

BỘ NÔNG NGHIỆP VÀ PHÁT TRIỂN NÔNG THÔN

TRƯỜNG ĐẠI HỌC THỦY LỢI



BÀI TẬP LỚN

HỌC PHẦN: PHÁT TRIỂN ỨNG DỤNG CHO CÁC THIẾT BỊ DI ĐỘNG

ĐỀ TÀI: Xây dựng ứng dụng đặt đơn cho nhà hàng 555

Mã Sinh Viên	Họ và Tên	Ngày Sinh	Lớp
2151173786	Lê Văn Khải	05/12/2003	63KTPM1
2151173763	Nguyễn Ngọc Đăng	02/11/2003	63KTPM1
2151170537	Nguyễn Trung Kiên	29/06/2003	63KTPM1
2151173790	Bùi Việt Linh	20/05/2003	63KTPM1

Hà Nội, năm 2024

BỘ NÔNG NGHIỆP VÀ PHÁT TRIỂN NÔNG THÔN

TRƯỜNG ĐẠI HỌC THỦY LỢI



BÀI TẬP LỚN

HỌC PHẦN: PHÁT TRIỂN ỨNG DỤNG CHO CÁC THIẾT BỊ DI ĐỘNG

ĐỀ TÀI: Xây dựng ứng dụng đặt đơn cho nhà hàng 555

Mã Sinh Viên	Họ và Tên	Ngày Sinh	Điểm	
			Bảng Số	Bảng Chữ
2151173786	Lê Văn Khải	1/11/2003		
2151173763	Nguyễn Ngọc Đăng	02/11/2003		
2151173790	Bùi Việt Linh	20/05/2003		
2151170537	Nguyễn Trung Kiên	29/06/2003		

CÁN BỘ CHẤM THI 1

CÁN BỘ CHẤM THI 2

Hà Nội, năm 2024

LỜI NÓI ĐẦU

Trong bối cảnh xã hội ngày càng phát triển, công nghệ thông tin không ngừng khẳng định vai trò quan trọng, không chỉ trong các lĩnh vực sản xuất mà còn trong ngành dịch vụ, đặc biệt là dịch vụ nhà hàng. Nhờ công nghệ, các nhà hàng có cơ hội nâng cao chất lượng phục vụ, tạo ra những trải nghiệm mượt mà và thuận tiện cho khách hàng. Đặt trong bối cảnh của thời đại số hóa, các ứng dụng hỗ trợ đặt bàn và gọi món trực tuyến đã trở thành một phần không thể thiếu trong chiến lược phát triển dịch vụ hiện đại. "Nhà hàng 555" chính là minh chứng sống động cho xu hướng này, khi công nghệ được ứng dụng để tối ưu hóa dịch vụ và nâng tầm trải nghiệm ẩm thực của khách hàng.

Ứng dụng "Nhà hàng 555" không chỉ đơn thuần là công cụ đặt bàn trực tuyến; nó còn mang đến khả năng cập nhật thực đơn một cách linh hoạt, gợi ý món ăn phù hợp với sở thích cá nhân của từng khách hàng. Với ứng dụng này, nhà hàng không chỉ giảm thiểu được thời gian chờ đợi mà còn tối ưu hóa không gian và nguồn lực, mang lại hiệu quả cao trong quá trình vận hành.

Với những tính năng thông minh và dễ dàng sử dụng, ứng dụng "Nhà hàng 555" là công cụ hỗ trợ toàn diện cho nhà hàng cũng như thực khách trong việc quản lý đơn hàng và thực đơn. Đồng thời, nó mang đến cho người dùng trải nghiệm ẩm thực tiện lợi, nhanh chóng, hiện đại. Dù ở bất kỳ đâu, chỉ với vài thao tác đơn giản, bạn có thể tận hưởng không gian ẩm thực phong phú mà không cần lo lắng về những bất tiện thường gặp khi đến nhà hàng.

MỤC LỤC

BẢNG CÁC TỪ VIẾT TẮT	4
CHƯƠNG 1. MÔ TẢ BÀI TOÁN	5
1.1. Giới thiệu	5
1.2. Chức năng chính	5
1. Xem trạng thái bàn và đặt bàn	5
2. Xem và đặt món ăn	5
3. Thanh toán	6
4. Quản lý tài khoản	6
5. Tìm kiếm và lọc món ăn	6
1.3. Yêu cầu phi chức năng	7
1. Hiệu năng	7
2. Khả năng mở rộng	7
3. Tính bảo mật	7
4. Tính ổn định và tin cậy	8
5. Khả năng tương thích	8
6. Khả năng sử dụng	8
7. Khả năng bảo trì	8
8. Khả năng giám sát	8
9. Tính năng cập nhật tự động	9
CHƯƠNG 2. PHÂN TÍCH YÊU CẦU VÀ THIẾT KẾ HỆ THỐNG	10
2.1. Phân tích yêu cầu:	10
2.2. Thiết kế hệ thống:	14
CHƯƠNG 3. KẾT QUẢ THỰC HIỆN	33
3.1. Công nghệ đã sử dụng	33
3.2. Tiến độ thực hiện	33
3.3. Hình ảnh sản phẩm	34
KẾT LUẬN	44

DANH MỤC HÌNH ẢNH

Hình 2.2.1: Biểu đồ lớp cho mô hình miền	15
Hình 2.2.2: Lược đồ mô hình quan hệ	29
Hình 3.3.1. Giao diện đăng ký	34
Hình 3.3.2. Giao diện xác thực OTP	35
Hình 3.3.3. Giao diện đăng nhập	36
Hình 3.3.4. Giao diện trang chủ	37
Hình 3.3.5. Giao diện tìm kiếm món ăn	38
Hình 3.3.6. Giao diện đã xem gần đây	39
Hình 3.3.7. Giao diện chi tiết món ăn và thêm vào giỏ	40
Hình 3.3.8. Giao diện giỏ hàng	41
Hình 3.3.9. Giao diện hồ sơ người dùng	42
Hình 3.3.10. Giao diện đặt bàn	43

BẢNG CÁC TỪ VIẾT TẮT

STT	TỪ VIẾT TẮT	VIẾT ĐẦY ĐỦ
1	CSDL	Cơ sở dữ liệu
2	GUI	Graphical User Interface
3	API	Application Programming Interface
4	IDE	Integrated Development Environment

CHƯƠNG 1. MÔ TẢ BÀI TOÁN

1.1. Giới thiệu

Ứng dụng đặt bàn và đặt món ăn online là giải pháp hoàn hảo cho những ai muốn tận hưởng trải nghiệm ẩm thực tiện lợi, nhanh chóng mà không cần phải đến trực tiếp nhà hàng. Chỉ với vài thao tác trên màn hình, người dùng có thể đặt bàn, lựa chọn món ăn yêu thích và lên kế hoạch bữa ăn theo ý muốn mọi lúc, mọi nơi. Được xây dựng trên nền tảng Java và XML cùng các thư viện hiện đại như JavaFX, ứng dụng hứa hẹn mang đến giao diện trực quan, sinh động và thân thiện, giúp người dùng dễ dàng thao tác và trải nghiệm một dịch vụ chất lượng cao ngay trên thiết bị di động của mình. Đây sẽ là công cụ không thể thiếu cho những ai yêu thích sự tiện ích và nhanh chóng trong nhịp sống hiện đại.

1.2. Chức năng chính

Ứng dụng Nhà hàng 555 cần có các chức năng sau:

1. Xem trạng thái bàn và đặt bàn

- Đặt bàn: Cho phép khách hàng đặt bàn trước qua ứng dụng. Chọn thời gian bắt đầu, loại bàn, số lượng bàn và thời gian sử dụng thêm (nếu có nhu cầu sử dụng thêm).
- Xem chi tiết bàn: Hiển thị thông tin chi tiết của từng bàn (số bàn, tình trạng).
- Trạng thái bàn được cập nhật theo thời gian thực, không làm ảnh hưởng đến trải nghiệm khách hàng.
- Lịch sử đặt bàn: Cho phép khách hàng xem lại lịch sử đặt bàn.

2. Xem và đặt món ăn

- Thêm món vào giỏ hàng: Khách hàng có thể chọn món ăn từ thực đơn và thêm vào giỏ hàng.
- Chỉnh sửa giỏ hàng: Cho phép thêm/xóa món, cập nhật số lượng món đã chọn trong giỏ hàng.

- Xác nhận đặt món: Hoàn tất đặt món ăn, chuyển trạng thái đơn hàng.
- Lịch sử món ăn: Cho phép khách hàng xem lại lịch sử các món đã xem hoặc đã gọi trước đó.

3. Thanh toán

- Xem chi tiết hóa đơn: Hiện thị chi tiết các món đã chọn và tổng số tiền phải thanh toán.
- Thanh toán trực tuyến: Cho phép khách hàng thanh toán hóa đơn qua các phương thức thanh toán trực tuyến (thẻ, ví điện tử).
- Thanh toán tại quầy: Cung cấp tùy chọn thanh toán trực tiếp tại quầy khi sử dụng dịch vụ tại nhà hàng.

4. Quản lý tài khoản

- Đăng ký/Đăng nhập: Cho phép khách hàng tạo tài khoản mới hoặc đăng nhập vào tài khoản hiện có.
- Đăng xuất: Cho phép khách hàng thoát tài khoản khỏi ứng dụng hiện tại.
- Cập nhật thông tin cá nhân: Khách hàng có thể cập nhật thông tin cá nhân như tên, gmail, ảnh đại diện.
- Đổi mật khẩu: Cho phép khách hàng đổi mật khẩu.
- Quên mật khẩu: Cung cấp lại mật khẩu mới cho khách hàng nếu quên.
- Lịch sử đặt bàn và món ăn: Cho phép khách hàng xem lại lịch sử đặt bàn và các món đã gọi trước đó.

5. Tìm kiếm và lọc món ăn

- Tìm kiếm món ăn: Cho phép khách hàng tìm kiếm món ăn theo tên hoặc loại món.
- Lọc món ăn: Hỗ trợ lọc món theo danh mục (món khai vị, món chính, tráng miệng).

- Gợi ý món ăn: Hiển thị các gợi ý món ăn theo những món ăn được mua nhiều nhất, những món ăn theo từng thời điểm trong ngày.
- Xem lại những món ăn đã từng xem.

1.3. Yêu cầu phi chức năng

1. Hiệu năng

- Ứng dụng phải có khả năng phản hồi nhanh, thời gian tải trang không vượt quá 3 giây.
- Ứng dụng phải xử lý được ít nhất 100 yêu cầu đặt bàn hoặc đặt món đồng thời trong giờ cao điểm.
- Ứng dụng phải có cơ chế dự phòng để tự động phục hồi từ các sự cố nhẹ, đảm bảo không làm gián đoạn dịch vụ cho người dùng.

2. Khả năng mở rộng

- Hệ thống phải có khả năng mở rộng linh hoạt để xử lý lượng người dùng tăng lên trong tương lai mà không ảnh hưởng đến hiệu suất.
- Cơ sở dữ liệu và máy chủ phải có khả năng nâng cấp khi cần.

3. Tính bảo mật

- Dữ liệu người dùng phải được bảo vệ và mã hóa (bao gồm mật khẩu và thông tin thanh toán).
- Ứng dụng phải có các biện pháp xác thực để đảm bảo chỉ người dùng hợp lệ mới có thể truy cập vào hệ thống.
- Các giao dịch thanh toán phải tuân thủ tiêu chuẩn bảo mật PCI-DSS.

4. Tính ổn định và tin cậy

- Ứng dụng phải có tính ổn định cao, với tỷ lệ uptime không thấp hơn 99.9%.
- Hệ thống phải đảm bảo không mất dữ liệu khi xảy ra lỗi hệ thống hoặc gián đoạn mạng.

5. Khả năng tương thích

- Ứng dụng phải tương thích với hệ điều hành Android.
- Giao diện người dùng phải hiển thị tốt trên nhiều kích thước màn hình khác nhau, bao gồm điện thoại và máy tính bảng.

6. Khả năng sử dụng

- Giao diện người dùng phải dễ sử dụng và trực quan, với các nút và chức năng dễ tìm thấy.

7. Khả năng bảo trì

- Mã nguồn của ứng dụng phải được tổ chức rõ ràng và dễ bảo trì, để dễ dàng sửa lỗi và nâng cấp.
- Cấu trúc mã phải có tài liệu chi tiết, với các ghi chú code cần thiết để dễ dàng hiểu và cập nhật.

8. Khả năng giám sát

- Ứng dụng phải cung cấp các công cụ giám sát và báo cáo hoạt động, bao gồm số lượng người dùng đang hoạt động, lượng đặt bàn/món, và các vấn đề xảy ra trong hệ thống.

9. Tính năng cập nhật tự động

- Ứng dụng phải có khả năng cập nhật tự động khi có phiên bản mới, giúp người dùng luôn sử dụng phiên bản mới nhất.

CHƯƠNG 2. PHÂN TÍCH YÊU CẦU VÀ THIẾT KẾ HỆ THỐNG

2.1. Phân tích yêu cầu:

Xác định người dùng:

Ứng dụng đặt bàn và đồ ăn nhà hàng được thiết kế để phục vụ các nhóm người dùng chính sau:

Khách hàng

Mô tả: Là những người sử dụng ứng dụng để đặt bàn, đặt món ăn và thanh toán trực tuyến.

Vai trò:

- Xem thực đơn, chọn món, và đặt bàn trước theo nhu cầu.
- Theo dõi trạng thái bàn đặt, đơn hàng, và lịch sử đặt bàn.
- Cập nhật thông tin tài khoản cá nhân và thay đổi mật khẩu.
- Đảm bảo an ninh, bảo mật cho hệ thống và người dùng, xem lịch sử hoạt động.

Thu thập yêu cầu:

Để xây dựng một hệ thống đáp ứng nhu cầu của các đối tượng người dùng trên, quá trình thu thập yêu cầu được thực hiện qua các phương pháp sau:

- **Phỏng vấn trực tiếp**

Đối tượng: Khách hàng, nhân viên, quản lý nhà hàng, và quản trị viên hệ thống.

Nội dung phỏng vấn:

1. Đối với khách hàng:

- Khám phá các vấn đề trong quá trình đặt bàn, đặt món, và thanh toán.

- Tìm hiểu về trải nghiệm tổng quan của khách hàng với giao diện đặt bàn và đặt món, mức độ tiện lợi, tốc độ xử lý, và các khó khăn gặp phải.
- Đánh giá mức độ hài lòng của khách hàng về dịch vụ, từ việc đặt bàn cho đến việc thanh toán và phản hồi sau khi sử dụng dịch vụ.
- Xem xét những tính năng mà khách hàng mong muốn có thêm trong hệ thống để nâng cao trải nghiệm như: tính năng đặt bàn trước nhiều ngày, theo dõi trạng thái đơn hàng theo thời gian thực.

2. Đối với nhân viên:

- Tìm hiểu về quy trình xử lý đơn hàng, quản lý bàn và món ăn, từ lúc nhận đơn đến lúc hoàn tất giao dịch với khách hàng.
- Đánh giá sự thuận tiện và hiệu quả của hệ thống quản lý đặt bàn, đơn hàng, và liệu có bất kỳ khó khăn nào trong quá trình sử dụng hệ thống hay không.
- Xem xét các khó khăn mà nhân viên gặp phải trong quá trình làm việc hàng ngày và đề xuất cải tiến để tăng cường hiệu quả hoạt động.

3. Đối với quản trị viên:

- Tìm hiểu yêu cầu về quản lý hệ thống, bảo trì, và bảo mật.
- Đánh giá các công cụ quản lý hiện tại, những yếu tố cần cải tiến để nâng cao sự ổn định và bảo mật của hệ thống, đồng thời đảm bảo rằng hệ thống có thể hoạt động liên tục và hỗ trợ tốt cho người dùng cuối.

● Khảo sát trực tuyến

Đối tượng: Khách hàng và nhân viên nhà hàng.

Nội dung khảo sát:

- Với khách hàng:
 - Trải nghiệm người dùng: Hỏi khách hàng về mức độ dễ sử dụng của ứng dụng, từ giao diện đến quy trình đặt bàn/đặt món.

- Phản hồi về tính năng hiện tại: Đánh giá các tính năng hiện có như đặt món trước, đặt bàn trước giờ ăn và tính năng ưu tiên.
 - Góp ý cải tiến: Mời khách hàng gợi ý các cải tiến mong muốn, có thể là về tốc độ ứng dụng, giao diện hay thêm tính năng tích điểm.
- Với nhân viên:
- Hiệu suất công việc: Khảo sát về mức độ hỗ trợ của ứng dụng đối với các tác vụ hàng ngày và yêu cầu cải tiến, như thêm thông báo đơn hàng hay tính năng tự động cập nhật trạng thái.
 - Đánh giá quy trình làm việc: Hỏi về mức độ thuận tiện của ứng dụng trong quy trình phục vụ (ví dụ, nhận đơn món và xác nhận đặt bàn).

Kết quả mong đợi:

- Các phản hồi sẽ giúp phát hiện những vấn đề về giao diện hoặc tính năng chưa tối ưu, từ đó đưa ra các thay đổi để làm cho ứng dụng thân thiện và trực quan hơn.
- Từ kết quả khảo sát, đội phát triển có thể xác định được các tính năng cần thiết mà khách hàng thực sự quan tâm, ví dụ như tính năng đặt bàn theo vị trí cụ thể, xem menu với hình ảnh món ăn hoặc tính năng khác.

● Quan sát thực tế tại nhà hàng

Đối tượng: Quan sát hoạt động của nhân viên (phục vụ, thu ngân, đầu bếp), quản lý và khách hàng tại nhà hàng.

Mục tiêu:

- Hiểu rõ quy trình hoạt động từ đặt bàn, gọi món, phục vụ món ăn, xử lý yêu cầu của khách hàng, thanh toán, và các dịch vụ bổ sung (như ghi nhận ý kiến phản hồi, khuyến mãi) để xác định các tính năng và luồng công việc hợp lý cho ứng dụng.
- Phát hiện các khó khăn, bất cập trong quy trình hiện tại để đề xuất cải tiến cho ứng dụng.

- Xác định các nhu cầu của khách hàng như việc ưu tiên tốc độ phục vụ, tính dễ dàng khi thanh toán, hoặc yêu cầu cá nhân hóa.

Phạm vi quan sát:

- Nhân viên phục vụ: Thao tác khi nhận order, phục vụ món ăn, xử lý yêu cầu bổ sung, và tương tác với khách hàng.
- Thu ngân: Quy trình thanh toán (tiền mặt, thẻ, ví điện tử), quản lý hóa đơn và cách ứng phó khi có yêu cầu từ khách hàng về thanh toán.
- Quản lý: Kiểm soát chất lượng dịch vụ, xử lý khiếu nại, phản hồi của khách hàng, và quản lý nhân sự trong ca làm việc.
- Khách hàng: Hành vi từ khi bước vào nhà hàng, lựa chọn chỗ ngồi, cách thức đặt món, phản hồi khi nhận món, và trải nghiệm trong suốt quá trình phục vụ.

Kết quả mong đợi:

- Thu thập dữ liệu để xây dựng các luồng công việc tối ưu và tích hợp vào ứng dụng, từ đó giúp tăng hiệu suất phục vụ và nâng cao trải nghiệm người dùng.
- Đề xuất các tính năng cải tiến, như tính năng đặt bàn trước, đặt món từ xa, thanh toán qua ứng dụng, hoặc phần mềm phản hồi tức thời từ khách hàng.

- **Phân tích hệ thống hiện tại**

- **Nội dung:** Phân tích các hệ thống quản lý nhà hàng hiện tại để tìm ra những hạn chế và điểm yếu. Qua đó, xem xét những chức năng và quy trình chưa hiệu quả, chẳng hạn như quản lý kho, đặt món, giao diện người dùng và tính năng bảo mật. Đánh giá khả năng tích hợp, tính tự động hóa, và sự thân thiện với người dùng của từng hệ thống. Cuối cùng, đưa ra các giải pháp cải tiến khả thi dựa trên những hạn chế này.
- **Mục tiêu:** Xác định các giải pháp nâng cao hiệu quả hoạt động và tối ưu hóa trải nghiệm của cả khách hàng và nhân viên trong nhà hàng. Cụ thể, đề xuất những cải tiến như đồng bộ hóa dữ liệu theo thời gian thực, nâng cấp tính năng bảo mật, cải thiện giao diện người dùng, và tối ưu quản lý kho hàng để

giúp nhà hàng hoạt động linh hoạt, nhanh chóng, đồng thời giảm chi phí và gia tăng sự hài lòng của khách hàng.

Phân tích yêu cầu:

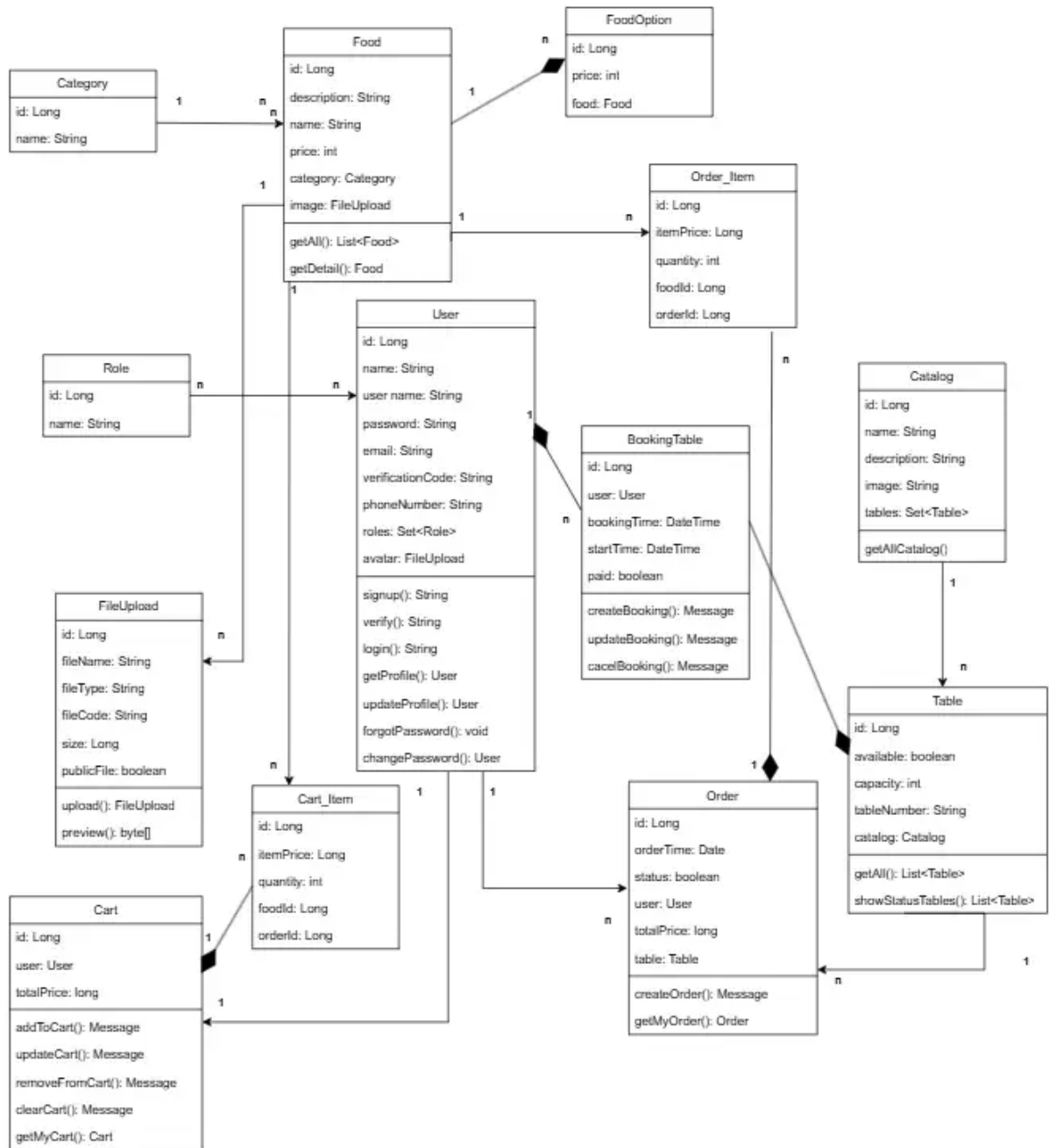
Dựa trên thông tin thu thập được, các yêu cầu được phân tích thành các yêu cầu chức năng và phi chức năng cụ thể:

1. Yêu cầu chức năng

- **Đặt bàn:** Cho phép khách hàng chọn và đặt bàn trực tuyến, cập nhật trạng thái bàn (trống, đã đặt, đang sử dụng).
- **Đặt món ăn:** Khách hàng có thể chọn món từ thực đơn, thêm vào giỏ hàng, chỉnh sửa số lượng, và xác nhận đơn hàng.
- **Thanh toán trực tuyến:** Hỗ trợ thanh toán qua thẻ, ví điện tử hoặc thanh toán tại nhà hàng.
- **Quản lý tài khoản:** Đăng ký, đăng nhập, cập nhật thông tin cá nhân và quản lý mật khẩu.

2.2. Thiết kế hệ thống:

Biểu đồ lớp cho mô hình miền:



Hình 2.2.1: Biểu đồ lớp cho mô hình miền

Biểu đồ lớp cho MVC:

- Model
 - User: Quản lý thông tin người dùng.
 - Role: Quản lý vai trò người dùng (ví dụ: "USER", "EMPLOYEE", "ADMIN").

- FileUpload: Thông tin các tệp tin được đăng tải trong ứng dụng.
- Food: Thông tin về món ăn, bao gồm tên, mô tả, giá, v.v.
- FoodOption: Thông tin về những lựa chọn thêm cho món ăn.
- Cart: Quản lý giỏ hàng của người dùng.
- CartItem: Đối tượng món ăn trong giỏ hàng.
- Category: Phân loại các món ăn.
- Catalog: Quản lý danh mục bàn.
- Table: Thông tin về các bàn trong nhà hàng.
- BookingTable: Thông tin đặt bàn.
- Order: Thông tin đơn đặt hàng.
- OrderItem: Thông tin món ăn trong đơn đặt.
- View:
 - activity_login.xml: Class hiển thị giao diện đăng nhập.
 - activity_main.xml: Class hiển thị giao diện trang chủ với menu.
 - activity_on_board_page.xml: Class hiển thị giao diện chọn đăng ký hay đăng nhập khi vào hồ sơ.
 - activity_otp.xml: Class hiển thị giao diện lấy mã và xác thực otp.
 - activity_sign_in.xml: Class hiển thị giao diện đăng ký.
 - activity_thankyou.xml: Class hiển thị thông báo cảm ơn sau khi thanh toán hàng.
 - detai_food.xml: Giao diện này sẽ bao gồm tên món ăn, giá, hình ảnh, mô tả, và các tùy chọn như số lượng và nút để thêm vào giỏ hàng.
 - fragment_cart.xml:
 - Hiển thị tổng quan số lượng món, lựa chọn thêm cho từng món, tổng quan giá, giúp người dùng biết chính xác số tiền cần thanh toán.
 - Thêm tính năng lưu giỏ hàng để người dùng có thể quay lại mua sau.
 - Thêm tính năng làm trống giỏ để người dùng có thể.
 - fragment_catalog.xml:
 - Hiển thị chi tiết 1 danh mục bàn, bao gồm tên, hình ảnh và số lượng người dùng khác.
 - Cho phép tìm kiếm để tìm ra danh mục bàn phù hợp.

- fragment_changepassword.xml:
 - Giao diện đổi mật khẩu mới cho người dùng.
- fragment_edit_profile.xml:
 - Thêm tùy chọn cho người dùng cập nhật ảnh đại diện.
 - Cho phép người dùng cập nhật thông tin liên lạc như họ tên và email để đảm bảo tính chính xác.
- fragment_food_detail.xml:
 - Hiển thị tổng quan món ăn, chi tiết món ăn, giá tiền, lựa chọn thêm món ăn liên quan, tổng quan giá.
 - Thêm tính năng thêm vào giỏ hàng
- fragment_food_options.xml:
 - Hiện thị món ăn, số lượng món ăn và giá tiền
- fragment_home.xml:
 - Gồm phần gợi ý các món ăn cho khách hàng lựa chọn
- fragment_profile.xml
 - Giao diện hiển thị thông tin cá nhân của người dùng, bao gồm tên, ảnh đại diện, email.
- fragment_recent.xml
 - Hiện thị danh sách các mục mà người dùng đã xem gần đây.
 - Cho phép người dùng truy cập nhanh vào các mục đã xem, giúp tiết kiệm thời gian tìm kiếm lại.
- fragment_search.xml:
 - Giao diện tìm kiếm để người dùng nhập từ khóa và tìm các mục trong hệ thống.
- fragment_table.xml
 - Giao diện sơ đồ bàn của nhà hàng, hiển thị tình trạng bàn trống hoặc đã đặt.
- fragment_order.xml: Hiện thị danh sách các đơn hàng. Giao diện này sẽ bao gồm một danh sách các đơn hàng, với các thông tin cơ bản như mã đơn hàng, ngày đặt, trạng thái, và tổng tiền.
- Controller:

- LoginActivity: Class tiếp nhận thông tin từ activity_login.xml sau đó lấy dữ liệu từ API để trả kết quả đăng nhập.
- SignupActivity: Class tiếp nhận thông tin từ activity_sign_in.xml sau đó lấy dữ liệu từ API để trả kết quả đăng ký vào cơ sở dữ liệu.
- OtpActivity: Class gửi OTP khi nhận yêu cầu từ máy chủ sau đó xác thực với thông tin từ activity_otp.xml.
- ThankyouActivity: Class gửi thông báo cảm ơn khi nhận được yêu cầu.
- CartFragment: Điều khiển các thao tác như thêm, xóa sản phẩm và tính tổng tiền. Đồng thời, gọi đến API để lưu, làm trống giỏ hàng khi cần.
- CatalogFragment: Điều khiển việc lấy dữ liệu từ API và cung cấp cho View. Đồng thời, xử lý các thao tác người dùng như tìm kiếm, lọc danh mục bán.
- ChangePasswordFragment: Kiểm tra tính hợp lệ của mật khẩu cũ, tính xác thực của mật khẩu mới và khớp của trường xác nhận. Xác thực mật khẩu cũ qua API và cập nhật mật khẩu mới nếu hợp lệ.
- EditProfileFragment: Điều khiển việc kiểm tra tính hợp lệ của dữ liệu, ví dụ như định dạng email, họ tên người dùng, ảnh đại diện và xử lý yêu cầu cập nhật thông tin qua API.
- FoodDetailFragment: Class gọi API để lấy thông tin chi tiết món ăn và hiển thị lên giao diện. Đồng thời, xử lý các thao tác của người dùng như chọn số lượng món ăn và thêm vào giỏ hàng.
- FoodOptionsBottomSheet: Class hiển thị dưới dạng pop-up từ cuối màn hình, cho phép người dùng tùy chọn các thuộc tính của món ăn, chẳng hạn như kích cỡ, loại topping, hoặc số lượng trước khi thêm vào giỏ hàng.
- HomeFragment: Class hiển thị giao diện chính của ứng dụng, nơi người dùng có thể xem danh sách các món ăn nổi bật, danh mục món ăn, hoặc các khuyến mãi hiện có.
- PaymentFragment: Class cho phép người dùng xem tổng quan đơn hàng và tiến hành thanh toán, bao gồm việc chọn phương thức thanh toán và xác nhận thông tin thanh toán.
- ProfileFragment: Quản lý và hiển thị giao diện thông tin cá nhân của người dùng, bao gồm các chi tiết như tên, ảnh đại diện, và thông tin liên hệ.

- RecentFragment: Hiện thị danh sách các mục mà người dùng đã xem gần đây từ API
- SearchFragment: Cung cấp công cụ tìm kiếm cho người dùng để tìm và lọc các mục trong hệ thống. Cho phép người dùng nhập từ khóa, chọn bộ lọc và hiện thị kết quả tìm kiếm theo tên món ăn.
- TableFragment: Cung cấp sơ đồ bàn chi tiết cho nhà hàng, giúp quản lý và sắp xếp chỗ ngồi hiệu quả. Fragment này hỗ trợ tính năng đặt chỗ trước, xem tình trạng bàn trống hoặc đã được đặt, giúp quản lý đặt chỗ thuận tiện hơn cho cả nhân viên và khách hàng.
- OrderFragment: Dùng để hiện thị danh sách các đơn hàng của người dùng, bao gồm thông tin về trạng thái đơn hàng, chi tiết các món ăn, và các thao tác như hủy đơn hoặc xem chi tiết đơn hàng.

Dựa trên kiến trúc MVC đã chọn, ta có thể xác định các lớp sau:

- Lớp User:
 - Thuộc tính
 - Long: id
 - name: String
 - username: String
 - password: String
 - email: String
 - verificationCode: String
 - phoneNumber: String
 - roles: Set<Role>
 - avatar: FileUpload
 - Phương thức
 - signup(): User
 - verify(): String
 - login(): String
 - getProfile(): User
 - updateProfile(): User
 - forgotPassword(): void
 - changePassword(): User
- Lớp Role:
 - Thuộc tính:
 - Long: id.
 - String: name.
 - Phương thức:
 - Getters và setters cho các thuộc tính.

- toString(): String
- Lớp FileUpload:
 - Thuộc tính:
 - Long: id
 - String: fileName
 - String: fileType
 - String: fileCode
 - Long: size
 - boolean: publicFile
 - Phương thức:
 - Getters và setters cho các thuộc tính.
 - isPublicFile(): boolean
 - setPublicFile(boolean publicFile): void
 - upload(): FileUpload
 - preview(): byte[]
 - toString(): String
- Lớp Food:
 - Thuộc tính:
 - Long: id
 - String: description
 - String: name
 - Int: price
 - Category: category
 - FileUpload: image
 - Phương thức:
 - Getters và setters cho các thuộc tính.
 - fetchFoodDetail(): Lấy thông tin đồ ăn
- Lớp FoodOption:
 - Thuộc tính:
 - Long: id
 - String: name
 - int: price
 - Food: food
 - Phương thức:
 - Getters và setters cho các thuộc tính.
- Lớp Category:
 - Thuộc tính:
 - Long: id
 - String: name
 - Phương thức:
 - Getters và setters cho các thuộc tính.

- Lớp Catalog:
 - Thuộc tính:
 - Long id
 - String name
 - String description
 - String image
 - Set<Table>
 - Phương thức:
 - getAllCatalog(String keyword): Lấy ra danh sách danh mục bàn theo từ khóa.
- Lớp Table:
 - Thuộc tính:
 - Long id
 - boolean available
 - int capacity
 - String tableNumber
 - Catalog catalog
 - Phương thức:
 - showStatusTables(Long catalogId, LocalDateTime startTime, int additionalTime): Lấy ra danh sách bàn cùng trạng thái (còn trống hoặc đã đặt) của bàn trong catalog và thời gian đã chọn.
- Lớp BookingTable:
 - Thuộc tính:
 - Long id
 - User user
 - LocalDateTime bookingTime
 - LocalDateTime startTime
 - boolean paid
 - Phương thức:
 - createBooking(List<Long> tableIds, LocalDateTime startTime, int additionalTime): Tạo đơn đặt bàn (có thể chọn một hoặc nhiều bàn), thời gian cho phép (nếu không nhập additionalTime) là 2 tiếng.
- Lớp Cart:
 - Thuộc tính
 - Long: id
 - User: user
 - Long: totalPrice
 - Phương thức
 - addToCart(Long foodId, int quantity, foodOptionIds): Xử lý yêu cầu thêm món vào giỏ hàng, xác thực người dùng bằng token và trả về kết quả cho người dùng

- `updateCart(Long foodId, int quantity, foodOptionIds)`: Xử lý yêu cầu cập nhật thông tin trong giỏ hàng của người dùng đã xác thực và trả về kết quả sau khi cập nhật
 - `removeFromCart()`: Message
 - `clearCart()`: Message
 - `getMyCart()`: Cart
- Lớp `Cart_Item`:
 - Thuộc tính
 - Long: id
 - Long: itemPrice
 - int: itemPrice
 - Long: foodId
 - Long: orderId
- Lớp `LoginActivity`:
 - Thuộc tính:
 - `EditText`: đối tượng `etUsername`.
 - `EditText`: đối tượng `etPassword`.
 - `Button`: đối tượng `btnLogin`.
 - `TextView`: Đối tượng `tvSignup`.
 - Phương thức:
 - `login(LoginForm loginForm)`: Xử lý thông tin đăng nhập thông qua API.
- Lớp `SignupActivity`: Class tiếp nhận thông tin từ `activity_sign_in.xml` sau đó lấy dữ liệu từ API để trả kết quả đăng ký vào cơ sở dữ liệu.
 - Thuộc tính:
 - `EditText fulNameInput` : đối tượng `fulNameInput`.
 - `EditText userNameInput` : đối tượng `userNameInput` .
 - `EditText emailInput` : đối tượng `emailInput` .
 - `EditText phoneInput` : đối tượng `phoneInput` .
 - `EditText passwordInput` : đối tượng `passwordInput` .
 - `EditText confirmPasswordInput` : đối tượng `confirmPasswordInput` .

- Button registerButton : đối tượng registerButton .
 - FirebaseAuth mAuth: đối tượng mAuth.
- Phương thức:
 - validateInputs(): Kiểm tra nội dung hợp lệ khi nhập vào EditText.
 - register(): Xử lý thông tin đăng ký thông qua API
- Lớp OtpActivity:
 - Thuộc tính:
 - TextView textViewCountdown: đối tượng textViewCountdown,
 - CountdownTimer countDownTimer: đối tượng countDownTimer,
 - Long timeLeftInMillis :đối tượng timeLeftInMillis
 - TextView textViewStartCountdown: đối tượng textViewStartCountdown
 - Phương thức:
 - setupOtpInputs(EditText... otpDigits): Tự động đổi EditText khi nhập từng số của OTP.
 - startCountdown(): Tự động đếm ngược thời gian hết hạn OTP từ khi gửi.
 - updateCountdownText(): Hiển thị thời gian đếm ngược của OTP.
- Lớp ThankyouActivity:
 - Thuộc tính:
 - Button buttonOk: đối tượng buttonOk.
 - Phương thức:
 - onCreate(Bundle savedInstanceState): Trả về giao diện thông báo cảm ơn khi gửi yêu cầu thông qua button.
- Lớp CartFragment:
 - Thuộc tính:
 - RecyclerView recyclerViewCart: đối tượng recyclerViewCart để hiển thị thông tin những gì trong giỏ hàng.
 - TextView tvEmptyCart, tvAddMoreItems: đối tượng tvEmptyCart, tvAddMoreItems

- private Button btnClear, btnContinue: đối tượng btnClear cho việc làm trống giỏ hàng, btnContinue cho việc xác nhận giỏ, tiến tới bước tiếp theo.
- Phương thức:
 - getCart(): lấy thông tin giỏ hàng
 - clearCart(): làm trống giỏ hàng
- Lớp CatalogFragment:
 - Thuộc tính:
 - RecyclerView rcvCatalogResults: đối tượng rcvCatalogResults để hiển thị danh sách danh mục bàn.
 - TextView tvNotFoundCatalog: đối tượng tvNotFoundCatalog để thông báo nếu không có danh mục bàn phù hợp.
 - EditText edtSearchCatalog: đối tượng edtSearchCatalog để nhập từ khóa cần tìm.
 - List<Catalog> catalogList: đối tượng catalogList để lấy danh sách các Catalog từ API.
 - CatalogAdapter catalogAdapter: đối tượng catalogAdapter được sử dụng để liên kết dữ liệu catalog với RecyclerView.
 - OnCatalogSelectedListener callback: đối tượng callback được sử dụng để giao tiếp với activity hoặc fragment cha khi một mục catalog được chọn.
 - Phương thức:
 - fetchCatalogs(String keyword): Lấy ra danh sách danh mục bàn theo từ khóa được nhập.
- Lớp EditProfileFragment:
 - Thuộc tính:
 - EditText nameEditText: đối tượng nameEditText để nhận họ tên mà người dùng nhập vào.
 - EditText emailEditText: đối tượng emailEditText để nhận email mà người dùng nhập vào.

- Button backButton: đối tượng backButton để quay lại fragment trước đó.
- Button saveButton: đối tượng saveButton để lưu lại thông tin người dùng vừa chỉnh sửa.
- ImageView profileImage: đối tượng profileImage để hiển thị ảnh đại diện của người dùng.
- File avatarFile: đối tượng avatarFile để nhận vào 1 tệp hình ảnh làm hình đại diện khác.
- Phương thức:
 - loadUserInfo(): lấy thông tin người dùng đang đăng nhập từ API và cập nhật lên giao diện.
 - updateProfile(): cập nhật thông tin người dùng.
- Lớp FoodDetailFragment:
 - Thuộc tính:
 - ViewPager2 viewPagerImages: đối tượng viewPagerImages cung cấp giao diện tương tác một cách mượt mà để người dùng duyệt qua nhiều hình ảnh của món ăn.
 - TextView tvDishName: đối tượng tvDishName hiển thị tên món ăn.
 - TextView tvDishDescription: đối tượng tvDishDescription hiển thị mô tả món ăn.
 - TextView tvPageIndicator: đối tượng tvPageIndicator hiển thị số trang hiện tại trên tổng số trang hình ảnh món ăn.
 - TextView tvFoodPrice: đối tượng tvFoodPrice hiển thị giá món ăn.
 - Button btnAddToCart: đối tượng btnAddToCart để thêm món ăn vào giỏ hàng.
 - RecyclerView recyclerViewRelatedDishes: đối tượng để xem những món ăn tương tự.
 - Phương thức:
 - fetchFoodDetails(): lấy ra thông tin chi tiết món ăn và danh sách các món liên quan từ API.

- displayFoodDetails(Food food): lấy và hiển thị danh sách các hình ảnh chi tiết của món ăn và hiển thị trên viewpage2.
- Lớp ProfileFragment:
 - Thuộc tính:
 - TextView editProfileText: đối tượng editProfileText.
 - ImageView profileImage: đối tượng profileImage.
 - TextView name: đối tượng name.
 - TextView email: đối tượng email.
 - ImageButton editProfileButton: đối tượng editProfileButton.
 - ListView settingsList: đối tượng settingsList.
 - Button logoutButton: đối tượng logoutButton
 - Phương thức:
 - getUserInfo(): Lấy thông tin người dùng từ API và cập nhật giao diện.
- Lớp RecentFragment:
 - Thuộc tính
 - RecyclerView recyclerView: đối tượng recyclerViewFoodHistory.
 - FoodHistoryAdapter adapter: đối tượng adapter.
 - TabLayout tabLayout: đối tượng tabLayout.
 - Phương thức
 - loadFoodHistory(): Lấy dữ liệu lịch sử món ăn của người dùng từ API và cập nhật vào
- Lớp SearchFragment:
 - Thuộc tính
 - RecyclerView recyclerView: đối tượng recyclerViewSearchResults.
 - FoodAdapter foodAdapter: đối tượng foodAdapter.
 - List<Food> foodList: danh sách foodList.
 - EditText searchEditText: đối tượng searchEditText.
 - ChipGroup chipGroupCategories: đối tượng chipGroupCategories.
 - TextView tvNotFound: đối tượng tvNotFound
 - Phương thức
 - onAttach(Context context): Gán callback cho sự kiện chọn món ăn.

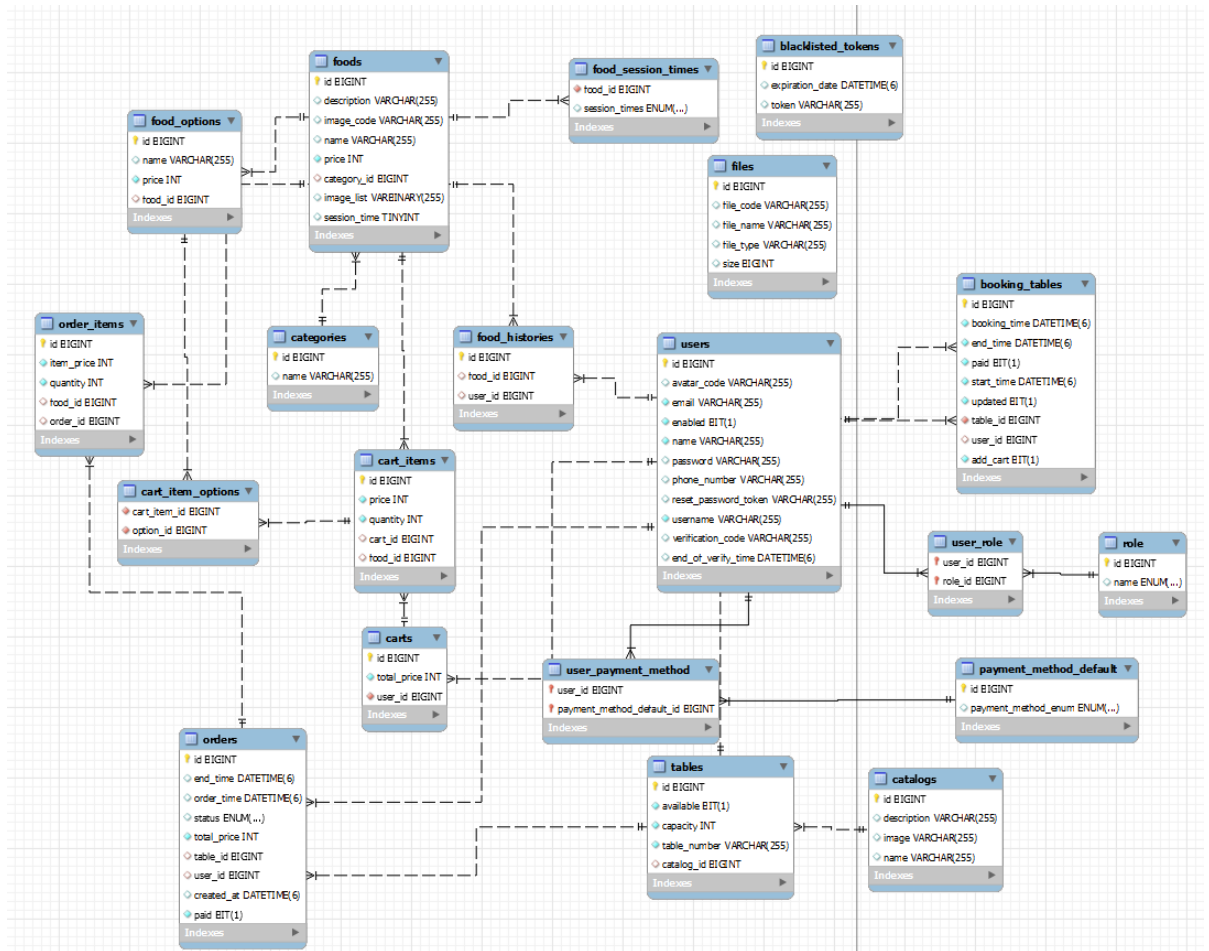
- searchFoods(String keyword): Gửi yêu cầu API để tìm kiếm các món ăn theo từ khóa và cập nhật kết quả vào RecyclerView
 - fetchCategories(): Lấy danh mục món ăn từ API và tạo Chip cho từng danh mục để người dùng có thể lọc món ăn theo danh mục.
- Lớp TableFragment:
 - Thuộc tính
 - Long catalogId: ID của danh mục bàn.
 - RecyclerView rcvTables: Đối tượng RecyclerView để hiển thị danh sách bàn.
 - TextView tvFreeTables: Đối tượng TextView để hiển thị số bàn trống.
 - TextView tvTotalTables: Đối tượng TextView để hiển thị tổng số bàn.
 - TextView tvTimeSelected: Đối tượng TextView để hiển thị thời gian đã chọn.
 - List<TableResponse> tableList: Danh sách các phản hồi bàn.
 - List<Long> selectedTableIds: Danh sách ID các bàn đã chọn.
 - List<String> selectedTableNumbers: Danh sách số bàn đã chọn.
 - Calendar calendar: Đối tượng Calendar để quản lý thời gian.
 - int selectedYear, selectedMonth, selectedDay, selectedHour, selectedMinute: Các thuộc tính để lưu trữ ngày và giờ đã chọn.
 - AlertDialog dialog: Đối tượng AlertDialog cho hộp thoại.
 - Toolbar toolbar: Đối tượng Toolbar.
 - Button btnConfirm: Đối tượng Button để xác nhận đặt bàn.
 - ImageView imgHelp: Đối tượng ImageView cho trợ giúp.
 - Spinner additionalTimeSpinner: Đối tượng Spinner cho thời gian bổ sung.
 - HashMap<String, Integer> additionalTimeMap: Bản đồ cho thời gian bổ sung.
 - int additionalTime: Thời gian bổ sung đã chọn.
 - StompClient stompClient: Đối tượng StompClient cho kết nối WebSocket.

- `List<TableResponse> tableListAfter`: Danh sách bàn sau khi cập nhật trạng thái.
- Phương thức
 - `onDestroy()`: Hủy kết nối WebSocket khi Activity bị hủy.
 - `showDateTimePicker()`: Hiện thị hộp thoại chọn ngày và giờ.
 - `getFormattedDateTime()`: Chuyển đổi ngày giờ thành định dạng `LocalDateTime`.
 - `shouldInterceptBackPressed()`: Kiểm tra điều kiện xử lý nút back.
 - `fetchTables()`: Gửi yêu cầu API để lấy trạng thái bàn theo thời gian đã chọn.
 - `updateTableStatus()`: Cập nhật trạng thái bàn sau khi gửi yêu cầu.
 - `onTableSelectionChanged(List<Long>selectedTableIds,List<String>selectedTableNumbers)`: Cập nhật danh sách các bàn đã chọn.
 - `createBookingTable(boolean addCart)`: Gửi yêu cầu API để tạo đặt bàn.
 - `nextToCreateOrder()`: Chuyển tiếp đến trang đặt hàng.
 - `callApiToDeclineItems()`: Gửi yêu cầu để từ chối món ăn.
 - `callApiToAddItemsToCart()`: Gửi yêu cầu để thêm món ăn vào giỏ hàng.
 - `goToCartFragment(BookingTableRequest bookingRequest)`: Chuyển đến `CartFragment` với yêu cầu đặt bàn.
- Lớp `OrderFragment`:
 - Thuộc tính:
 - `RecyclerView rvDishes`: đối tượng `recyclerViewDish` để hiển thị thông tin món ăn đã chọn trong giỏ hàng.
 - `RecyclerView rvTables`: đối tượng `recyclerViewTables` để hiển thị thông tin bàn đã chọn trong giỏ hàng.
 - `TextView tvTotalAmount`: đối tượng `tvTotalAmount`
 - `private Button btnConfirmOrder`: đối tượng `btnConfirmOrder` cho phép quay lại trang chủ

Thiết kế cơ sở dữ liệu:

- Tạo một database mysql có tên “restaurant555” để lưu trữ dữ liệu.
- Tạo các bảng để lưu trữ dữ liệu liên quan.

Lược đồ mô hình quan hệ:



Hình 2.2.2: Lược đồ mô hình quan hệ

Thiết kế giao diện:

- Hiện thị trang chủ với menu chính các lựa chọn:
 - Hiện thị đặt đồ ăn.
 - Hiện thị đặt bàn.
 - Xem giỏ hàng.
 - Tìm kiếm.
 - Hồ sơ cá nhân.
 - Phương thức thanh toán.
 - Đã xem gần đây.

- Nếu chưa đăng nhập, ở trang hồ sơ sẽ có đăng nhập và đăng ký.
- Khi vào phần hồ sơ sau khi đăng nhập ta có thể chọn các mục:
 - Chỉnh sửa hồ sơ.
 - Đổi mật khẩu
 - Đăng xuất.
- Ở màn hình hiển thị chi tiết đồ ăn ngoài bao gồm thông tin, có thể thêm trực tiếp món ăn vào giỏ hàng.
- Tại màn hình đặt bàn, có các lựa chọn.
- Màn hình đơn hàng cho phép người dùng xem thông tin các đơn hàng đã tạo như trạng thái, các món hoặc bàn (nếu có).
- Khi vào màn hình phương thức thanh toán, ta có thể thêm phương thức thanh toán.
- Minh họa: <https://www.figma.com/design/fAzxfJr4CJvkpuZIEesE2/555?node-id=1-936>

2.3. Triển khai:

- Viết code:
 - Thiết lập cấu trúc dự án
Tạo các thư mục cần thiết cho mã nguồn, tài nguyên giao diện người dùng (UI), và cấu hình ứng dụng.
Cấu hình các thư viện và phụ thuộc cần thiết như Retrofit cho API.
 - Xây dựng giao diện người dùng (UI)
Thiết kế các màn hình chính của ứng dụng, bao gồm màn hình đặt bàn, chọn món, giỏ hàng và thanh toán.
Sử dụng XML để tạo các layout và áp dụng Java để quản lý tương tác với người dùng, đảm bảo giao diện thân thiện và dễ sử dụng.
 - Xây dựng backend web service
Sử dụng spring boot để viết các rest api và mysql cho việc lưu trữ dữ liệu.
 - Kết nối với API
Kết nối ứng dụng với cơ sở dữ liệu để lưu trữ và truy xuất thông tin như danh sách món ăn và thông tin đặt bàn,...

Thiết lập các API để lấy dữ liệu từ server, đồng bộ hóa thông tin từ hệ thống nhà hàng và xử lý các yêu cầu đặc biệt là yêu cầu đặt bàn từ người dùng.

- Tối ưu hóa hiệu năng và kiểm tra lỗi

Tối ưu mã nguồn, giảm thiểu thời gian tải và xử lý dữ liệu để tăng tốc độ phản hồi của ứng dụng.

- Kiểm thử: Viết các test case để kiểm tra các chức năng của ứng dụng, đảm bảo ứng dụng hoạt động đúng theo yêu cầu.

2.4. Vận hành và bảo trì:

Cài đặt và triển khai:

- Đảm bảo ứng dụng được triển khai đầy đủ các tính năng, từ đặt bàn, quản lý giỏ hàng đến xử lý thanh toán.
- Cung cấp hướng dẫn chi tiết cho quản trị viên về cách cài đặt, cấu hình và quản lý hệ thống.

Bảo trì hệ thống:

- Theo dõi hiệu suất của hệ thống định kỳ để đảm bảo tính ổn định và phát hiện các sự cố tiềm ẩn.
- Cập nhật các tính năng và sửa lỗi định kỳ để nâng cao trải nghiệm người dùng.
- Đảm bảo bảo mật dữ liệu người dùng bằng cách sử dụng các phương pháp mã hóa và xác thực chặt chẽ.

Hỗ trợ người dùng:

- Cung cấp tài liệu hướng dẫn sử dụng cho khách hàng
- Xây dựng kênh hỗ trợ trực tuyến để giải quyết các thắc mắc và sự cố mà người dùng gặp phải trong quá trình sử dụng ứng dụng.

Nâng cấp hệ thống:

- Đánh giá định kỳ các yêu cầu mới từ người dùng để cập nhật và bổ sung tính năng.

- Cải thiện giao diện và luồng công việc dựa trên phản hồi thực tế từ khách hàng và nhân viên nhà hàng.

CHƯƠNG 3. KẾT QUẢ THỰC HIỆN

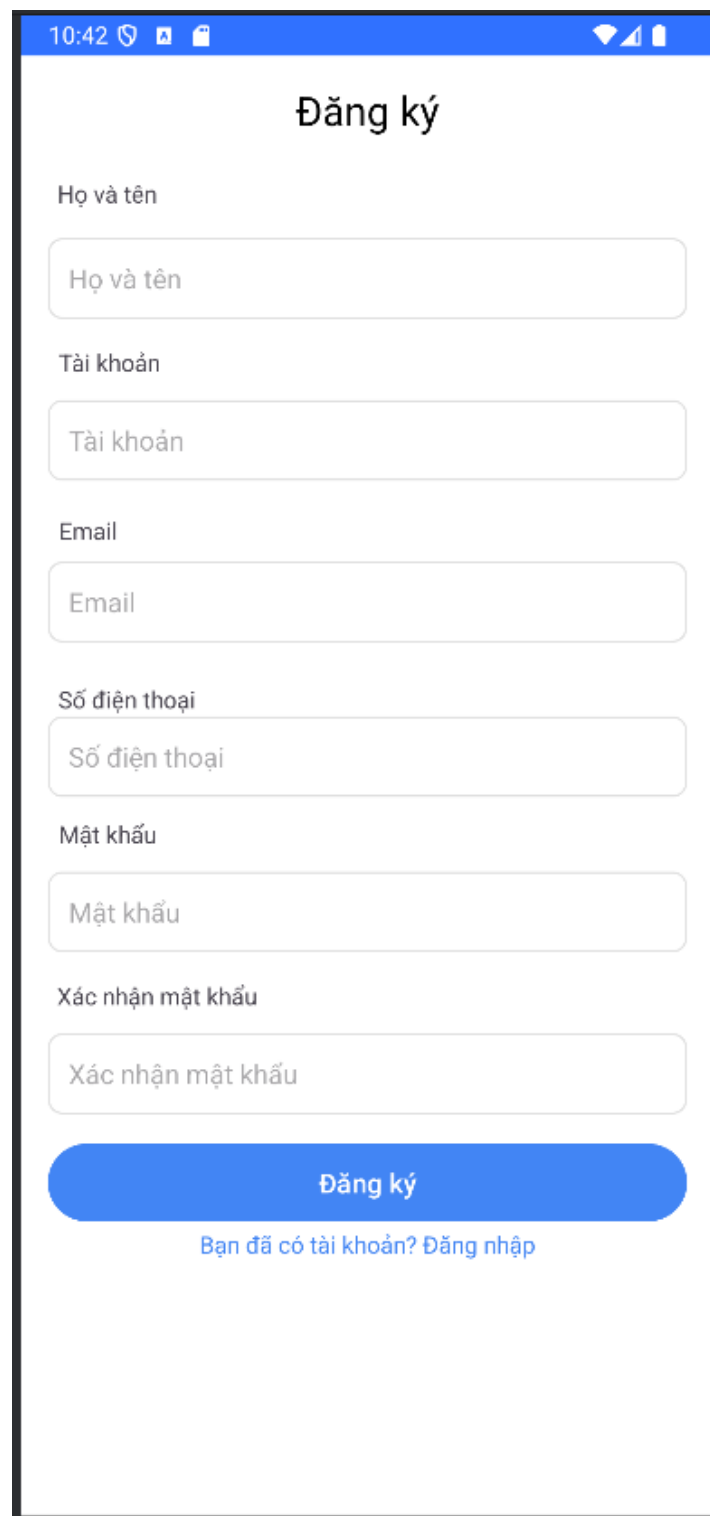
3.1. Công nghệ đã sử dụng

- Ngôn ngữ lập trình: Java
- Công cụ: IDE Android Studio
- Thư viện:
 1. Lombok (org.projectlombok:lombok:1.18.34): Giúp đơn giản hóa mã nguồn bằng cách tự động tạo các phương thức getter, setter, constructor và các phương thức khác thông qua các annotation.
 2. Gson (com.google.code.gson:gson:2.11.0): Thư viện của Google giúp chuyển đổi dữ liệu JSON thành các đối tượng Java và ngược lại.
 3. Retrofit (com.squareup.retrofit2:retrofit:2.11.0): Thư viện của Square để gọi các API RESTful một cách dễ dàng, hỗ trợ các thao tác HTTP.
 4. Glide (com.github.bumptech.glide:glide:4.16.0): Thư viện dùng để tải và hiển thị hình ảnh hiệu quả trong ứng dụng Android.
 5. ViewPager2 (androidx.viewpager2:viewpager2:1.1.0): Thành phần UI cho phép người dùng vuốt giữa các trang khác nhau, được cải tiến từ ViewPager.
 6. Material Components (com.google.android.material:material:1.9.0): Thư viện chứa các thành phần giao diện theo tiêu chuẩn Material Design.
 7. CircleIndicator (me.relex:circleindicator:2.1.6): Thư viện để hiển thị các chấm tròn chỉ số trang trong ViewPager.
 8. OkHttp (com.squareup.okhttp3:okhttp:4.12.0): Thư viện HTTP của Square cho phép kết nối mạng một cách linh hoạt.
 9. StompProtocol (com.github.NaikSoftware:StompProtocolAndroid:1.6.6): Thư viện để làm việc với giao thức STOMP.

3.2. Tiến độ thực hiện

Link github tới dự án: https://github.com/dannhanong/CSE441_PROJECT

3.3. Hình ảnh sản phẩm



The image shows a mobile application interface for registration. At the top, there is a status bar with the time 10:42 and various icons. The main title 'Đăng ký' is centered at the top of the form. Below the title, there are six input fields, each with a label above it: 'Họ và tên', 'Tài khoản', 'Email', 'Số điện thoại', 'Mật khẩu', and 'Xác nhận mật khẩu'. Each input field contains a placeholder text matching its label. At the bottom of the form, there is a large blue button with the text 'Đăng ký'. Below this button, there is a link in blue text that says 'Bạn đã có tài khoản? Đăng nhập'.

10:42

Đăng ký

Họ và tên

Họ và tên

Tài khoản

Tài khoản

Email

Email

Số điện thoại

Số điện thoại

Mật khẩu

Mật khẩu

Xác nhận mật khẩu

Xác nhận mật khẩu

Đăng ký

[Bạn đã có tài khoản? Đăng nhập](#)

Hình 3.3.1. Giao diện đăng ký

The image shows a mobile application screen for OTP verification. At the top, there is a blue status bar with the time 10:48 and various icons. The main content area has a white background. The title 'Xác thực OTP' is centered at the top in a large, bold, black font. Below the title, there is a paragraph of text: 'Vui lòng nhập mã số chúng tôi gửi cho bạn qua số điện thoại'. Underneath this, a smaller line of text indicates the validity of the code: 'Mã xác thực có tác dụng trong 02:51'. Below this text is a horizontal line with six dashes, representing the input field for the OTP code. Below the input field is a large, rounded blue button with the text 'Xác thực OTP' in white. At the bottom of the screen, there is a line of text: 'Bạn chưa nhận được mã? [Gửi lại?](#)'.

10:48

Xác thực OTP

Vui lòng nhập mã số chúng tôi gửi cho bạn qua số điện thoại

Mã xác thực có tác dụng trong 02:51

Xác thực OTP

Bạn chưa nhận được mã? [Gửi lại?](#)

Hình 3.3.2. Giao diện xác thực OTP

10:43

Đăng nhập

Tài khoản

Tài khoản

Mật khẩu

Mật khẩu

Đăng nhập

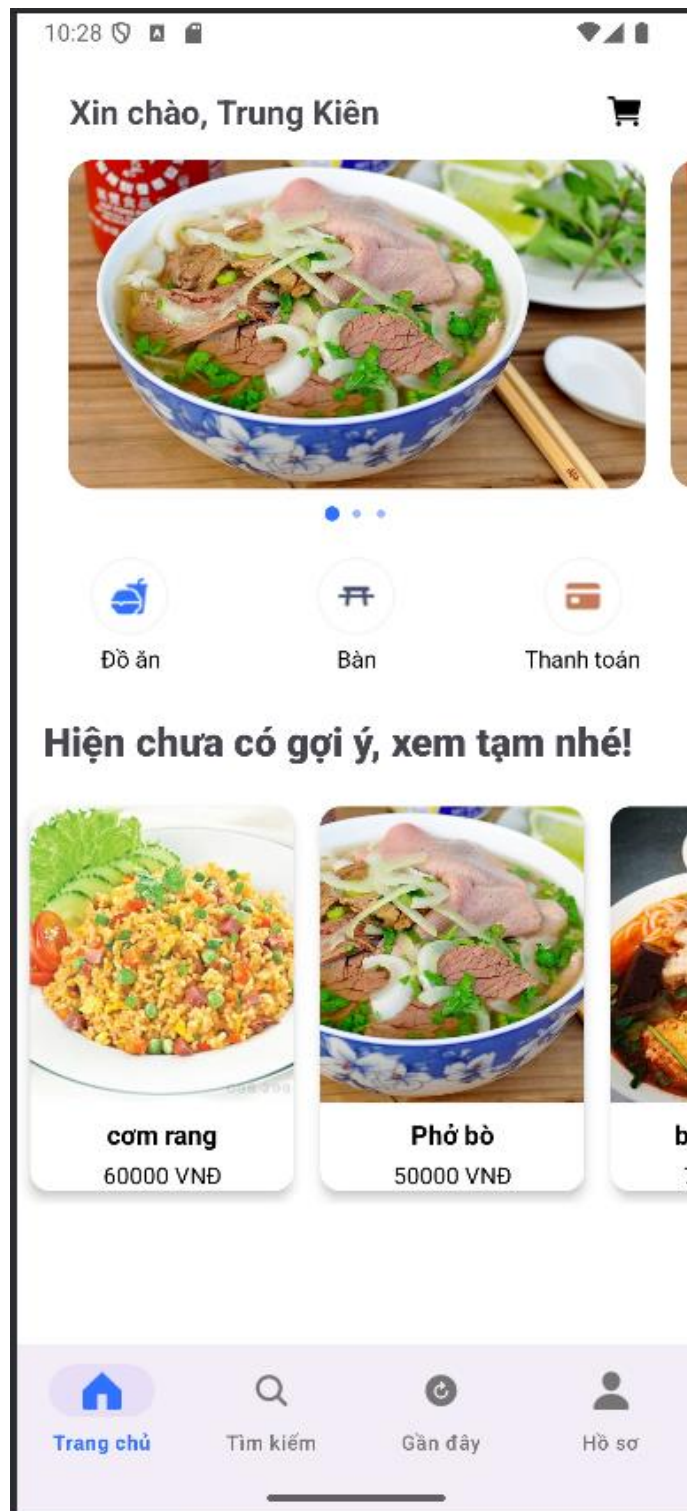
Đăng nhập bằng phương thức khác

f

G

Bạn chưa có tài khoản? [Đăng ký](#)

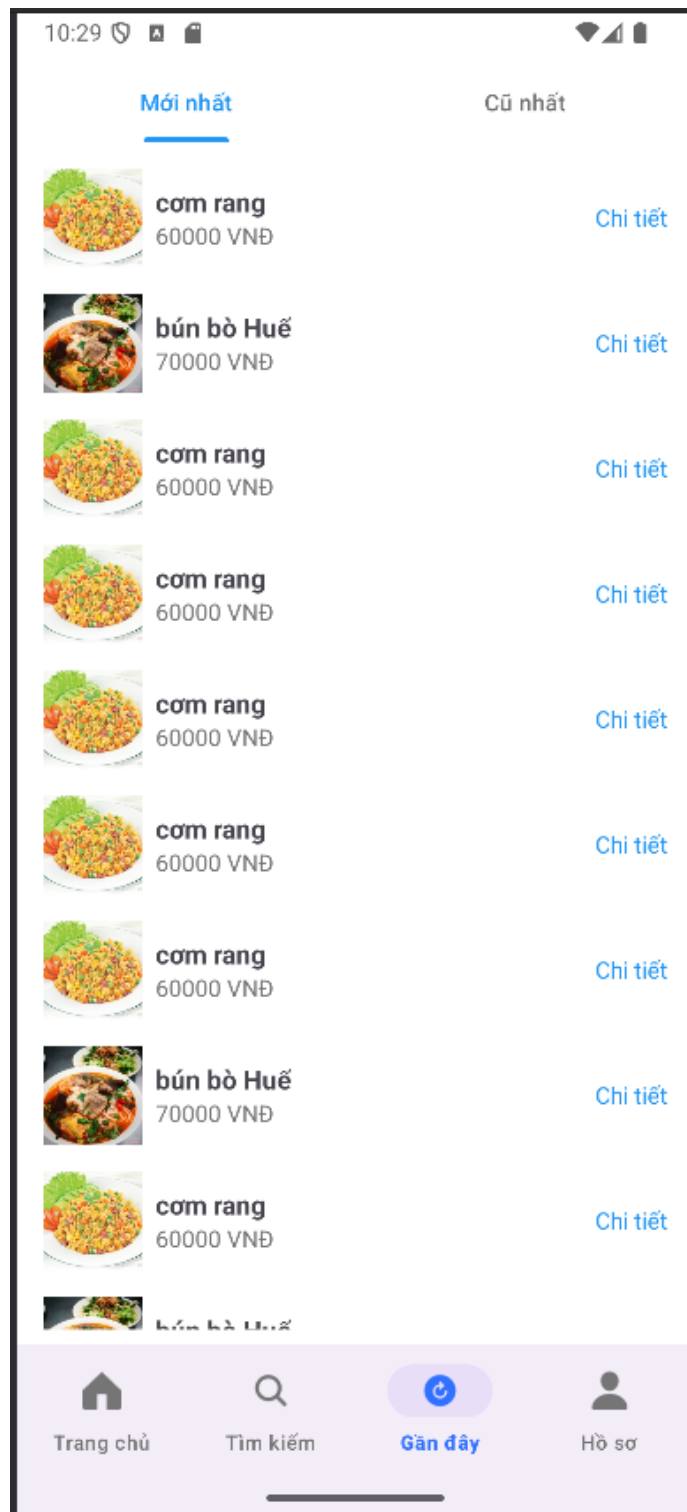
Hình 3.3.3. Giao diện đăng nhập



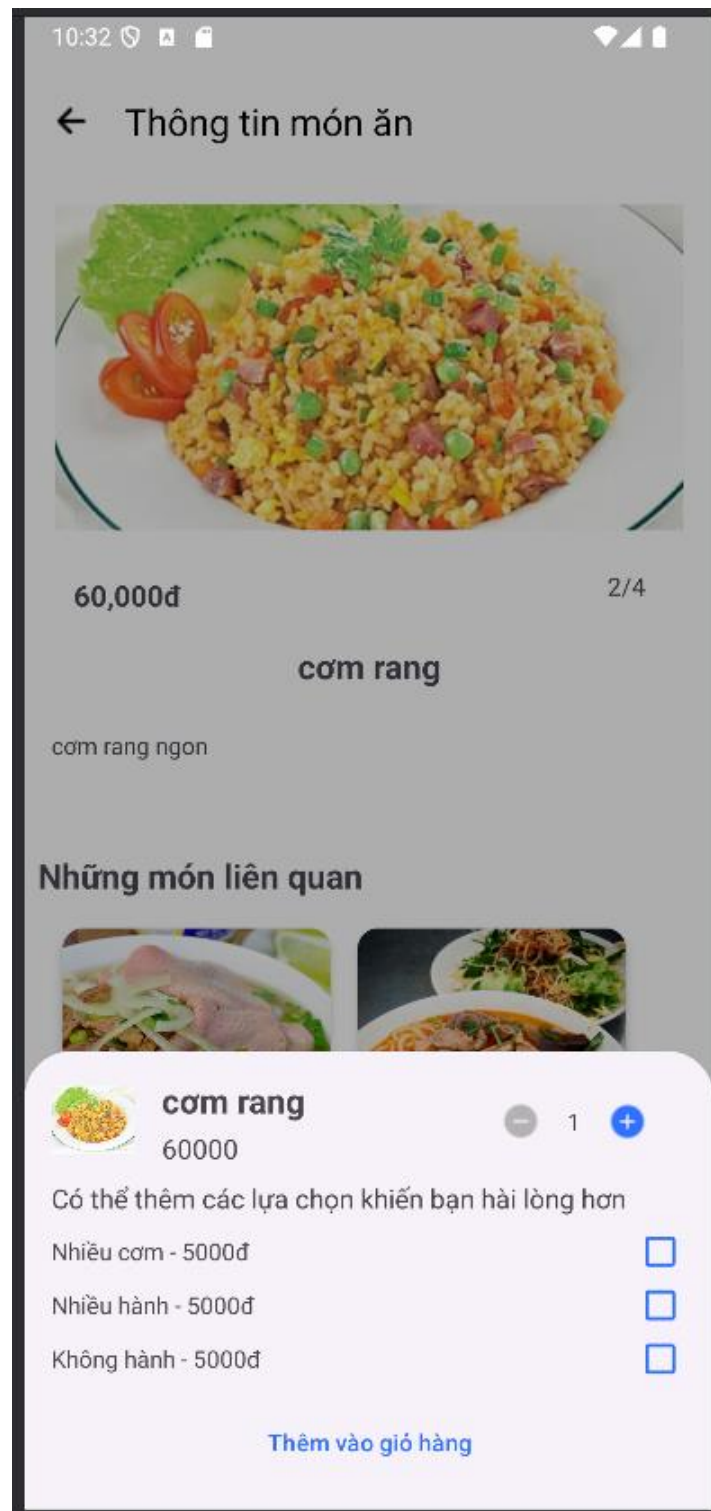
Hình 3.3.4. Giao diện trang chủ



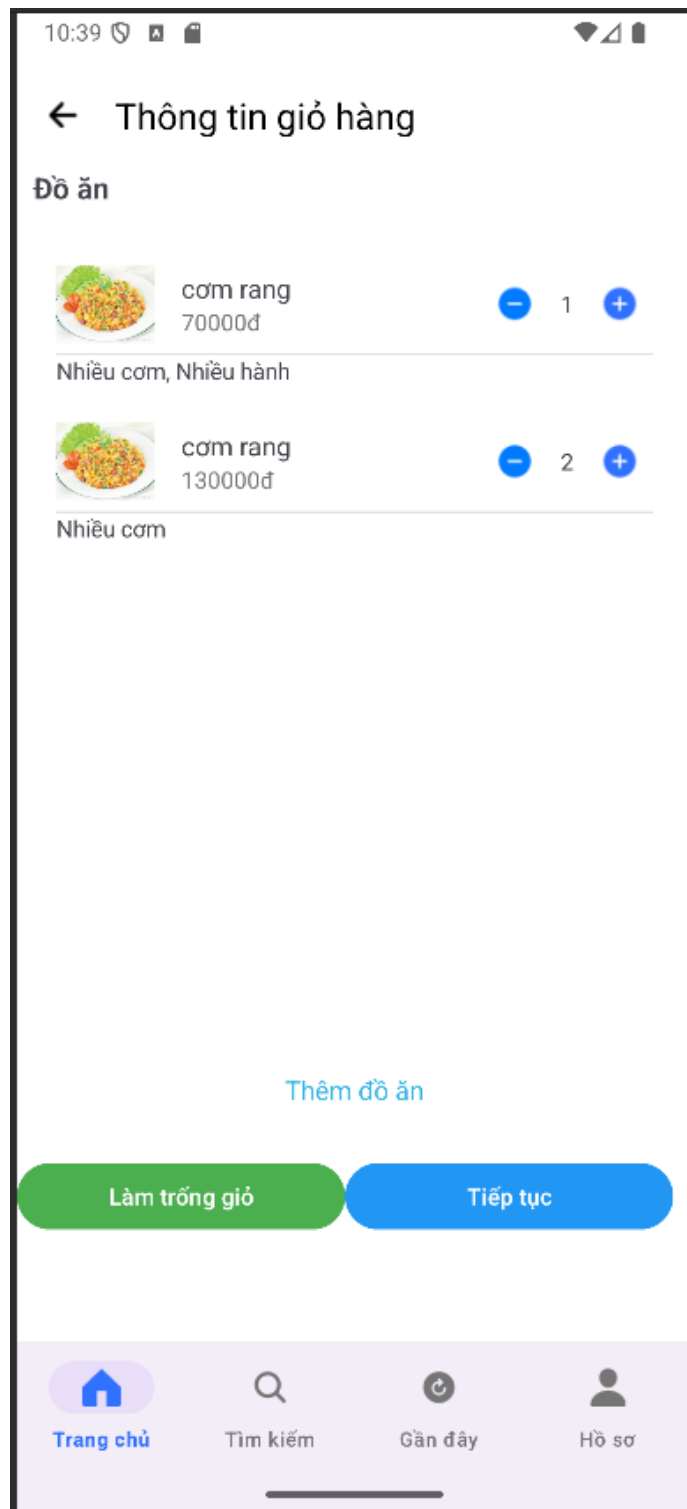
Hình 3.3.5. Giao diện tìm kiếm món ăn



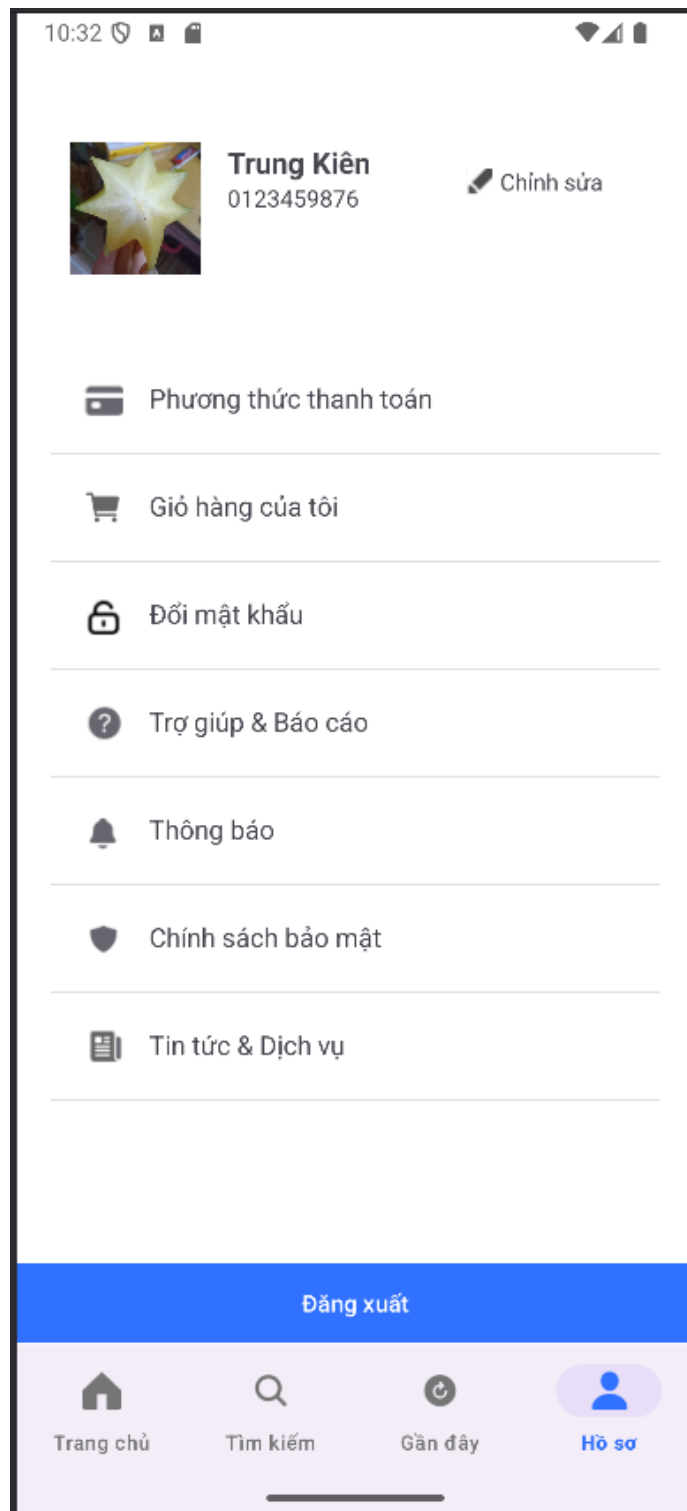
Hình 3.3.6. Giao diện đã xem gần đây



Hình 3.3.7. Giao diện chi tiết món ăn và thêm vào giỏ



Hình 3.3.8. Giao diện giỏ hàng



Hình 3.3.9. Giao diện hồ sơ người dùng

10:38

←

Thông tin bàn

Thời gian chọn: 1/11/2024 13:34

Đặt thêm: 1 giờ

▼

Bàn trống: 10/10

1

Trống

2

Trống

3

Trống

4

Trống

5

Đã chọn

6

Trống

7

Đã chọn

8

Trống

9

Trống

10

Trống

Hướng dẫn ?

Xác nhận

Trang chủ

Tìm kiếm

Gần đây

Hồ sơ

Hình 3.3.10. Giao diện đặt bàn

KẾT LUẬN

Ứng dụng đặt đơn của nhà hàng 555 trên Android mang lại nhiều tiện lợi cho khách hàng, giúp họ dễ dàng đặt bàn và gọi món từ xa mà không cần phải đến trực tiếp nhà hàng. Với giao diện thân thiện và dễ sử dụng, khách hàng có thể xem thực đơn, gợi ý, chọn món ăn yêu thích, đặt bàn trước, xem thông tin giỏ hàng cũng như đơn đã đặt chỉ trong vài thao tác. Ứng dụng cũng giúp giảm tải cho nhân viên nhà hàng vào những giờ cao điểm, đồng thời hạn chế các sai sót khi ghi nhận đơn hàng bằng tay, từ đó nâng cao hiệu quả phục vụ và chất lượng trải nghiệm khách hàng.

Tuy nhiên, ứng dụng cũng tồn tại một số hạn chế. Một số khách hàng có thể gặp khó khăn khi sử dụng nếu họ không quen với công nghệ hoặc không có thiết bị Android tương thích. Ngoài ra, khi nhu cầu sử dụng tăng cao, có khả năng ứng dụng sẽ gặp tình trạng quá tải nếu không được tối ưu hóa tốt về mặt hiệu năng và cơ sở hạ tầng. Việc bảo trì và cập nhật ứng dụng cũng cần thường xuyên thực hiện để khắc phục lỗi và cải tiến tính năng, nếu không sẽ dẫn đến trải nghiệm không ổn định và gây mất lòng tin ở khách hàng.

Trong tương lai, ứng dụng có thể phát triển thêm nhiều tính năng hiện đại như tích hợp AI để gợi ý món ăn theo sở thích của khách hàng, cung cấp chương trình tích điểm và ưu đãi cá nhân hóa để thu hút khách hàng trung thành. Nhà hàng cũng có thể mở rộng ứng dụng sang hệ điều hành khác để tăng lượng khách hàng tiếp cận. Bên cạnh đó, việc cải thiện hệ thống bảo mật để bảo vệ thông tin cá nhân của khách hàng và nâng cấp cơ sở hạ tầng để đáp ứng nhu cầu truy cập cao sẽ là hướng phát triển quan trọng, đảm bảo ứng dụng không chỉ tiện lợi mà còn an toàn và đáng tin cậy trong mắt khách hàng.

DANH MỤC TÀI LIỆU THAM KHẢO

- [1]. Thầy Kiều Tuấn Dũng, *Bộ giáo án lập trình Android cơ bản*, Đại học Thủy Lợi.
- [2]. Margaret Kozak Polk - Coding Android Apps (2024, Chapman and Hall_CRC).