẫn là thách thức, đặc biệt với du khách nước ngoài hoặc những người mới tiếp cận. Các ứng dụng nhận diện hình ảnh hiện tại (như Google Lens) chưa được tối ưu hóa cho ẩm thực Việt Nam và hiếm có giải pháp nào tích hợp cả nhận diện món ăn lẫn tìm kiếm quán ăn dựa trên vị trí.

Sự phát triển của trí tuệ nhân tạo (AI), đặc biệt là các mô hình nhận diện hình ảnh, cùng với các API bản đồ như Google Maps, mở ra cơ hội xây dựng một ứng dụng di động đáp ứng nhu cầu thực tiễn này. Đề tài “Ứng dụng nhận diện món ăn Việt Nam và tìm kiếm quán ăn” được lựa chọn nhằm:

- Phát triển một giải pháp công nghệ tích hợp, giúp người dùng dễ dàng nhận diện món ăn Việt Nam và khám phá quán ăn phù hợp.
- Góp phần quảng bá văn hóa ẩm thực Việt Nam thông qua công nghệ hiện đại.
- Ứng dụng các kiến thức về AI, phát triển ứng dụng di động và tích hợp API, phù hợp với định hướng nghiên cứu và đào tạo của Trường Đại học Bách Khoa – Đại học Đà Nẵng.

Với ý nghĩa thực tiễn và tiềm năng khoa học, đề tài không chỉ đáp ứng nhu cầu của người dùng mà còn đóng góp vào việc ứng dụng công nghệ vào lĩnh vực văn hóa và du lịch.

2. Mục đích

Mục đích của đề tài là xây dựng một ứng dụng di động thông minh, hỗ trợ người dùng nhận diện các món ăn Việt Nam thông qua hình ảnh và cung cấp thông tin chi tiết về món ăn, đồng thời tích hợp chức năng tìm kiếm quán ăn gần vị trí hiện tại. Ứng dụng góp phần quảng bá văn hóa ẩm thực Việt Nam bằng công nghệ hiện đại, nâng cao trải nghiệm khám phá ẩm thực cho người dùng trong nước và du khách quốc tế.

2.1. Sản phẩm

Sản phẩm của đề tài là một ứng dụng di động (Vietfood Lens) chạy trên nền tảng Android, với các chức năng chính gồm: nhận diện món ăn Việt Nam qua hình ảnh chụp từ camera hoặc chọn từ thư viện; hiển thị thông tin chi tiết về món ăn như tên, nguồn gốc, nguyên liệu, cách chế biến, giá trị dinh dưỡng; tìm kiếm, gợi ý quán ăn gần vị trí người dùng thông qua Google Maps API; quản lý tài khoản cá nhân, lưu trữ lịch sử nhận

diện và danh sách yêu thích; hỗ trợ đa ngôn ngữ (Tiếng Việt, Tiếng Anh), gửi thông báo đẩy, cùng các tính năng xã hội phục vụ nhu cầu chia sẻ và kết nối người dùng.

2.2. Mục tiêu hệ thống

- Đạt độ chính xác cao trong nhận diện món ăn qua ảnh.
- Cung cấp thông tin món ăn đầy đủ, tin cậy, dễ tra cứu.
- Hỗ trợ tìm kiếm, chỉ đường đến quán ăn phục vụ món ăn đã nhận diện.
- Giao diện thân thiện, dễ sử dụng cho nhiều đối tượng người dùng.
- Tích hợp các tính năng cá nhân hóa: lưu lịch sử, yêu thích, thông báo.
- Đảm bảo an toàn dữ liệu người dùng và bảo mật hệ thống.

2.3. Tính năng

Nhận diện món ăn:

- Chụp ảnh món ăn bằng camera hoặc chọn ảnh từ thư viện.
- Nhận diện món ăn Việt Nam bằng mô hình AI (CNN), hiển thị kết quả kèm độ tin cậy.

Thông tin món ăn:

- Hiển thị thông tin chi tiết về món ăn (tên, nguyên liệu, công thức, nguồn gốc, giá trị dinh dưỡng).
- Cung cấp danh sách tất cả các món ăn có thể nhận diện để người dùng tham khảo.

Tìm kiếm và khám phá nhà hàng:

- Tìm kiếm nhà hàng phục vụ món ăn đã nhận diện trong bán kính 5km, sử dụng Google Maps API (Places API).
- Hiển thị thông tin nhà hàng (địa chỉ, đánh giá, giờ mở cửa), hỗ trợ lọc theo đánh giá.
- Tích hợp Google Maps API (Directions API) để xem vị trí nhà hàng trên bản đồ và chỉ đường.

Quản lý tài khoản người dùng:

- Đăng ký, đăng nhập bằng email/mật khẩu hoặc Google Sign-In.
- Xác thực qua OTP, đặt lại mật khẩu, chỉnh sửa thông tin cá nhân, thay đổi email/mật khẩu, xóa tài khoản.

Tính năng xã hội và cá nhân hoá:

- Kết nối bạn bè và chia sẻ thông tin món ăn/nhà hàng với người dùng khác.
- Lưu món ăn, nhà hàng yêu thích; xem lịch sử nhận diện.

Hỗ trợ và giao diện:

- Hỗ trợ đa ngôn ngữ (Tiếng Việt/Anh)
- Nhận thông báo qua Firebase Messaging. Giao diện trực quan, dễ sử dụng.

2.4. Đối tượng người dùng

- Người yêu thích ẩm thực Việt Nam: Những cá nhân muốn tìm hiểu về món ăn, công thức, hoặc giá trị dinh dưỡng của các món ăn Việt Nam.
- Du khách quốc tế: Những người cần nhận diện món ăn và tìm kiếm quán ăn địa phương khi du lịch tại Việt Nam.
- Người mới tiếp cận ẩm thực Việt Nam muốn tra cứu thông tin, công thức, địa điểm ăn uống và các cá nhân yêu thích công nghệ nhận diện hình ảnh, ứng dụng AI vào đời sống.

2.5. Phạm vi

Ứng dụng tập trung vào việc nhận diện 40 món ăn Việt Nam qua hình ảnh, đồng thời hỗ trợ tìm kiếm các quán ăn phục vụ những món này trong bán kính khoảng 5km từ vị trí người dùng. Việc tìm kiếm và chỉ đường đến quán ăn được thực hiện thông qua tích hợp Google Maps API. Hệ thống hiện triển khai trên nền tảng Android và có khả năng mở rộng về số lượng món ăn cũng như nền tảng sử dụng trong tương lai. Ứng dụng không bao gồm chức năng đặt món hoặc thanh toán trực tuyến.

2.6. Công nghệ phát triển

- Quản lý và lưu trữ mã nguồn: Github
- Ngôn ngữ: Dart (Flutter), Python (FastAPI)
- Framework/Library: Flutter, FastAPI, EfficientNet B4 (pretrained), PyTorch
- Lưu trữ ảnh: Cloudinary
- Cơ sở dữ liệu: PostgreSQL
- API, dịch vụ tích hợp: Google Maps API, Google Sign-in, Zoho Mail
- Thông báo đẩy: Firebase Cloud Messaging (FCM)
- Triển khai backend: Digital Ocean
- Triển khai ứng dụng di động: Google Play Store

Chương 1: CƠ SỞ LÝ THUYẾT

1.1. Tổng quan về trí tuệ nhân tạo và nhận diện hình ảnh

1.1.1. Khái niệm trí tuệ nhân tạo (AI), học sâu (deep learning)

Trí tuệ nhân tạo (AI) là lĩnh vực khoa học máy tính tập trung vào việc phát triển các hệ thống có khả năng thực hiện các nhiệm vụ đòi hỏi trí thông minh của con người, như học hỏi, phân tích và nhận diện hình ảnh. AI cho phép xử lý khối lượng lớn dữ liệu, tự động phát hiện các mẫu và quy luật, từ đó đưa ra quyết định hoặc dự đoán. Trong đề tài Vietfood Lens, AI được ứng dụng để nhận diện 40 món ăn Việt Nam phổ biến (phở, bánh mì, bún bò Huế, v.v.) từ hình ảnh, hỗ trợ người dùng khám phá ẩm thực và tìm kiếm quán ăn.

Học sâu (deep learning), một nhánh của học máy, sử dụng mạng nơ-ron nhân tạo với nhiều tầng để mô phỏng khả năng xử lý thông tin của con. Mạng nơ-ron sâu (Deep Neural Network - DNN) tự động trích xuất các đặc trưng phức tạp từ dữ liệu, đặc biệt hiệu quả trong các nhiệm vụ như nhận diện hình ảnh và phân loại. Sự phát triển của công nghệ xử lý dữ liệu lớn và phần cứng (như GPU) đã thúc đẩy học sâu trở thành công cụ mạnh mẽ trong AI hiện đại. Trong Vietfood Lens, học sâu được áp dụng thông qua mô hình EfficientNet B4 để nhận diện món ăn với độ chính xác cao, đáp ứng nhu cầu thực tiễn và góp phần quảng bá văn hóa ẩm thực Việt Nam.

1.1.2. Nhận diện hình ảnh bằng AI

Nhận diện hình ảnh là một lĩnh vực quan trọng thuộc Thị giác Máy tính (Computer Vision), tập trung vào việc phát triển các hệ thống và thuật toán cho phép máy tính hiểu và diễn giải nội dung của hình ảnh kỹ thuật số hoặc video. Mục tiêu của nhận diện hình ảnh là xác định, phân loại và đôi khi là định vị các đối tượng, con người, địa điểm, hoặc các mẫu cụ thể trong một bức ảnh.

Quá trình nhận diện hình ảnh bằng AI, đặc biệt khi sử dụng học sâu, thường bao gồm các bước chính:

- Thu thập và Tiền xử lý Dữ liệu (Data Collection and Preprocessing): Giai đoạn này bao gồm việc thu thập một tập dữ liệu hình ảnh đủ lớn và đa dạng. Đối với đề án Vietfood Lens, đây là quá trình xây dựng bộ dữ liệu gồm hình ảnh của 40 món ăn Việt Nam đặc trưng. Dữ liệu sau đó được tiền xử lý thông qua các kỹ thuật như thay đổi kích thước, chuẩn hóa (normalization) và tăng cường dữ liệu (data augmentation) để cải thiện khả năng tổng quát hóa của mô hình.

- Trích xuất Đặc trưng (Feature Extraction): Trong các phương pháp truyền thống, các đặc trưng như màu sắc, kết cấu, hình dạng được trích xuất thủ công. Tuy nhiên, với

học sâu, đặc biệt là các Mạng Nơ-ron Tích chập (CNNs), quá trình này được tự động hóa. Các lớp tích chập trong CNN tự học các đặc trưng phân cấp, từ các chi tiết đơn giản như cạnh, góc ở các lớp đầu, đến các cấu trúc phức tạp hơn như bộ phận của đối tượng hay toàn bộ đối tượng ở các lớp sâu hơn, giúp mô hình "hiểu" được nội dung hình ảnh.

- Huấn luyện Mô hình (Model Training): Dữ liệu đã tiền xử lý và các đặc trưng (tự động học được) được sử dụng để huấn luyện một mô hình học máy/học sâu. Trong Vietfood Lens, mô hình EfficientNet B4 được huấn luyện trên bộ dữ liệu 40 món ăn. Quá trình này bao gồm việc tối ưu hóa các tham số của mô hình dựa trên một hàm mất mát (loss function) để giảm thiểu sai số giữa dự đoán của mô hình và nhãn thực tế của dữ liệu.

- Đánh giá Mô hình (Model Evaluation): Sau khi huấn luyện, mô hình được đánh giá trên một tập dữ liệu kiểm thử riêng biệt (dữ liệu mà mô hình chưa từng thấy trong quá trình huấn luyện) bằng các độ đo như độ chính xác (accuracy), precision, recall, F1-score. Việc này nhằm đảm bảo mô hình có khả năng hoạt động tốt trên dữ liệu thực tế và không bị quá khớp (overfitting) với dữ liệu huấn luyện.

Các bài toán chính trong nhận diện hình ảnh bao gồm:

- Phân loại ảnh (Image Classification): Gán một nhãn duy nhất cho toàn bộ bức ảnh (ví dụ: xác định bức ảnh này chứa "phở", "bánh mì" hay "bún bò Huế"). Đây chính là nhiệm vụ cốt lõi của đề án Vietfood Lens.

- Phát hiện đối tượng (Object Detection): Không chỉ phân loại mà còn xác định vị trí (thường bằng hộp giới hạn - bounding box) của một hoặc nhiều đối tượng trong ảnh.

- Phân đoạn ảnh (Image Segmentation): Phân loại từng pixel trong ảnh thuộc về đối tượng nào, tạo ra một bản đồ chi tiết về các đối tượng trong ảnh, cho phép hiểu sâu hơn về bố cục và nội dung của ảnh.

Sự ra đời và phát triển của học sâu, đặc biệt là kiến trúc Mạng Nơ-ron Tích chập (CNN), đã tạo ra một cuộc cách mạng trong lĩnh vực nhận diện hình ảnh. CNN có khả năng học các biểu diễn đặc trưng mạnh mẽ trực tiếp từ dữ liệu pixel thô, vượt trội hơn hẳn các phương pháp truyền thống và đạt được hiệu suất đột phá trên nhiều bộ dữ liệu và bài toán thách thức, điển hình là trong việc nhận diện các đối tượng phức tạp như món ăn.

1.2. Mô hình EfficientNet B4 trong nhận diện hình ảnh

1.2.1. Tổng quan kiến trúc EfficientNet

1.2.2. Đặc điểm và ưu điểm của EfficientNet B4

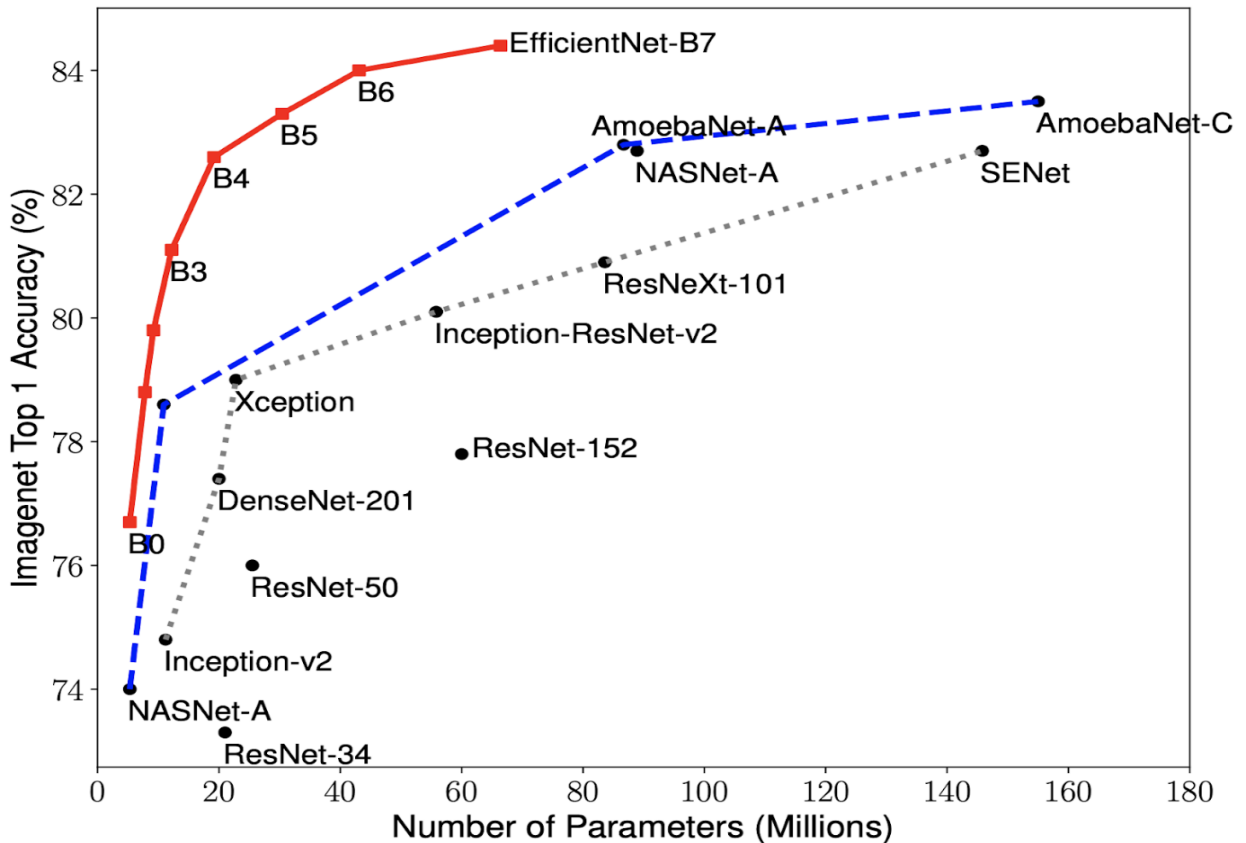
EfficientNet-B4 là một trong những phiên bản mở rộng của EfficientNet, được thiết kế để cân bằng giữa độ chính xác và hiệu quả tính toán. Một số đặc điểm nổi bật của EfficientNet-B4 bao gồm:

Hiệu suất cao: EfficientNet-B4 đạt được độ chính xác vượt trội trên các bộ dữ liệu chuẩn như ImageNet, với số lượng tham số và FLOPs (số phép toán dấu chấm động) thấp hơn đáng kể so với các mô hình CNN truyền thống.

Khả năng tổng quát hóa tốt: Nhờ vào kiến trúc tối ưu và phương pháp Compound Scaling, EfficientNet-B4 có khả năng học và tổng quát hóa tốt trên nhiều loại dữ liệu khác nhau, bao gồm cả các bộ dữ liệu nhỏ và phức tạp.

Hiệu quả tính toán: Với số lượng tham số và yêu cầu tính toán thấp hơn, EfficientNet-B4 phù hợp để triển khai trên các thiết bị có tài nguyên hạn chế như điện thoại di động hoặc các hệ thống nhúng.

Khả năng mở rộng: EfficientNet-B4 có thể được mở rộng hoặc thu nhỏ tùy theo yêu cầu về độ chính xác và tài nguyên tính toán, nhờ vào phương pháp Compound Scaling linh hoạt.



Hình 1.3: Biểu đồ so sánh hiệu suất và kích thước mô hình EfficientNet và các kiến trúc CNN phổ biến khác.

1.3. Phát triển ứng dụng di động với Flutter

1.3.1. Giới thiệu về Flutter

Flutter là một framework giao diện người dùng (UI) mã nguồn mở, miễn phí do Google phát triển, được ra mắt lần đầu vào năm 2017. Flutter cho phép xây dựng các ứng dụng di động chất lượng cao, có thể chạy trên cả hai nền tảng iOS và Android từ một mã nguồn duy nhất. Đây không chỉ là một framework mà còn là một bộ công cụ phát triển phần mềm (SDK) hoàn chỉnh, bao gồm các công cụ biên dịch mã nguồn Dart thành mã máy native cho từng nền tảng. Flutter sử dụng ngôn ngữ lập trình Dart, một ngôn ngữ hướng đối tượng, nhanh và hiện đại, giúp phát triển giao diện người dùng một cách hiệu quả và linh hoạt. Với Flutter, các nhà phát triển có thể tạo ra các giao diện đẹp mắt, mượt mà và có hiệu suất gần như ứng dụng native.

1.3.2. Ưu điểm của Flutter

Flutter sở hữu nhiều ưu điểm nổi bật giúp nó trở thành lựa chọn hàng đầu trong phát triển ứng dụng di động đa nền tảng:

Phát triển đa nền tảng (Cross-platform): Viết một lần, chạy trên nhiều nền tảng như iOS, Android, web, Windows, MacOS và Linux, giúp tiết kiệm thời gian và chi phí phát triển.

Hiệu suất cao: Flutter không dựa vào các thành phần UI của nền tảng mà sử dụng công cụ kết xuất riêng, giúp ứng dụng chạy mượt mà, hình ảnh động sắc nét, thời gian phản hồi nhanh như ứng dụng native. Flutter cũng dùng biên dịch Ahead-of-Time (AOT) và Just-in-Time (JIT) để tối ưu tốc độ khởi chạy và phát triển.

Giao diện người dùng đẹp và linh hoạt: Flutter cung cấp bộ thư viện widget phong phú, cho phép tạo các giao diện tùy chỉnh, bắt mắt và tương tác cao. Mọi thành phần trong Flutter đều là widget, giúp dễ dàng xây dựng và tái sử dụng UI.

Hot Reload: Tính năng này cho phép lập trình viên thấy ngay kết quả thay đổi mã nguồn trên giao diện mà không cần khởi động lại ứng dụng, tăng tốc quá trình phát triển và sửa lỗi.

Cộng đồng và hỗ trợ mạnh mẽ: Flutter được Google phát triển và duy trì, có cộng đồng đông đảo, tài liệu phong phú, cùng nhiều plugin hỗ trợ tích hợp các tính năng như camera, GPS, cảm biến... giúp phát triển ứng dụng đa chức năng dễ dàng hơn.

Tối ưu chi phí phát triển: Nhờ sử dụng một mã nguồn duy nhất cho nhiều nền tảng, Flutter giúp giảm đáng kể chi phí và thời gian phát triển, đồng thời dễ dàng bảo trì và mở rộng ứng dụng.

1.3.3. Cấu trúc dự án Flutter sử dụng GetX

- lib/

Thư mục chứa mã nguồn chính, được chia thành các thư mục con như sau:

- **bindings/**

Chứa các lớp Binding kế thừa từ Bindings của GetX, dùng để khai báo và nạp các Controller (dependency injection) cho từng màn hình hoặc module. Việc này giúp tách biệt và quản lý phụ thuộc một cách rõ ràng.

- **controllers/**

Chứa các lớp Controller kế thừa từ GetxController, quản lý trạng thái và logic nghiệp vụ cho từng màn hình hoặc module. Mỗi màn hình thường có một controller riêng để xử lý dữ liệu và trạng thái.

- **views/**

Chứa các widget giao diện người dùng (UI) của ứng dụng. Các widget này thường kế thừa từ GetWidget hoặc sử dụng GetBuilder/Obx để lắng nghe và phản hồi trạng thái từ Controller.

- **models/**

Chứa các lớp mô hình dữ liệu (data models) dùng trong ứng dụng.

- **services/**

Chứa các lớp xử lý nghiệp vụ liên quan đến API, cơ sở dữ liệu, hoặc các dịch vụ nền tảng khác.

- **utils/ hoặc helpers/**

Chứa các hàm tiện ích, hằng số hoặc các helper functions dùng chung.

- **main.dart**

Điểm vào ứng dụng, sử dụng GetMaterialApp thay cho MaterialApp để kích hoạt các tính năng của GetX như quản lý route, dependency injection và snackbar, dialog không cần context.

Ví dụ mô hình hoạt động của GetX trong dự án

- Mỗi màn hình bao gồm 3 thành phần chính:

Controller: kế thừa GetxController, quản lý trạng thái và logic.

Binding: kế thừa Bindings, dùng để đăng ký Controller với GetX.

View (Widget): kế thừa GetWidget hoặc widget bình thường sử dụng Obx để tự động cập nhật UI khi trạng thái thay đổi.

- Ưu điểm của cấu trúc này

Tách biệt rõ ràng giữa UI và business logic, giúp code sạch, dễ đọc và bảo trì.

Quản lý trạng thái hiệu quả với các biến .obs và widget Obx.

Dependency injection tự động, không cần truyền context hay phải khởi tạo thủ công.

Quản lý route và điều hướng đơn giản, không cần context.

Dễ dàng mở rộng và tái sử dụng các module.

1.4. Xây dựng backend với FastAPI

1.4.1. Giới thiệu về FastAPI

FastAPI là một framework web hiện đại và mạnh mẽ được phát triển bằng ngôn ngữ Python, chuyên dùng để xây dựng các API RESTful với hiệu suất cao. FastAPI tận dụng các tiêu chuẩn hiện đại như OpenAPI và JSON Schema để tự động tạo tài liệu API trực quan, đồng thời hỗ trợ lập trình bất đồng bộ (async/await) giúp tối ưu hiệu quả xử lý các yêu cầu đồng thời. Nhờ thiết kế hướng đến trải nghiệm lập trình viên, FastAPI giúp viết code nhanh chóng, dễ đọc, dễ bảo trì và giảm thiểu lỗi nhờ hỗ trợ gợi ý kiểu dữ liệu (type hints) chuẩn xác.

1.4.2. Ưu điểm của FastAPI

Hiệu suất cao: FastAPI có tốc độ xử lý rất nhanh, ngang ngửa với các nền tảng như NodeJS và Go, nhờ sử dụng Starlette và Pydantic cùng khả năng lập trình bất đồng bộ của Python.

Dễ sử dụng và phát triển nhanh: Việc viết API với FastAPI rất đơn giản, chỉ cần vài dòng code để tạo ra một API đầy đủ chức năng, đồng thời giảm thiểu lỗi lập trình nhờ kiểm tra kiểu dữ liệu tự động.

Tài liệu API tự động: FastAPI tự động sinh ra tài liệu API tương tác thông qua giao diện Swagger UI và Redoc, giúp dễ dàng thử nghiệm và hiểu rõ các endpoint mà không cần viết thêm tài liệu thủ công.

Hỗ trợ kiểm tra và xác thực dữ liệu: Sử dụng Pydantic, FastAPI giúp xác nhận và kiểm tra dữ liệu đầu vào một cách tự động, đảm bảo dữ liệu hợp lệ trước khi xử lý.

Hỗ trợ async/await: Tối ưu cho các ứng dụng cần xử lý nhiều yêu cầu đồng thời hoặc các tác vụ I/O-bound như truy cập cơ sở dữ liệu, gọi API bên ngoài.

Cộng đồng và tài liệu phong phú: FastAPI có cộng đồng phát triển năng động, tài liệu chi tiết và nhiều ví dụ thực tế giúp người dùng dễ dàng làm quen và áp dụng.

1.4.3. Ứng dụng của FastAPI trong hệ thống

Trong hệ thống Vietfood Lens, FastAPI đóng vai trò là backend chính, cung cấp các API phục vụ cho ứng dụng di động và giao tiếp với các dịch vụ liên quan:

- Xây dựng các API: Quản lý người dùng (đăng ký, đăng nhập, xác thực), xử lý upload ảnh, quản lý lịch sử nhận diện, yêu thích, thông báo, v.v.
- Tích hợp với mô hình AI: FastAPI đảm nhận vai trò trung gian nhận ảnh từ ứng dụng mobile, chuyển tiếp đến server nhận diện ảnh (EfficientNet B4), nhận kết quả và trả về client.

- Tích hợp dịch vụ ngoài: Quản lý xác thực bằng Google Sign-in, gửi email qua Zoho Mail, gửi thông báo đẩy qua Firebase Cloud Messaging.

- Quản lý dữ liệu: Kết nối với cơ sở dữ liệu PostgreSQL để lưu trữ thông tin người dùng, món ăn, lịch sử hoạt động,...

- Đảm bảo bảo mật và hiệu năng: Thực hiện xác thực, phân quyền người dùng (OAuth2, JWT), xử lý bất đồng bộ để đảm bảo hệ thống hoạt động ổn định, mở rộng dễ dàng khi số lượng người dùng tăng cao.

Việc sử dụng FastAPI giúp hệ thống backend của Vietfood Lens đảm bảo hiệu suất, độ ổn định và khả năng mở rộng, đồng thời hỗ trợ tích hợp tốt với các dịch vụ hiện đại phục vụ cho cả AI lẫn người dùng cuối.

1.5. Hệ quản trị cơ sở dữ liệu PostgreSQL

1.5.1. Giới thiệu về PostgreSQL

PostgreSQL là một hệ quản trị cơ sở dữ liệu quan hệ-đối tượng (Object-Relational Database Management System - ORDBMS) mã nguồn mở, được phát triển dựa trên dự án POSTGRES tại Đại học California, Berkeley. Đây là một trong những hệ quản trị cơ sở dữ liệu tiên tiến và phổ biến nhất hiện nay, có khả năng chạy trên nhiều nền tảng như Linux, Windows, MacOS và các hệ điều hành UNIX khác. PostgreSQL được cung cấp dưới giấy phép mã nguồn mở tự do, cho phép người dùng sử dụng, sửa đổi và phân phối mà không bị giới hạn.

1.5.2. Ưu điểm của PostgreSQL

- Hỗ trợ kiểu dữ liệu đa dạng: PostgreSQL hỗ trợ nhiều loại kiểu dữ liệu bao gồm kiểu nguyên thủy (số nguyên, số thực, chuỗi, boolean), kiểu cấu trúc (ngày/giờ, mảng, phạm vi, UUID), kiểu tài liệu (JSON/JSONB, XML), kiểu hình học và các kiểu dữ liệu tùy chỉnh, giúp lưu trữ và xử lý dữ liệu linh hoạt và phong phú.

- Tính toàn vẹn và bảo mật dữ liệu: Hệ thống hỗ trợ các ràng buộc như khóa chính, khóa ngoại, ràng buộc duy nhất, not-null, check constraint, cùng với các cơ chế xác thực và kiểm soát truy cập mạnh mẽ như LDAP, SCRAM-SHA-256, certificate, giúp đảm bảo tính chính xác và an toàn của dữ liệu.

- Hiệu suất và khả năng mở rộng: PostgreSQL có trình tối ưu hóa truy vấn phức tạp, hỗ trợ các loại chỉ mục đa dạng (b-tree, GiST, GIN...), truy vấn song song, streaming replication và khả năng mở rộng theo chiều ngang (sharding) và chiều dọc, đáp ứng tốt các ứng dụng có khối lượng dữ liệu lớn và số lượng người dùng đồng thời cao.

- Hỗ trợ các tính năng nâng cao: Bao gồm thủ tục sự kiện (trigger), view, transaction với tính toàn vẹn cao, multiversion concurrency control (MVCC) giúp xử lý đồng thời hiệu quả, cùng khả năng mở rộng với các ngôn ngữ thủ tục như PL/pgSQL, Perl, Python.

- Mã nguồn mở và cộng đồng phát triển mạnh: PostgreSQL được phát triển và duy trì bởi cộng đồng rộng lớn, liên tục cập nhật và cải tiến, đồng thời có nhiều module mở rộng như PostGIS cho dữ liệu không gian.

1.5.3. Ứng dụng của PostgreSQL trong hệ thống

Trong hệ thống Vietfood Lens, hệ quản trị cơ sở dữ liệu PostgreSQL được sử dụng làm nền tảng lưu trữ và quản lý dữ liệu quan hệ chính của ứng dụng. Cụ thể:

- Lưu trữ dữ liệu người dùng, món ăn, lịch sử nhận diện và các thông tin nghiệp vụ: PostgreSQL đảm bảo tính toàn vẹn dữ liệu nhờ hỗ trợ các ràng buộc khóa chính, khóa ngoại và các kiểm tra dữ liệu (constraints).

- Hỗ trợ truy vấn phức tạp và báo cáo: Khả năng xử lý các câu truy vấn đa dạng, tối ưu hóa truy vấn giúp hệ thống truy xuất, tổng hợp dữ liệu nhanh chóng, hiệu quả, phục vụ các chức năng như thống kê lịch sử, tìm kiếm món ăn, quán ăn

- Bảo mật dữ liệu: PostgreSQL cung cấp cơ chế xác thực người dùng và phân quyền truy cập chi tiết, đảm bảo dữ liệu được bảo vệ an toàn khỏi các truy cập trái phép.

- Khả năng mở rộng và chịu tải cao: Khi quy mô hệ thống tăng, PostgreSQL có thể mở rộng linh hoạt, đáp ứng tốt nhu cầu lưu trữ và truy cập dữ liệu của nhiều người dùng đồng thời mà vẫn đảm bảo hiệu năng.

- Tích hợp hiệu quả với backend FastAPI: PostgreSQL dễ dàng tích hợp với FastAPI thông qua các thư viện ORM như SQLAlchemy hoặc asyncpg, hỗ trợ phát triển backend một cách nhanh chóng, ổn định, thuận tiện cho việc triển khai các API dữ liệu.

1.6. Tích hợp các dịch vụ ngoài trong hệ thống

1.6.1. Google Maps API (Places, Directions)

Google Maps API cung cấp các dịch vụ bản đồ mạnh mẽ giúp ứng dụng tích hợp các tính năng định vị, tìm kiếm địa điểm và chỉ đường.

- Places API cho phép truy cập dữ liệu về hơn 200 triệu địa điểm toàn cầu, hỗ trợ tìm kiếm địa điểm theo tên, địa chỉ, số điện thoại và cung cấp thông tin chi tiết như đánh giá, hình ảnh, giờ mở cửa. Ngoài ra, tính năng autocomplete giúp người dùng nhập địa chỉ hoặc tên địa điểm chính xác, thuận tiện hơn.

- Directions API hỗ trợ tính toán lộ trình tối ưu giữa điểm xuất phát và điểm đến, áp dụng cho nhiều loại phương tiện như đi bộ, xe đạp, xe hơi, phương tiện công cộng. API sử dụng thuật toán định tuyến tiên tiến kết hợp dữ liệu giao thông thực tế, trả về hướng dẫn chi tiết từng bước, khoảng cách và thời gian dự kiến di chuyển.

Việc tích hợp Google Maps API giúp Vietfood Lens nâng cao trải nghiệm người dùng nhờ các tính năng bản đồ tương tác, tìm kiếm địa điểm và chỉ đường chính xác, nhanh chóng.

1.6.2. Google Sign-in

Google Sign-in là dịch vụ xác thực người dùng qua tài khoản Google, đơn giản hóa quy trình đăng nhập và tăng cường bảo mật cho ứng dụng.

- Tích hợp thông qua plugin `google_sign_in`, hỗ trợ đa nền tảng (Android, iOS, web, macOS).

- Quá trình tích hợp bao gồm đăng ký ứng dụng trên Google Cloud Platform, bật OAuth APIs, cấu hình consent screen và sử dụng API để thực hiện đăng nhập.

- Cho phép người dùng đăng nhập nhanh chóng bằng tài khoản Google, đồng thời truy cập các thông tin cơ bản như email, tên người dùng và ảnh đại diện.

- Hỗ trợ single sign-on, giúp người dùng không phải nhập lại thông tin đăng nhập nhiều lần trên các ứng dụng khác nhau.

Google Sign-in giúp tăng sự tiện lợi, đồng nhất trải nghiệm người dùng và bảo mật hệ thống.

1.6.3. Firebase Cloud Messaging (FCM)

Firebase Cloud Messaging (FCM) là dịch vụ gửi thông báo đẩy (push notification) miễn phí, đa nền tảng cho Android, iOS và web.

- Cho phép backend gửi thông báo hoặc dữ liệu đến thiết bị của người dùng ngay cả khi ứng dụng đang tắt hoặc ở chế độ nền.

- Hỗ trợ gửi thông báo theo nhóm người dùng, theo chủ đề, hoặc gửi trực tiếp đến từng thiết bị riêng lẻ.

- Tích hợp FCM giúp ứng dụng duy trì tương tác hiệu quả với người dùng, gửi cảnh báo, thông báo cập nhật hoặc các thông tin quan trọng kịp thời, tăng sự gắn kết và chủ động của hệ thống.

1.6.4. Zoho Mail

Zoho Mail là dịch vụ email doanh nghiệp cung cấp hệ thống gửi/nhận email chuyên nghiệp, bảo mật cao, dễ dàng tích hợp vào hệ thống backend.

- Được sử dụng để gửi email xác nhận, thông báo, hoặc các thông tin liên quan đến tài khoản và hoạt động của người dùng.

- Hỗ trợ giao thức SMTP, IMAP, POP3 giúp dễ dàng tích hợp với các backend framework như FastAPI.

- Cung cấp các tính năng quản lý hộp thư, lọc thư rác, bảo mật và lưu trữ email lâu dài, phù hợp với ứng dụng yêu cầu giao tiếp qua email chuyên nghiệp và an toàn.

1.6.5. Cloudinary

Cloudinary là nền tảng lưu trữ, quản lý và phân phối hình ảnh/video trên nền tảng đám mây, giúp tối ưu hóa hiệu suất và bảo mật khi làm việc với dữ liệu hình ảnh trong ứng dụng di động.

Lưu trữ và quản lý ảnh: Cloudinary cho phép upload, lưu trữ, quản lý, phân loại và truy cập ảnh nhanh chóng thông qua API hoặc dashboard.

Xử lý ảnh động và tiện lợi: Hỗ trợ các thao tác tự động như nén ảnh, thay đổi kích thước, chuyển định dạng, crop, watermark... trực tiếp qua URL hoặc API, giúp tiết kiệm tài nguyên và tối ưu hiển thị trên thiết bị di động.

Phân phối qua CDN: Hình ảnh sau khi lưu trữ được phân phối qua mạng lưới CDN toàn cầu, tăng tốc độ tải ảnh cho người dùng ở mọi khu vực địa lý.

Bảo mật: Hỗ trợ xác thực, phân quyền truy cập ảnh, đảm bảo dữ liệu ảnh không bị rò rỉ ngoài ý muốn.

Dễ tích hợp với backend: Cloudinary cung cấp thư viện cho nhiều ngôn ngữ, đặc biệt dễ tích hợp với backend FastAPI bằng Python thông qua các SDK hoặc REST API.

Trong hệ thống Vietfood Lens, Cloudinary được dùng để lưu trữ ảnh món ăn do người dùng upload và ảnh kết quả nhận diện, đảm bảo dữ liệu ảnh luôn an toàn, dễ dàng truy xuất, chia sẻ và mở rộng theo nhu cầu phát triển của ứng dụng.

1.7. Kết chương

Trong chương này, các khái niệm và công nghệ nền tảng chính của đề tài “Ứng dụng nhận diện món ăn Việt Nam và tìm kiếm quán ăn” (Vietfood Lens) đã được trình bày, gồm: trí tuệ nhân tạo (AI), học sâu với EfficientNet B4, Flutter & GetX cho phát triển di động, FastAPI & PostgreSQL cho backend, Cloudinary cho lưu trữ ảnh, cùng các dịch vụ ngoài như Google Maps API, Google Sign-In, FCM và Zoho Mail. Sự kết hợp các công nghệ này giúp đáp ứng yêu cầu nhận diện món ăn và tìm kiếm quán ăn, đồng thời tạo nền tảng mở rộng cho hệ thống trong tương lai. Những nội dung này là cơ sở lý thuyết quan trọng cho các chương phân tích, thiết kế và triển khai tiếp theo.

Chương 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

2.1. Phân tích yêu cầu

2.1.1. Giới thiệu bài toán

Trong bối cảnh hội nhập quốc tế và sự phát triển mạnh mẽ của công nghệ số, ẩm thực Việt Nam ngày càng khẳng định vị thế trên bản đồ thế giới với những món ăn đặc trưng như phở, bánh mì, bún bò Huế, bánh xèo... Tuy nhiên, sự đa dạng và phong phú về món ăn, cùng với cách chế biến và tên gọi đặc trưng vùng miền, lại là thách thức không nhỏ đối với du khách quốc tế, người mới tiếp cận ẩm thực Việt Nam, cũng như ngay cả người Việt khi di chuyển giữa các vùng miền. Họ thường gặp khó khăn trong việc nhận diện chính xác tên món ăn, tìm hiểu thành phần, nguồn gốc hay cách chế biến, đặc biệt khi chỉ có hình ảnh thực tế. Ngoài ra, việc tìm kiếm các quán ăn phù hợp với món ăn mong muốn tại địa phương cũng là vấn đề còn nhiều bất tiện.

Mặc dù hiện nay đã có các ứng dụng nhận diện hình ảnh phổ biến như Google Lens, nhưng các giải pháp này chưa tối ưu cho đặc thù ẩm thực Việt Nam; đồng thời các ứng dụng tìm kiếm quán ăn thường không tích hợp chức năng nhận diện món ăn qua ảnh, gây hạn chế trải nghiệm cho người dùng.

Trước thực trạng đó, đề tài “Ứng dụng nhận diện món ăn Việt Nam và tìm kiếm quán ăn” (Vietfood Lens) được đề xuất, với mục tiêu xây dựng một ứng dụng di động thông minh, kết hợp trí tuệ nhân tạo (AI) và bản đồ số để giúp người dùng nhận diện chính xác món ăn Việt Nam từ ảnh chụp hoặc ảnh tải lên, đồng thời gợi ý các quán ăn gần nhất phục vụ món ăn đó dựa trên vị trí thực tế. Hệ thống không chỉ hỗ trợ nhận diện món ăn mà còn cung cấp các thông tin chi tiết như nguyên liệu, nguồn gốc, cách chế biến, giá trị dinh dưỡng tham khảo; đồng thời cho phép lưu lại lịch sử nhận diện, đánh dấu món/quán ăn yêu thích, chia sẻ thông tin và tương tác xã hội, từ đó nâng cao trải nghiệm cá nhân hóa và thúc đẩy quảng bá văn hóa ẩm thực Việt Nam.

Cụ thể, hệ thống hướng tới các nhóm người dùng chính:

- Du khách quốc tế và nội địa: Những người mong muốn tìm hiểu, khám phá ẩm thực Việt Nam trong hành trình du lịch, trải nghiệm mới lạ.
- Người yêu ẩm thực, food blogger, cộng đồng mạng xã hội: Các cá nhân thường xuyên chụp ảnh, chia sẻ món ăn và cần một công cụ nhận diện nhanh, chính xác, cung cấp thông tin kịp thời.

Phạm vi của bài toán trong đề án này tập trung vào việc phát triển chức năng cốt lõi là nhận diện 40 món ăn Việt Nam cụ thể từ hình ảnh (chi tiết danh sách món ăn được trình bày tại **Phụ lục 1**). Đầu vào của hệ thống là hình ảnh món ăn do người dùng tải

lên hoặc chụp trực tiếp. Đầu ra là tên món ăn được nhận diện, kèm theo độ tin cậy của dự đoán và các thông tin liên quan đến món ăn đó. Hệ thống có thể gặp giới hạn trong việc nhận diện các món ăn bị che khuất quá nhiều, hình ảnh chất lượng quá thấp, hoặc các món ăn không nằm trong danh sách 40 món đã được huấn luyện. Trong phiên bản hiện tại, hệ thống tập trung vào nhận diện từng món ăn riêng lẻ trong một bức ảnh. Với những giải pháp này, Vietfood Lens kỳ vọng sẽ góp phần mang lại trải nghiệm tiện lợi, hiện đại cho người dùng, đồng thời nâng cao giá trị quảng bá và bảo tồn văn hóa ẩm thực Việt Nam trong thời đại số hóa.

2.1.2. Đặc tả yêu cầu chức năng

Chức năng dành cho người dùng:

Nhóm chức năng Quản lý tài khoản & xác thực:

- Đăng ký tài khoản:

Người dùng đăng ký bằng email, hệ thống gửi mã xác thực OTP qua email. Sau khi xác thực OTP thành công, người dùng tiếp tục điền các thông tin cá nhân còn lại để hoàn tất tạo tài khoản.

- Đăng nhập:

Cho phép đăng nhập bằng email và mật khẩu đã đăng ký. Hỗ trợ đăng nhập bằng tài khoản Google (Google Sign-In) và sử dụng thông tin của tài khoản Google để tạo tài khoản mới. Hỗ trợ đăng nhập bằng Google ngay cả với tài khoản đã đăng ký thủ công trước đó (sau khi liên kết).

- Liên kết/hủy liên kết tài khoản Google:

Người dùng có thể liên kết tài khoản tạo thủ công (đăng ký qua email) với tài khoản Google để tiện lợi cho việc đăng nhập về sau. Hỗ trợ hủy liên kết với Google nếu người dùng không muốn sử dụng chức năng này nữa.

- Quản lý hồ sơ cá nhân:

Cập nhật thông tin cá nhân như tên hiển thị, ảnh đại diện, đổi mật khẩu, đổi email.

- Khôi phục mật khẩu:

Cung cấp chức năng lấy lại mật khẩu qua email (gửi mã xác thực OTP cho phép đặt lại mật khẩu mới).

- Xóa tài khoản:

Cho phép người dùng yêu cầu xóa tài khoản và toàn bộ dữ liệu liên quan sẽ được xóa hoặc ẩn theo chính sách bảo vệ quyền riêng tư.

Nhóm chức năng Nhận diện món ăn:

- Tải ảnh/Chụp ảnh:

Cho phép người dùng chọn hình ảnh món ăn từ thư viện trên thiết bị hoặc sử dụng camera để chụp ảnh mới. Có thể lưu ảnh vừa chụp vào thư viện.

- Gửi yêu cầu nhận diện:

Khi người dùng xác nhận gửi ảnh, hệ thống gửi ảnh lên backend. Backend tiếp nhận và gửi ảnh đến server AI model (EfficientNet B4) qua API, nhận kết quả nhận diện, xử lý lưu lịch sử và chuyển lại cho client.

- Hiển thị kết quả nhận diện:

Sau khi nhận diện, ứng dụng hiển thị kết quả cho người dùng”

+ Nếu độ tin cậy (confidence) từ 80% trở lên:

Hiển thị label tên món ăn, độ tin cậy.

Có nút “Xem thông tin chi tiết” chuyển đến màn hình chi tiết món ăn (Tên, nguồn gốc, hình minh hoạ, nguyên liệu, cách chế biến, giá trị dinh dưỡng tham khảo).

Có nút “Tìm kiếm nhà hàng” gợi ý các quán ăn phục vụ món này gần vị trí người dùng.

+ Nếu độ tin cậy dưới 80%:

Hiển thị label “Không chắc đây là món gì...”Gợi ý món ăn có độ tin cậy cao nhất (dưới 80%).

Không có nút “Xem thông tin chi tiết”.

Vẫn có nút “Tìm kiếm nhà hàng” cho món được gợi ý (dành cho người dùng vẫn muốn thử tìm kiếm quán ăn liên quan).

Nhóm chức năng Tìm kiếm & Thông tin quán ăn, nhà hàng

- Tìm kiếm quán ăn (Google Maps API):

Dựa trên kết quả nhận diện món ăn, hệ thống sử dụng Google Maps Places API để tìm kiếm và gợi ý các quán ăn/nhà hàng gần vị trí hiện tại của người dùng có phục vụ món đó (yêu cầu quyền truy cập vị trí).

- Hiển thị thông tin quán ăn:

Lấy thông tin chi tiết từ Google Maps API (tên quán, địa chỉ, số điện thoại, trạng thái mở cửa, hình ảnh, đánh giá...).

- Chỉ đường đến quán ăn:

Tích hợp Google Maps Directions API hoặc mở ứng dụng bản đồ để hướng dẫn đường đi từ vị trí người dùng tới quán ăn được chọn.

Nhóm chức năng Tương tác & Cá nhân hoá:

- Quản lý yêu thích:

Cho phép người dùng đánh dấu "yêu thích" các món ăn và quán ăn.

- Xem danh sách yêu thích:

Hiển thị danh sách các món ăn và quán ăn đã được người dùng đánh dấu yêu thích.

- Lịch sử nhận diện:

Tự động lưu lại và cho phép người dùng xem lại lịch sử các món ăn đã được nhận diện thành công.

- Kết bạn, tương tác xã hội:

Cho phép người dùng tìm kiếm và kết bạn với các người dùng khác trong hệ thống để tăng cường tương tác cộng đồng.

- Chia sẻ món ăn/quán ăn:

Người dùng có thể chia sẻ thông tin về món ăn hoặc quán ăn với bạn bè trong hệ thống thông qua chức năng chia sẻ có ở màn hình “Chi tiết món ăn” và “Chi tiết quán ăn/nhà hàng”.

Nhóm chức năng Thông báo:

- Nhận thông báo:

Người dùng có thể nhận các thông báo từ hệ thống và các thông báo tương tác (người dùng khác gửi lời mời kết bạn, chấp nhận kết bạn hoặc chia sẻ món ăn, quán ăn/nhà hàng).

Chức năng dành cho quản trị viên:

- Thêm, sửa, xóa thông tin món ăn, quán ăn/nhà hàng.

- Quản lý tài khoản người dùng (kích hoạt, vô hiệu hóa, xóa).

- Gửi thông báo tới người dùng:

Cập nhật món ăn mới.

Cập nhật phiên bản mới.

Thông báo hệ thống.

2.1.3. Đặc tả yêu cầu phi chức năng

Hiệu năng:

Ứng dụng phản hồi nhanh, thời gian nhận diện món ăn và tìm kiếm quán ăn dưới 3 giây.

Bảo mật:

Mã hóa dữ liệu nhạy cảm, xác thực OTP/Google, bảo vệ quyền riêng tư người dùng. Hỗ trợ xác thực bằng access token/refresh token, cho phép làm mới phiên đăng nhập an toàn mà không phải nhập lại mật khẩu thường xuyên.

Khả năng mở rộng:

Có thể mở rộng số lượng món ăn, hỗ trợ nhiều người dùng mà không giảm hiệu năng.

Khả năng tích hợp:

Dễ tích hợp các dịch vụ ngoài (Google Maps, Cloudinary, FCM...).

Tương thích:

Hỗ trợ các thiết bị Android phổ biến, giao diện thích ứng nhiều kích thước màn hình.

Giao diện và trải nghiệm người dùng:

Giao diện trực quan, dễ sử dụng cho mọi lứa tuổi. Hỗ trợ đa ngôn ngữ (Tiếng Việt/ Tiếng Anh) mở rộng phục vụ người dùng quốc tế.

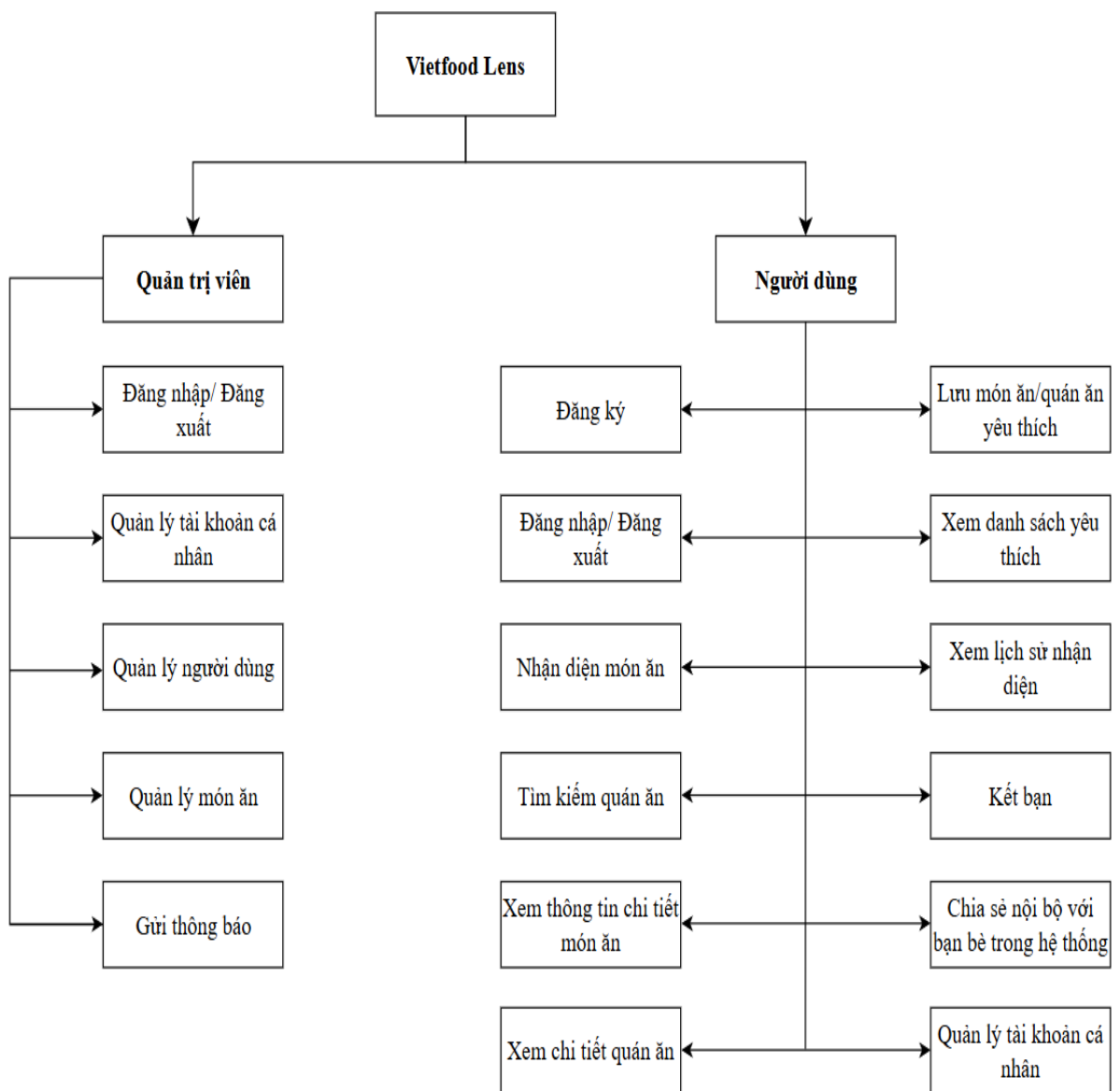
Bảo trì & phát triển:

Mã nguồn rõ ràng, có tài liệu API, dễ dàng cập nhật mô hình AI, tính năng mới.

2.2. Thiết kế hệ thống

2.2.1. Sơ đồ phân rã chức năng

Sơ đồ phân rã chức năng được mô tả như hình 2.1

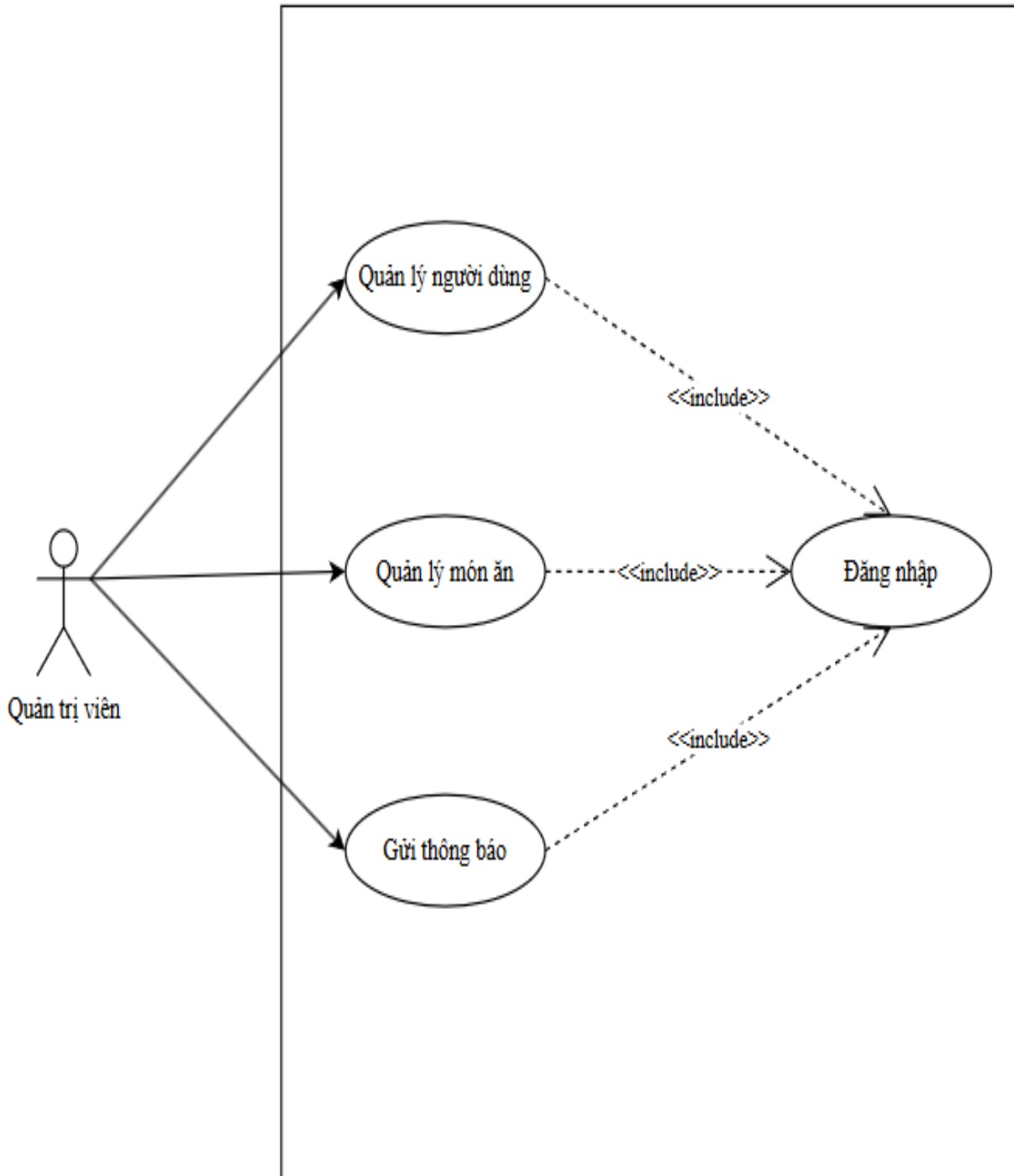


Hình 2.1: Sơ đồ phân rã chức năng

2.2.2. Biểu đồ ca sử dụng

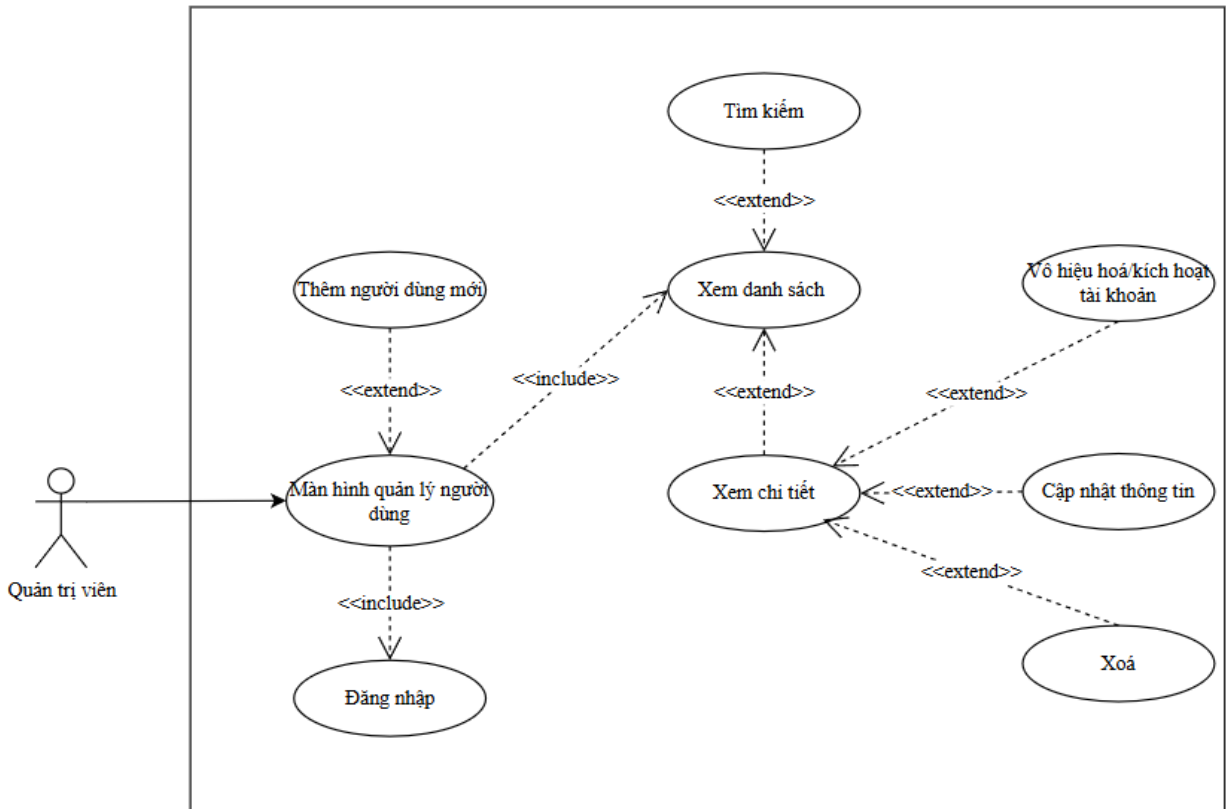
2.2.2.1. Quản trị viên

Biểu đồ ca sử dụng của quản trị viên được mô tả như hình 2.2:



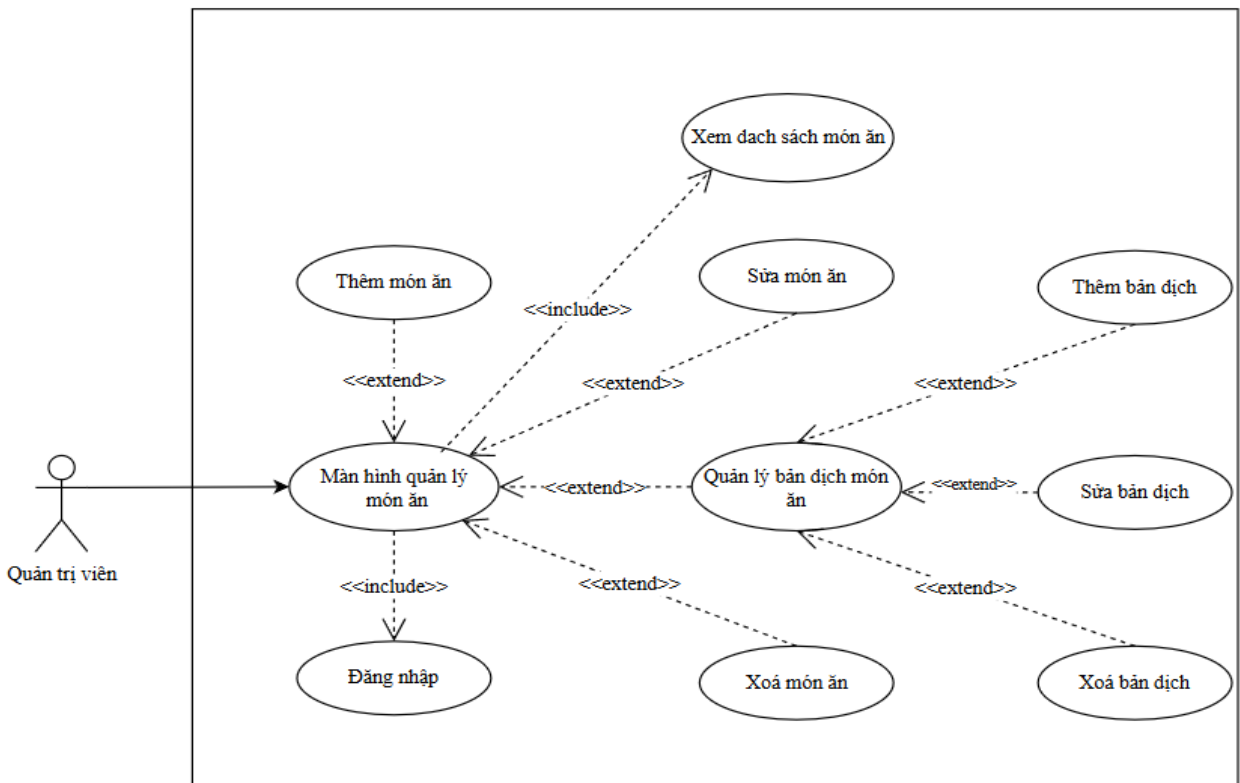
Hình 2.2: Biểu đồ ca sử dụng của quản trị viên

Biểu đồ ca sử dụng màn hình quản lý người dùng được mô tả như hình 2.3:



Hình 2.3: Biểu đồ ca sử dụng màn hình quản lý người dùng

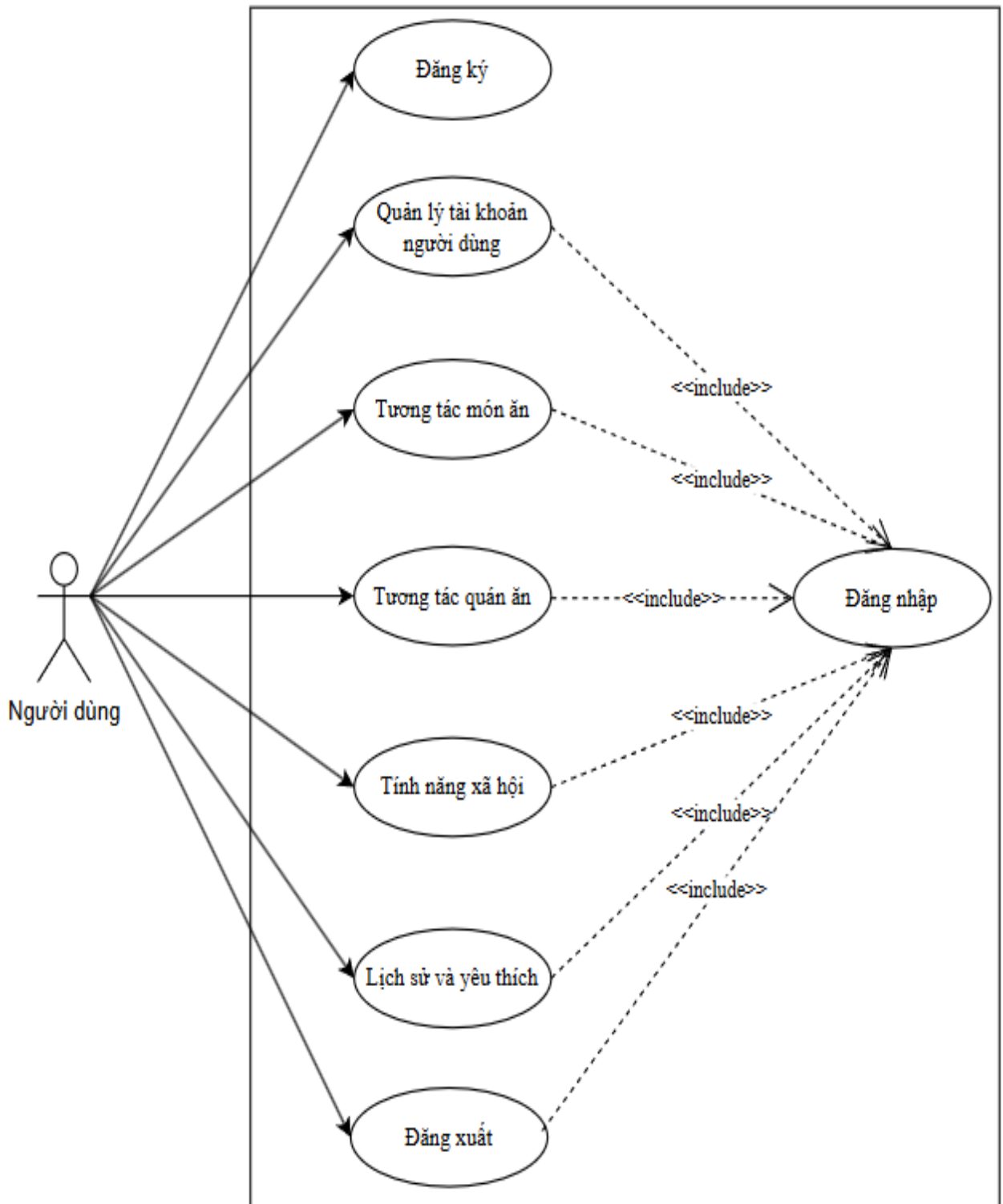
Biểu đồ ca sử dụng màn hình quản lý món ăn được mô tả như hình 2.4:



Hình 2.4: Biểu đồ ca sử dụng màn hình quản lý món ăn

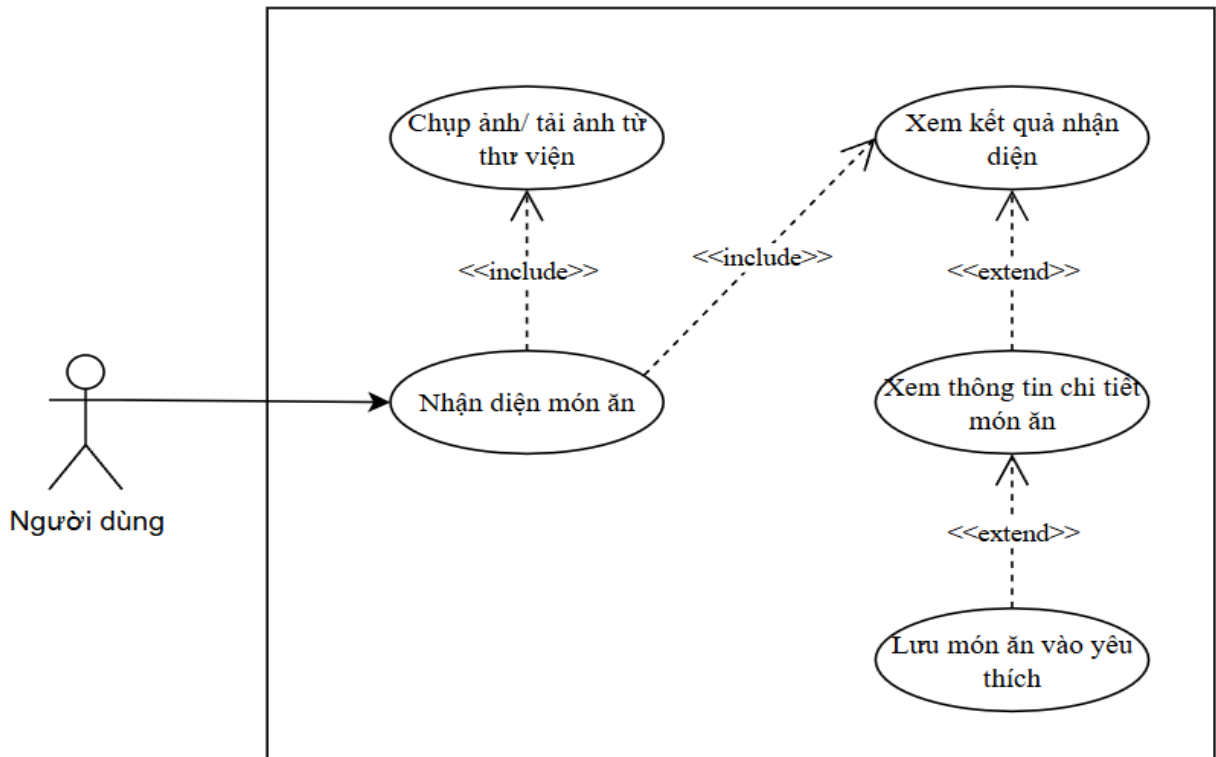
2.2.2.2. Người dùng

Biểu đồ ca sử dụng của người dùng được mô tả như hình 2.5:



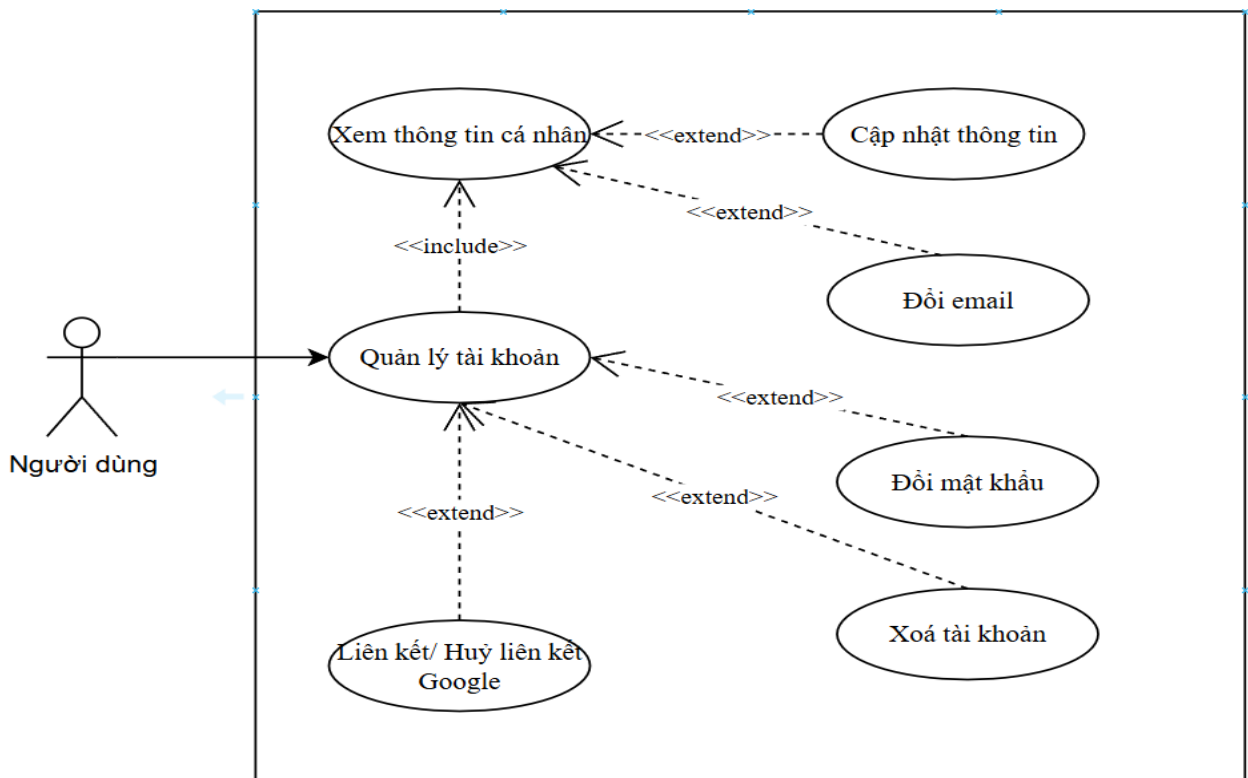
Hình 2.5: Biểu đồ ca sử dụng của người dùng

Biểu đồ ca sử dụng của chức năng tương tác món ăn được mô tả như hình 2.6:



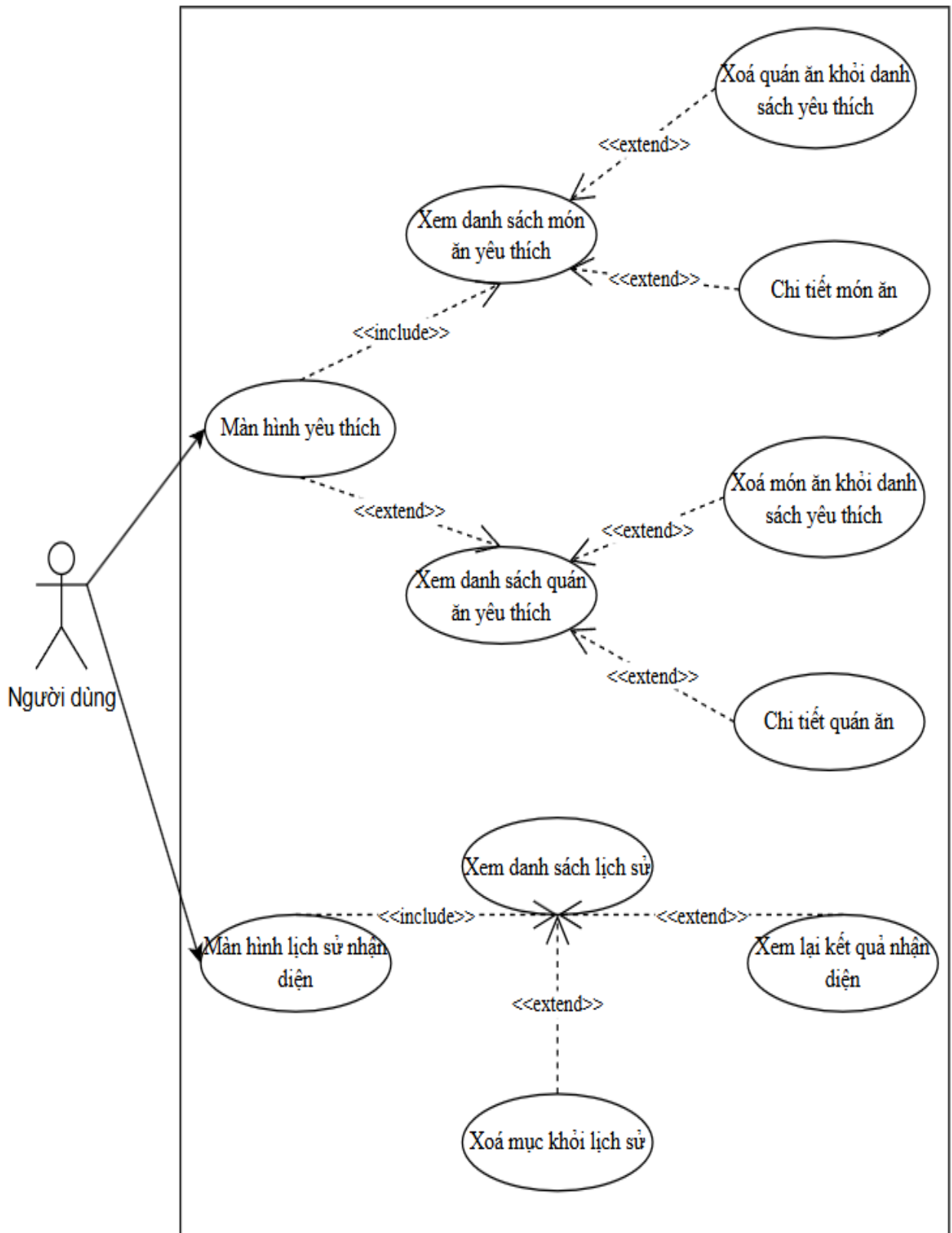
Hình 2.6: Biểu đồ ca sử dụng của chức năng tương tác món ăn

Biểu đồ ca sử dụng của chức năng quản lý tài khoản được mô tả như hình 2.7:



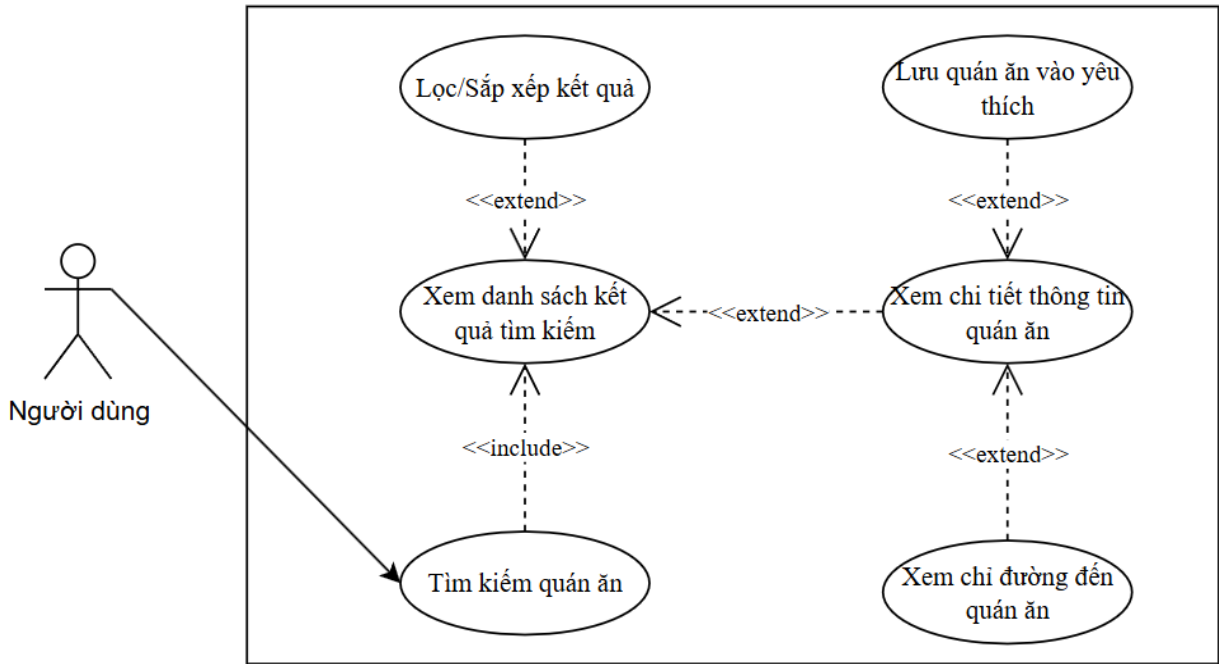
Hình 2.7: Biểu đồ ca sử dụng của chức năng quản lý tài khoản

Biểu đồ ca sử dụng của chức năng lịch sử và yêu thích được mô tả như hình 2.8:



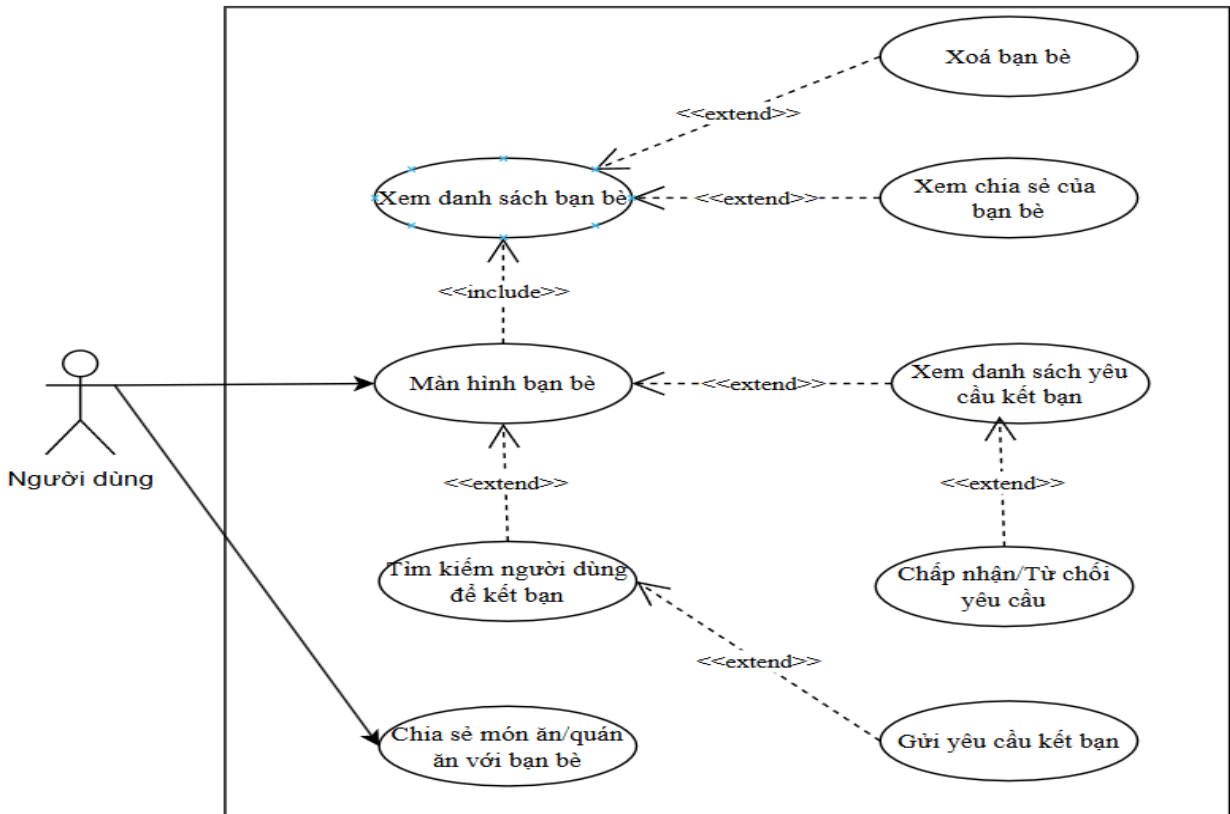
Hình 2.8: Biểu đồ ca sử dụng của chức năng Lịch sử và yêu thích

Biểu đồ ca sử dụng của chức năng tương tác quán ăn được mô tả như hình 2.9:



Hình 2.9: Biểu đồ ca sử dụng của chức năng tương tác quán ăn

Biểu đồ ca sử dụng của chức năng tính năng xã hội được mô tả như hình 2.10:

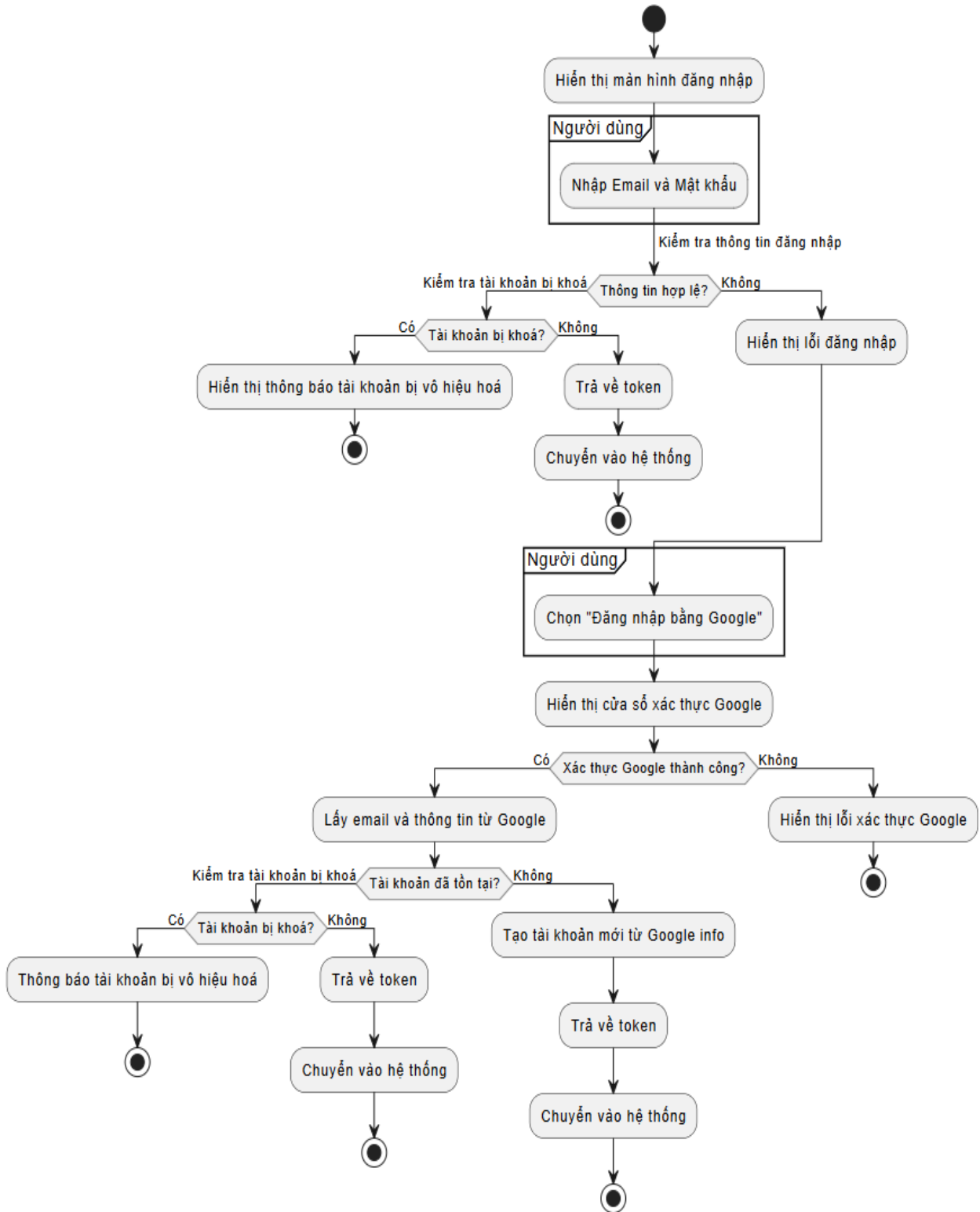


Hình 2.10: Biểu đồ ca sử dụng của tính năng xã hội

2.2.3. Biểu đồ hoạt động

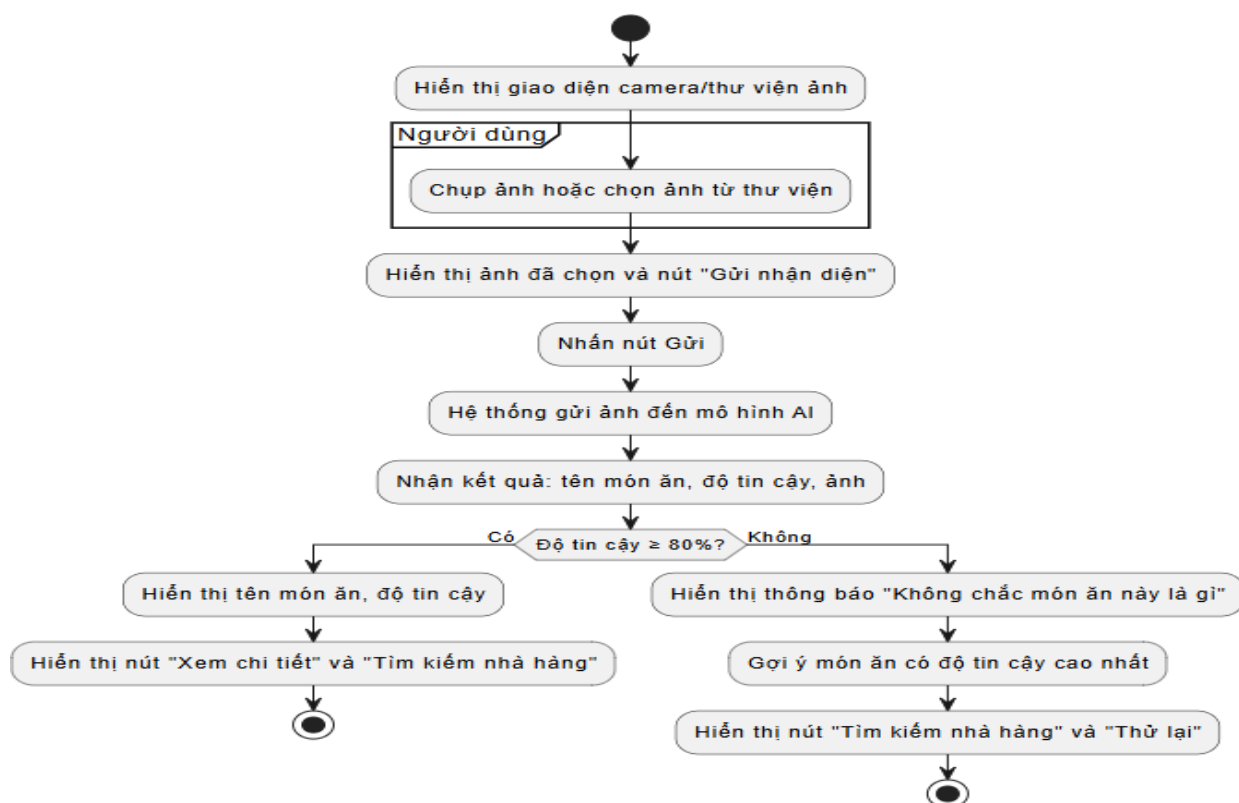
Dưới đây là một số biểu đồ giúp minh họa một số chức năng của hệ thống:

Biểu đồ hoạt động chức năng đăng nhập của hệ thống được mô tả như hình 2.11 :



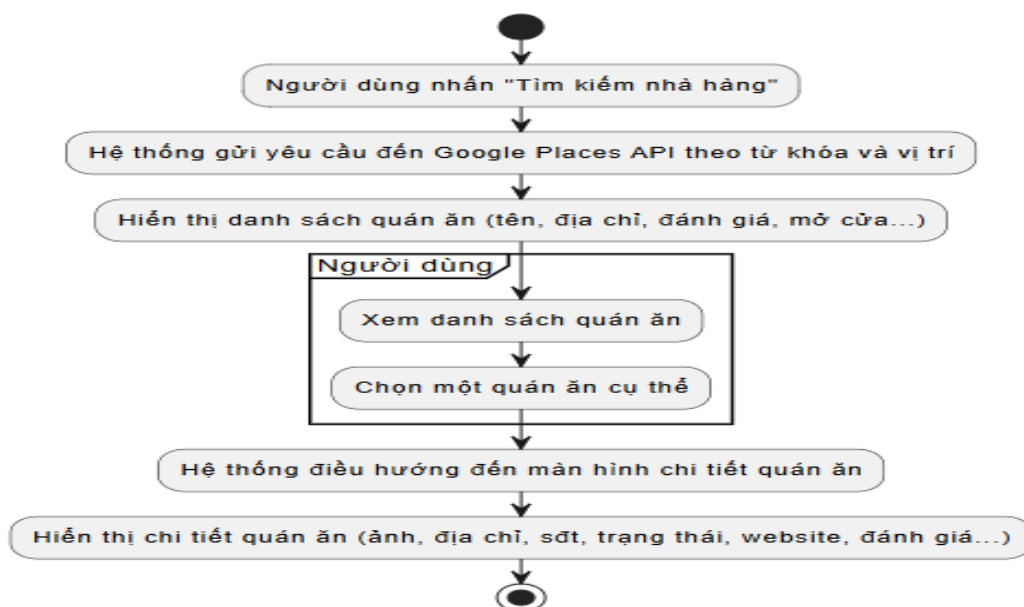
Hình 2.11: Biểu đồ hoạt động chức năng đăng nhập

Biểu đồ hoạt động mô tả chức năng nhận diện món ăn được mô tả như hình 2.12 :



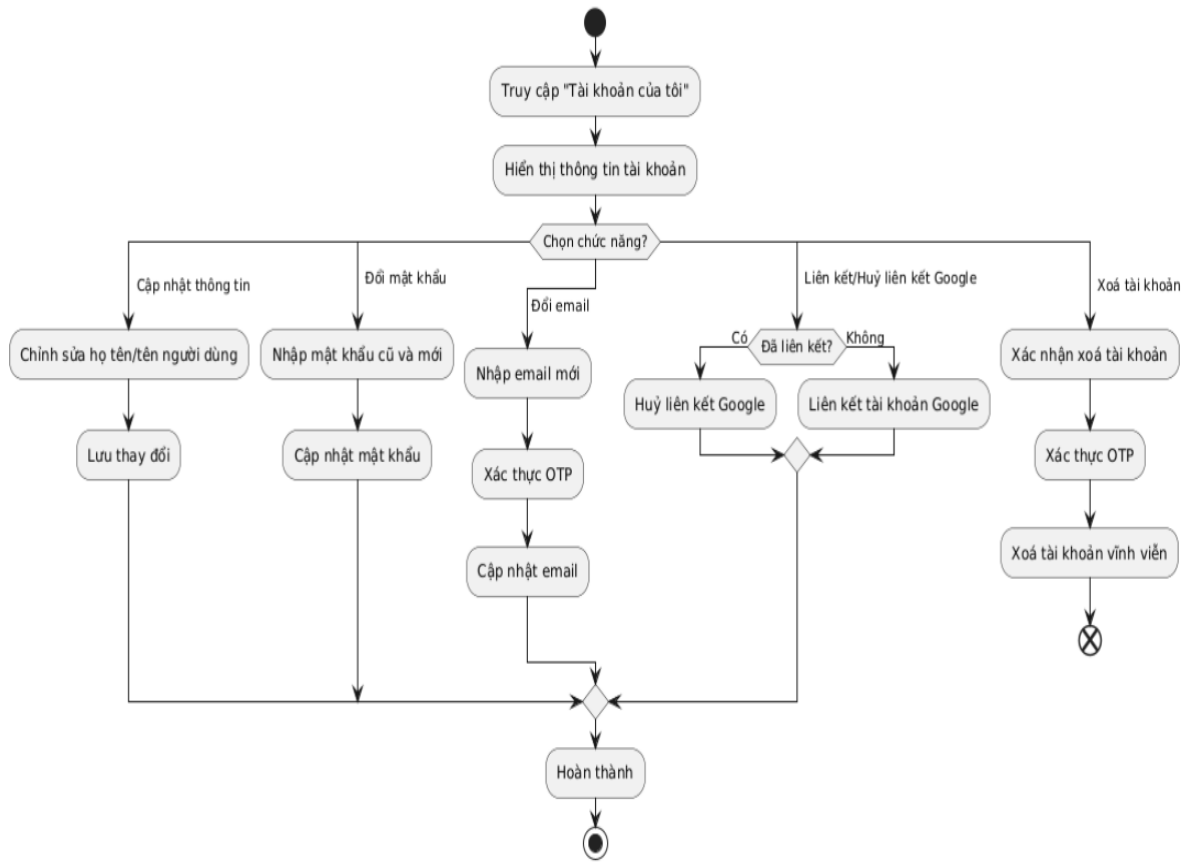
Hình 2.12: Biểu đồ hoạt động của chức năng nhận diện món ăn

Biểu đồ hoạt động mô tả chức năng tìm kiếm và xem chi tiết quán ăn hình 2.13:



Hình 2.13: Biểu đồ hoạt động chức năng tìm kiếm và xem chi tiết quán ăn

Biểu đồ hoạt động mô tả chức năng quản lý tài khoản được mô tả như hình 2.14:

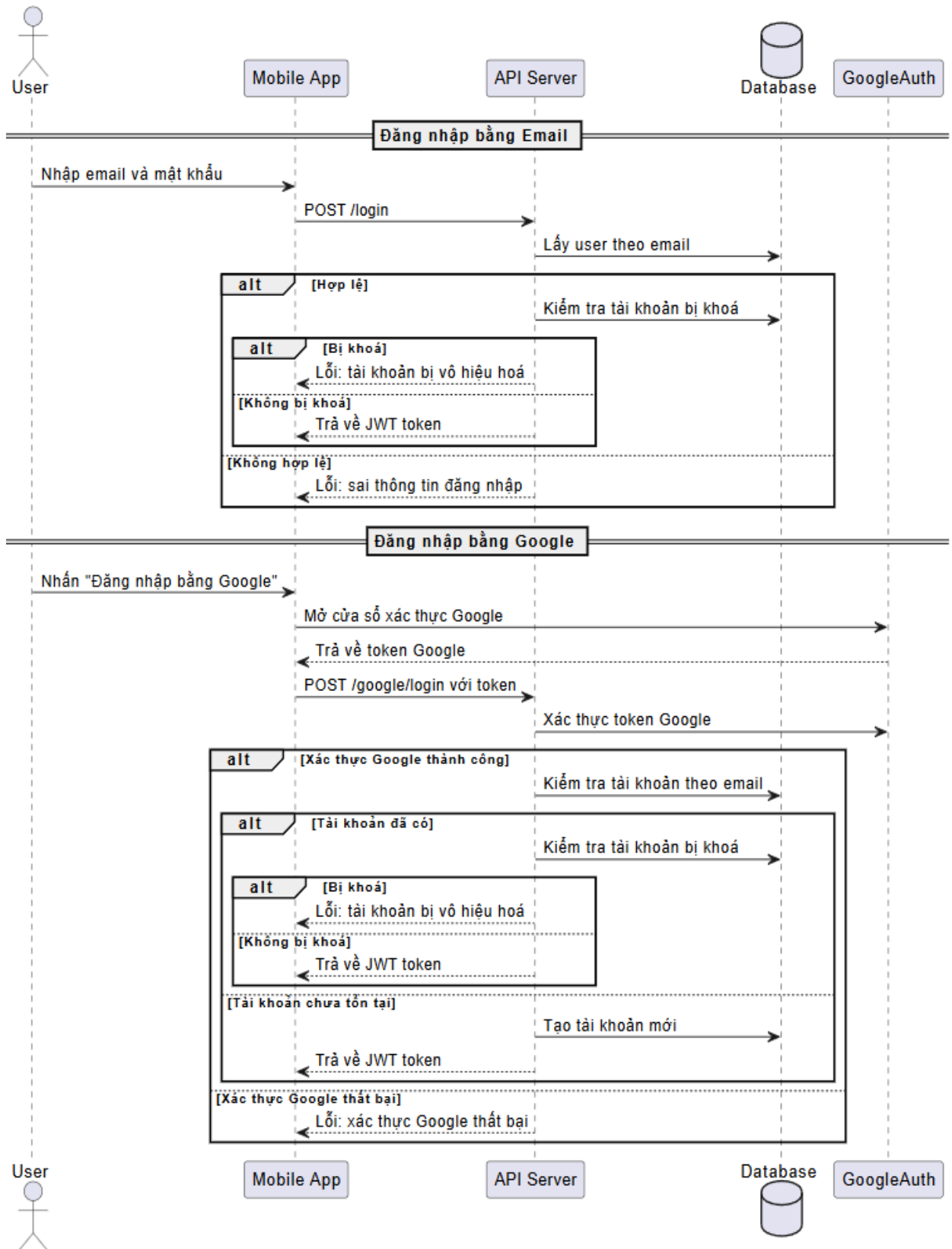


Hình 2.14: Biểu đồ hoạt động mô tả chức năng quản lý tài khoản

2.2.4. Biểu đồ tuần tự

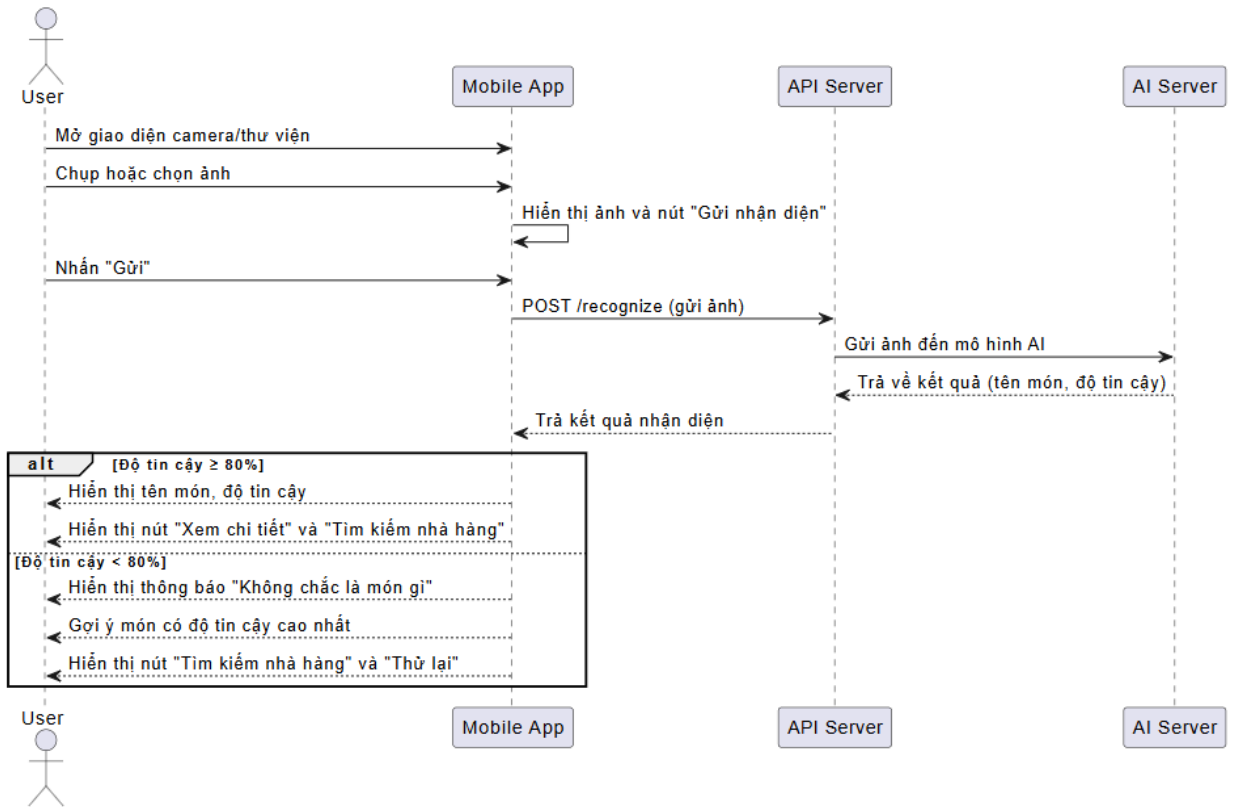
Một số biểu đồ tuần tự tổng quát sau sẽ cho thấy sự tương tác cơ bản giữa các thành phần của hệ thống rõ hơn trong một số chức năng quan trọng:

Biểu đồ tuần tự chức năng đăng nhập được mô tả như hình 2.15:



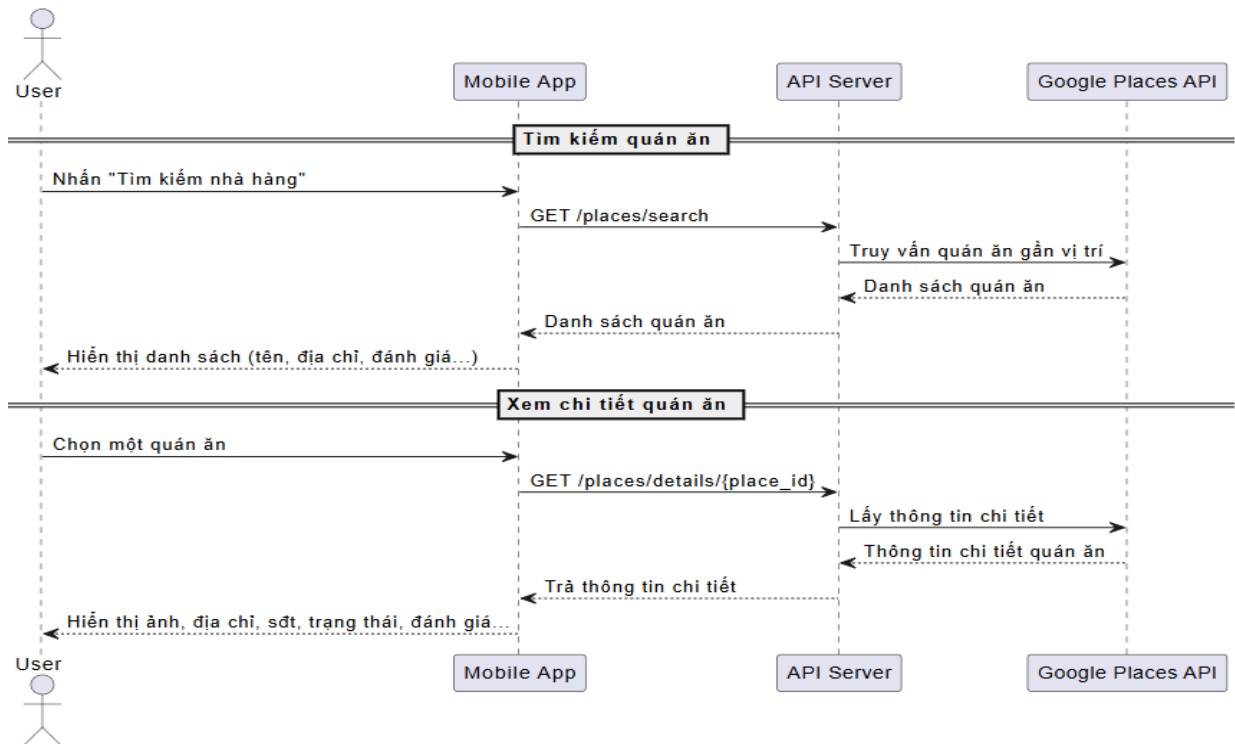
Hình 2.15: Biểu đồ tuần tự chức năng đăng nhập

Biểu đồ tuần tự chức năng nhận diện món ăn được mô tả như hình 2.16:



Hình 2.16: Biểu đồ tuần tự chức năng nhận diện món ăn

Biểu đồ tuần tự chức năng tìm kiếm và xem chi tiết quán ăn được mô tả như hình 2.17:

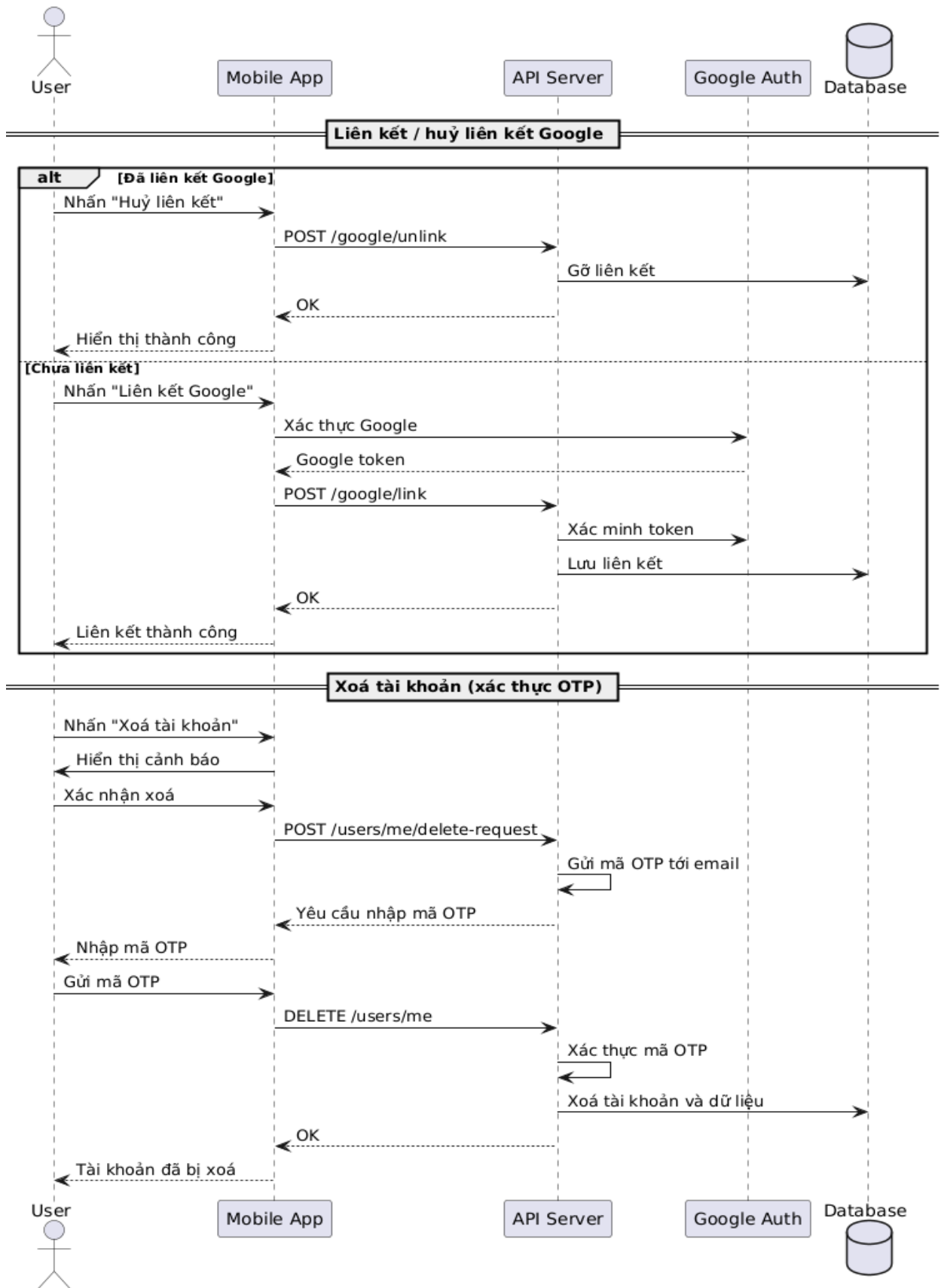


Hình 2.17: Biểu đồ tuần tự chức năng tìm kiếm và xem chi tiết quán ăn

Biểu đồ tuần tự chức năng quản lý tài khoản được mô tả như hình 2.18 và hình 2.19:



Hình 2.18: Biểu đồ tuần tự cho chức năng quản lý tài khoản gồm xem và cập nhật thông tin, đổi mật khẩu, đổi email

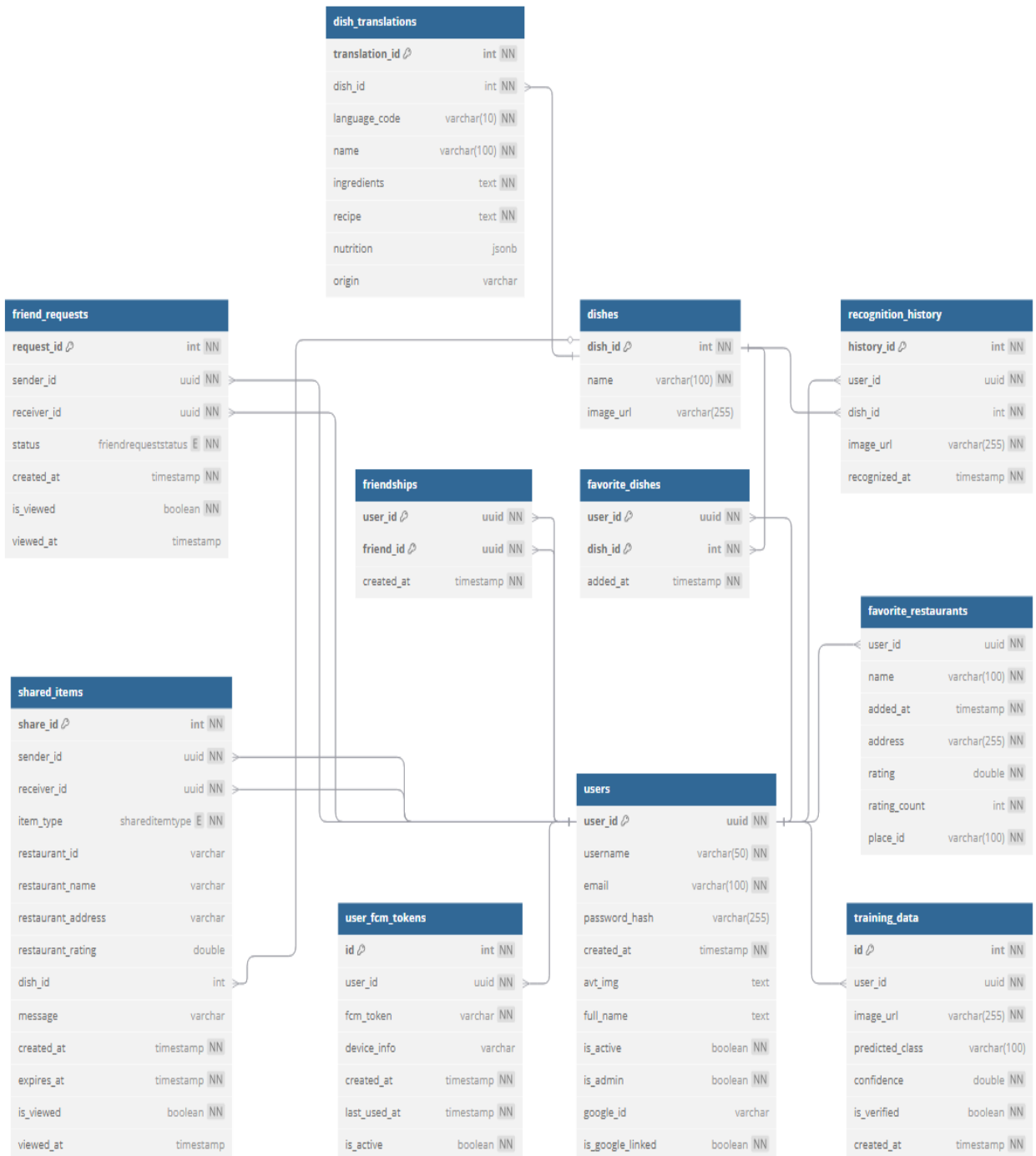


Hình 2.19: Biểu đồ tuần tự cho chức năng quản lý tài khoản gồm liên kết/hủy liên kết tài khoản Google và xoá tài khoản

2.2.5. Thiết kế cơ sở dữ liệu

2.2.5.1. Lược đồ ERD

Lược đồ ERD của hệ thống được mô tả tổng quát thông qua hình 2.19:



Hình 2.20: Lược đồ ERD

2.2.5.2. Các bảng dữ liệu

Mô tả các bảng trong cơ sở dữ liệu

Bảng 2.1: Bảng Users

Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
user_id	uuid	pk, not null	Mã định danh duy nhất của người dùng
username	varchar(50)	unique, not null	Tên người dùng
email	varchar (100)	unique, not null	Địa chỉ email người dùng
password_hash	varchar(255)	nullable	Mật khẩu đã mã hoá
created_at	timestamp	not null	Thời điểm tạo tài khoản
avt_img	text	nullable	URL ảnh đại diện
full_name	text	Nullable	Họ tên đầy đủ của người dùng
is_active	boolean	not null	Trạng thái của tài khoản
is_admin	boolean	not null	Tài khoản có quyền quản trị hay không
google_id	varchar	nullable	ID tài khoản Google (nếu có liên kết)
is_google_linked	boolean	not null, default: false	Tài khoản có liên kết với Google hay không

Bảng 2.2: Bảng Dishes

Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
dish_id	int	pk, increment, not null	Mã định danh món ăn
name	varchar(100)	unique, not null	Tên món ăn
image_url	varchar(255)	nullable	URL ảnh đại diện

Bảng 2.3: Bảng Dish_translations

Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
translation_id	int	pk, increment, not null	Mã định danh bản dịch
dish_id	int	not null	Tham chiếu tới món ăn
language_code	varchar(10)	not null	Mã ngôn ngữ (ví dụL ‘vi’, ‘en’)
name	varchar(100)	not null	Tên món ăn theo ngôn ngữ
ingredients	text	not null	Nguyên liệu
recipe	text	not null	Cách chế biến
nutrition	jsonb	nullable	Thông tin dinh dưỡng
origin	varchar	nullable	Nguồn gốc món ăn

Bảng 2.4: Bảng Favorite_dishes

Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
user_id	uuid	pk, not null	Mã định danh người dùng
dish_id	int	pk, not null	Mã món ăn yêu thích
added_at	timestamp	not null	Thời điểm thêm vào yêu thích

Bảng 2.5: Bảng Favorite_restaurants

Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
user_	uuid	not null	Mã định danh người dùng
name	varchar(100)	not null	Tên quán ăn
added_at	timestamp	not null	Thời điểm thêm vào yêu thích
address	varchar(255)	not null	Địa chỉ quán ăn
rating	double	not null	đánh giá trung bình
rating_count	int	not null	Số lượng đánh giá
place_id	varchar(100)	not null	ID địa điểm từ Google Places

Bảng 2.6: Bảng Friend_requests

Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
request_id	int	pk, increment, not null	Mã yêu cầu kết bạn
sender_id	uuid	not null	Người gửi lời mời
receiver_id	uuid	not null	Người nhận lời mời
status	enum (friendrequeststatus)	not null	Trạng thái yêu cầu (pending/accepted/rejected)
created_at	timestamp	not null	Thời điểm gửi yêu cầu
is_viewed	boolean	not null	Đã xem hay chưa
viewed_at	timestamp	nullable	Thời điểm đã xem

Bảng 2.7: Bảng Friendships

Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
user_id	uuid	pk, not null	Mã người dùng
friend_id	uuid	pk, not null	Mã người dùng bạn bè
created_at	timestamp	not null	Thời điểm trở thành bạn bè

Bảng 2.8: Bảng Recognition_history

Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
history_id	int	pk, increment, not null	Mã lịch sử nhận diện
user_id	uuid	not null	Mã người dùng
dish_id	int	not null	Mã món ăn đã nhận diện
image_url	varchar(255)	not null	URL ảnh được nhận diện
recognized_at	timestamp	not null	Thời điểm nhận diện

Bảng 2.9: Bảng Shared_items

Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
share_id	int	pk, increment, not null	Mã chia sẻ
sender_id	uuid	not null	Người gửi chia sẻ
receiver_id	uuid	not null	Người nhận chia sẻ
item_type	enum (shareditemtype)	not null	Loại mục được chia sẻ (dish/restaurant)
restaurant_id	varchar	nullable	ID quán ăn
restaurant_name	varchar	nullable	Tên quán ăn
restaurant_address	varchar	nullable	Địa chỉ quán ăn
restaurant_rating	double	nullable	Đánh giá quán ăn
dish_id	int	nullable	Mã món ăn
message	varchar	nullable	Tin nhắn đính kèm
created_at	timestamp	not null	Thời điểm chia sẻ
expires_at	timestamp	not null	Thời điểm hết hạn chia sẻ
is_viewed	boolean	not null	Đã xem hay chưa
viewed_at	timestamp	nullable	Thời điểm đã xem

Bảng 2.10: Bảng Training_data

Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
id	int	pk, increment, not null	Mã dòng dữ liệu
user_id	uuid	not null	Người đóng góp dữ liệu
image_url	varchar(100)	nullable	URL ảnh món ăn
predicted_class	varchar(100)	nullable	Kết quả dự đoán của mô hình
confidence	double	not null	Độ tin cậy của kết quả
is_verified	boolean	not null	Gán true nếu độ tin cậy ≥ 0.8 , false nếu < 0.8
created_at	timestamp	not null	Thời điểm gửi dữ liệu

Bảng 2.11: Bảng User_fcm_tokens

Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
id	int	pk, increment, not null	Mã định danh token FCM
user_id	uuid	not_null	Mã người dùng sở hữu token
fcm_token	varchar	not null	Mã token Firebase Cloud Messaging
device_info	varchar	nullable	Thông tin thiết bị
created_at	timestamp	not null	Thời điểm tạo token
last_used_at	timestamp	not null	Thời điểm sử dụng cuối cùng
is_active	boolean	not null	Token còn hoạt động hay không

2.2.6. Xây dựng mô hình nhận diện món ăn

Phần này trình bày quá trình xây dựng mô hình nhận diện món ăn cho hệ thống, nhằm nhận diện chính xác 40 món ăn Việt Nam phổ biến (phở, bánh mì, bún bò Huế, v.v.). Mô hình sử dụng kiến trúc EfficientNet B4, được huấn luyện trên bộ dữ liệu ảnh món ăn Việt Nam, với các bước tiền xử lý dữ liệu, huấn luyện và đánh giá hiệu suất.

2.2.6.1. Tiền xử lý dữ liệu

Thu thập và tổ chức dữ liệu:

- Nguồn dữ liệu: Tập dữ liệu hình ảnh các món ăn Việt Nam được sử dụng trong đồ án này được tổng hợp từ 2 nguồn chính:

Bộ dữ liệu công khai trên Kaggle: Một phần lớn dữ liệu (30 lớp món ăn) được lấy từ bộ dữ liệu "Vietnamese Foods" có sẵn trên nền tảng Kaggle, được đóng góp bởi tác giả Quang Dang (liên kết: <https://www.kaggle.com/datasets/quandang/vietnamese-foods/data>)

Dữ liệu tự thu thập: Để mở rộng phạm vi và đặc thù của bộ dữ liệu, 10 lớp món ăn khác đã được tự thu thập thông qua kỹ thuật crawling dữ liệu từ các nguồn trực tuyến (Google, Bing).

- Tổ chức dữ liệu: Toàn bộ dữ liệu sau khi thu thập và tổng hợp được tổ chức thành cấu trúc thư mục tiêu chuẩn phục vụ cho việc huấn luyện mô hình học sâu. Dữ liệu được chia thành ba tập riêng biệt: Tập Huấn luyện (Train), Tập Kiểm định (Validate) và Tập

Kiểm thử (Test). Mỗi tập này lại chứa các thư mục con, mỗi thư mục con đặt tên theo một lớp món ăn cụ thể và chứa tất cả hình ảnh thuộc về lớp đó.

Tổng số lớp: Hệ thống được thiết kế để nhận diện tổng cộng 40 lớp món ăn của Việt Nam.

Làm sạch dữ liệu:

- Loại bỏ ảnh hỏng hoặc không thể đọc được (dùng OpenCV).
- Loại bỏ ảnh quá nhỏ (kích thước cạnh nhỏ hơn 150 pixel).
- Loại bỏ ảnh trùng lặp thông qua kiểm tra mã băm MD5 (dùng hashlib)
- Thống kê dữ liệu: Sau khi làm sạch, bộ dữ liệu 30,848 ảnh phân chia như sau:
 - Tập Train: 2,527 ảnh (chiếm 73%), với số lượng ảnh mỗi món dao động từ 310 (Bánh pía) đến 1,065 (Bún bò Huế).
 - Tập Validate: 2,906 ảnh (chiếm 9.4%), từ 24 (Chè đậu đen) đến 153 (Bún bò Huế).
 - Tập Test: 5,415 ảnh (chiếm 17.6%), từ 26 (Chè đậu đen) đến 305 (Bún bò Huế).

Cân bằng dữ liệu:

Do số lượng ảnh giữa các món ăn không đồng đều, WeightedRandomSampler được sử dụng để cân bằng dữ liệu trong tập Train. Trọng số (weights) được tính dựa trên số lượng ảnh của từng món, đảm bảo mô hình không bị thiên lệch về các lớp có nhiều dữ liệu hơn.

Biến đổi dữ liệu: Tăng cường dữ liệu (Data augmentation) được áp dụng để cải thiện khả năng tổng quát hoá của mô hình, sử dụng thư viện Albumentations.

• Tập train:

- RandomResizedCrop: Cắt ngẫu nhiên với tỷ lệ (0.8, 1.0) và kích thước 380x380.
- HorizontalFlip: Lật ngang ảnh với xác suất 0.5.
- ColorJitter: Điều chỉnh màu sắc (độ sáng, độ tương phản, v.v.) với xác suất 0.8.
- Xoay ảnh ngẫu nhiên trong khoảng ± 15 độ.
- CoarseDropout: Che ngẫu nhiên 1-4 vùng ảnh, tăng tính mạnh mẽ của mô hình.
- Normalize: Chuẩn hóa ảnh với mean (0.485, 0.456, 0.406) và std (0.229, 0.224, 0.225).

• Tập Validate và Test:

- LongestMaxSize: Điều chỉnh kích thước cạnh dài nhất thành 380 pixel.
- PadIfNeeded: Đệm ảnh để đạt kích thước 380x380.
- Normalize: Chuẩn hóa ảnh với mean (0.485, 0.456, 0.406) và std (0.229, 0.224, 0.225).

2.2.6.2. Xây dựng và huấn luyện mô hình

Tải dữ liệu:

Dữ liệu được tải bằng DataLoader của Pytorch, với các biến đổi Albumentations đã áp dụng:

- Batch size: 32
- Số worker: 2 , để tăng tốc độ tải dữ liệu.
- WeightedRandomSample: Áp dụng cho tập Train để cân bằng dữ liệu.

Khởi tạo mô hình:

Mô hình EfficientNet B4 được sử dụng với trọng số pretrained từ ImageNet

- Lớp đầu ra (classifier) được thay đổi để phù hợp với 40 món ăn Việt Nam

Huấn luyện mô hình:

Quá trình huấn luyện được thực hiện trong 30 epoch với các tham số:

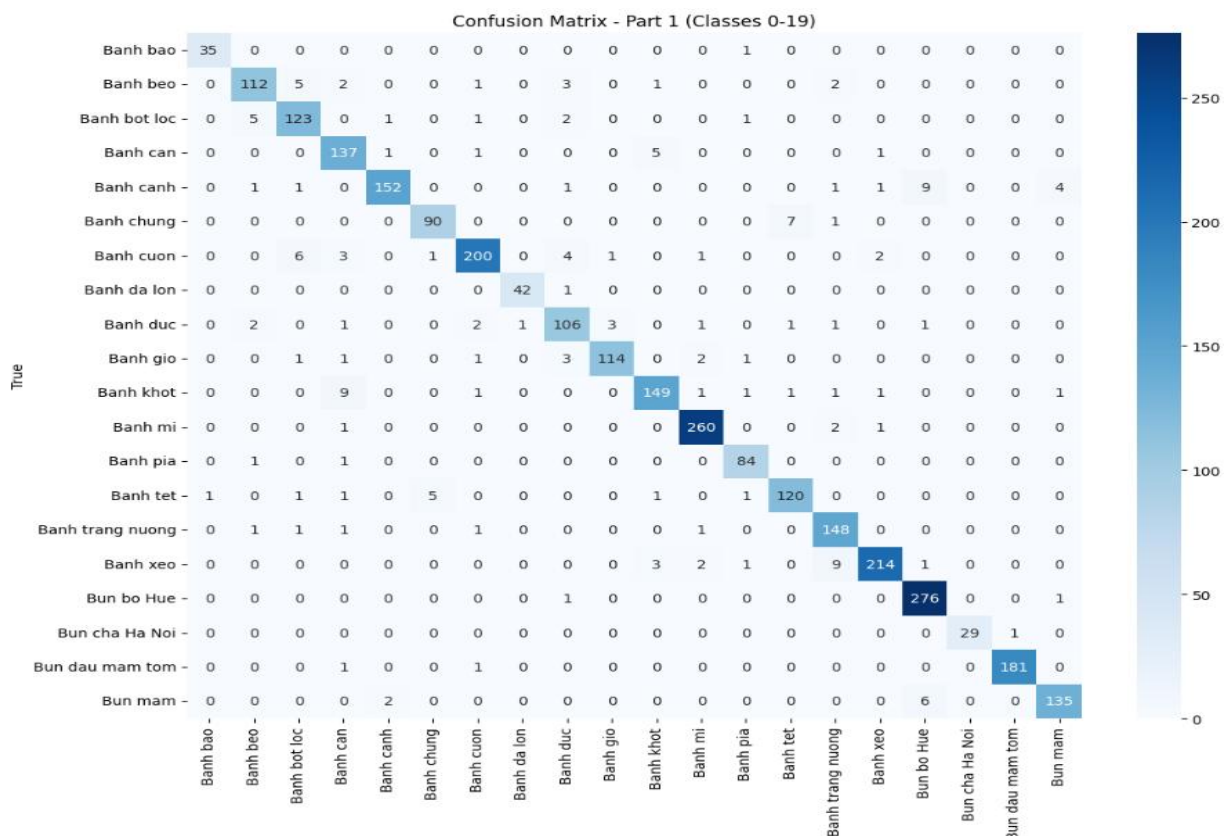
- Hàm mất mát: CrossEntropyLoss để phân loại đa lớp.
- Bộ tối ưu hóa: AdamW với learning rate ban đầu $3e-4$ và weight decay $1e-4$.
- Lập lịch learning rate: ReduceLROnPlateau, giảm learning rate khi độ chính xác trên tập Validate không cải thiện sau 3 epoch.
- Mixed-precision training: Sử dụng GradScaler và autocast để tăng tốc độ huấn luyện và tiết kiệm bộ nhớ GPU.
- Early stopping: Dừng huấn luyện nếu độ chính xác không cải thiện sau 8 epoch, lưu mô hình tốt nhất (best_b4.pth).
- Quá trình huấn luyện bao gồm hai pha: Pha huấn luyện (Training) tính toán loss, cập nhật trọng số mô hình bằng backpropagation; Pha đánh giá (Validation) tính loss và accuracy trên tập Validate, điều chỉnh learning rate và lưu mô hình tốt nhất.

2.2.6.3. Đánh giá mô hình

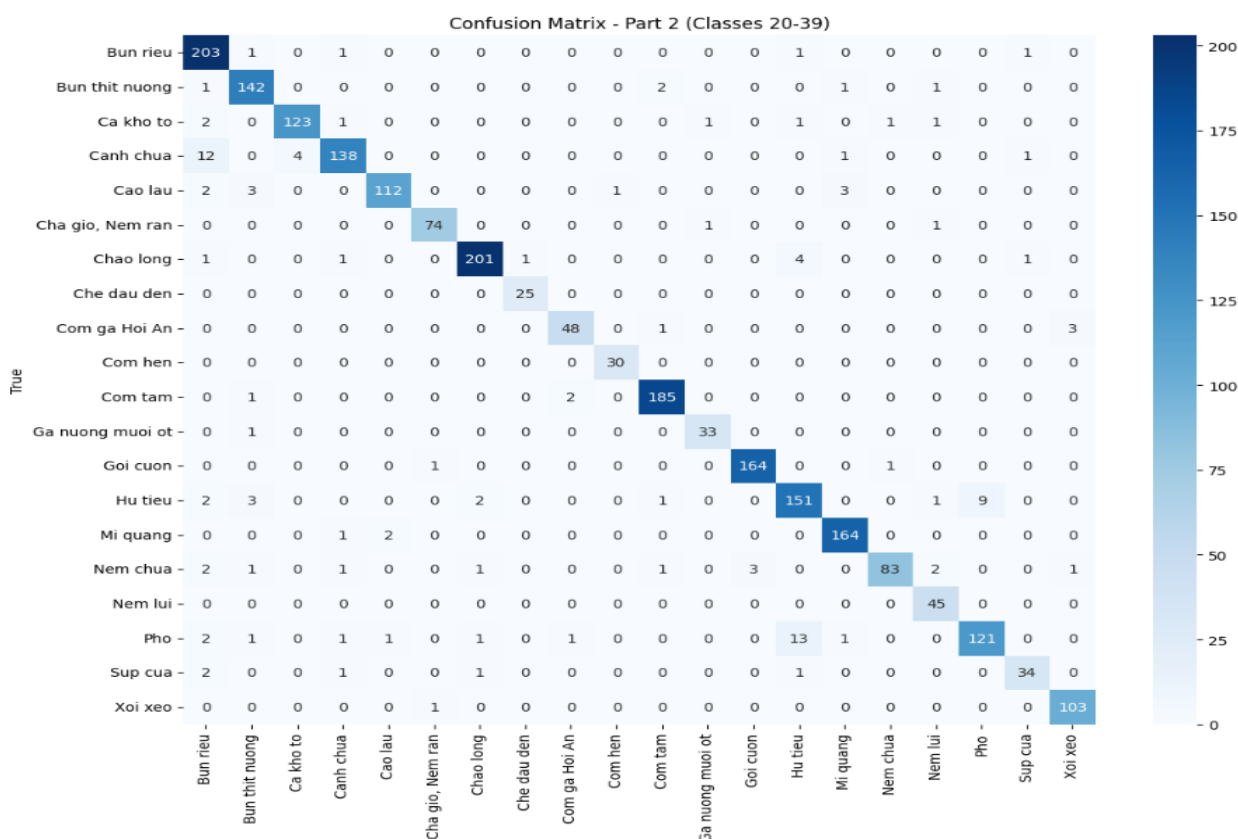
Tính toán Loss và Accuracy:

Sử dụng hàm mất mát CrossEntropyLoss để tính Test Loss và so sánh dự đoán với nhãn thực tế để tính Test Accuracy. Mô hình được đặt ở chế độ đánh giá (eval), tắt tính toán gradient để tối ưu hiệu suất. Dự đoán được thực hiện trên từng batch ảnh trong tập Test, với kết quả tổng hợp để đánh giá hiệu suất chung của mô hình.

Confusion Matrix:



Hình 2.21: Ma trận nhầm lẫn mô hình EfficientNet B4 – các lớp từ 0-19



Hình 2.22: Ma trận nhầm lẫn mô hình EfficientNet B4 – các lớp từ 20-39

Classification Report:

Các chỉ số Precision, Recall và F1-score được tính cho từng món ăn, sử dụng thư viện Scikit-learn. Báo cáo này đảm bảo mô hình không bị thiên lệch giữa các lớp, đặc biệt với các món ăn có số lượng ảnh ít.

	precision	recall	f1-score	support
Banh bao	0.97	0.97	0.97	36
Banh beo	0.92	0.88	0.90	128
Banh bot loc	0.85	0.87	0.86	142
Banh can	0.85	0.93	0.89	147
Banh canh	0.87	0.80	0.83	191
Banh chung	0.93	0.92	0.92	98
Banh cuon	0.93	0.88	0.91	226
Banh da lon	0.95	0.98	0.97	43
Banh duc	0.84	0.83	0.83	128
Banh gio	0.97	0.89	0.93	128
Banh khot	0.92	0.90	0.91	165
Banh mi	0.96	0.97	0.97	268
Banh pia	0.93	0.98	0.95	86
Banh tet	0.90	0.88	0.89	137
Banh trang nuong	0.88	0.93	0.90	159
Banh xeo	0.97	0.91	0.94	235
Bun bo Hue	0.80	0.90	0.85	305
Bun cha Ha Noi	0.94	0.83	0.88	35
Bun dau mam tom	0.99	0.98	0.99	184
Bun mam	0.90	0.87	0.89	155
Bun rieu	0.83	0.89	0.86	228
Bun thit nuong	0.89	0.95	0.92	150
Ca kho to	0.92	0.92	0.92	134
Canh chua	0.94	0.84	0.89	164
Cao lau	0.97	0.90	0.93	124
Cha gio, Nem ran	0.89	0.95	0.92	78
Chao long	0.96	0.93	0.95	215
Che dau đen	0.96	0.96	0.96	26
Com ga Hoi An	0.94	0.91	0.92	53
Com hen	0.94	1.00	0.97	30
Com tam	0.97	0.98	0.97	189
Ga nuong muoi ot	0.89	0.97	0.93	34
Goi cuon	0.93	0.96	0.95	170
Hu tieu	0.80	0.78	0.79	194
Mi quang	0.93	0.96	0.95	171
Nem chua	0.88	0.77	0.82	108
Nem lui	0.80	1.00	0.89	45
Pho	0.81	0.75	0.78	162
Sup cua	0.89	0.87	0.88	39
Xoi xeo	0.89	0.98	0.93	105
accuracy			0.90	5415
macro avg	0.91	0.91	0.91	5415
weighted avg	0.90	0.90	0.90	5415

Hình 2.23: Báo cáo các chỉ số precision, recall, f1-score theo từng lớp món ăn.

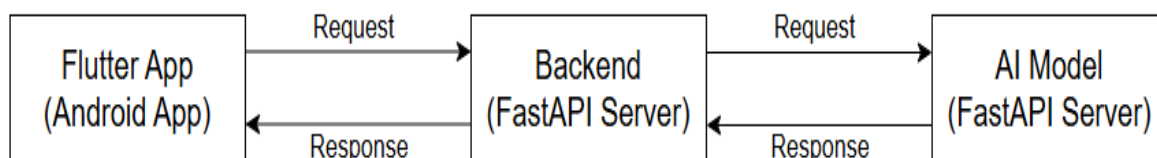
Nhận xét:

- Mô hình đạt accuracy 90% trên 5.415 mẫu test với 40 lớp, cho thấy hiệu suất phân loại tốt.
- Hiệu suất giữa các lớp khá đồng đều, với *Bún bò Huế* đạt F1-score cao nhất (0.99) và *Xôi xéo* thấp nhất (0.78). Tuy vậy, một số lớp có ít mẫu test (như Chè đậu đen, Nem lụi, Cơm hến) có thể khiến đánh giá ở các lớp này kém tin cậy hơn.
- Confusion matrix cho thấy một số cặp lớp thường bị nhầm lẫn, điển hình như: **Bánh căn** ↔ **Bánh khọt** (14 trường hợp); **Canh chua** ↔ **Bún riêu** (13 trường hợp) vì đều là món nước dùng đỏ với rau thơm; **Phở** ↔ **Bún bò Huế** (29 trường hợp) và **Phở** ↔ **Hủ tiếu** (22 trường hợp).
- Mô hình thể hiện hiệu suất ổn định với weighted average precision và recall đều đạt 0.90, không có dấu hiệu thiên lệch nghiêm trọng ở bất kỳ lớp nào.
- Tổng thể, mô hình EfficientNet B4 cho thấy khả năng học đặc trưng hình ảnh hiệu quả, với độ chính xác và khả năng phân loại ổn định trên 40 lớp món ăn Việt Nam.

2.2.7. Kiến trúc hệ thống

Hệ thống được xây dựng theo mô hình Client – Server, trong đó toàn bộ các thành phần giao tiếp với nhau thông qua giao thức REST API. Hệ thống bao gồm ba thành phần chính:

- Ứng dụng di động (Client): Được phát triển bằng Flutter, chạy trên nền tảng Android. Đây là nơi người dùng thực hiện các thao tác như đăng nhập, nhận diện món ăn, tìm kiếm nhà hàng, xem lịch sử, kết bạn,...
- Máy chủ backend (Server): Được xây dựng bằng FastAPI, đóng vai trò trung tâm xử lý nghiệp vụ, giao tiếp với cơ sở dữ liệu, mô hình AI và các API bên ngoài.
- Máy chủ mô hình AI: Được triển khai bằng FastAPI, chứa mô hình EfficientNet B4 dùng để nhận diện món ăn từ ảnh. Mô hình này được triển khai trên máy chủ độc lập và giao tiếp với backend thông qua API.



Hình 2.24: Sơ đồ kiến trúc hệ thống theo mô hình Client – Server

Ưu điểm của kiến trúc:

- Tách biệt rõ ràng các vai trò: Giao diện người dùng, xử lý nghiệp vụ và xử lý AI độc lập.
- Dễ mở rộng và bảo trì: Có thể cập nhật mô hình AI hoặc backend mà không ảnh hưởng tới client.
- Tối ưu triển khai: Mô hình AI có thể chạy trên máy chủ cấu hình cao trong khi backend nhẹ và linh hoạt.
- Bảo mật: Client không gọi trực tiếp AI hay dịch vụ ngoài, mọi giao tiếp đi qua backend.

Chương 3: TRIỂN KHAI VÀ KẾT QUẢ ĐẠT ĐƯỢC

3.1. Môi trường phát triển

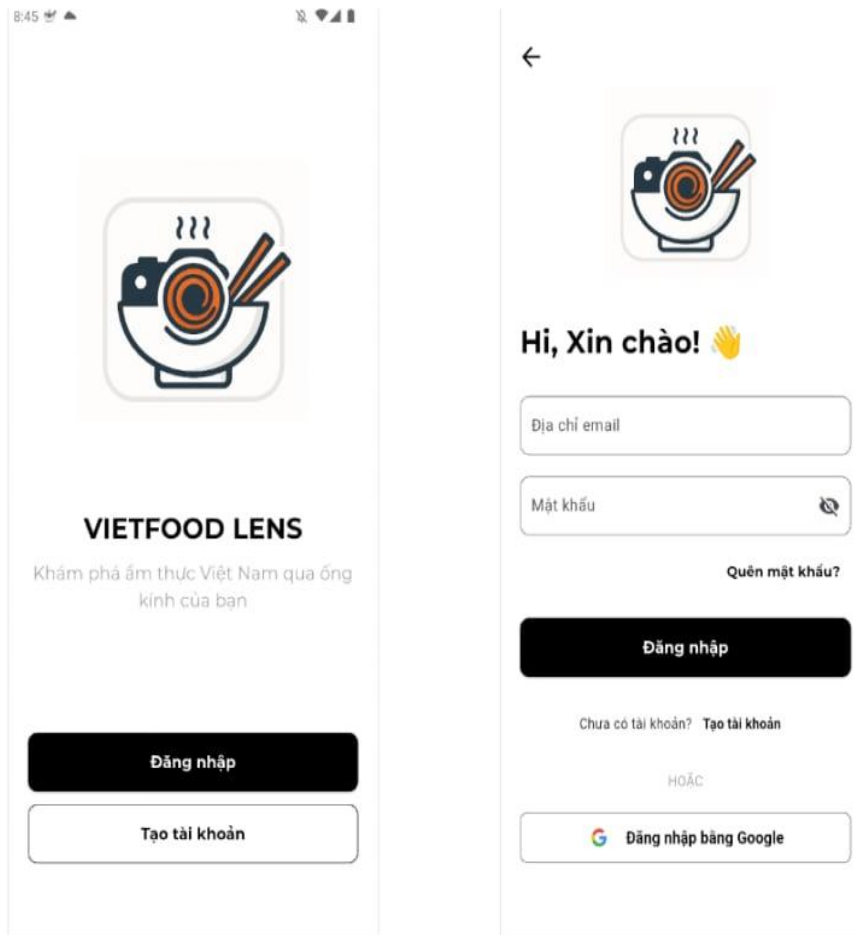
Hệ thống được triển khai với các công nghệ và môi trường phát triển cụ thể, được trình bày trong Bảng 3.1 dưới đây.

Bảng 3.1: Môi trường và công nghệ triển khai hệ thống

Thành phần	Công nghệ sử dụng
Ngôn ngữ backend	Python
Framework backend	FastAPI
Fronten	Flutter
Hệ quản trị CSDL	PostgreSQL
Mô hình AI	EfficientNet B4 (Pytorch)
Công cụ huấn luyện	Kaggle
Triển khai backend	App Platform - DigitalOcean (1GB RAM, 1 vCPU)
Triển khai model	Cloud VPS (Droplet - 2GB RAM, 2 vCPUs) – DigitalOcean
Triển khai CSDL	Aiven Cloud PostgreSQL – Dịch vụ DBaaS (<i>Database as a Service</i>)
Triển khai ứng dụng Android	Google Play Store

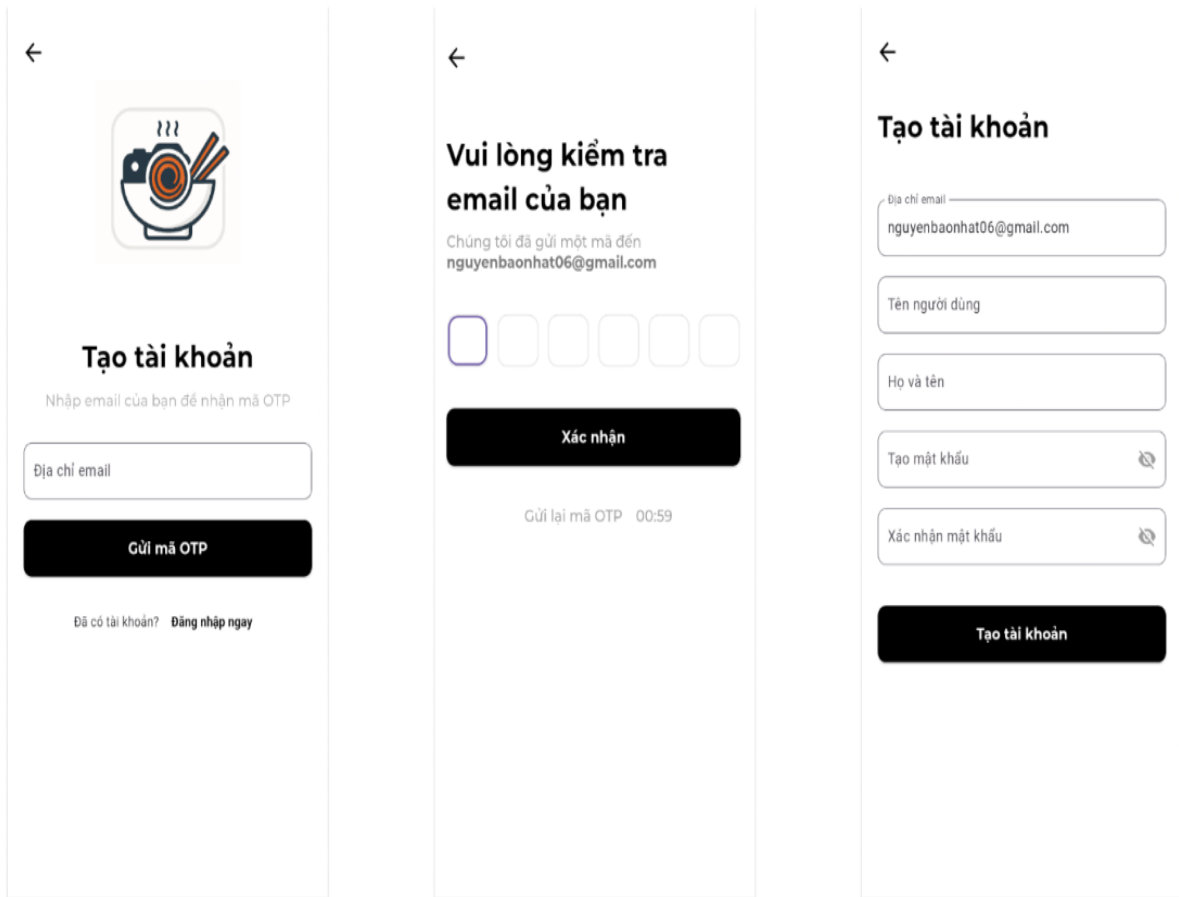
3.2. Kết quả đạt được

3.2.1. Giao diện Người dùng



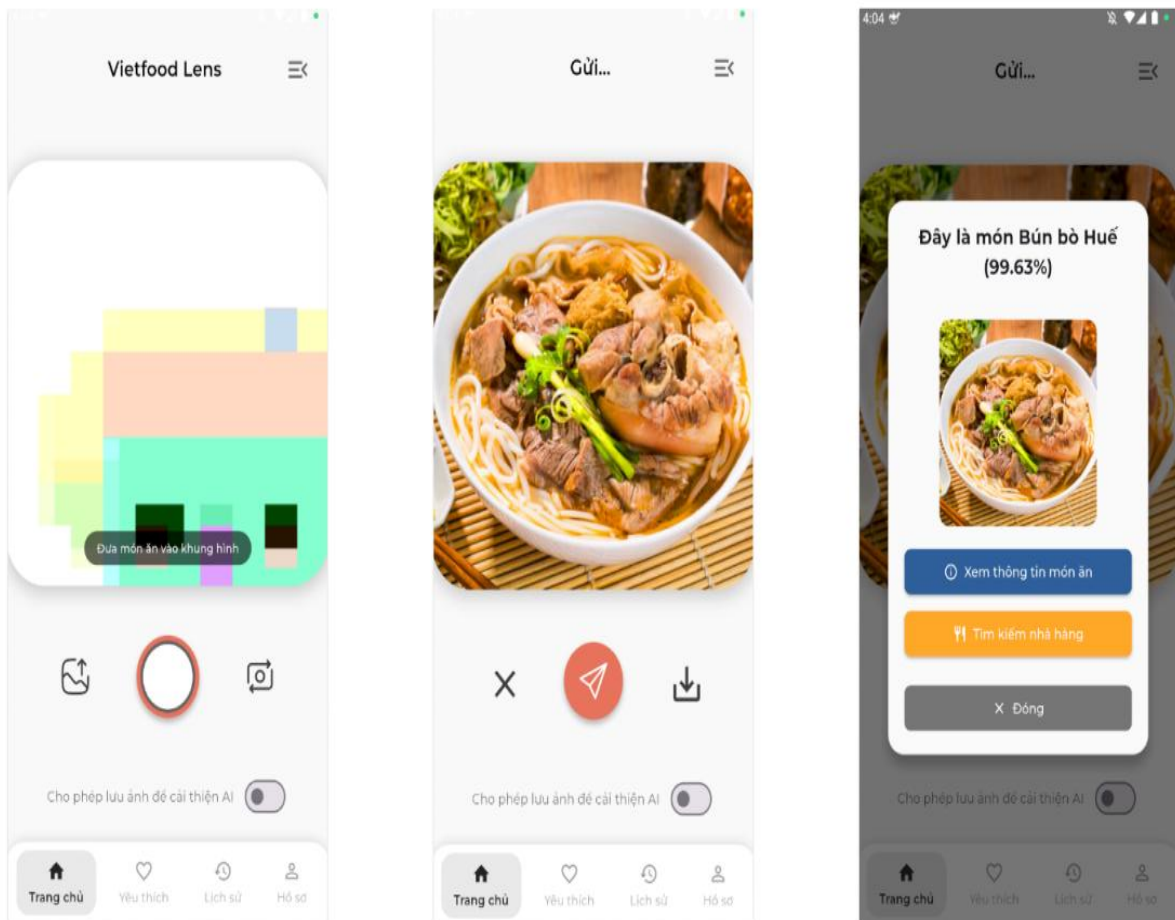
Hình 3.1: Giao diện màn hình chào mừng và đăng nhập của ứng dụng Vietfood Lens

Khi người dùng mở ứng dụng, màn hình chào mừng sẽ hiển thị kèm logo và mô tả ngắn gọn về chức năng chính của ứng dụng. Từ đây, người dùng có thể chọn "Đăng nhập" hoặc "Tạo tài khoản". Nếu chọn đăng nhập, người dùng sẽ được chuyển sang màn hình đăng nhập, nơi họ có thể nhập thông tin tài khoản hoặc sử dụng tính năng đăng nhập bằng Google.



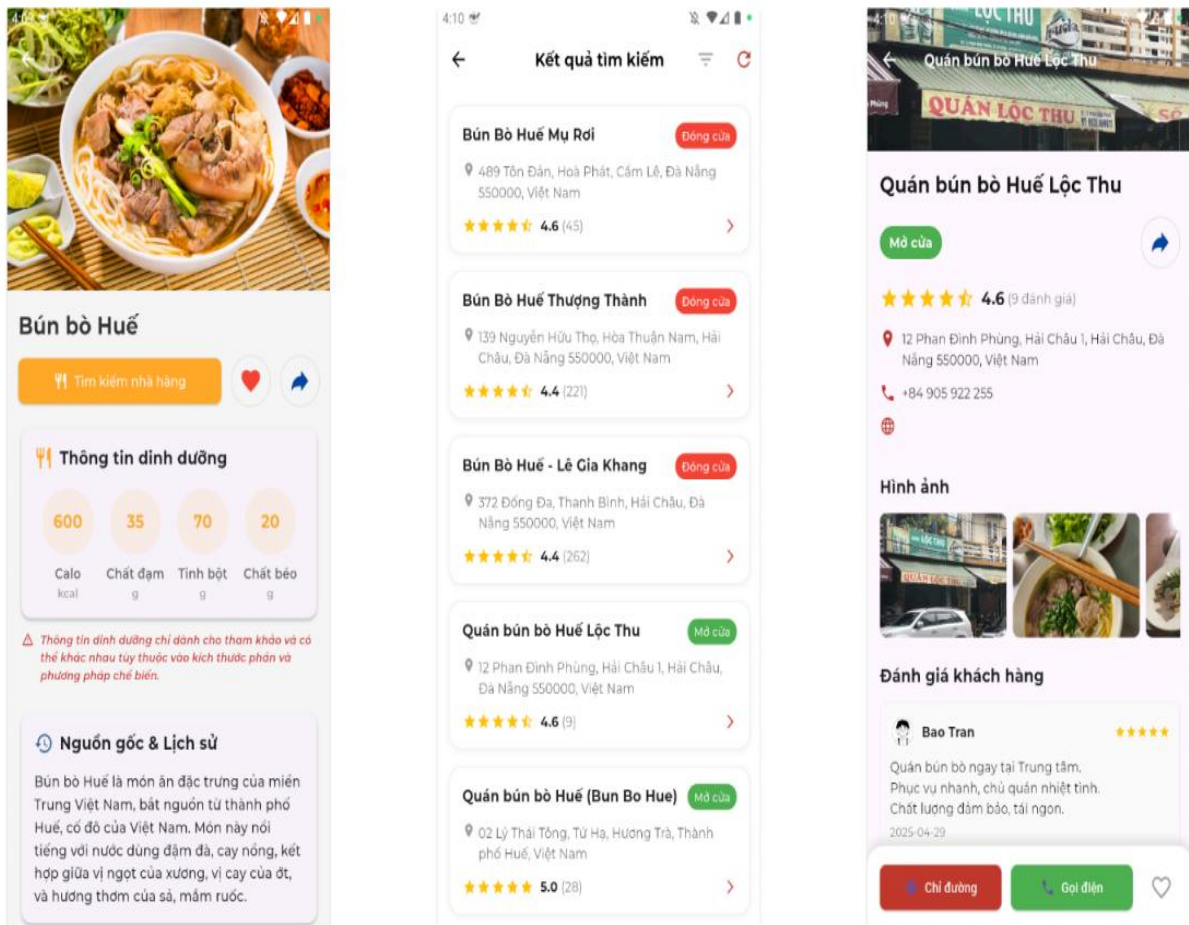
Hình 3.2: Giao diện quy trình đăng ký của ứng dụng Vietfood Lens

Khi người dùng tạo tài khoản, hệ thống sẽ yêu cầu nhập email để gửi mã OTP. Sau khi xác thực mã OTP thành công, người dùng tiếp tục điền thông tin cá nhân và mật khẩu để hoàn tất quá trình đăng ký. Giao diện được thiết kế trực quan, phân thành từng bước nhỏ giúp tối ưu trải nghiệm và đảm bảo tính bảo mật.



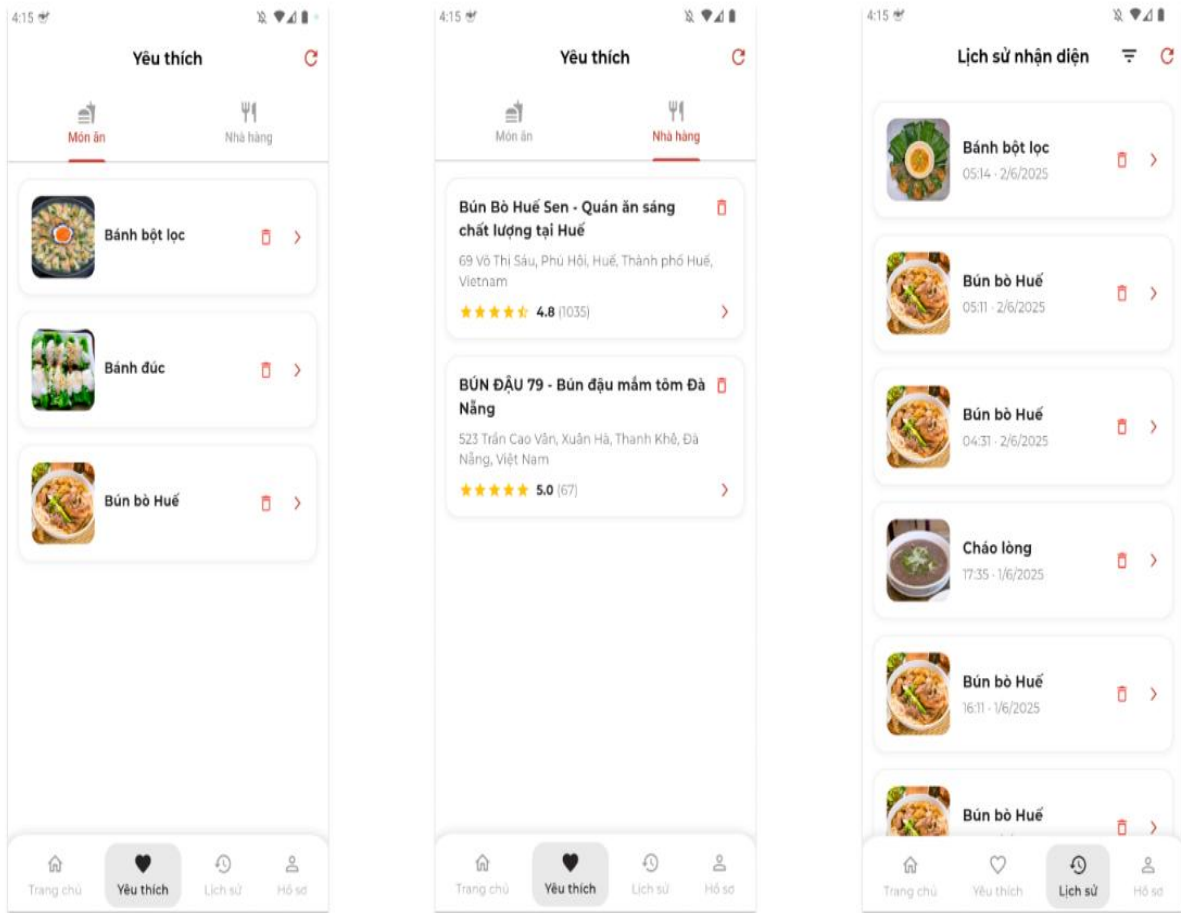
Hình 3.3: Giao diện quá trình nhận diện món ăn

Đây là chức năng chính của hệ thống, cho phép người dùng nhận diện món ăn Việt Nam thông qua ảnh chụp hoặc tải lên từ thư viện. Kết quả dự đoán được hiển thị trực quan, kèm theo các hành động tiếp theo như xem thông tin món ăn hoặc tìm kiếm nhà hàng gần đó.



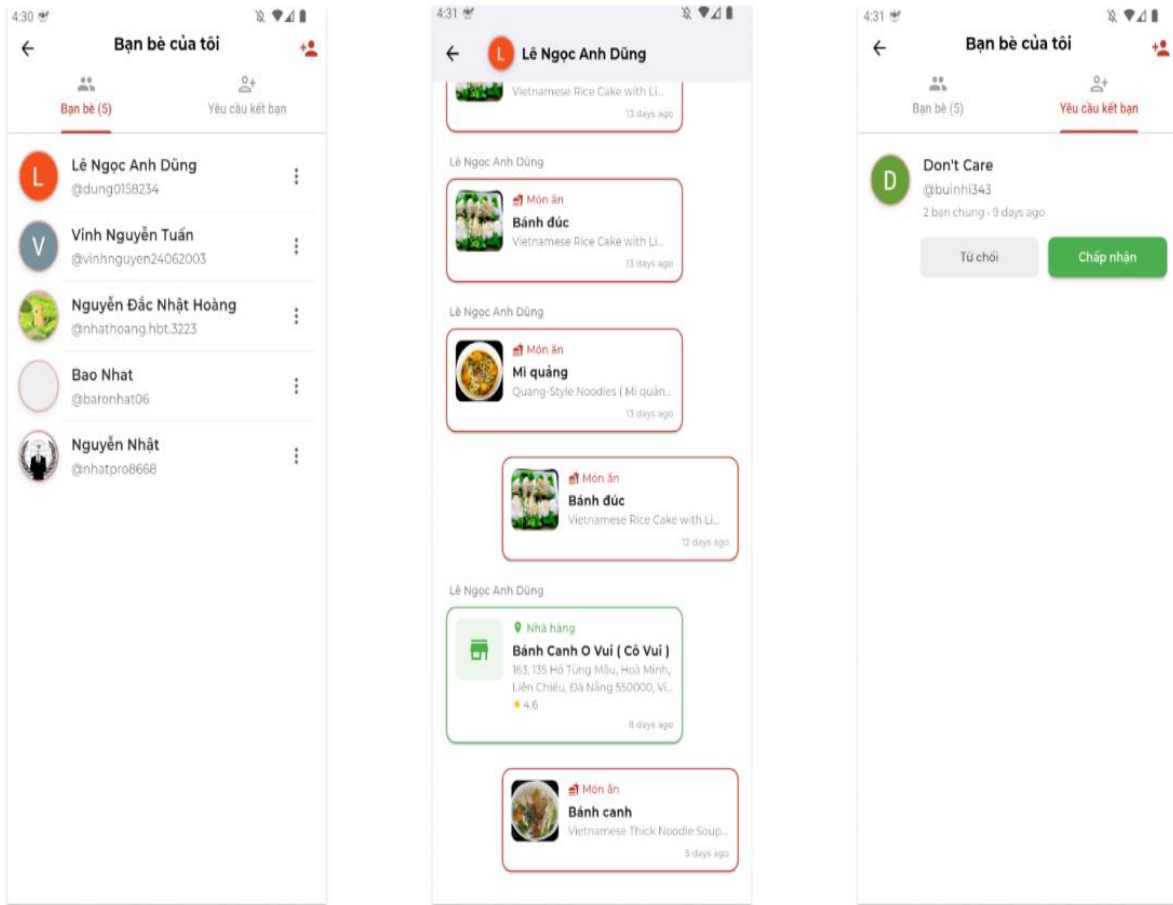
Hình 3.4: Giao diện xem thông tin món ăn và tìm kiếm quán ăn gần đó

Sau khi nhận diện món ăn thành công, người dùng có thể xem thông tin chi tiết về món ăn bao gồm giá trị dinh dưỡng tham khảo, nguồn gốc và lịch sử, nguyên liệu, cách chế biến. Từ màn hình này, người dùng có thể chọn chức năng "Tìm kiếm nhà hàng", hệ thống sẽ trả về danh sách các quán ăn gần đó phù hợp với món ăn. Khi chọn một nhà hàng cụ thể, người dùng có thể xem chi tiết địa điểm, đánh giá, ảnh thực tế và sử dụng các chức năng như gọi điện, chỉ đường hoặc lưu vào danh sách yêu thích.



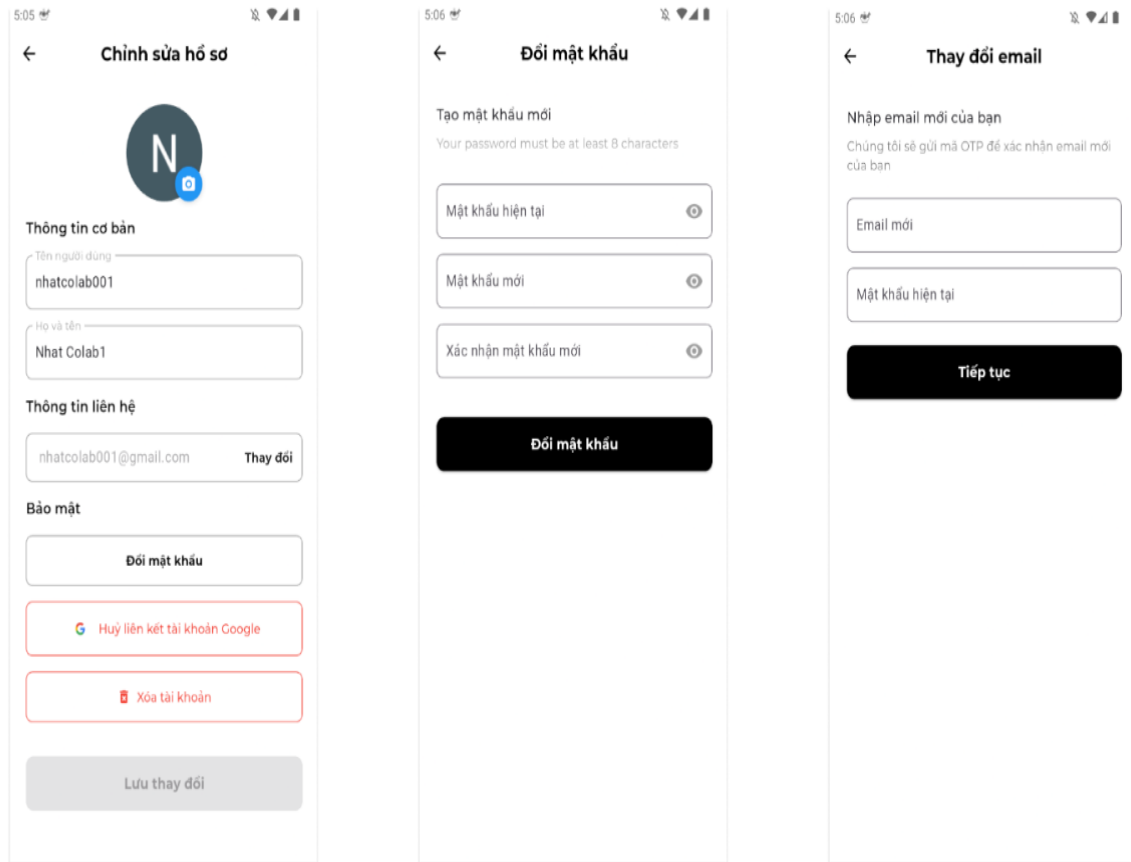
Hình 3.5: Giao diện chức năng yêu thích và lịch sử nhận diện

Người dùng có thể truy cập nhanh vào danh sách các món ăn và nhà hàng yêu thích thông qua tab “Yêu thích”. Giao diện phân thành hai mục rõ ràng: “Món ăn” và “Nhà hàng”. Ngoài ra, món ăn được nhận diện thành công cũng được lưu lại trong tab “Lịch sử”. Các mục đều hỗ trợ thao tác xoá khi không còn cần thiết.



Hình 3.6: Giao diện quản lý bạn bè

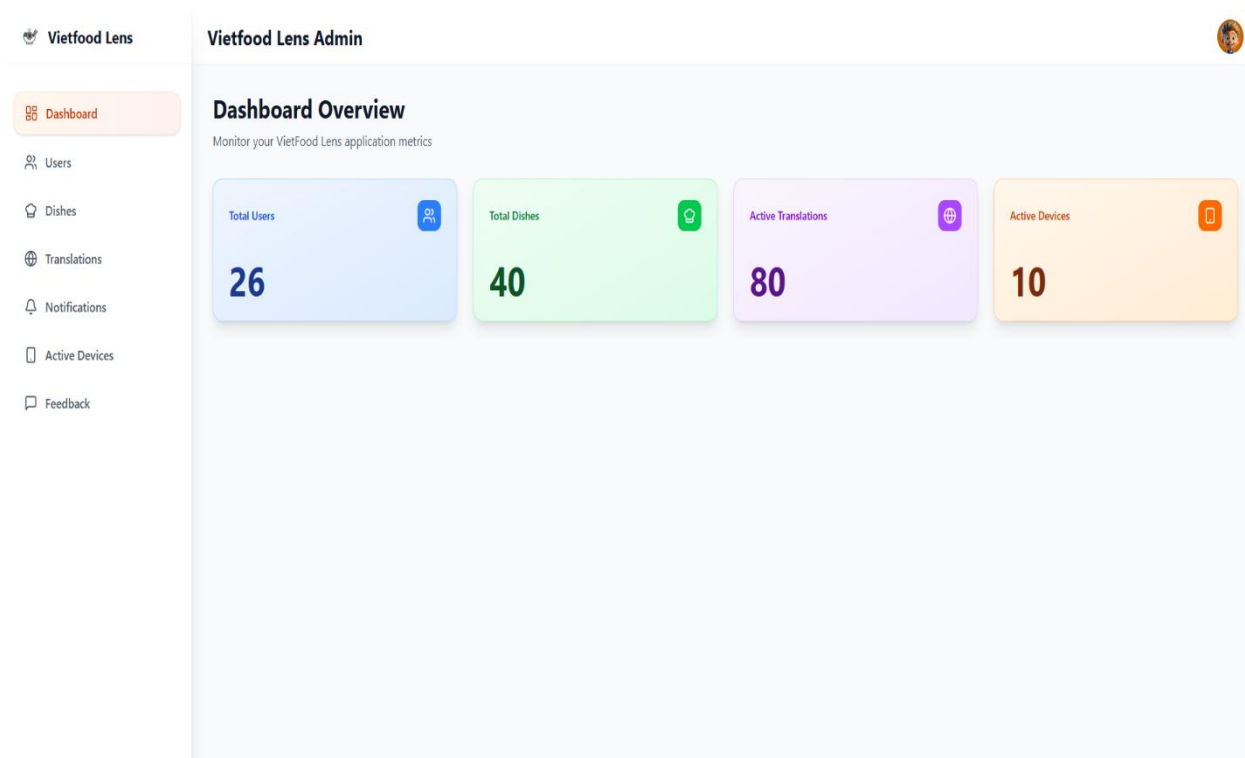
Ứng dụng hỗ trợ chức năng kết nối người dùng nội bộ thông qua tính năng bạn bè. Người dùng có thể tìm kiếm, gửi lời mời, chấp nhận hoặc từ chối yêu cầu kết bạn. Sau khi trở thành bạn bè, người dùng có thể chia sẻ cho bạn bè những món ăn hoặc quán ăn. Tính năng này tạo sự tương tác xã hội, giúp người dùng khám phá món ăn và địa điểm một cách thú vị hơn.



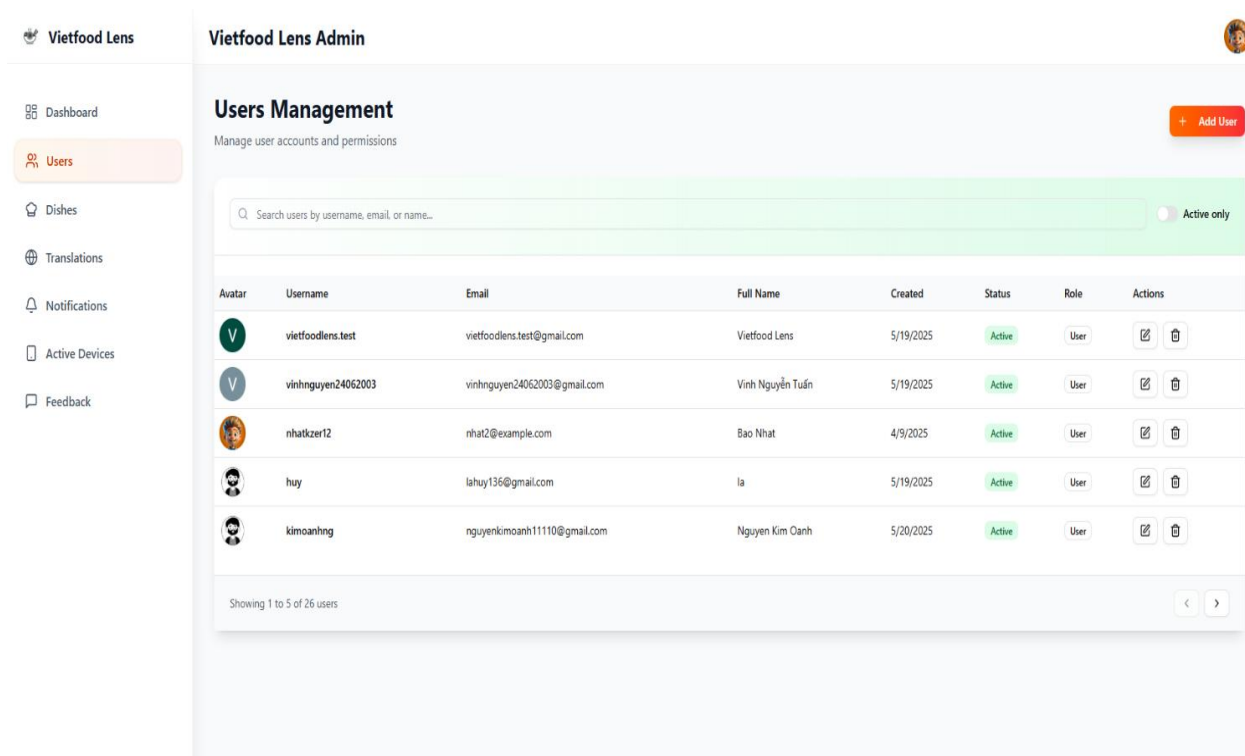
Hình 3.7: Giao diện quản lý tài khoản người dùng

Ứng dụng cung cấp giao diện quản lý hồ sơ người dùng thân thiện, cho phép chỉnh sửa thông tin cá nhân, cập nhật ảnh đại diện và thay đổi thông tin đăng nhập. Ngoài ra, người dùng có thể thực hiện các thao tác bảo mật như đổi mật khẩu, thay đổi email, hoặc xóa tài khoản nếu không còn sử dụng. Việc liên kết tài khoản Google cũng có thể được hủy bất kỳ lúc nào để đảm bảo quyền kiểm soát dữ liệu cá nhân.

3.2.2. Giao diện Quản trị viên



Hình 3.8: Giao diện trang quản trị hệ thống



Hình 3.9: Giao diện trang quản lý người dùng

Quản trị viên có thể quản lý tài khoản người dùng, bao gồm việc tìm kiếm, chỉnh sửa, xóa hoặc thay đổi trạng thái hoạt động (kích hoạt/vô hiệu hóa) của người dùng.

3.3. Đánh giá kết quả

Sau quá trình triển khai, hệ thống đã hoàn thiện đầy đủ các chức năng chính bao gồm: nhận diện món ăn qua hình ảnh, tìm kiếm quán ăn, lưu yêu thích, lịch sử, kết bạn và quản lý hồ sơ cá nhân. Mô hình AI hoạt động ổn định, thời gian phản hồi nhanh. Giao diện người dùng thân thiện, trực quan. Ngoài ra hệ thống cũng cung cấp giao diện quản trị cho phép theo dõi và quản lý người dùng.

KẾT LUẬN

1. Kết quả

Trong thời gian tìm hiểu, nghiên cứu cơ sở lý thuyết, phân tích hệ thống và triển khai ứng dụng công nghệ, đồ án đã đạt được những kết quả sau:

Về mặt lý thuyết:

- Nắm được quy trình xây dựng một ứng dụng di động hoàn chỉnh.
- Vận dụng được các kiến thức đã học để phân tích và xây dựng hệ thống.
- Lập trình backend với FastAPI: xây dựng API, quản lý cơ sở dữ liệu quan hệ, xử lý bảo mật.
- Xây dựng ứng dụng di động bằng Flutter.
- Nâng cao hiểu biết về cách triển khai hệ thống trên nền tảng cloud và tối ưu trải nghiệm người dùng.

Về mặt thực tiễn:

- Hệ thống đã được xây dựng và triển khai thành công, đáp ứng đầy đủ các chức năng cơ bản cần có.
- Người dùng có thể nhận diện món ăn từ ảnh chụp, tìm kiếm nhà hàng gần đó, lưu lại yêu thích, theo dõi lịch sử và tương tác với bạn bè.
- Mô hình AI hoạt động ổn định, giao diện dễ sử dụng.

2. Hạn chế

Mặc dù hệ thống đã hoàn thiện và hoạt động ổn định, nhưng vẫn còn một số hạn chế cần cải thiện:

- Độ chính xác của mô hình AI chưa thực sự cao với các món ăn ít phổ biến hoặc có hình dạng tương tự nhau.
- Tập dữ liệu dữ liệu huấn luyện còn hạn chế, hiện hệ thống mới hỗ trợ nhận diện 40 món ăn. Số lượng ảnh cho mỗi lớp chưa đồng đều, ảnh chưa đủ đa dạng về góc chụp, hình dạng món ăn và điều kiện ánh sáng.
- Kết quả nhận diện phụ thuộc vào chất lượng ảnh đầu vào: ảnh mờ, thiếu sáng hoặc ảnh không phải món ăn (như vật dụng, tay người, hình in...) có thể gây dự đoán sai.
- Tìm kiếm quán ăn dựa trên Google Maps API có thể thiếu thông tin ở một số khu vực ít dữ liệu hoặc quán ăn nhỏ chưa hiển thị.
- Ứng dụng hiện chỉ hỗ trợ trên Android, chưa triển khai trên nền tảng iOS.

3. Hướng phát triển

- Mở rộng số lượng món ăn: Bổ sung thêm nhiều món ăn Việt Nam vào tập dữ liệu để tăng độ bao phủ và tính ứng dụng thực tế.
- Nâng cấp mô hình AI: Thu thập thêm dữ liệu đa dạng hơn để cải thiện độ chính xác, đặc biệt với các món dễ gây nhầm lẫn.
- Triển khai trên nền tảng iOS: Mở rộng phạm vi sử dụng sang hệ điều hành iOS để tiếp cận thêm người dùng.

TÀI LIỆU THAM KHẢO

- [1]. FastAPI. (n.d.). <https://fastapi.tiangolo.com/>
- [2]. EfficientNet: Improving Accuracy and Efficiency through AutoML and Model Scaling. (n.d.).
<https://research.google/blog/efficientnet-improving-accuracy-and-efficiency-through-automl-and-model-scaling/>
- [3]. Tan, M., & Le, Q., V. (2019, May 28). EfficientNet: Rethinking model scaling for convolutional neural networks. arXiv.org. <https://arxiv.org/abs/1905.11946>
- [4]. Google Maps Platform Documentation. (n.d.). Google for Developers. <https://developers.google.com/maps/documentation/places/android-sdk>
- [5]. Mitra, S. (2025, January 20). Integrating Google Authentication with FastAPI: A Step-by-Step Guide. FutureSmart AI Blog.
<https://blog.futuresmart.ai/integrating-google-authentication-with-fastapi-a-step-by-step-guide>
- [6]. Set up a Firebase Cloud Messaging client app on Android. (n.d.). Firebase. <https://firebase.google.com/docs/cloud-messaging/android/client>
- [7]. App Platform | DigitalOcean Documentation. (n.d.).
<https://docs.digitalocean.com/products/app-platform/>
- [8]. Zoho Mail - Apps documentation. (n.d.). <https://apps.make.com/zoho-mail>
- [9]. Converting Images with Python | Cloudinary. (2025, May 17). Cloudinary.
<https://cloudinary.com/guides/web-performance/converting-images-with-python>

PHỤ LỤC 1

Danh sách các món ăn được sử dụng trong hệ thống

STT	Tên món ăn
1	Bánh bao
2	Bánh bèo
3	Bánh bột lọc
4	Bánh căn
5	Bánh canh
6	Bánh chưng
7	Bánh cuốn
8	Bánh da lợn
9	Bánh đúc
10	Bánh giò
11	Bánh khọt
12	Bánh mì
13	Bánh pía
14	Bánh tét
15	Bánh tráng nướng
16	Bánh xèo
17	Bún bò Huế
18	Bún chả Hà Nội
19	Bún đậu mắm tôm
20	Bún mắm
21	Bún riêu
22	Bún thịt nướng
23	Cá kho tộ
24	Canh chua
25	Cao lầu
26	Chả giò/Nem rán
27	Cháo lòng
28	Chè đậu đen
29	Cơm gà Hội An
30	Cơm hến
31	Cơm tấm
32	Gà nướng muối ớt
33	Gỏi cuốn
34	Hủ tiếu
35	Mì quảng
36	Nem chua
37	Nem lụi
38	Phở
39	Súp cua
40	Xôi xéo