

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



ĐỒ ÁN TỐT NGHIỆP
NGÀNH: CÔNG NGHỆ THÔNG TIN

ĐỀ TÀI:

**XÂY DỰNG ỨNG DỤNG ĐIỆN THOẠI
DI ĐỘNG QUẢN LÝ QUÁN CAFE
BÁN HÀNG ONLINE**

Người hướng dẫn: ThS. Nguyễn Thị Xuân Thủy
Nhóm sinh viên thực hiện: Nguyễn Toàn Thắng - 2151220172
Lê Trung Tư Thương - 2151220207
Lớp: 21CT4

Đà Nẵng, 12/2025

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



ĐỒ ÁN TỐT NGHIỆP
NGÀNH: CÔNG NGHỆ THÔNG TIN

ĐỀ TÀI:

XÂY DỰNG ỨNG DỤNG ĐIỆN THOẠI
DI ĐỘNG QUẢN LÝ QUÁN CAFE
BÁN HÀNG ONLINE

Người hướng dẫn:

Nhóm sinh viên thực hiện:

Lớp:

ThS. Nguyễn Thị Xuân Thủy

Nguyễn Toàn Thắng - 2151220172

Lê Trung Tư Thương - 2151220207

21CT4

Đà Nẵng, 12/2025

TÓM TẮT

Tên đề tài: **Xây dựng ứng dụng điện thoại di động quản lý quán cafe bán hàng Online.**

Nhóm sinh viên thực hiện: Nguyễn Toàn Thắng – Lê Trung Tư Thương

Lớp: 21CT4

Đề tài tập trung vào việc xây dựng một ứng dụng di động Android quản lý quán cafe bán hàng online, sử dụng ngôn ngữ lập trình Kotlin và kiến trúc MVVM. Hệ thống được thiết kế để hỗ trợ cả chủ quán và khách hàng với các chức năng chính như: đăng ký/đăng nhập, quản lý sản phẩm và menu, đặt hàng trực tuyến, quản lý giỏ hàng, thanh toán online (VNPay, Blockchain), áp dụng voucher, đánh giá sản phẩm, quản lý đơn hàng và theo dõi trạng thái giao hàng.

Ứng dụng sử dụng Firebase (Realtime Database, Authentication, Storage) để lưu trữ dữ liệu và xác thực người dùng, tích hợp Google Maps API để tính khoảng cách và phí vận chuyển, hỗ trợ thanh toán qua VNPAY và Blockchain (Ethereum Sepolia). Hệ thống còn bao gồm tính năng quản lý hạng thành viên dựa trên tổng chi tiêu, quản lý địa chỉ giao hàng, và giao diện quản trị dành cho chủ quán.

Ứng dụng quản lý quán cafe bán hàng online sẽ là công cụ hỗ trợ đắc lực cho cả chủ quán và khách hàng, giúp tự động hóa quy trình bán hàng, giảm thiểu công việc quản lý thủ công, nâng cao trải nghiệm người dùng và tối ưu hóa hiệu quả kinh doanh.

LỜI NÓI ĐẦU

Để hoàn thành được Đồ án tốt nghiệp này em xin chân thành cảm ơn quý thầy cô trong khoa Công Nghệ Thông Tin - Trường Đại Học Kiến Trúc Đà Nẵng, đặc biệt là cô **ThS. Nguyễn Thị Xuân Thủy** đã tận tình truyền đạt kiến thức trong những năm học vừa qua. Với vốn kiến thức được tiếp thu trong quá trình học không chỉ là nền tảng cho quá trình làm đồ án tốt nghiệp mà còn là hành trang quý báu để em bước vào đời một cách vững chắc và tự tin.

Em cũng xin gửi lời cảm ơn chân thành và sự tri ân sâu sắc của cô Nguyễn Thị Xuân Thủy trong suốt thời gian vừa qua đã nhiệt tình chỉ dạy, giúp đỡ em để có thể hoàn thành tốt đồ án tốt nghiệp.

Trong quá trình làm bài đồ án tốt nghiệp, do trình độ lý luận cũng như kinh nghiệm thực tiễn còn hạn chế nên không thể tránh khỏi những thiếu sót, em rất mong nhận được ý kiến đóng góp quý báu của các thầy cô để em có thể học hỏi thêm được nhiều kinh nghiệm, kiến thức.

Em xin trân trọng cảm ơn!

CAM ĐOAN

Em xin cam đoan kết quả đạt được trong đồ án tốt nghiệp là sản phẩm của riêng cá nhân dưới sự hướng dẫn của ThS. Nguyễn Thị Xuân Thủy, không sao chép lại của người khác. Trong toàn bộ nội dung của đồ án, những điều được trình bày hoặc là của bản thân hoặc là được tổng hợp từ nhiều nguồn tài liệu. Tất cả các tài liệu tham khảo đều có xuất xứ rõ ràng và được trích dẫn hợp pháp. Em xin hoàn toàn chịu trách nhiệm và chịu mọi hình thức kỷ luật theo quy định cho lời cam đoan của mình.

Sinh viên thực hiện

Nguyễn Toàn Thắng

Lê Trung Tư Thương

MỤC LỤC

TÓM TẮT	i
LỜI NÓI ĐẦU	ii
CAM ĐOAN	iii
MỤC LỤC.....	iv
DANH SÁCH HÌNH ẢNH	vii
DANH SÁCH HÌNH BẢNG	x
DANH SÁCH CÁC KÝ HIỆU, CHỮ VIẾT TẮT.....	xi
Chương 1. GIỚI THIỆU TỔNG QUAN.....	1
1.1. Lý do chọn đề tài	1
1.2. Lịch sử nghiên cứu	1
1.3. Mục tiêu đề tài.....	2
1.4. Đối tượng, phạm vi nghiên cứu	3
1.4.1. Đối tượng nghiên cứu	3
1.4.2. Phạm vi nghiên cứu	3
Chương 2. CƠ SỞ LÝ THUYẾT VÀ CÔNG CỤ XÂY DỰNG HỆ THỐNG	5
2.1. Cơ sở lý thuyết.....	5
2.1.1. Giới thiệu hệ điều hành Android	5
2.1.2. Firebase	10
2.2. Công cụ xây dựng.....	14
2.2.1. Android Studio	14
2.2.2. Kotlin.....	16
2.3. Tích hợp API và dịch vụ bên thứ ba	19
2.3.1. Firebase Authentication	19
2.3.2. Firebase Realtime Database	19
2.3.3. Google Distance Matrix API.....	20

2.3.4. CoinGecko API.....	20
2.3.5. VNPay SDK	20
2.3.6. Sepolia RPC API (Blockchain Payment)	21
2.3.7. Google Play Services Location	21
Chương 3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG THÔNG TIN	22
3.1. Phân tích yêu cầu hệ thống	22
3.1.1. Yêu cầu chức năng (Functional Requirements).....	22
3.1.2. Yêu cầu phi chức năng (Non-Functional Requirements).....	23
3.2. Biểu đồ use case.....	24
3.2.1. Biểu đồ Use Case tổng quát	24
3.2.2. Biểu đồ Use Case chi tiết.....	26
3.3. Biểu đồ hoạt động.....	37
3.3.1. Đặt hàng	37
3.3.2. Quản lý đơn hàng.....	38
3.3.3. Thanh toán	39
3.3.4. Đánh giá	40
3.3.5. Xem sản phẩm	41
3.3.6. Đăng ký/Đăng nhập	42
3.3.7. Quản lý sản phẩm (Admin).....	43
3.3.8. Quản lý đơn hàng (Admin)	44
3.3.9. Quản lý voucher	45
3.3.10. Quản lý danh mục	46
3.3.11. Quản lý topping	47
3.3.12. Phản hồi và thống kê.....	48
3.4. Biểu đồ tuần tự (Sequence Diagram)	49
3.4.1. Biểu đồ tuần tự đăng nhập	49
3.4.2. Biểu đồ tuần tự đăng ký.....	50
3.4.3. Biểu đồ tuần tự xem chi tiết đồ uống	51
3.4.4. Biểu đồ tuần tự thêm vào giỏ hàng	52
3.4.5. Biểu đồ tuần tự đặt hàng	54
3.4.6. Biểu đồ tuần tự thanh toán tổng quát	56
3.4.7. Biểu đồ tuần tự thanh toán VNPay.....	58
3.4.8. Biểu đồ tuần tự thanh toán blockchain.....	59

3.4.9. Biểu đồ tuần tự theo dõi đơn hàng.....	61
3.4.10. Biểu đồ tuần tự đánh giá đơn hàng.....	62
3.5. Thiết kế cơ sở dữ liệu	64
3.5.1. Biểu đồ ERD.....	64
3.5.2. Cấu trúc các bảng.....	65
3.6. Thiết kế bảo mật.....	70
3.6.1. Bảo mật xác thực người dùng	70
3.6.2. Bảo mật dữ liệu	70
3.6.3. Bảo mật API và giao tiếp.....	71
3.6.4. Bảo mật ứng dụng.....	72
3.6.5. Bảo mật dữ liệu local	72
3.6.6. Bảo mật giao tiếp mạng.....	72
3.6.7. Input Validation và Sanitization	73
3.6.8. Rate Limiting và DDoS Protection	73
3.6.9. Logging và Monitoring	73
3.6.10. Backup và Recovery	73
Chương 4. TRIỂN KHAI ỨNG DỤNG VÀ KẾT QUẢ ĐẠT ĐƯỢC	74
4.1. Môi trường phát triển và công cụ lập trình.....	74
4.2. Triển khai ứng dụng.....	74
4.3. Kết quả đạt được.....	75
4.3.1. Giao diện người dùng	75
4.3.2. Giao diện người quản trị.....	93
4.4. Kiểm thử hệ thống.....	95
4.4.1. Kiểm thử chức năng (Functional Testing).....	95
4.4.2. Kiểm thử hiệu năng (Performance Testing)	99
4.4.3. Kiểm thử bảo mật (Security Testing)	99
4.4.4. Kết quả kiểm thử tổng hợp.....	100
KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	101
TÀI LIỆU THAM KHẢO	103

DANH SÁCH HÌNH ẢNH

Hình 2.1 Hệ điều hành Android	6
Hình 2.1 Giao diện hệ điều hành Android	7
Hình 2.2 Kho ứng dụng đa dạng.....	8
Hình 2.3 Tối ưu hoá bộ nhớ RAM.....	9
Hình 2.4 Virus xâm nhập	9
Hình 2.5 Firebase	10
Hình 2.6 Realtime Database	11
Hình 2.7 Authentication.....	12
Hình 2.8 Cloud Firestore	12
Hình 2.9 Cloud Storage	14
Hình 2.10 Mô-đun và tệp tài nguyên	15
Hình 2.11 Kotlin	17
Hình 3.1 Sơ đồ chức năng đặt hàng.....	37
Hình 3.2 Sơ đồ chức năng quản lý đơn hàng.....	38
Hình 3.3 Sơ đồ chức năng thanh toán.....	39
Hình 3.7 Sơ đồ chức năng quản lý sản phẩm	43
Hình 3.9 Sơ đồ chức năng voucher.....	45
Hình 3.10 Sơ đồ chức năng quản lý danh mục	46
Hình 3.11 Sơ đồ chức năng quảng lý topping	47
Hình 3.12 Sơ đồ chức năng phản hồi và thống kê	48
Hình 3.13 Biểu đồ Use Case tổng quát Khách hàng	24
Hình 3.14 Biểu đồ Use Case tổng quát Người quản trị	25
Hình 3.15 Use case cho chức năng Đặt hàng	26
Hình 3.16 Use case cho chức năng đăng ký, đăng nhập, đăng xuất.....	27
Hình 3.17 Use case cho chức năng Use case cho chức năng Thanh toán	28

Hình 3.18 Use case cho chức năng Quản lý đơn hàng	30
Hình 3.10 Use case cho chức năng Đánh giá	31
Hình 3.20 Use case cho chức năng Quản lý sản phẩm	32
Hình 3.21 Use case cho chức năng Quản lý voucher	34
Hình 3.22 Use case cho chức năng Sử dụng voucher.....	35
Hình 3.23 Sơ đồ tuần tự chức năng đăng nhập.....	49
Hình 3.24 Sơ đồ tuần tự chức năng đăng ký.....	50
Hình 3.25 Sơ đồ tuần tự chức năng xem chi tiết đồ uống	51
Hình 3.26 Sơ đồ tuần tự chức năng thêm vào giỏ hàng.....	52
Hình 3.27 Sơ đồ tuần tự chức năng đặt hàng.....	54
Hình 3.28 Sơ đồ tuần tự chức năng thanh toán tổng quát	56
Hình 3.29 Sơ đồ tuần tự chức năng thanh toán VNPay.....	58
Hình 3.30 Sơ đồ tuần tự chức năng thanh toán blockchain.....	59
Hình 3.31 Sơ đồ tuần tự chức năng theo dõi đơn hàng	61
Hình 3.32 Sơ đồ tuần tự chức năng đánh giá đơn hàng.....	62
Hình 3.33 Biểu đồ ERD.....	64
Hình 4.1 Giao diện màn hình đăng nhập	75
Hình 4.2 Giao diện màn hình đăng ký	76
Hình 4.3 Giao diện màn hình trang chủ.....	77
Hình 4.4 Giao diện màn hình đơn hàng.....	78
Hình 4.5 Giao diện màn hình danh mục sản phẩm.....	79
Hình 4.6 Giao diện màn hình chi tiết sản phẩm	80
Hình 4.7 Giao diện màn hình giỏ hàng.....	81
Hình 4.8 Giao diện màn hình thanh toán.....	82
Hình 4.9 Giao diện màn hình chọn địa chỉ giao hàng	83
Hình 4.10 Giao diện màn hình xác nhận đơn hàng	84
Hình 4.11 Giao diện màn hình đặt hàng thành công	85
Hình 4.12 Giao diện màn hình lịch sử đơn hàng.....	86

Hình 4.13 Giao diện màn hình theo dõi đơn hàng.....	87
Hình 4.14 Giao diện màn hình đánh giá sản phẩm.....	88
Hình 4.15 Giao diện màn hình voucher.....	89
Hình 4.16 Giao diện màn hình tài khoản.....	90
Hình 4.17 Giao diện màn hình thông tin tài khoản.....	91
Hình 4.18 Giao diện màn hình đổi mật khẩu.....	92
Hình 4.19 Giao diện trang chủ người quản trị.....	93
Hình 4.20 Giao diện trang quản lý sản phẩm	93
Hình 4.21 Giao diện màn hình quản lý voucher.....	94
Hình 4.22 Giao diện màn hình quản lý phản hồi.....	94

DANH SÁCH HÌNH BẢNG

Bảng 3.1. Use case cho chức năng Đặt hàng.....	26
Bảng 3.2. Use case cho chức năng đăng ký, đăng nhập, đăng xuất	27
Bảng 3.3. Use case cho chức năng Use case cho chức năng Thanh toán.....	29
Bảng 3.4. Use case cho chức năng Quản lý đơn hàng.....	30
Bảng 3.5. Use case cho chức năng Đánh giá.....	31
Bảng 3.6. Use case cho chức năng Quản lý sản phẩm	33
Bảng 3.7. Use case cho chức năng Quản lý voucher.....	34
Bảng 3.8. Use case cho chức năng Sử dụng voucher	35
Bảng 3.9. Bảng user.....	65
Bảng 3.10. Bảng Category.....	65
Bảng 3.11. Bảng Drink	65
Bảng 3.12. Bảng Order	66
Bảng 3.13 Bảng DrinkOrder.....	67
Bảng 3.14 Bảng Address.....	68
Bảng 3.15 Bảng Voucher	69
Bảng 3.16 Bảng Rating.....	70

DANH SÁCH CÁC KÝ HIỆU, CHỮ VIẾT TẮT

Từ viết tắt	Tiếng Anh	Tiếng Việt
MVVM	Model-View-ViewModel	Mô hình-Khung nhìn-Mô hình khung nhìn
API	Application Programming Interface	Giao diện lập trình ứng dụng
SDK	Software Development Kit	Bộ công cụ phát triển phần mềm
IDE	Integrated Development Environment	Môi trường phát triển tích hợp
UI/UX	User Interface/User Experience	Giao diện người dùng/Trải nghiệm người dùng
GPS	Global Positioning System	Hệ thống định vị toàn cầu
JSON	JavaScript Object Notation	Ký hiệu đối tượng JavaScript
SQL	Structured Query Language	Ngôn ngữ truy vấn có cấu trúc
DAO	Data Access Object	Đối tượng truy cập dữ liệu
CRUD	Create, Read, Update, Delete	Tạo, Đọc, Cập nhật, Xóa
HTTPS	HyperText Transfer Protocol Secure	Giao thức truyền siêu văn bản bảo mật

HTTP	HyperText Transfer Protocol	Giao thức truyền siêu văn bản
URL	Uniform Resource Locator	Định vị tài nguyên thống nhất
ERD	Entity Relationship Diagram	Sơ đồ quan hệ thực thể
ID	Identifier	Định danh
SSL	Secure Sockets Layer	Lớp cổng bảo mật
OS	Operating System	Hệ điều hành
POS	Point of Sale	Điểm bán hàng
QR	Quick Response	Phản hồi nhanh
F&B	Food and Beverage	Thực phẩm và Đồ uống

CHƯƠNG 1. GIỚI THIỆU TỔNG QUAN

1.1. Lý do chọn đề tài

Ngày nay, cùng với sự phát triển nhanh chóng của công nghệ thông tin và Internet, các mô hình kinh doanh truyền thống đang dần chuyển mình sang nền tảng trực tuyến. Đặc biệt trong lĩnh vực F&B (Food and Beverage), việc kết hợp giữa quản lý cửa hàng và bán hàng online đang trở thành xu hướng phổ biến.

Đối với các quán cafe, việc quản lý đơn hàng, thanh toán, tương tác với khách hàng và theo dõi doanh thu thường gặp nhiều khó khăn nếu chỉ làm thủ công. Trong khi đó, khách hàng lại có nhu cầu đặt hàng nhanh chóng, tiện lợi và có thể theo dõi trạng thái đơn hàng trực tuyến.

Chính vì vậy, việc xây dựng một ứng dụng di động quản lý quán Cafe bán hàng online là cần thiết. Ứng dụng không chỉ giúp chủ quán dễ dàng quản lý hoạt động kinh doanh, mà còn giúp khách hàng đặt hàng, thanh toán và nhận ưu đãi thuận tiện hơn.

Từ những lý do trên, em quyết định chọn đề tài: “Xây dựng ứng dụng di động quản lý quán Cafe bán hàng online”, nhằm góp phần ứng dụng công nghệ thông tin vào thực tế hoạt động kinh doanh, mang lại hiệu quả cao và trải nghiệm hiện đại cho người dùng.

1.2. Lịch sử nghiên cứu

Trong nước:

Hiện nay ở Việt Nam, các ứng dụng quản lý quán cafe và bán hàng online còn chưa phổ biến rộng rãi. Phần lớn các cửa hàng nhỏ lẻ vẫn sử dụng hình thức quản lý truyền thống như ghi chép thủ công hoặc sử dụng bảng tính Excel, gây mất thời gian và dễ sai sót.

Một số ứng dụng và hệ thống đã được phát triển như CukCuk, KiotViet, Sapo POS, hay POS365,... tuy nhiên các ứng dụng này thường chỉ tập trung vào quản lý bán hàng tổng quát (đa ngành nghề), chưa thực sự được thiết kế chuyên biệt cho mô hình kinh doanh quán cafe. Ngoài ra, một số tính năng như đặt hàng online, định vị cửa hàng, áp dụng mã giảm giá, hay đo khoảng cách giữa khách hàng và quán vẫn còn hạn chế hoặc chưa được tích hợp đầy đủ.

Do đó, nhu cầu về một ứng dụng di động đơn giản, dễ sử dụng, chi phí thấp và phù hợp với đặc thù của quán cafe nhỏ là rất cần thiết.

Ngoài nước:

Ở nước ngoài, các ứng dụng hỗ trợ quản lý và bán hàng trong lĩnh vực đồ uống và cafe đã phát triển sớm hơn, tiêu biểu như Starbucks App, Square POS, Toast POS, hay Clover. Các ứng dụng này thường được tích hợp nhiều chức năng hiện đại như đặt hàng trực tuyến, thanh toán không tiền mặt, tích điểm khách hàng thân thiết, và theo dõi vị trí giao hàng.

Tuy nhiên, các ứng dụng trên chủ yếu phục vụ cho hệ thống cửa hàng lớn hoặc chuỗi thương hiệu quốc tế, không phù hợp với các mô hình quán cafe độc lập hoặc quy mô nhỏ tại Việt Nam.

Vì vậy, đề tài “Xây dựng ứng dụng di động quản lý quán Cafe bán hàng online” hướng tới việc phát triển một giải pháp nội địa, tối ưu cho người dùng Việt, giao diện thân thiện, tích hợp các tính năng cần thiết như:

Đăng ký, đăng nhập tài khoản;

Đặt và quản lý đơn hàng;

Định vị vị trí và đo khoảng cách từ khách hàng đến cửa hàng;

Thanh toán trực tuyến và áp dụng mã giảm giá.

1.3. Mục tiêu đề tài

Mục tiêu chung:

Mục tiêu của đề tài là phát triển một ứng dụng di động quản lý quán Cafe bán hàng online chạy trên nền tảng Android, được lập trình bằng Android Studio với ngôn ngữ Kotlin.

Hệ thống được thiết kế để hỗ trợ cả chủ quán và khách hàng, cung cấp các chức năng quan trọng như:

- Đăng nhập và đăng ký tài khoản;
- Đặt hàng và quản lý đơn hàng;
- Thanh toán trực tuyến và áp dụng mã giảm giá;
- Định vị và đo khoảng cách từ khách hàng đến quán;
- Quản lý sản phẩm, menu và doanh thu.

Ứng dụng hướng đến mục tiêu tự động hóa quy trình bán hàng, giảm tải công việc quản lý thủ công, và nâng cao trải nghiệm người dùng, đồng thời đảm bảo bảo mật dữ liệu, tính thân thiện trong giao diện, và khả năng mở rộng trong tương lai.

Mục tiêu cụ thể:

Để đạt được mục tiêu chung nêu trên, đề tài “Xây dựng ứng dụng di động quản lý quán Cafe bán hàng online” sẽ thực hiện các mục tiêu cụ thể sau:

- Xây dựng giao diện người dùng (UI/UX) trực quan, thân thiện, dễ sử dụng trên nền tảng Android, giúp cả chủ quán và khách hàng thao tác nhanh chóng, thuận tiện.
- Phát triển chức năng đăng nhập và đăng ký tài khoản, cho phép người dùng (khách hàng và quản lý quán) truy cập vào hệ thống với quyền hạn riêng biệt.
- Tích hợp tính năng định vị và đo khoảng cách giữa quán và khách hàng nhằm hỗ trợ việc xác định vị trí giao hàng và tính toán thời gian phục vụ hợp lý.
- Phát triển chức năng đặt hàng và quản lý đơn hàng trực tuyến, giúp khách hàng dễ dàng lựa chọn món, xem giỏ hàng và theo dõi trạng thái đơn hàng.
- Xây dựng chức năng thanh toán trực tuyến (qua ví điện tử hoặc phương thức mô phỏng), đồng thời hỗ trợ áp dụng mã giảm giá, khuyến mãi.
- Phát triển chức năng quản lý sản phẩm, menu, doanh thu và khách hàng dành cho chủ quán.
- Cải tiến và mở rộng hệ thống trong tương lai, đảm bảo khả năng cập nhật thêm các tính năng như đánh giá món, quản lý nhân viên, hoặc báo cáo doanh thu tự động.

1.4. Đối tượng, phạm vi nghiên cứu

1.4.1. Đối tượng nghiên cứu

Nội dung nghiên cứu:

Nghiên cứu và phát triển ứng dụng di động quản lý quán Cafe bán hàng online trên nền tảng Android Studio, sử dụng ngôn ngữ lập trình Kotlin, cơ sở dữ liệu Firebase (Realtime Database và Cloud Firestore) và Room Database, và tích hợp các API hỗ trợ như Google Maps API để định vị và đo khoảng cách.

Đối tượng sử dụng:

Chủ quán Cafe: có nhu cầu quản lý menu, đơn hàng, doanh thu, thông tin khách hàng và các chương trình khuyến mãi.

Khách hàng: có nhu cầu đặt hàng, theo dõi đơn hàng, áp dụng mã giảm giá, thanh toán trực tuyến và xác định vị trí quán gần nhất.

1.4.2. Phạm vi nghiên cứu

❖ Phạm vi nội dung:

Đề tài tập trung vào việc xây dựng ứng dụng di động Android với các chức năng chính gồm:

- Đăng nhập, đăng ký tài khoản người dùng;
- Đặt và quản lý đơn hàng trực tuyến;
- Tích hợp thanh toán và mã giảm giá;
- Định vị và đo khoảng cách giữa quán và khách hàng;
- Quản lý menu, doanh thu và thông tin khách hàng.

Đề tài không đi sâu vào phát triển hệ thống máy chủ phức tạp hoặc ứng dụng trên nền tảng iOS.

❖ Phạm vi không gian:

Ứng dụng được thiết kế và thử nghiệm trong môi trường mô phỏng một quán Cafe thực tế tại Việt Nam, có thể triển khai trong các quán nhỏ, quán nhượng quyền hoặc hộ kinh doanh cá nhân.

❖ Phạm vi thời gian:

Đề tài được thực hiện trong thời gian từ [tháng 10] đến [tháng kết 12], bao gồm các giai đoạn: phân tích yêu cầu, thiết kế hệ thống, lập trình, kiểm thử và hoàn thiện ứng dụng.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT VÀ CÔNG CỤ XÂY DỰNG HỆ THỐNG

2.1. Cơ sở lý thuyết

2.1.1. Giới thiệu hệ điều hành Android

2.1.1.1. Giới thiệu Android

Hệ điều hành Android là một hệ điều hành dựa trên nền tảng Linux, được thiết kế dành cho các thiết bị di động có màn hình cảm ứng như điện thoại thông minh và máy tính bảng. Ban đầu, Android được phát triển bởi Android Inc. với sự hỗ trợ tài chính từ Google và sau này được chính Google mua lại vào năm 2005.

Android ra mắt vào năm 2007 cùng với tuyên bố thành lập Liên minh thiết bị cầm tay mở: một hiệp hội gồm các công ty phần cứng, phần mềm, và viễn thông với mục tiêu đẩy mạnh các tiêu chuẩn mở cho các thiết bị di động.

Android có mã nguồn mở và Google phát hành mã nguồn theo Giấy phép Apache. Chính mã nguồn mở cùng với một giấy phép không có nhiều ràng buộc đã cho phép các nhà phát triển thiết bị, mạng di động và các lập trình viên nhiệt huyết được điều chỉnh và phân phối Android một cách tự do.

Những yếu tố này đã giúp Android trở thành nền tảng điện thoại thông minh phổ biến nhất thế giới, vượt qua Symbian OS vào quý 4 năm 2010, và được các công ty công nghệ lựa chọn khi họ cần một hệ điều hành không nặng nề, có khả năng tinh chỉnh, và giá rẻ rẻ chạy trên các thiết bị công nghệ cao thay vì tạo dựng từ đầu.

Từ năm 2011, Android đã là hệ điều hành bán chạy nhất trên toàn cầu trên điện thoại thông minh và từ năm 2013 trên máy tính bảng. Android có hơn ba tỷ người dùng hàng tháng tính đến tháng 5 năm 2021, là hệ điều hành có cài đặt nhiều nhất trên thế giới, và Cửa hàng Google Play có hơn 3 triệu ứng dụng tính đến tháng 1 năm 2021.

2.1.1.2. Lịch sử phát triển

Tháng 7 năm 2005, Google mua lại Android Inc., một công ty nhỏ có trụ sở tại Palo Alto, California, Mỹ. Lúc đó, thông tin về công việc của Android còn rất hạn chế, chỉ cho biết họ đang phát triển phần mềm cho điện thoại di động. Điều này đã gieo rắc những tin đồn về việc Google có ý định bước vào thị trường điện thoại di động.

Tháng 9 năm 2007, tờ InformationWeek đăng tải một nghiên cứu của Evalueserve cho biết Google đã nộp một số đơn xin cấp bằng sáng chế trong lĩnh vực điện thoại di động. Điều này tạo thêm sự hào hứng và tò mò về tương lai của Android.

Ngày 9 tháng 12 năm 2008, thêm 14 thành viên mới gia nhập dự án Android được công bố, gồm có ARM Holdings, Atheros Communications, Asustek Computer Inc, Garmin Ltd, Softbank, Sony Ericsson, Toshiba Corp và Vodafone Group Plc.

Android đã trải qua một hành trình dài từ khi thiết bị đầu tiên sử dụng hệ điều hành này xuất hiện, chiếc HTC Dream (hay còn gọi là T-Mobile G1). Trong thời gian đó, người dùng đã thấy sự xuất hiện của nhiều phiên bản Android, giúp nó dần biến đổi thành một nền tảng di động mạnh mẽ và đa dạng như ngày hôm nay.

Giới thiệu Hệ điều hành Android

Android là hệ điều hành được phát triển bởi Tổng công ty Android, với sự hỗ trợ tài chính từ Google. Mãi đến năm 2005, được chính Google mua lại và cho ra mắt vào năm 2007.



Hình 2.1 Hệ điều hành Android

Android là hệ điều hành nguồn mở dựa trên nền tảng Linux, chủ yếu dành cho các thiết bị có màn hình cảm ứng như điện thoại, máy tính bảng.

Với mã nguồn mở và giấy phép không có nhiều ràng buộc nên Android ngày càng trở thành nền tảng điện thoại thông minh phổ biến nhất thế giới.

Vào quý 3 năm 2012, Android chiếm 75% thị phần về điện thoại thông minh trên toàn thế giới với con số tổng cộng khoảng 500 triệu thiết bị đã được kích hoạt và 1,3 triệu lượt kích hoạt mỗi ngày.

Tuy nhiên, hiện nay Android đang có sự cạnh tranh với iOS từ Apple nhưng Android vẫn dẫn vị thế thượng phong trên cuộc chiến điện thoại thông minh.

2.1.1.3. Giao diện và ứng dụng Android

Giao diện

Giao diện của Android sử dụng cảm ứng chạm, tác động trực tiếp lên màn hình như vuốt, chạm, phóng to và thu lại để xử lý các đối tượng trên màn hình.

Sau khi khởi động các thiết bị Android màn hình chính sẽ hiển thị gồm nhiều biểu tượng (icon) và tiện ích (widget). Giao diện của Android cho phép người dùng tự do sắp xếp hình dáng, biểu tượng, tiện ích tùy theo ý thích của mình.

Những ứng dụng do bên thứ 3 cung cấp có trên Google Play cho phép người dùng thay đổi chủ đề của màn hình chính tương tự như Windows Phone hay iOS.

Tuy nhiên, những nhà sản xuất đều thay đổi hình dáng Android một cách linh hoạt để dễ dàng phân biệt chúng với các hệ điều hành khác.



Hình 2.1 Giao diện hệ điều hành Android

Các phiên bản Android sẽ được Google cập nhật theo chu kỳ từ 6 đến 9 tháng. Hiện nay (tính đến tháng 08/2021), bản cập nhật chính thức mới nhất của Android là Android 11.

Ứng dụng

Các ứng dụng do bên thứ ba có trên Google Play để người dùng có thể tải về. Các ứng dụng trên Play Store cho phép người dùng tải về và cập nhật các ứng dụng do Google và các nhà phát triển phát hành.

Đối với những ứng dụng mất phí tải về, nếu người dùng mua một ứng dụng mà họ cảm thấy không hài lòng thì họ được hoàn trả tiền sau 15 phút kể từ lúc tải về.

Tính đến tháng 10/2012, đã có hơn 700.000 ứng dụng trên Android và số lượt tải về từ cửa hàng ứng dụng chính của Android (Google Play) chiếm khoảng 25 tỷ lượt.

Đến nay, con số này đã lên tới 3 triệu ứng dụng.

Các ứng dụng cho Android được phát triển bằng ngôn ngữ Java và sử dụng Bộ phát triển phần mềm Android (SDK). Bộ phát triển này gồm có công cụ gỡ lỗi, thư viện phần mềm... hỗ trợ với công suất tối đa cho nhu cầu của các thiết bị.

Ngoài ra, các công cụ phát triển khác như bộ phát triển gốc dành cho các ứng dụng cũng được tích hợp sẵn.

2.1.1.4. Ưu điểm và nhược điểm của hệ điều hành Android

Ưu điểm:

Kho ứng dụng đa dạng: Với hệ thống cửa hàng ứng dụng Google Play, hệ điều hành Android có thể đáp ứng các nhu cầu từ chơi game cho đến làm việc với hơn 3 triệu ứng dụng để người dùng lựa chọn.



Hình 2.2 Kho ứng dụng đa dạng

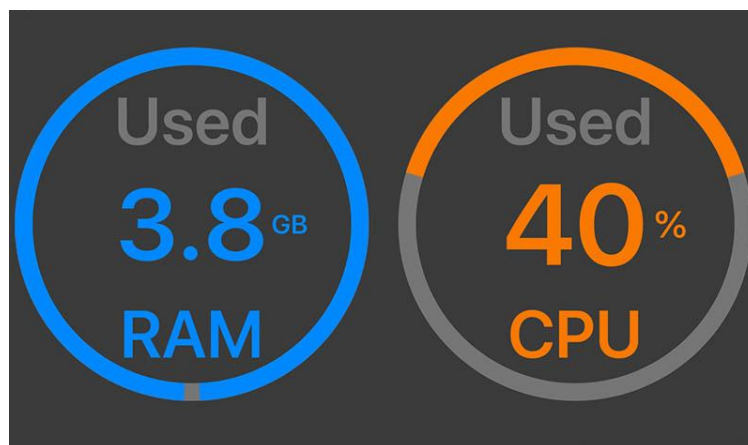
Mẫu mã đa dạng: Với nhiều nhà sản xuất lớn như Samsung, OPPO, Xiaomi, Huawei, Sony, Nokia... người dùng có thể lựa chọn giữa nhiều mẫu mã thiết bị khác nhau, từ các mẫu giá rẻ cho đến các mẫu cao cấp.

Khả năng tùy biến cao có thể chỉnh sửa mà không có sự can thiệp hay cấm cản từ Google: Do bản chất nguồn mở của hệ điều hành Android, ai cũng có thể lấy được mã nguồn của hệ điều hành này. Điều này cũng đồng nghĩa là các nhà sản xuất, cũng như là các lập trình viên độc lập, có thể tự do tùy biến Android để có được hiệu năng tốt nhất hoặc bỏ đi những tính năng không cần thiết.

Người dùng ưa chuộng nhiều: Android có cộng đồng người dùng và lập trình viên độc lập khá lớn, nên khi gặp vấn đề về thiết bị hay về phiên bản Android, người dùng sẽ được hỗ trợ rất nhiệt tình từ phía cộng đồng.

Nhược điểm:

Nhiều ứng dụng chạy ngầm làm chậm máy: So với iOS, Android tối ưu hóa bộ nhớ RAM có phần kém hơn, dẫn đến việc nhiều ứng dụng chạy ngầm gây chậm máy hoặc thậm chí là đơ máy.



Hình 2.3 Tối ưu hoá bộ nhớ RAM

Một số ứng dụng chưa được tối ưu hóa tốt: Do có quá nhiều mẫu mã khác nhau, các nhà phát triển không thể tối ưu hóa ứng dụng cho tất cả các mẫu thiết bị Android trên thị trường, nên các ứng dụng có thể gặp các lỗi như không hiển thị được toàn màn hình hoặc không thể tận dụng hết sức mạnh phần cứng của máy.

Chất lượng một số ứng dụng còn kém: Một số ứng dụng trên Google Play có chất lượng khá kém với hàng loạt các quảng cáo khó chịu và không có các chức năng hữu dụng, gây cản trở cho công việc hay thời gian giải trí của người dùng.

Dễ bị virus xâm nhập: Nếu người dùng tải các ứng dụng ở các nguồn khác ngoài Google Play nhưng lại không cài ứng dụng diệt virus, máy của họ có khả năng cao sẽ bị nhiễm virus từ các ứng dụng ngoài này.

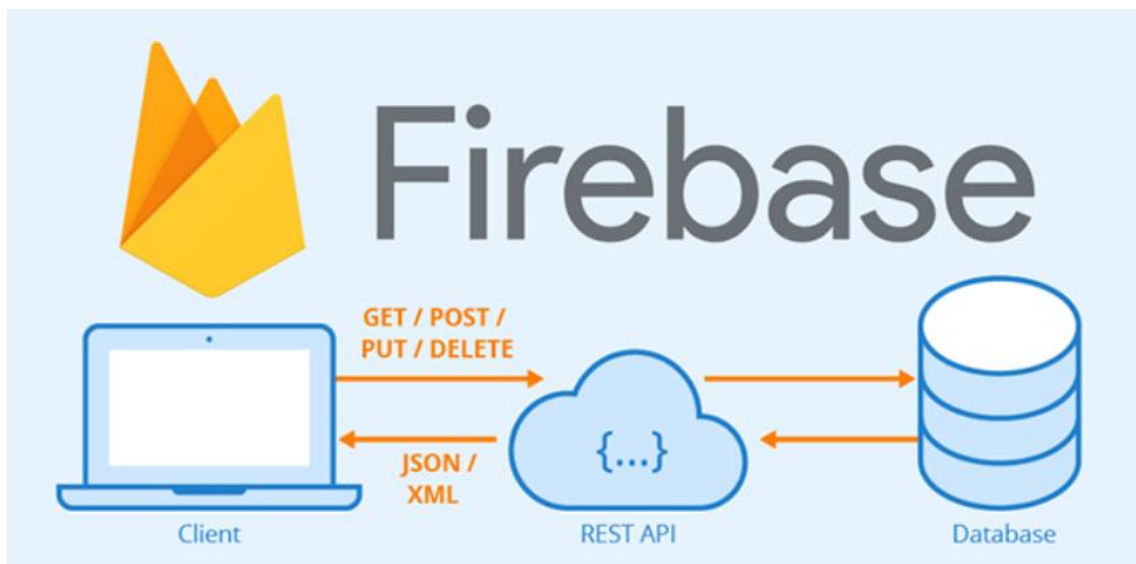


Hình 2.4 Virus xâm nhập

2.1.2. Firebase

2.1.2.1. Tổng quan

Firebase là một nền tảng giúp phát triển các ứng dụng di động trong web. Bên cạnh đó, Firebase còn được hiểu là một dịch vụ cơ sở dữ liệu hoạt động trên nền tảng đám mây cloud với hệ thống máy chủ mạnh mẽ của Google.



Hình 2.5 Firebase

Firebase chứa cơ sở dữ liệu mang đến khả năng code nhanh và thuận tiện hơn. Lập trình viên có thể dễ dàng lập trình ứng dụng bằng cách đơn giản hóa các thao tác với cơ sở dữ liệu sẵn có.

2.1.2.2. Lịch sử hình thành và phát triển nền tảng Firebase:

Vào năm 2011, James Tamplin và Andrew Lee đã cho ra mắt Evolve. Đây là một nền tảng có cấu trúc khá đơn giản chuyên cung cấp các API cần thiết để tích hợp tính năng trò chuyện vào các trang web. Tuy nhiên, họ nhận ra rằng nền tảng này đang được sử dụng để truyền dữ liệu ứng dụng chứ không đơn giản là chat. Sau đó, họ đã phát triển Envole và tạo nên Firebase.

Đến tháng 4 năm 2012, Firebase đã được công bố trên toàn cầu dưới dạng một công ty riêng biệt. Những năm sau đó, Firebase đã thực hiện nhiều cuộc huy động vốn và ra mắt các sản phẩm mới.

Đến tháng 10 năm 2014, Firebase đã chính thức được Google mua lại và trở thành một ứng dụng đa năng trên nền tảng di động và website.

2.1.2.3. Tính năng của firebase

❖ *Realtime Database:*

Realtime Database là một cơ sở dữ liệu thời gian thực. Ngay sau khi bạn đăng ký tài khoản trên Firebase, bạn sẽ nhận được Realtime Database được lưu trữ dưới dạng JSON và được đồng bộ hóa theo thời gian thực đối với mọi kết nối.



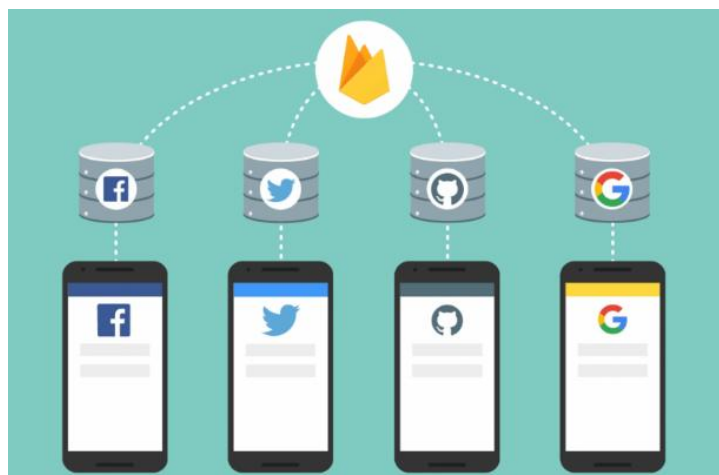
Hình 2.6 Realtime Database

Đối với các ứng dụng được xây dựng trên đa nền tảng như Android, IOS và WebApp, tất cả client sẽ cùng sử dụng một cơ sở dữ liệu. Bên cạnh đó, hệ thống dữ liệu này sẽ tự động cập nhật khi lập trình viên phát triển ứng dụng. Sau đó, tất cả dữ liệu này sẽ được truyền tải thông qua các kết nối SSL có 2048 bit.

❖ Các khả năng chính của Realtime Database:

- Realtime: Firebase Realtime Database sử dụng đồng bộ dữ liệu mỗi khi dữ liệu có thay đổi, mọi thiết bị được kết nối sẽ nhận được thay đổi trong vài mili giây.
- Offline: Khi người dùng ngoại tuyến, dữ liệu sẽ được lưu trên bộ nhớ cache của thiết bị và tự động đồng bộ khi bạn trực tuyến. Tất cả là tự động
- Accessible from Client Devices: Firebase Realtime Database có thể truy cập từ một thiết bị mobile hoặc trình duyệt web. Nó không cần một ứng dụng server nào cả. Bảo mật và xác thực dữ liệu có thể thông qua các Rule bảo mật của Firebase Realtime Database, các rule được thực thi khi dữ liệu được đọc hoặc ghi.

Authentication:

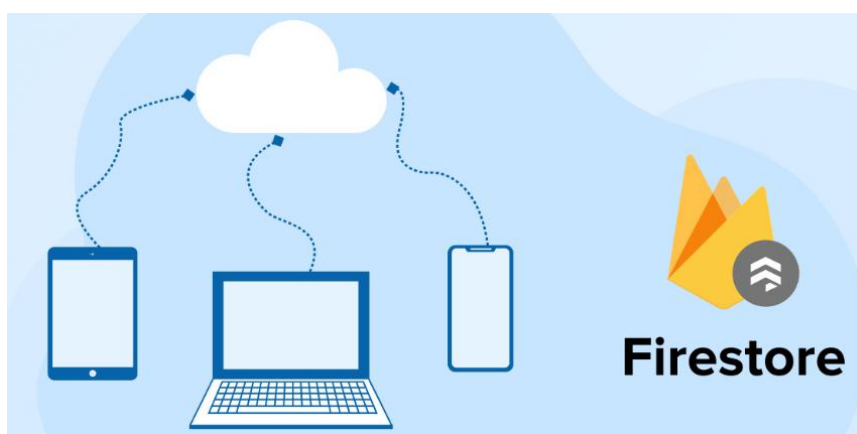


Hình 2.7 Authentication

Authentication là tính năng giúp xác thực danh tính của người dùng ứng dụng. Firebase cung cấp các bước xác thực thông qua Email, Facebook, Twitter, GitHub hay Google. Điều này giúp cho các thông tin cá nhân của khách hàng được bảo vệ một cách tốt nhất, hạn chế được tình trạng bị hacker đánh cắp. Đồng thời việc xác thực danh tính qua Firebase sẽ giúp người dùng tiếp cận sản phẩm nhanh chóng và an toàn hơn.

Cloud Firestore

Cloud Firestore được phát triển từ tính năng Realtime Database. Trải qua nhiều lần nâng cấp, Cloud Firestore có giao diện trực quan và khả năng mở rộng ưu việt hơn so với Realtime Database. Tính năng này của Firebase giúp đồng bộ mọi dữ liệu trên các ứng dụng thông qua việc đăng ký thời gian thực và cung cấp hỗ trợ ngoại tuyến cho thiết bị di động cũng như website.



Hình 2.8 Cloud Firestore

❖ Tính năng firestore:

Được thiết kế để mở rộng quy mô: Firestore được thiết kế để mở rộng quy mô cấu trúc dữ liệu rất lớn. Nó có khả năng xử lý khối lượng công việc lớn. Nó cung cấp cơ sở hạ

tăng tốt nhất với tính năng sao chép dữ liệu đa vùng, tính nhất quán, hỗ trợ giao dịch và hoạt động hàng loạt.

Cập nhật theo thời gian thực: Firestore cung cấp cho doanh nghiệp tính năng đồng bộ hóa dữ liệu theo thời gian thực để cập nhật dữ liệu kịp thời trên tất cả các thiết bị được kết nối.

Hỗ trợ ngoại tuyến: Firestore cung cấp hỗ trợ ngoại tuyến cho cơ sở dữ liệu. Vì vậy, bạn có thể đọc, ghi và truy vấn cơ sở dữ liệu ngay cả khi bạn ngoại tuyến. Nó sẽ đồng bộ hóa lại dữ liệu khi bạn trực tuyến và kết nối.

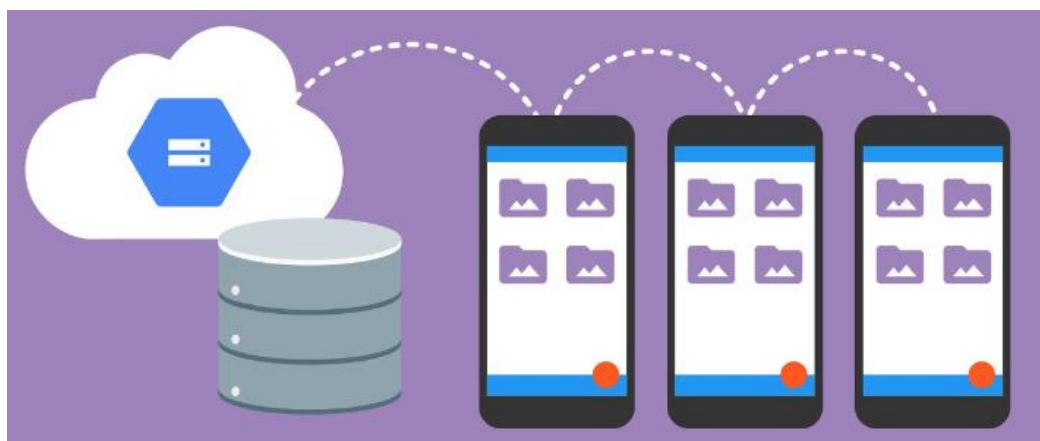
Truy vấn biểu cảm: Firestore cho phép truy vấn cơ sở dữ liệu, nơi bạn có thể đọc/ghi một tài liệu cũng như có thể tìm nạp danh sách tài liệu bằng cách sắp xếp và lọc. Bạn có thể bao gồm nhiều bộ lọc trong truy vấn của mình. Firestore sử dụng các chỉ mục trong cơ sở dữ liệu truy vấn để có hiệu suất tốt hơn.

Danh sách kiểu dữ liệu firestore hỗ trợ:

- String
- Number
- Boolean
- Map
- Array
- Null
- Timestamp
- Geo point
- Reference

Cloud Storage:

Cloud Storage cho Firebase là dịch vụ lưu trữ đối tượng mạnh mẽ, đơn giản và tiết kiệm chi phí được xây dựng cho quy mô Google. SDK Firebase dành cho Cloud Storage bổ sung tính năng bảo mật của Google cho các tệp tải lên và tải xuống cho ứng dụng Firebase của bạn, bất kể chất lượng mạng.



Hình 2.9 Cloud Storage

❖ **Firestore Cloud Messaging:**

Sử dụng FCM, bạn có thể thông báo cho ứng dụng khách rằng email mới hoặc dữ liệu khác có sẵn để đồng bộ hóa. Bạn có thể gửi tin nhắn thông báo để thúc đẩy tương tác lại và giữ chân người dùng. Đối với các trường hợp sử dụng như nhắn tin tức thời, một tin nhắn có thể chuyển tải trọng lên tới 4000 byte tới ứng dụng khách.

2.2. Công cụ xây dựng

2.2.1. Android Studio

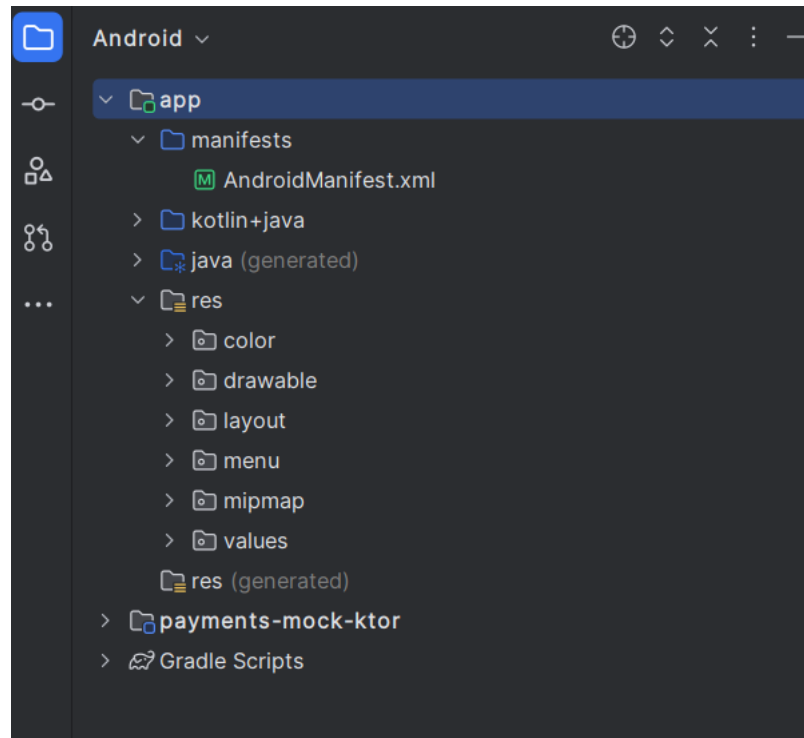
Android Studio là Môi trường phát triển tích hợp (IDE) chính thức để phát triển ứng dụng Android. Nhờ có công cụ cho nhà phát triển và trình soạn thảo mã mạnh mẽ của IntelliJ IDEA, Android Studio cung cấp thêm nhiều tính năng giúp bạn nâng cao năng suất khi xây dựng ứng dụng Android, chẳng hạn như:

- Một hệ thống xây dựng linh hoạt dựa trên Gradle
- Một trình mô phỏng nhanh và nhiều tính năng
- Một môi trường hợp nhất nơi bạn có thể phát triển cho mọi thiết bị Android
- Tính năng Live Edit (Chỉnh sửa trực tiếp) để cập nhật các thành phần kết hợp trong trình mô phỏng và thiết bị thực theo thời gian thực
- Mã mẫu và quá trình tích hợp GitHub để giúp bạn xây dựng các tính năng ứng dụng phổ biến cũng như nhập mã mẫu
- Đa dạng khung và công cụ thử nghiệm
- Công cụ tìm lỗi mã nguồn (lint) để nắm bắt hiệu suất, khả năng hữu dụng, khả năng tương thích với phiên bản và các vấn đề khác
- Hỗ trợ C++ và NDK
- Tích hợp sẵn tính năng hỗ trợ Google Cloud Platform, giúp dễ dàng tích hợp Google Cloud Messaging và App Engine.

Cấu trúc dự án:

Mỗi dự án trong Android Studio chứa một hoặc nhiều mô-đun có tệp mã nguồn và tệp tài nguyên. Có các loại mô-đun sau:

- Mô-đun ứng dụng Android
- Mô-đun thư viện
- Mô-đun Google App Engine



Hình 2.10 Mô-đun và tệp tài nguyên

Mỗi mô-đun ứng dụng có chứa các thư mục sau:

- manifests (tệp kê khai): Chứa tệp AndroidManifest.xml.
- java: Chứa các tệp mã nguồn Java và Kotlin, bao gồm cả mã kiểm thử JUnit.
- res: Chứa mọi tài nguyên không phải đoạn mã, chẳng hạn như chuỗi giao diện người dùng và hình ảnh bitmap.

Cấu trúc dự án Android trên ổ đĩa khác với cách trình bày ở đây. Để xem cấu trúc tệp của dự án thực tế, hãy chọn Project (Dự án) thay vì Android trên trình đơn Project (Dự án).

2.2.1.1. Hệ thống xây dựng Gradle

Android Studio sử dụng Gradle làm nền tảng cho hệ thống xây dựng với nhiều tính năng dành riêng cho Android do Trình hỗ trợ Android cho Gradle cung cấp. Hệ thống xây dựng này hoạt động như một công cụ tích hợp trên trình đơn Android Studio và độc lập với dòng lệnh. Bạn có thể sử dụng các tính năng của hệ thống xây dựng để làm những việc sau:

Tuỳ chỉnh, định cấu hình và mở rộng quy trình xây dựng.

Tạo nhiều tệp APK cho ứng dụng với nhiều tính năng trong khi sử dụng cùng một dự án và mô-đun.

Sử dụng lại mã và tài nguyên trên các nhóm tài nguyên (source set).

2.2.1.2. Công cụ gỡ lỗi và phân tích tài nguyên

Gỡ lỗi cùng dòng:

Bạn có thể dùng tính năng gỡ lỗi cùng dòng để cải thiện kết quả kiểm tra toàn diện về mã (code walkthrough) trong khung hiển thị trình gỡ lỗi nhờ tính năng xác minh cùng dòng đối với nội dung tham chiếu, biểu thức và giá trị biến.

Thông tin gỡ lỗi cùng dòng bao gồm:

Giá trị biến cùng dòng

Các đối tượng tham chiếu đến một đối tượng đã chọn

Giá trị trả về của phương thức

Biểu thức toán tử và lambda

Giá trị trong chú giải công cụ

Truy cập vào tệp dữ liệu:

Bộ công cụ SDK Android, chẳng hạn như Systrace và Logcat, tạo ra dữ liệu về hiệu suất và quá trình gỡ lỗi cho bản phân tích ứng dụng chi tiết.

Cách xem các tệp dữ liệu đã tạo hiện có:

Mở cửa sổ công cụ Captures (Ghi lại).

Trong danh sách các tệp đã tạo, hãy nhấp đúp vào một tệp để xem dữ liệu tương ứng.

Nhấp chuột phải vào tệp HPROF bất kỳ để chuyển đổi các tệp đó thành tệp chuẩn.

Kiểm tra định dạng tệp sử dụng RAM.

Kiểm tra mã:

Công cụ tìm lỗi mã nguồn (lint tool) kiểm tra các tệp nguồn dự án Android để tìm ra các lỗi có thể xảy ra và cải thiện khả năng tối ưu hoá, nhằm đảm bảo tính chính xác, tính bảo mật, hiệu suất, khả năng hữu dụng, khả năng hỗ trợ tiếp cận và khả năng quốc tế hoá.

2.2.2. Kotlin

2.2.2.1. Tổng quan

Kotlin là một ngôn ngữ lập trình nguồn mở, kiểu tĩnh, hỗ trợ cả lập trình chức năng lẫn hướng đối tượng. Kotlin cung cấp cú pháp và khái niệm tương tự trong các ngôn ngữ

khác, bao gồm cả C#, Java và Scala cùng nhiều ngôn ngữ khác. Kotlin không phải là độc nhất – mà Kotlin lấy cảm hứng từ nhiều thập kỷ để phát triển ngôn ngữ. Ngôn ngữ này tồn tại trong các biến thể nhắm đến JVM (Kotlin/VM), JavaScript (Kotlin/JS) và mã gốc (Kotlin/mã gốc).



Hình 2.11 Kotlin

2.2.2.2. Lịch sử hình thành của kotlin

Năm 2010: JetBrains bắt đầu nghiên cứu và phát triển Kotlin như một ngôn ngữ mới dựa trên JVM.

Tháng 7 năm 2011: Phiên bản M1 của Kotlin được giới thiệu công khai lần đầu tiên.

Tháng 6 năm 2013: JetBrains công bố rằng Kotlin sẽ được phát triển dưới dạng dự án mã nguồn mở.

Tháng 2 năm 2016: Phiên bản 1.0 của Kotlin được phát hành, đánh dấu bước ngoặt quan trọng trong lịch sử của ngôn ngữ.

Tháng 5 năm 2017: Google công bố hỗ trợ chính thức Kotlin trong việc phát triển ứng dụng Android.

Tháng 10 năm 2017: Phiên bản Kotlin 1.2 được phát hành với nhiều cải tiến như hỗ trợ cho multiplatform projects và `kotlinx.coroutines`.

Tháng 5 năm 2019: Kotlin 1.3 ra mắt với tính năng chính là Kotlin/Native, cho phép phát triển ứng dụng đa nền tảng không cần JVM.

Tháng 12 năm 2020: Kotlin 1.4 được phát hành với các tính năng mới như `kotlinx.serialization` và JVM IR compiler.

Tháng 8 năm 2021: Kotlin 1.5 ra mắt với nhiều cải tiến về tương thích ngược với Java và tính năng mới như sealed interfaces và implicit conversions.

2.2.2.3. Những điểm nổi bật và hạn chế của kotlin

Ưu điểm:

Tương thích ngược với Java: Kotlin được thiết kế để tương thích hoàn toàn với mã nguồn Java. Điều này cho phép lập trình viên dễ dàng tích hợp Kotlin vào các dự án Java hiện có và sử dụng lại mã nguồn Java đã có mà không cần phải viết lại từ đầu.

Cú pháp đơn giản và dễ đọc: Kotlin có cú pháp ngắn gọn và dễ đọc, giúp lập trình viên viết mã một cách dễ dàng và nhanh chóng. Cú pháp Kotlin cũng giúp làm giảm sự lặp lại mã và tăng cường tính rõ ràng trong việc hiểu mã nguồn.

Hỗ trợ lập trình hàm: Kotlin hỗ trợ lambda expression và higher-order functions, cho phép lập trình viên sử dụng các khái niệm lập trình hàm để viết mã ngắn gọn, linh hoạt và dễ dàng đọc hiểu. Điều này giúp tăng cường tính linh hoạt và mạnh mẽ của ngôn ngữ.

Cộng đồng phát triển sáng tạo và hỗ trợ mạnh mẽ: Kotlin có một cộng đồng lập trình viên đông đảo, sáng tạo và nhiệt tình. JetBrains, công ty phát triển Kotlin, cung cấp sự hỗ trợ mạnh mẽ và liên tục cho ngôn ngữ này, và cộng đồng cũng đóng góp vào việc phát triển và cải thiện Kotlin.

Được ưu tiên hỗ trợ trong Android Studio và IDE: Việc này mang lại lợi ích cho lập trình viên bằng cách cung cấp tính năng như hoàn thành mã thông minh, phân tích mã, refactoring dễ dàng, gỡ rối hiệu quả và cập nhật liên tục.

Nhược điểm:

Kích thước file và tài nguyên: Kotlin có xu hướng tạo ra các file.class lớn hơn so với mã nguồn Java tương đương. Điều này có thể làm tăng kích thước ứng dụng và tiêu tốn nhiều tài nguyên hơn.

Tài liệu và nguồn học tập: Mặc dù Kotlin đang trở nên phổ biến, nhưng so với Java, tài liệu và nguồn học tập về Kotlin vẫn còn hạn chế. Điều này có thể làm cho việc học và nắm bắt Kotlin khó khăn hơn đối với một số lập trình viên mới.

Tốc độ biên dịch: Kotlin có thể có tốc độ biên dịch chậm hơn so với Java, đặc biệt khi xử lý các tệp mã lớn. Điều này có thể ảnh hưởng đến thời gian phát triển và kiểm thử của ứng dụng.

Thời gian khởi động ứng dụng: Ứng dụng Kotlin có thể mất thời gian khởi động lâu hơn so với ứng dụng Java tương đương. Điều này có thể gây khó khăn đối với trải nghiệm người dùng khi khởi động ứng dụng.

2.3. Tích hợp API và dịch vụ bên thứ ba

2.3.1. *Firestore Authentication*

2.3.1.1. Giới thiệu

Dịch vụ xác thực người dùng của Google. Cung cấp xác thực email/password, bảo mật mật khẩu bằng hash, quản lý session và phân quyền.

2.3.1.2. Mục đích

- Xác thực đăng nhập và đăng ký tài khoản
- Quản lý tài khoản và bảo mật mật khẩu
- Phân quyền user và admin

2.3.1.3. Ứng dụng trong dự án

- Đăng nhập: `firebaseAuth.signInWithEmailAndPassword()`
- Đăng ký: `firebaseAuth.createUserWithEmailAndPassword()`
- Kiểm tra email đã tồn tại
- Hash và lưu trữ mật khẩu an toàn

2.3.2. *Firestore Realtime Database*

2.3.2.1. Giới thiệu

Database NoSQL của Google. Lưu trữ dữ liệu dạng JSON, đồng bộ realtime giữa client và server. Tự động cập nhật khi dữ liệu thay đổi.

2.3.2.2. Mục đích

- Lưu trữ dữ liệu ứng dụng
- Đồng bộ realtime giữa các thiết bị
- Theo dõi trạng thái đơn hàng realtime

2.3.2.3. Ứng dụng trong dự án

- Lưu thông tin: users, orders, drinks, categories, toppings, vouchers, addresses, ratings
- Theo dõi trạng thái đơn hàng với `ValueEventListener`
- Cập nhật realtime khi có thay đổi từ admin
- Lưu đánh giá và rating của khách hàng

2.3.3. Google Distance Matrix API

2.3.3.1. Giới thiệu

Google Distance Matrix API là dịch vụ của Google Maps. Tính khoảng cách và thời gian di chuyển giữa các điểm dựa trên tọa độ GPS. Sử dụng dữ liệu bản đồ để tính đường đi tối ưu.

2.3.3.2. Mục đích

- Tính khoảng cách thực tế giữa cửa hàng và địa chỉ giao hàng
- Tính phí vận chuyển dựa trên khoảng cách

2.3.3.3. Ứng dụng trong dự án

- Tính khoảng cách từ cửa hàng (tọa độ cố định) đến địa chỉ giao hàng
- Tính phí vận chuyển theo block (3km/block, 5.000 VND/block)
- Hiển thị khoảng cách cho khách hàng khi đặt hàng
- Sử dụng endpoint: <https://maps.googleapis.com/maps/api/distancematrix/json>

2.3.4. CoinGecko API

2.3.4.1. Giới thiệu

CoinGecko API là dịch vụ công khai cung cấp tỷ giá cryptocurrency theo thời gian thực. Hỗ trợ nhiều loại tiền tệ và coin, cập nhật realtime.

2.3.4.2. Mục đích

- Lấy tỷ giá ETH/VND realtime
- Tính số lượng ETH tương ứng với số tiền VND cần thanh toán

2.3.4.3. Ứng dụng trong dự án

- Lấy tỷ giá ETH/VND khi khách hàng chọn thanh toán blockchain
- Tính số lượng ETH (chuyển sang Wei) cần thanh toán
- Tạo QR code thanh toán với số tiền chính xác
- Sử dụng endpoint: <https://api.coingecko.com/api/v3/simple/price>

2.3.5. VNPay SDK

2.3.5.1. Giới thiệu

VNPay SDK là bộ công cụ thanh toán trực tuyến của VNPay. Hỗ trợ thanh toán qua ngân hàng, ví điện tử. Cung cấp SDK cho Android, hỗ trợ sandbox để test.

2.3.5.2. Mục đích

- Cho phép khách hàng thanh toán đơn hàng qua ngân hàng
- Tích hợp phương thức thanh toán phổ biến tại Việt Nam

2.3.5.3. Ứng dụng trong dự án

- Tạo URL thanh toán với thông tin đơn hàng (sử dụng VnPayHelper)
- Mở VNPay SDK để khách hàng thanh toán
- Nhận kết quả thanh toán qua deep link callback (shopfee://vnpay-result)
- Xác thực chữ ký HMAC-SHA512 để đảm bảo an toàn
- Hỗ trợ sandbox mode để test

2.3.6. Sepolia RPC API (Blockchain Payment)

2.3.6.1. Giới thiệu

Sepolia RPC API là API tương tác với Ethereum Sepolia Testnet .Cho phép đọc giao dịch, trạng thái blockchain, và kiểm tra transaction. Sepolia là testnet của Ethereum dùng để kiểm thử.

2.3.6.2. Mục đích

- Kiểm tra giao dịch blockchain đã được xác nhận
- Xác thực thanh toán bằng cryptocurrency
- Theo dõi transaction trên blockchain

2.3.6.3. Ứng dụng trong dự án

- Kiểm tra transaction hash trên Sepolia testnet
- Xác minh số tiền và địa chỉ nhận
- Auto-polling để phát hiện giao dịch thanh toán
- Lưu transaction hash vào đơn hàng
- Sử dụng RPC methods

2.3.7. Google Play Services Location

2.3.7.1. Giới thiệu

Google Play Services Location là dịch vụ vị trí của Google. Cung cấp vị trí GPS của thiết bị, tích hợp với Google Maps, độ chính xác cao.

2.3.7.2. Mục đích

- Lấy vị trí GPS hiện tại của người dùng
- Hỗ trợ tính toán phí vận chuyển chính xác

2.3.7.3. Ứng dụng trong dự án

- Lấy tọa độ GPS khi khách hàng chọn địa chỉ giao hàng
- Tính khoảng cách từ cửa hàng đến địa chỉ giao hàng
- Hỗ trợ tính phí vận chuyển khi không có tọa độ địa chỉ

CHƯƠNG 3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG THÔNG TIN

3.1. Phân tích yêu cầu hệ thống

3.1.1. Yêu cầu chức năng (Functional Requirements)

3.1.1.1. Quản lý người dùng

- Hệ thống cho phép người dùng đăng ký tài khoản mới với email và mật khẩu
- Hệ thống xác thực người dùng khi đăng nhập
- Hệ thống hỗ trợ chức năng quên mật khẩu
- Hệ thống phân quyền User và Admin (Admin có email format @admin.com)
- Hệ thống quản lý hạng thành viên (Thường, Bạc, Vàng, Kim cương) dựa trên tổng chi tiêu

3.1.1.2. Quản lý sản phẩm

- Hệ thống hiển thị danh sách sản phẩm theo danh mục
- Hệ thống hiển thị banner sản phẩm nổi bật (featured)
- Hệ thống cho phép xem chi tiết sản phẩm (ảnh, mô tả, giá, đánh giá)
- Hệ thống hỗ trợ tìm kiếm sản phẩm
- Hệ thống cho phép Admin thêm, sửa, xóa sản phẩm và danh mục

3.1.1.3. Quản lý giỏ hàng

- Hệ thống lưu trữ giỏ hàng local bằng Room Database
- Hệ thống cho phép thêm, xóa, cập nhật số lượng sản phẩm trong giỏ hàng
- Hệ thống tính tổng tiền giỏ hàng tự động
- Hệ thống cho phép tùy chỉnh sản phẩm (size, đá, đường, topping)

3.1.1.4. Quản lý đơn hàng

- Hệ thống cho phép tạo đơn hàng từ giỏ hàng
- Hệ thống tính phí vận chuyển dựa trên khoảng cách (Google Maps API)
- Hệ thống cho phép áp dụng voucher khi đặt hàng
- Hệ thống quản lý trạng thái đơn hàng (Mới, Đang xử lý, Đang giao, Hoàn thành)
- Hệ thống cho phép xem lịch sử đơn hàng
- Hệ thống cho phép Admin xem và cập nhật trạng thái đơn hàng

3.1.1.5. Thanh toán

- Hệ thống tích hợp thanh toán VNPay

- Hệ thống tích hợp thanh toán Blockchain (Ethereum Sepolia)
- Hệ thống quy đổi tỉ giá ETH/VND theo thời gian thực (CoinGecko API)
- Hệ thống xác nhận thanh toán và cập nhật trạng thái đơn hàng

3.1.1.6. Đánh giá và phản hồi

- Hệ thống cho phép đánh giá sản phẩm (sao)
- Hệ thống cho phép viết review cho đơn hàng
- Hệ thống hiển thị đánh giá trung bình và số lượng review

3.1.1.7. Quản lý voucher

- Hệ thống quản lý voucher (thêm, sửa, xóa - Admin)
- Hệ thống hỗ trợ voucher theo phần trăm và số tiền cố định
- Hệ thống kiểm tra điều kiện áp dụng voucher (số tiền tối thiểu, hạng thành viên)
- Hệ thống quản lý số lượng voucher (tổng số và đã sử dụng)

3.1.1.8. Quản lý địa chỉ

- Hệ thống cho phép thêm, sửa, xóa địa chỉ giao hàng
- Hệ thống lưu tọa độ GPS của địa chỉ
- Hệ thống cho phép đặt địa chỉ mặc định

3.1.2. Yêu cầu phi chức năng (Non-Functional Requirements)

3.1.2.1. Hiệu năng

- Ứng dụng phải phản hồi trong vòng 2 giây cho các thao tác cơ bản
- Ảnh sản phẩm được cache để tải nhanh
- Dữ liệu giỏ hàng được lưu local để hoạt động offline

3.1.2.2. Bảo mật

- Mật khẩu được mã hóa bằng Firebase Auth
- API keys được bảo vệ và không commit lên Git
- Phân quyền rõ ràng giữa User và Admin

3.1.2.3. Khả năng mở rộng

- Kiến trúc MVVM dễ bảo trì và mở rộng
- Sử dụng Dependency Injection (Hilt) để quản lý dependencies

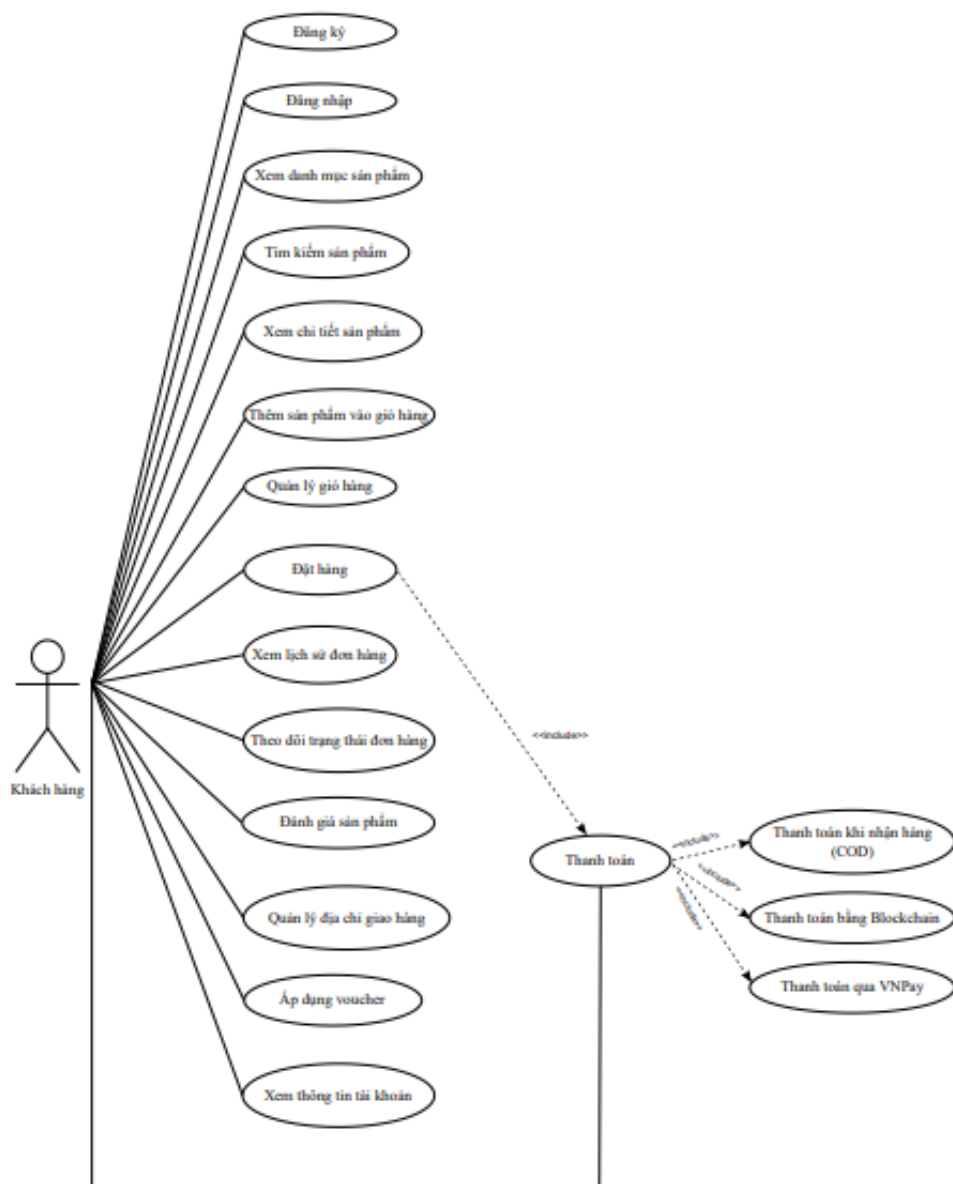
3.1.2.4. Giao diện

- Giao diện tuân thủ Material Design Guidelines
- Hỗ trợ nhiều kích thước màn hình (responsive)

3.2. Biểu đồ use case

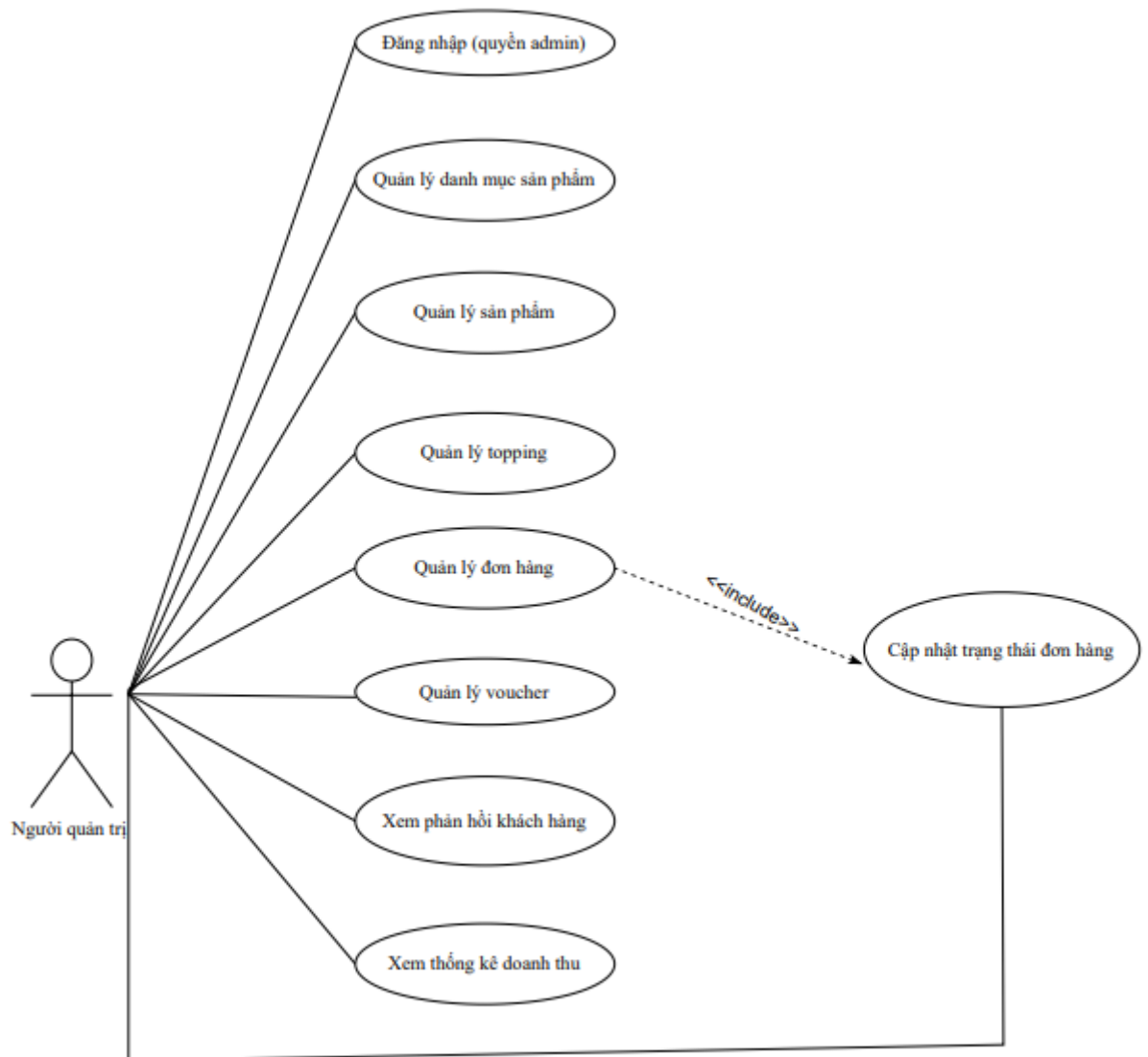
3.2.1. Biểu đồ Use Case tổng quát

Khách hàng:



Hình 3.13 Biểu đồ Use Case tổng quát Khách hàng

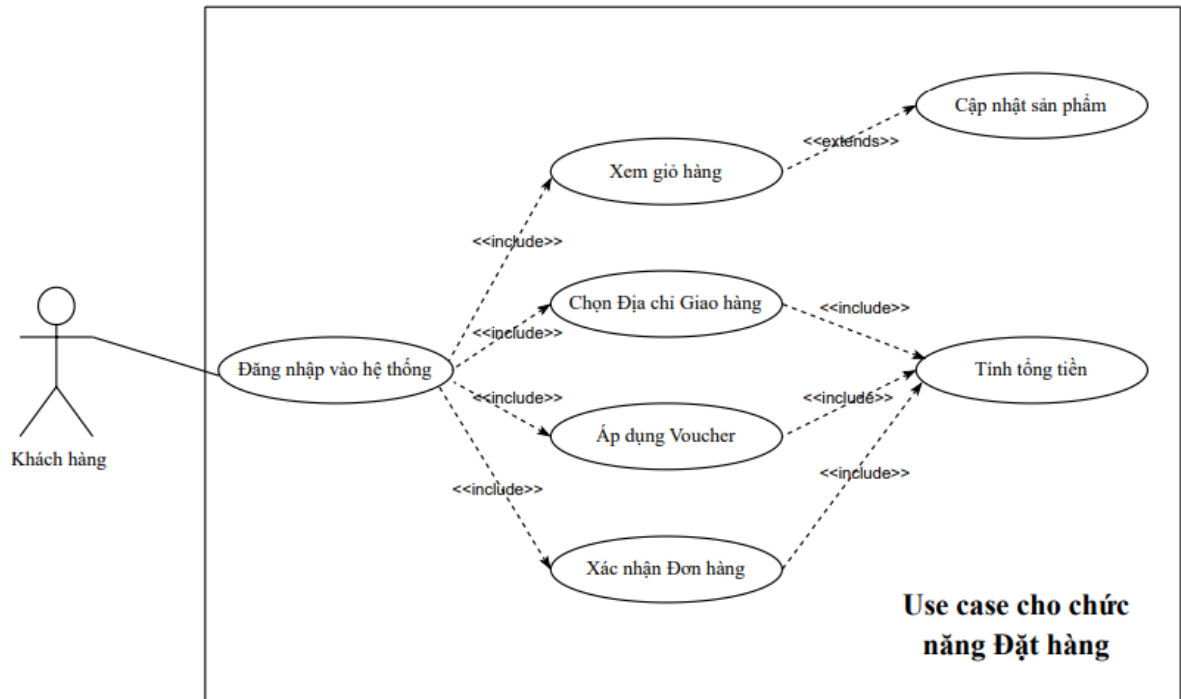
Người quản trị:



Hình 3.14 Biểu đồ Use Case tổng quát Người quản trị

3.2.2. Biểu đồ Use Case chi tiết

3.2.2.1. Use case cho chức năng Đặt hàng



Hình 3.15 Use case cho chức năng Đặt hàng

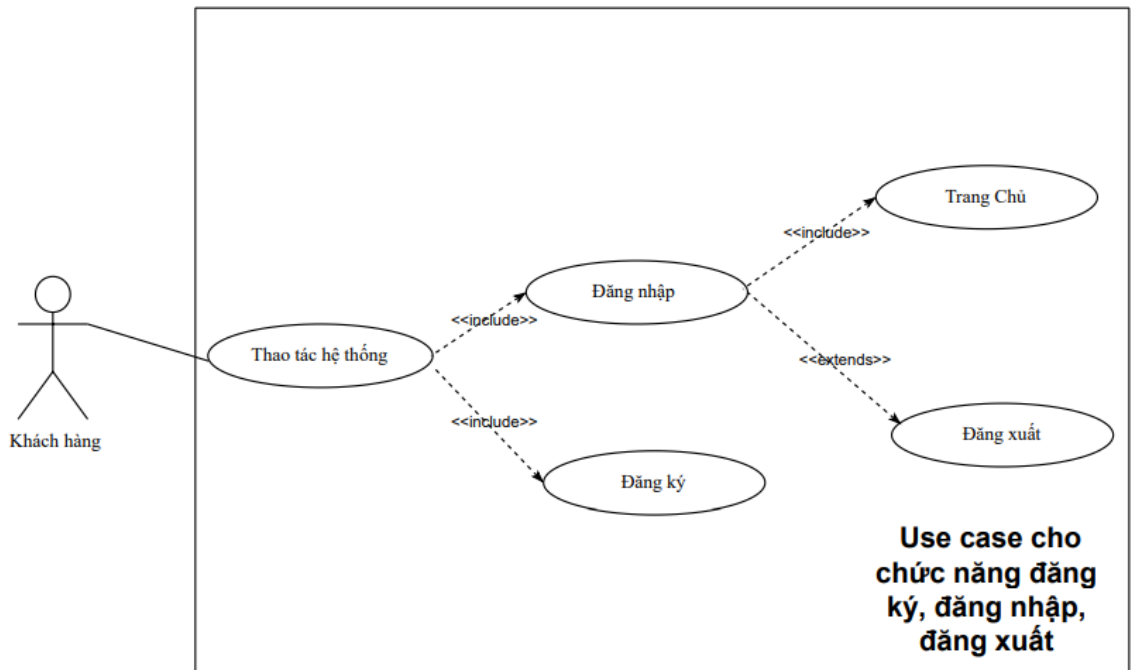
Bảng 3.1. Use case cho chức năng Đặt hàng

Tác nhân	Khách hàng
Mô tả	Người dùng hệ thống có nhiệm vụ thực hiện đặt hàng, bao gồm các thao tác: xem giỏ hàng, chọn địa chỉ giao hàng, áp dụng voucher, xem tổng tiền và xác nhận đơn hàng
Tiền điều kiện	Khách hàng đã đăng nhập thành công vào hệ thống và giỏ hàng có ít nhất 1 sản phẩm
Luồng xử lý chính	1. Khách hàng đăng nhập vào hệ thống 2. Khách hàng xem giỏ hàng của mình 3. Khách hàng có thể cập nhật số lượng hoặc xóa sản phẩm 4. Khách hàng chọn địa chỉ giao hàng hoặc thêm địa chỉ mới 5. Hệ thống tính khoảng cách và phí vận chuyển dựa trên địa chỉ

	6. Khách hàng có thể áp dụng voucher để giảm giá 7. Hệ thống kiểm tra điều kiện voucher 8. Hệ thống tính lại tổng tiền 9. Khách hàng xem tổng tiền và xác nhận đơn hàng
--	--

Mô tả quá trình khách hàng hoàn thành đặt hàng trên hệ thống. Khách hàng trước tiên đăng nhập vào hệ thống, sau đó xem giỏ hàng, chọn địa chỉ giao hàng, áp dụng voucher (nếu có), xem tổng tiền và cuối cùng xác nhận đơn hàng để hệ thống ghi nhận. Quy trình này thường được triển khai trên các nền tảng thương mại điện tử, nơi mọi bước từ xem giỏ hàng, chọn địa chỉ đến xác nhận đơn hàng đều được tự động hóa và quản lý chặt chẽ. Use case này không chỉ đơn thuần là một quy trình, mà còn đảm bảo tính liền mạch, minh bạch và dễ sử dụng cho người dùng trong môi trường mua sắm trực tuyến.

3.2.2.2. Use case cho chức năng đăng ký, đăng nhập, đăng xuất



Hình 3.16 Use case cho chức năng đăng ký, đăng nhập, đăng xuất

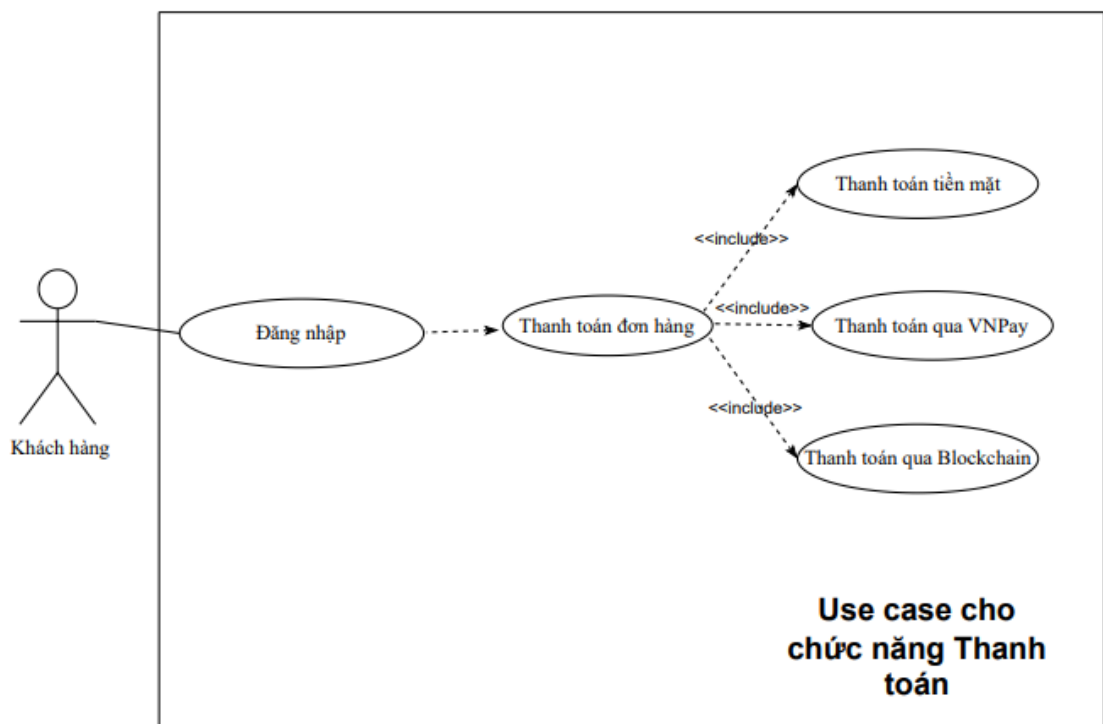
Bảng 3.2. Use case cho chức năng đăng ký, đăng nhập, đăng xuất

Tác nhân	Khách hàng
Mô tả	Người sử dụng hệ thống để thực hiện các thao tác liên quan đến việc đăng nhập, đăng ký tài khoản, và đăng xuất
Tiền điều kiện	Người dùng cần có kết nối Internet để truy cập hệ thống. Nếu thực hiện đăng nhập, cần có tài khoản hợp lệ

Luồng xử lý chính	1. Người dùng truy cập giao diện hệ thống 2. Người dùng có thể chọn một trong hai thao tác sau • Đăng ký tài khoản - Nhập email, mật khẩu và thông tin cá nhân • Đăng nhập hệ thống - Nhập email và mật khẩu 3. Hệ thống xử lý yêu cầu và phản hồi kết quả tương ứng
--------------------------	--

Đây là thao tác cơ bản và bắt buộc mà người dùng thực hiện để tương tác với hệ thống. Khách hàng có thể bắt đầu bằng việc đăng ký tài khoản, cung cấp thông tin cần thiết để tạo một hồ sơ người dùng mới. Sau khi có tài khoản, người dùng tiến hành đăng nhập để truy cập hệ thống và thực hiện các chức năng như xem sản phẩm, đặt hàng hoặc quản lý (nếu là Admin). Khi hoàn tất các thao tác, người dùng có thể đăng xuất để kết thúc phiên làm việc. Quy trình này đảm bảo việc quản lý truy cập của người dùng được thực hiện an toàn và rõ ràng, thường được triển khai trong các hệ thống thương mại điện tử hoặc các ứng dụng trực tuyến.

3.2.2.3. Use case cho chức năng Thanh toán



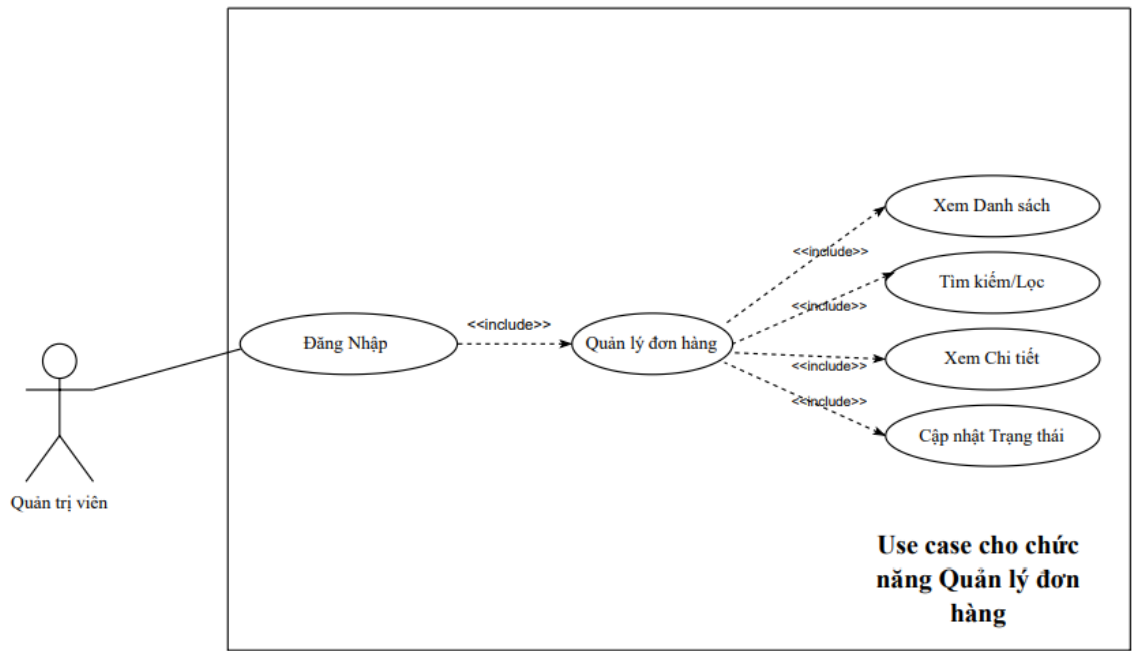
Hình 3.17 Use case cho chức năng Use case cho chức năng Thanh toán

Bảng 3.3. Use case cho chức năng Use case cho chức năng Thanh toán

Tác nhân	Khách hàng
Mô tả	Người sử dụng hệ thống để thực hiện thanh toán cho đơn hàng đã đặt, có thể chọn thanh toán khi nhận hàng hoặc qua VNPAY hoặc Blockchain
Tiền điều kiện	Khách hàng đã đăng nhập vào hệ thống, đã đặt hàng thành công và đơn hàng có trạng thái
Luồng xử lý chính	1. Khách hàng xem thông tin đơn hàng cần thanh toán 2. Phương thức khi nhận hàng 3. Phương thức VNPAY: - Khách hàng chọn "VNPAY" - Hệ thống tạo Payment URL (có chữ ký bảo mật) - Mở VNPAY SDK - Khách hàng thực hiện thanh toán - VNPAY gửi callback (deep link) - Hệ thống xác minh chữ ký & cập nhật trạng thái 4. Phương thức Blockchain (ETH) - Khách hàng chọn "Blockchain" - Hệ thống lấy tỉ giá ETH/VND (CoinGecko API) - Tính số lượng ETH & tạo QR code (địa chỉ ví, số lượng ETH) - Khách hàng quét QR code và xác nhận giao dịch (MetaMask, Trust Wallet) - Hệ thống poll blockchain để xác nhận - Hệ thống lưu transaction hash & cập nhật trạng thái 5. Hiện thị thông báo "Thanh toán thành công" và hóa đơn

Hệ thống hỗ trợ phương thức thanh toán hiện đại: Thanh toán khi nhận hàng, thanh toán VNPAY (thanh toán truyền thống qua ví điện tử) và Blockchain (thanh toán bằng tiền điện tử Ethereum). Mỗi phương thức có quy trình riêng nhưng đều đảm bảo tính bảo mật và minh bạch trong giao dịch. Use case này đảm bảo khách hàng có nhiều lựa chọn thanh toán phù hợp với nhu cầu của mình.

3.2.2.4. Use case cho chức năng Quản lý đơn hàng (Admin)



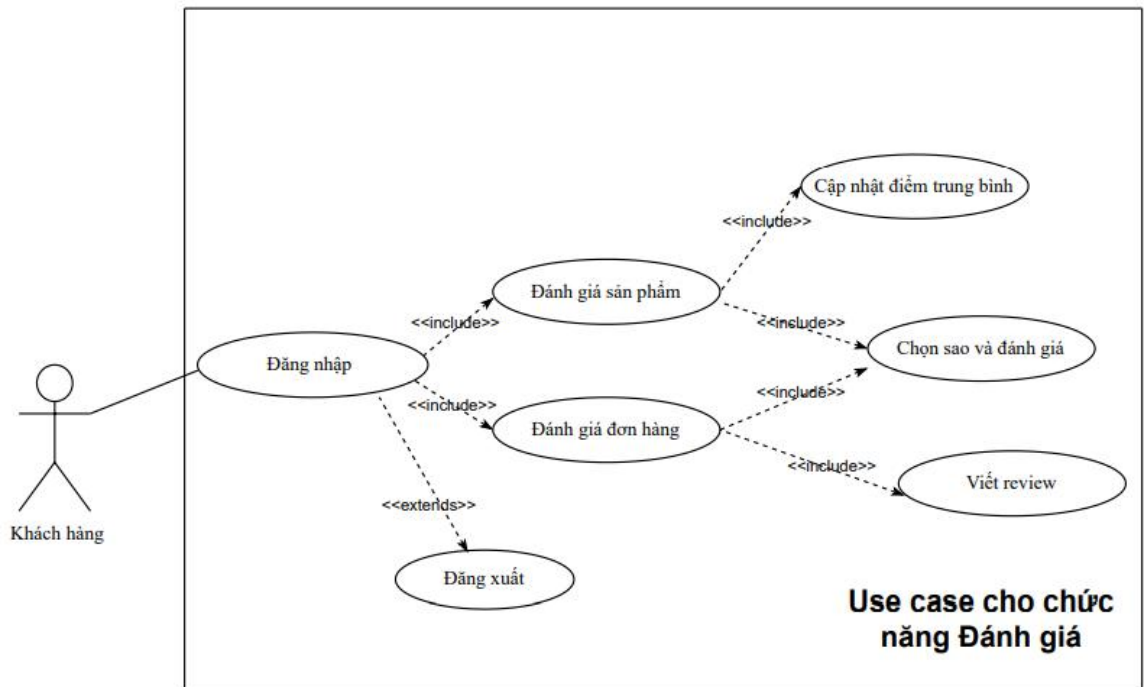
Hình 3.18 Use case cho chức năng Quản lý đơn hàng

Bảng 3.4. Use case cho chức năng Quản lý đơn hàng

Tác nhân	Quản trị viên
Mô tả	Cho phép Quản trị viên xem, tìm kiếm, lọc, và cập nhật trạng thái của tất cả đơn hàng trong hệ thống
Tiền điều kiện	Quản trị viên đã đăng nhập thành công với quyền quản trị viên
Luồng xử lý chính	1. Quản trị viên truy cập "Quản lý đơn hàng" 2. Hệ thống hiển thị danh sách đơn hàng (sắp xếp mới nhất). 3. Quản trị viên thực hiện các tác vụ - Xem Danh sách - Tìm kiếm - Lọc - Xem Chi tiết - Cập nhật Trạng thái 4. Hệ thống cập nhật trạng thái và gửi thông báo cho khách hàng 5. Hệ thống hiển thị thông báo "Cập nhật thành công"

Đây là một trong những chức năng quan trọng mà quản trị viên có thể thực hiện để quản lý thông tin liên quan đến đơn hàng. Sau khi đăng nhập, quản trị viên truy cập vào mục quản lý đơn hàng, nơi họ có thể xem danh sách các đơn hàng hiện có, tìm kiếm đơn hàng cụ thể, lọc theo trạng thái, xem chi tiết và cập nhật trạng thái đơn hàng. Quy trình này giúp đảm bảo hệ thống tổ chức đơn hàng một cách rõ ràng, đồng thời cho phép cập nhật và xử lý thông tin một cách linh hoạt. Đây là một tính năng quan trọng trong các hệ thống quản lý bán hàng trực tuyến, hỗ trợ quản lý tốt các đơn hàng và theo dõi tiến trình giao hàng.

3.2.2.5. Use case cho chức năng Đánh giá



Hình 3.19 Use case cho chức năng Đánh giá

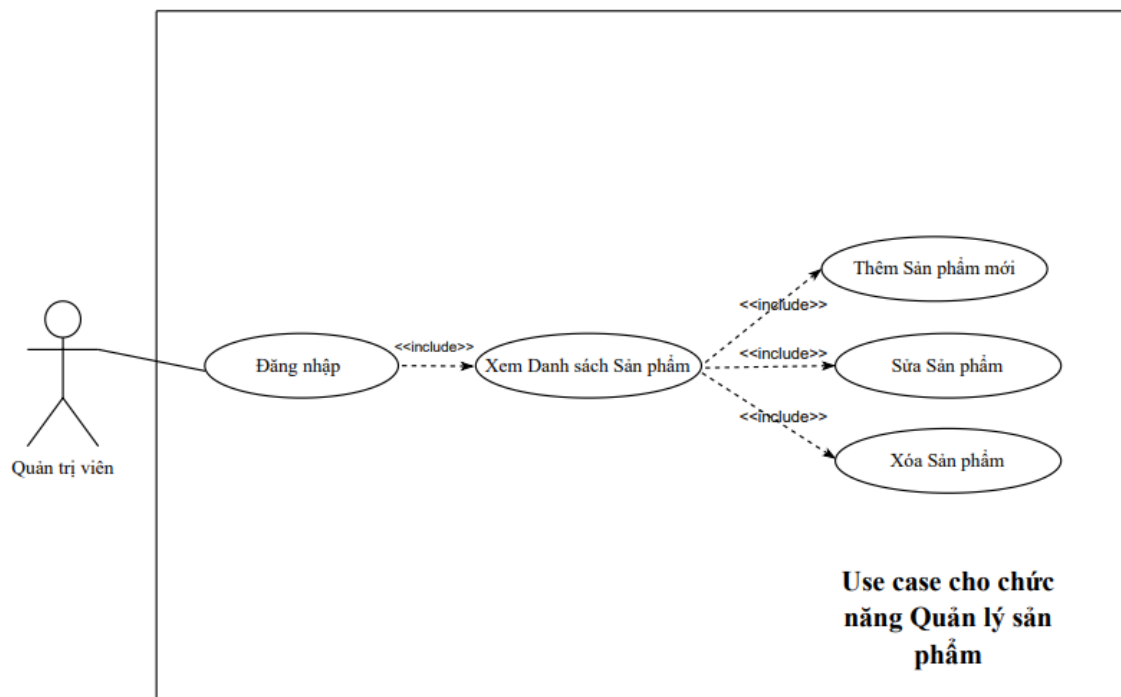
Bảng 3.5. Use case cho chức năng Đánh giá

Tác nhân	Khách hàng
Mô tả	Cho phép Khách hàng cung cấp phản hồi dưới dạng số sao (1-5) và bài viết (review) cho sản phẩm hoặc đơn hàng đã hoàn thành.
Tiền điều kiện	Khách hàng đã đăng nhập thành công Đối với đánh giá Đơn hàng: Đơn hàng phải ở trạng thái Hoàn thành

Luồng xử lý chính	1. Đánh giá Sản phẩm - Khách hàng chọn sản phẩm - Nhấn "Đánh giá sản phẩm" - Chọn số sao và viết review - Nhấn "Gửi" 2. Đánh giá Đơn hàng - Khách hàng chọn đơn hàng đã hoàn thành - Nhấn "Đánh giá đơn hàng" - Chọn số sao và viết review - Nhấn "Gửi" 3. Hiện thị thông báo "Đánh giá đã được gửi thành công"
--------------------------	---

Nhằm tương tác với hệ thống để đánh giá chất lượng sản phẩm và dịch vụ. Sau khi đăng nhập, khách hàng có thể đánh giá sản phẩm khi xem chi tiết hoặc đánh giá đơn hàng sau khi nhận hàng. Hệ thống cho phép khách hàng chọn số sao và viết review để chia sẻ trải nghiệm của mình. Quy trình này giúp khách hàng dễ dàng phản hồi về sản phẩm và dịch vụ, đảm bảo sự rõ ràng và minh bạch trong việc đánh giá chất lượng. Đây là một tính năng phổ biến trong các hệ thống thương mại điện tử, giúp khách hàng khác tham khảo trước khi quyết định mua.

3.2.2.6. Use case cho chức năng Quản lý sản phẩm (Admin)



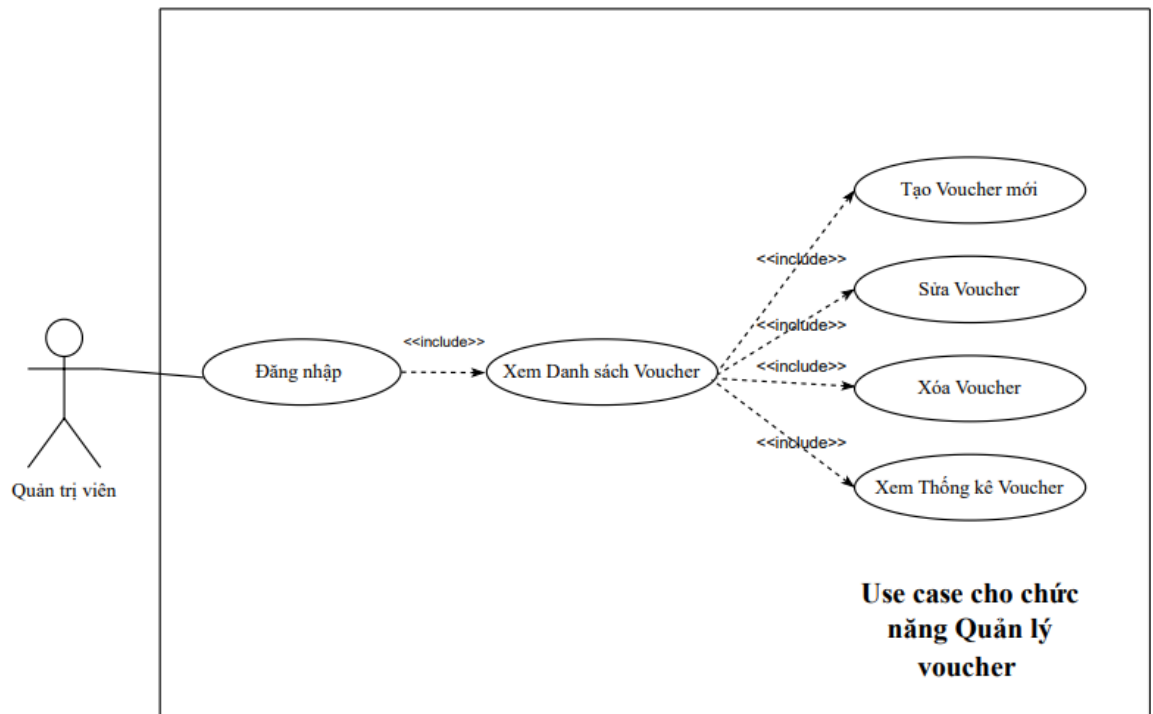
Hình 3.20 Use case cho chức năng Quản lý sản phẩm

Bảng 3.6. Use case cho chức năng Quản lý sản phẩm

Tác nhân	Quản trị viên
Mô tả	Cho phép Quản trị viên thực hiện các thao tác Thêm (Create), Sửa (Update), Xóa (Delete) sản phẩm, quản lý ảnh và thiết lập giá sản phẩm
Tiền điều kiện	Quản trị viên đã đăng nhập thành công với quyền quản trị viên
Luồng xử lý chính	1. Quản trị viên truy cập chức năng "Quản lý sản phẩm" 2. Hệ thống hiển thị danh sách sản phẩm hiện có 3. Quản trị viên thực hiện các tác vụ - Thêm mới - Sửa - Xóa 4. Hệ thống xử lý yêu cầu và cập nhật kết quả vào Firebase 5. Quản trị viên có thể đăng xuất

Đây là cách quản trị viên sử dụng hệ thống để quản lý các sản phẩm. Sau khi đăng nhập, quản trị viên có thể truy cập chức năng quản lý sản phẩm, từ đó thực hiện các tác vụ như xem danh sách sản phẩm hiện có, thêm sản phẩm mới, sửa đổi thông tin sản phẩm, xóa sản phẩm không cần thiết, upload ảnh và thiết lập giá. Use case này đảm bảo tính tổ chức và tiện lợi trong việc quản lý danh sách sản phẩm, giúp quản trị viên duy trì cơ sở dữ liệu sản phẩm chính xác và dễ dàng sử dụng trong các hoạt động như hiển thị cho khách hàng hoặc cập nhật thông tin.

3.2.2.7. Use case cho chức năng Quản lý voucher (Admin)



Hình 3.21 Use case cho chức năng Quản lý voucher

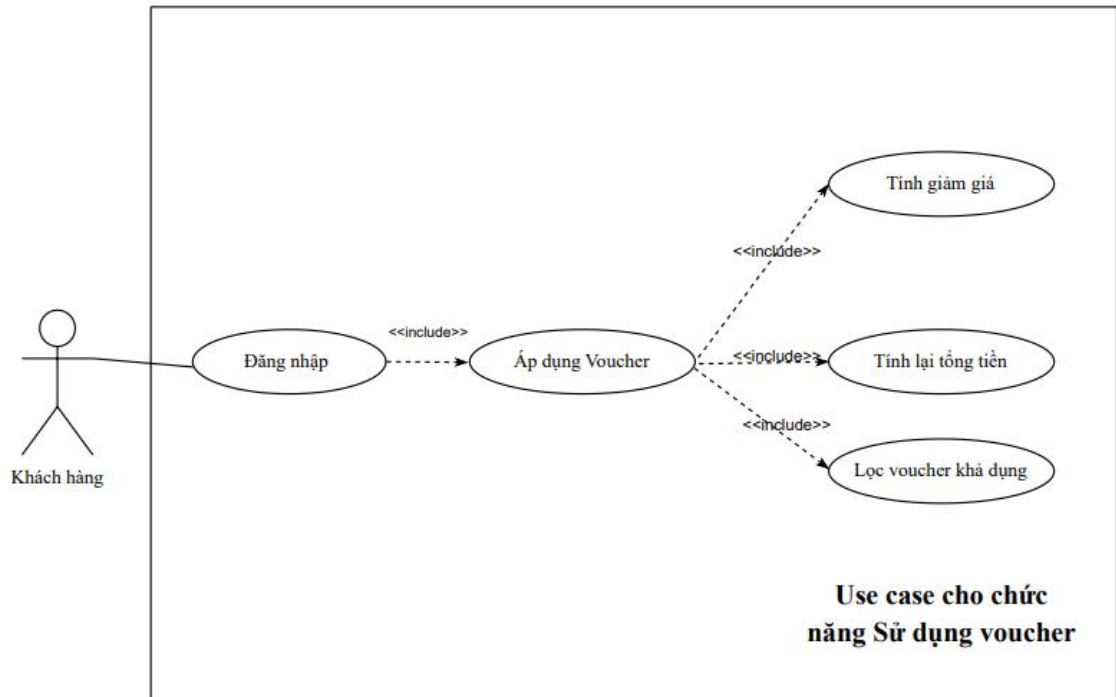
Bảng 3.7. Use case cho chức năng Quản lý voucher

Tác nhân	Quản trị viên
Mô tả	Cho phép Quản trị viên thực hiện các thao tác Thêm (Create), Sửa (Update), Xóa (Delete), và thiết lập điều kiện cho các mã giảm giá (voucher)
Tiền điều kiện	Quản trị viên đã đăng nhập thành công với quyền quản trị viên
Luồng xử lý chính	1. Quản trị viên truy cập chức năng "Quản lý voucher" 2. Hệ thống hiển thị danh sách voucher hiện có. 3. Quản trị viên thực hiện các tác vụ: - Tạo mới - Sửa - Xóa - Xem thống kê 4. Hệ thống xử lý yêu cầu và cập nhật kết quả vào Firebase. 5. Quản trị viên có thể đăng xuất

Đây là một trong những chức năng mà quản trị viên có thể thực hiện để quản lý thông tin liên quan đến voucher. Sau khi đăng nhập, quản trị viên truy cập vào mục quản lý

voucher, nơi họ có thể xem danh sách các voucher hiện có, tạo mới voucher, chỉnh sửa thông tin của voucher đã tồn tại, hoặc xóa voucher không còn phù hợp. Quy trình này giúp đảm bảo hệ thống tổ chức voucher một cách rõ ràng, đồng thời cho phép cập nhật và xử lý thông tin một cách linh hoạt. Đây là một tính năng quan trọng trong các hệ thống thương mại điện tử, hỗ trợ quản lý tốt các chương trình khuyến mãi và giảm giá.

3.2.2.8. Usecase cho chức năng Sử dụng voucher (Khách hàng)



Hình 3.22 Use case cho chức năng Sử dụng voucher

Bảng 3.8. Use case cho chức năng Sử dụng voucher

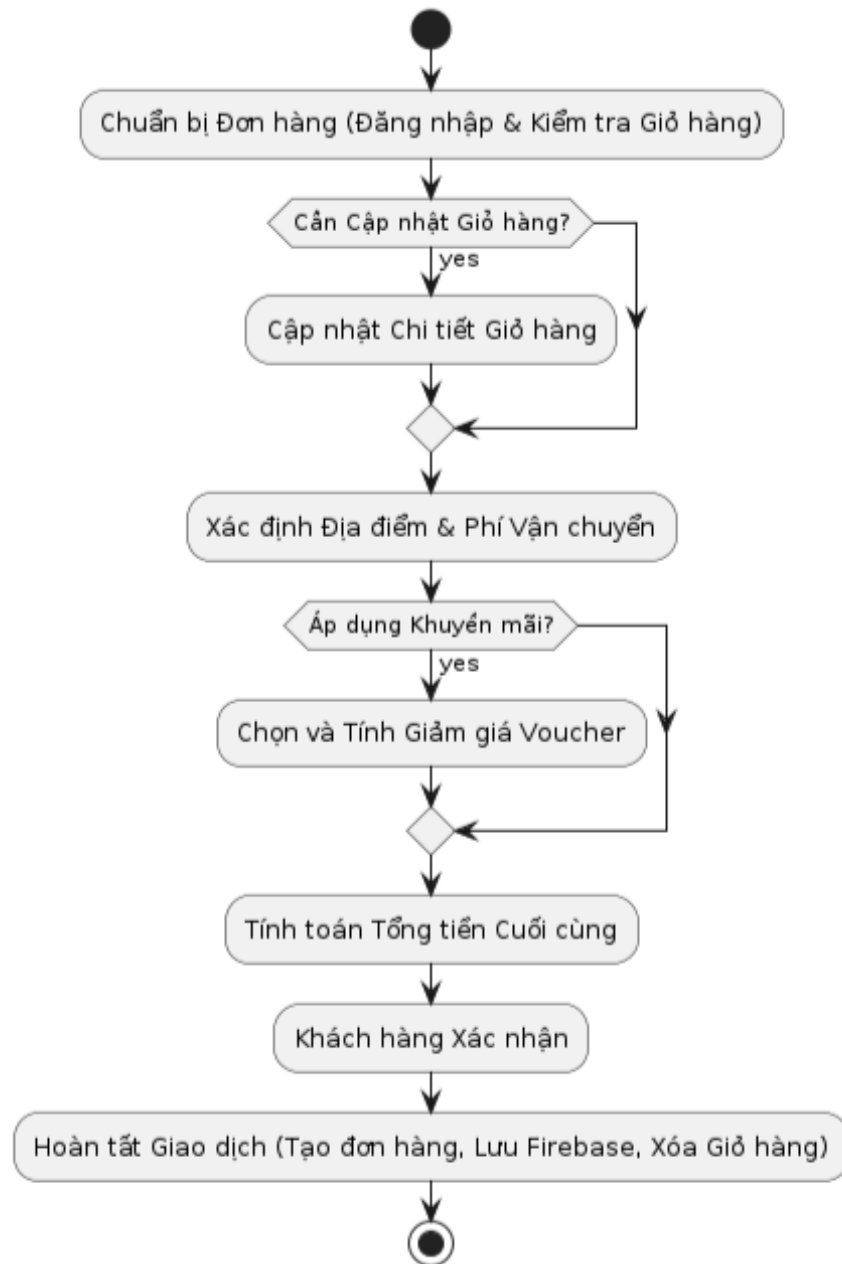
Tác nhân	Khách hàng
Mô tả	Cho phép Khách hàng xem các voucher đủ điều kiện và sử dụng một voucher để giảm giá trị tổng đơn hàng
Tiền điều kiện	1. Khách hàng đã đăng nhập và đang trong quá trình đặt hàng 2. Tổng tiền đơn hàng đáp ứng điều kiện tối thiểu của voucher

Luồng xử lý chính	1 Khách hàng nhấn "Áp dụng voucher" 2 Hệ thống hiển thị danh sách voucher khả dụng. 3 Hệ thống Lọc Voucher dựa trên: - Tổng tiền - Hạng thành viên - Số lượng 4 Khách hàng chọn một voucher 5 Hệ thống kiểm tra lại điều kiện voucher. 6 Hệ thống Tính giảm giá 7 Hệ thống Tính lại tổng tiền 8 Khách hàng xác nhận áp dụng 9 Hệ thống lưu voucherId vào đơn hàng 10 Sau khi thanh toán thành công, hệ thống tăng usedquantity của voucher
--------------------------	---

Nhằm tương tác với hệ thống để sử dụng voucher khi đặt hàng. Sau khi đăng nhập và trong quá trình đặt hàng, khách hàng có thể xem danh sách voucher khả dụng, chọn voucher phù hợp và áp dụng để được giảm giá. Hệ thống tự động kiểm tra điều kiện voucher và tính toán giá trị giảm giá. Quy trình này giúp khách hàng dễ dàng sử dụng voucher để tiết kiệm chi phí, đảm bảo sự rõ ràng và minh bạch trong việc áp dụng khuyến mãi. Đây là một tính năng phổ biến trong các hệ thống thương mại điện tử, giúp tăng trải nghiệm mua sắm và khuyến khích khách hàng quay lại.

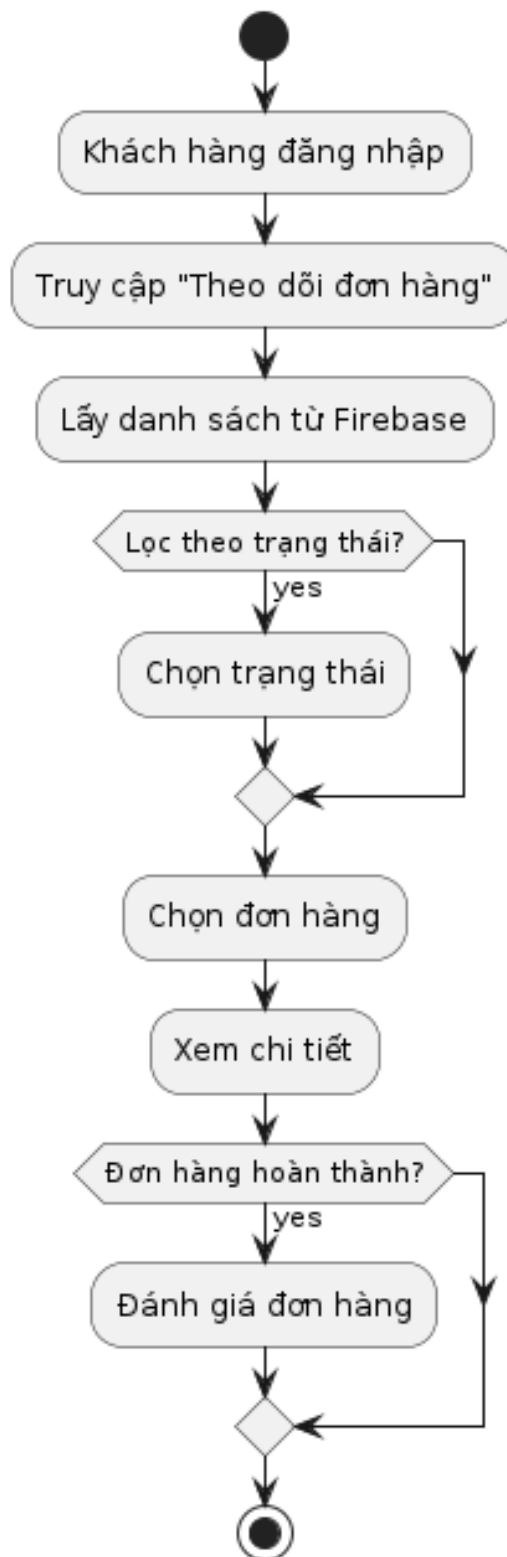
3.3. Biểu đồ hoạt động

3.3.1. Đặt hàng



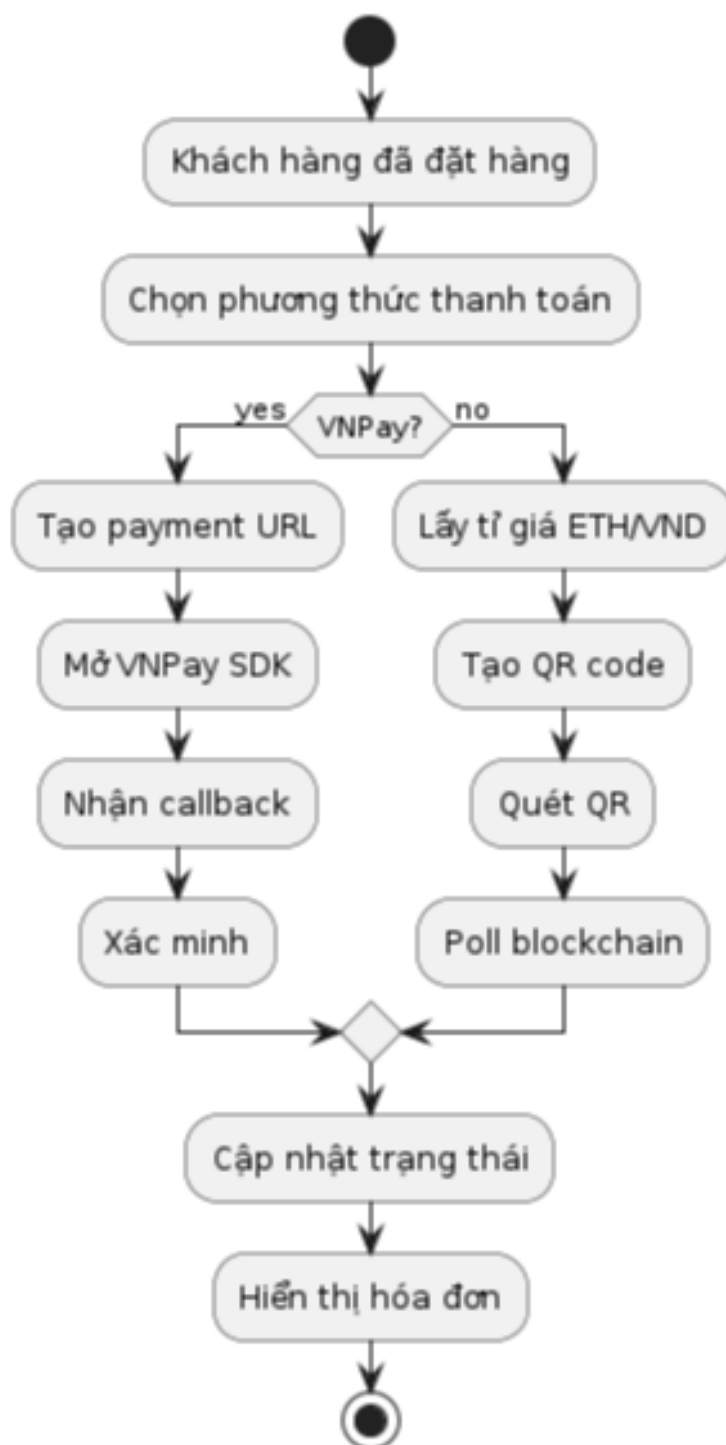
Hình 3.1 biểu đồ chức năng đặt hàng

3.3.2. Quản lý đơn hàng



Hình 3.2 biểu đồ chức năng quản lý đơn hàng

3.3.3. Thanh toán



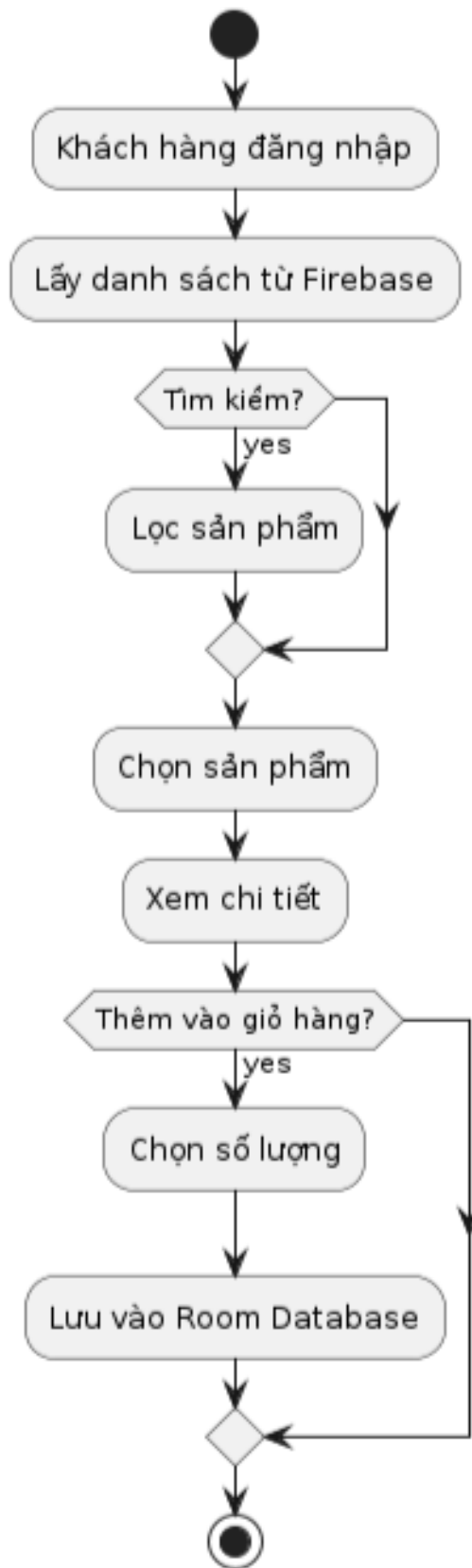
Hình 3.3 biểu đồ chức năng thanh toán

3.3.4. Đánh giá



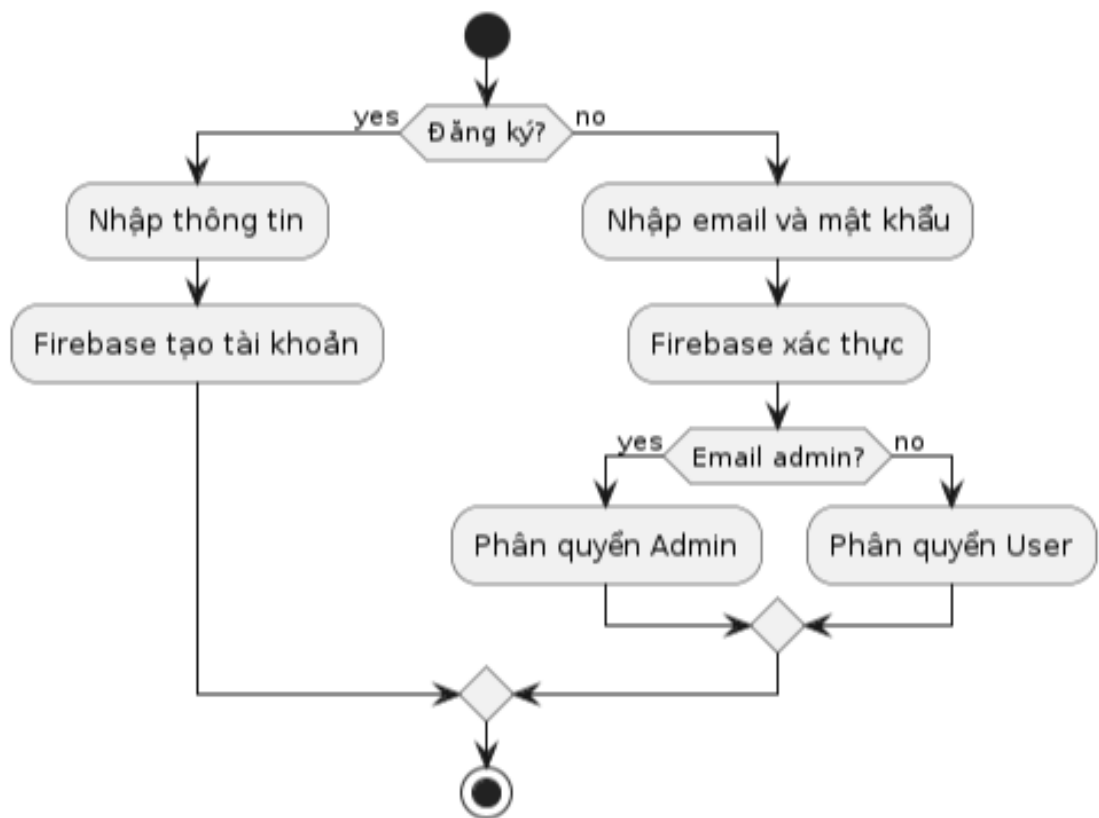
Hình 3.4 biểu đồ chức năng đánh giá

3.3.5. Xem sản phẩm



Hình 3.5 biểu đồ chức năng xem sản phẩm

3.3.6. Đăng ký/Đăng nhập



Hình 3.6 biểu đồ chức năng đăng ký, đăng nhập

3.3.7. Quản lý sản phẩm (Admin)



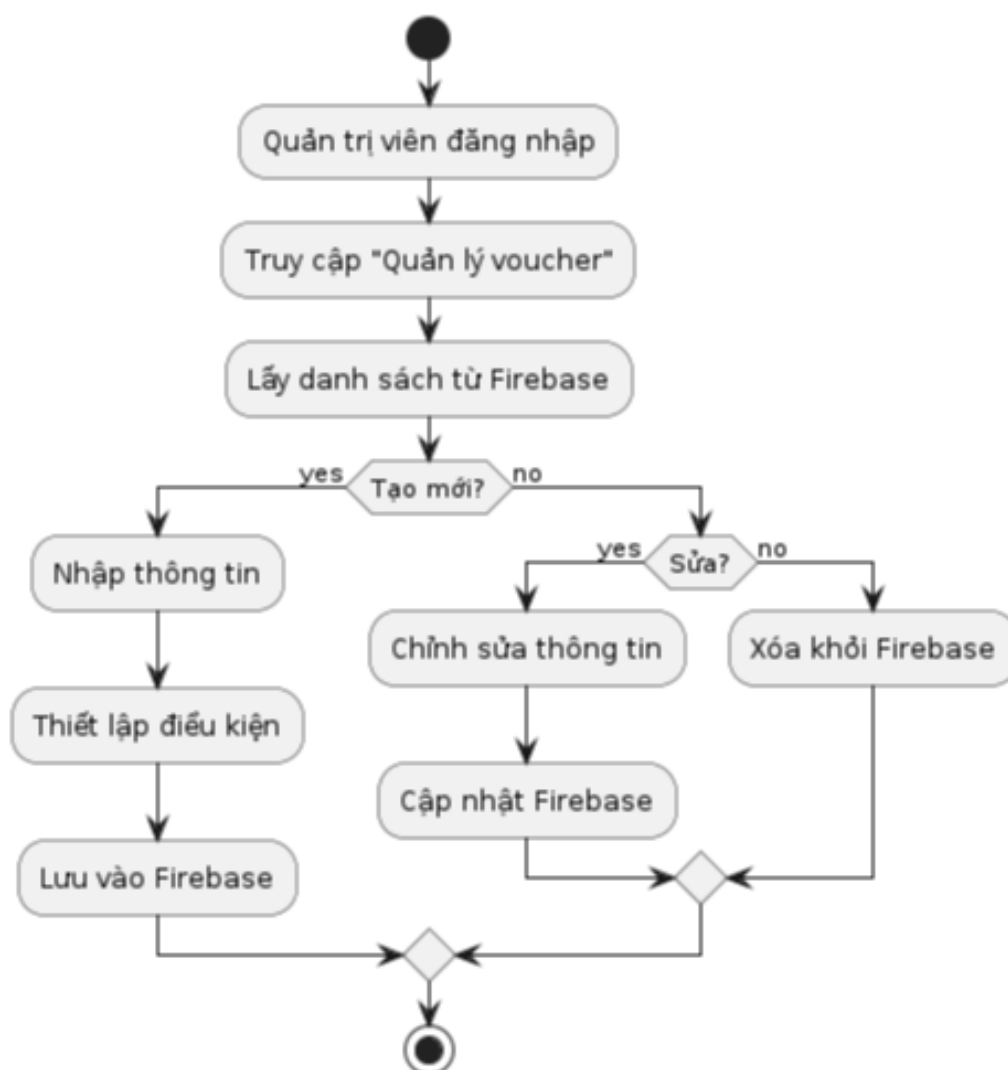
Hình 3.7 biểu đồ chức năng quản lý sản phẩm

3.3.8. Quản lý đơn hàng (Admin)



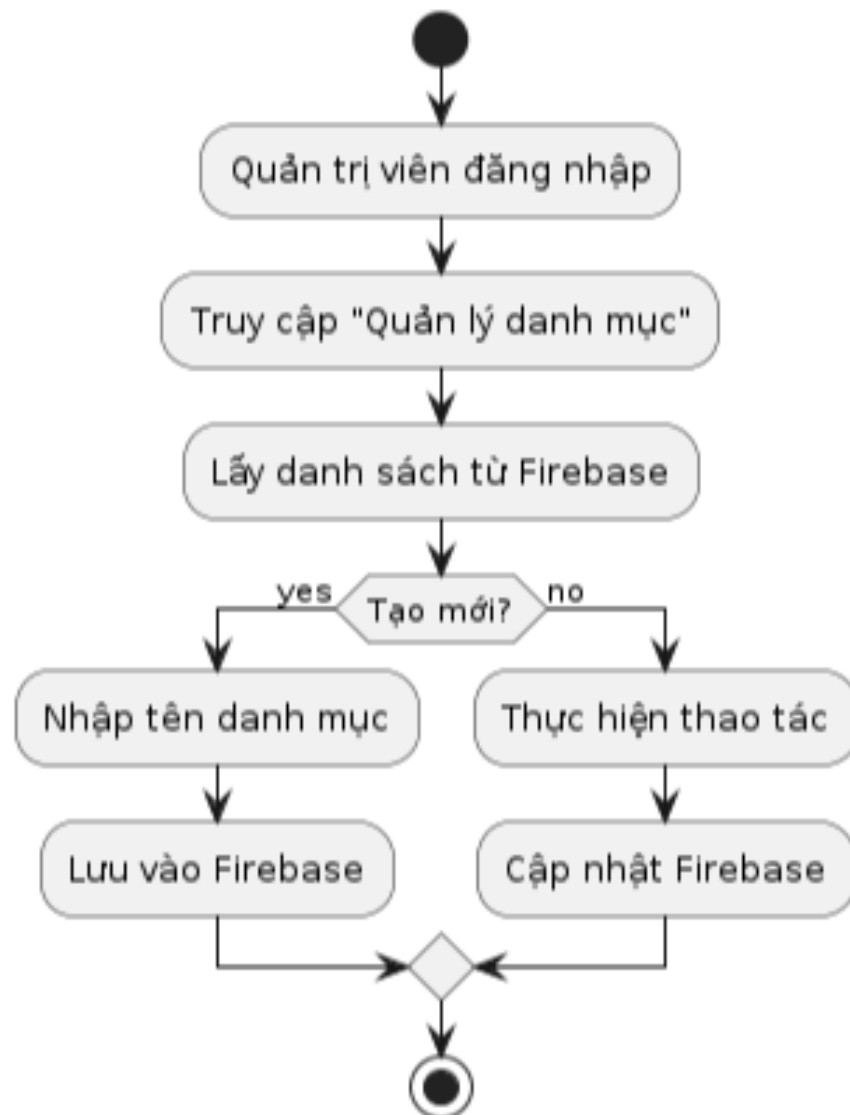
Hình 3.8 biểu đồ chức năng quản lý đơn hàng

3.3.9. Quản lý voucher



Hình 3.9 biểu đồ chức năng voucher

3.3.10. Quản lý danh mục



Hình 3.10 biểu đồ chức năng quản lý danh mục

3.3.11. Quản lý topping



Hình 3.11 biểu đồ chức năng quản lý topping

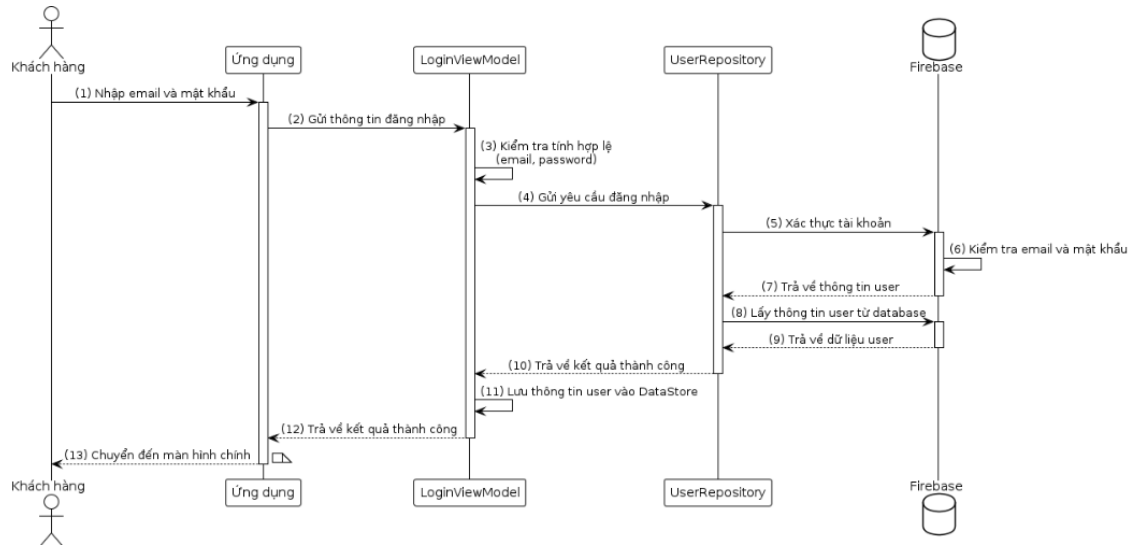
3.3.12. Phản hồi và thống kê



Hình 3.12 biểu đồ chức năng phản hồi và thống kê

3.4. Biểu đồ tuần tự (Sequence Diagram)

3.4.1. Biểu đồ tuần tự đăng nhập

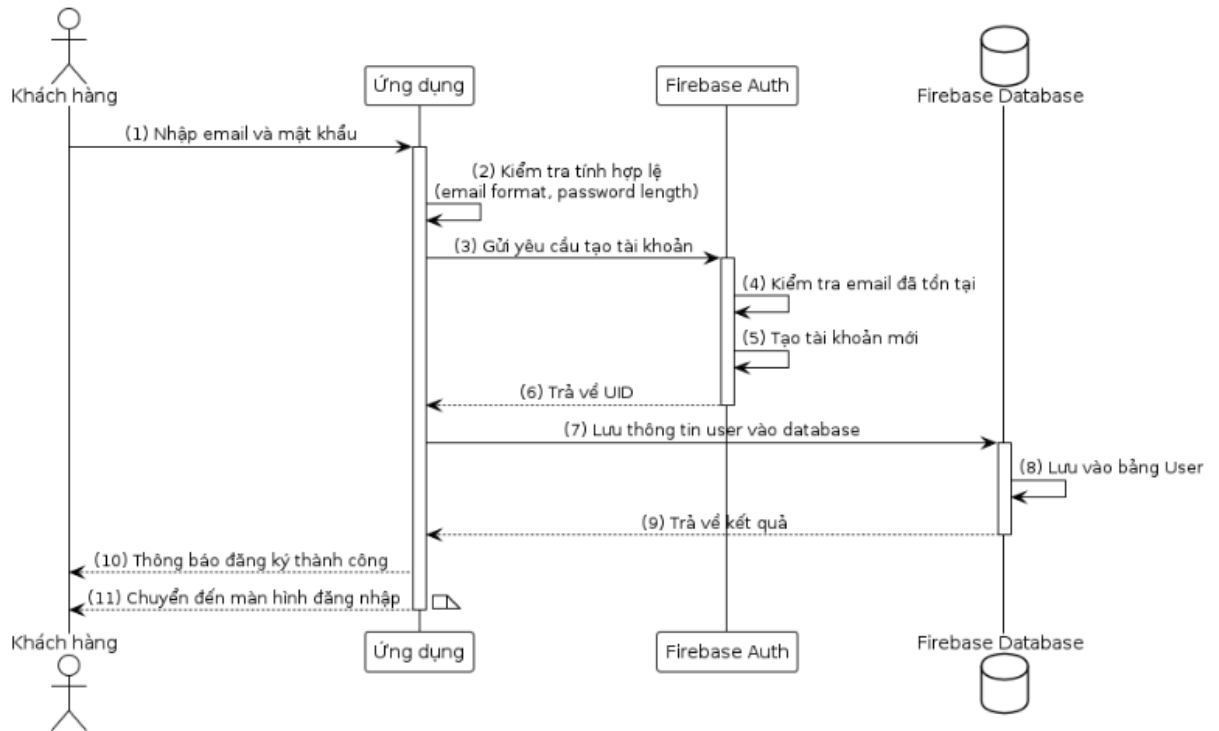


Hình 3.23 Sơ đồ tuần tự chức năng đăng nhập

Người dùng muốn thực hiện chức năng đăng nhập thì thực hiện như sau:

- (1) Nhập email và mật khẩu — Khách hàng nhập thông tin
- (2) Gửi thông tin đăng nhập — App gửi đến ViewModel
- (3) Kiểm tra tính hợp lệ — ViewModel validate email/password
- (4) Gửi yêu cầu đăng nhập — ViewModel gửi đến Repository
- (5) Xác thực tài khoản — Repository gọi Firebase Authentication
- (6) Kiểm tra email và mật khẩu — Firebase xác thực
- (7) Trả về thông tin user — Firebase trả về FirebaseUser
- (8) Lấy thông tin user từ database — Repository truy vấn Realtime Database
- (9) Trả về dữ liệu user — Firebase trả về User đầy đủ
- (10) Trả về kết quả thành công — Repository trả về User cho ViewModel
- (11) Lưu thông tin user vào DataStore — Lưu local để dùng sau
- (12) Trả về kết quả thành công — ViewModel thông báo UI
- (13) Chuyển đến màn hình chính — Điều hướng theo role (Admin/User)

3.4.2. Biểu đồ tuần tự đăng ký

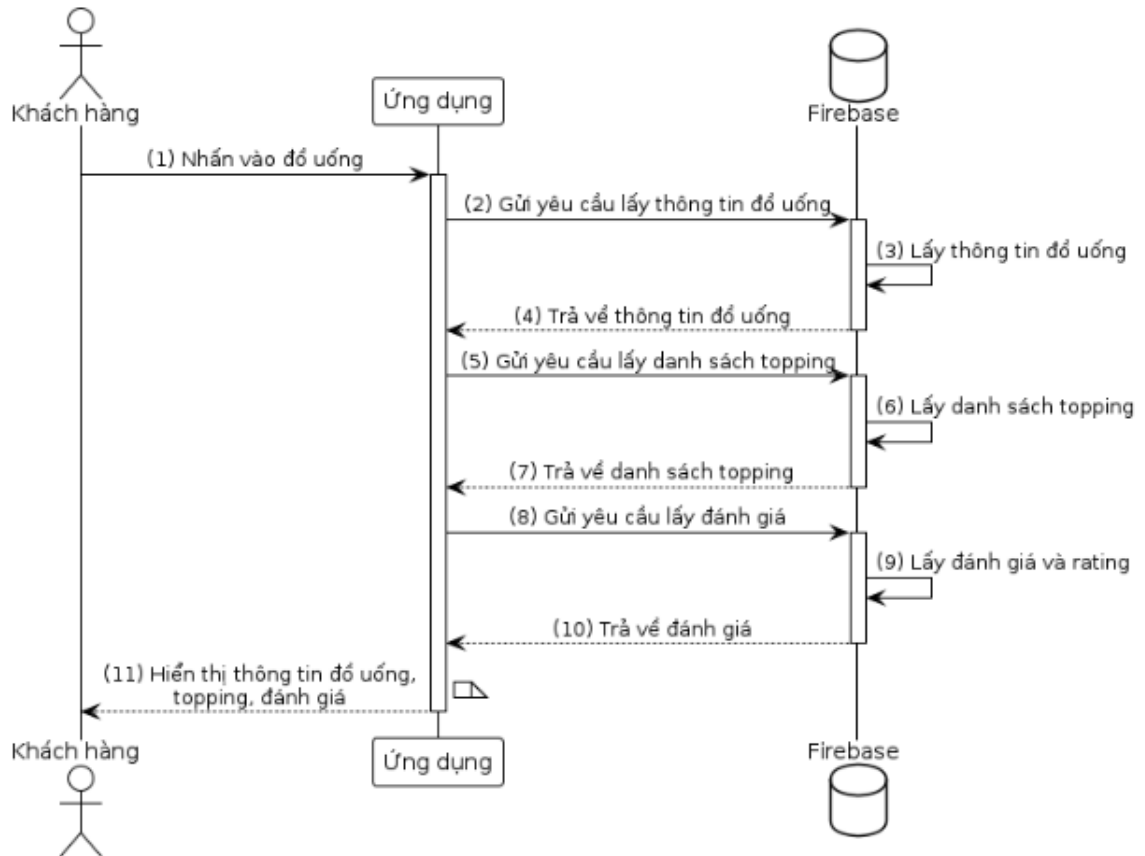


Hình 3.24 Sơ đồ tuần tự chức năng đăng ký

Người dùng muốn thực hiện chức năng đăng ký thì thực hiện như sau:

- (1) Nhập email và mật khẩu — Khách hàng nhập thông tin vào form đăng ký
- (2) Kiểm tra tính hợp lệ — App validate email format và password (email không trống, password không trống, email đúng format, kiểm tra admin/user format)
- (3) Gửi yêu cầu tạo tài khoản — App gửi đến Firebase Authentication
- (4) Kiểm tra email đã tồn tại — Firebase Authentication kiểm tra email đã được sử dụng
- (5) Tạo tài khoản mới — Firebase Authentication tạo tài khoản và hash mật khẩu
- (6) Trả về thông tin user — Firebase Authentication trả về FirebaseUser (UID, email)
- (7) Tạo đối tượng User — App tạo User object từ FirebaseUser và xác định isAdmin
- (8) Lưu thông tin user vào database — App lưu User vào Firebase Realtime Database
- (9) Trả về kết quả — Firebase Database trả về kết quả sau khi lưu
- (10) Lưu thông tin user vào DataStore — App lưu User vào DataStore để dùng sau
- (11) Chuyển đến màn hình chính — Điều hướng theo role (Admin/User)

3.4.3. Biểu đồ tuần tự xem chi tiết đồ uống



Hình 3.25 Sơ đồ tuần tự chức năng xem chi tiết đồ uống

Người dùng muốn thực hiện chức năng xem chi tiết đồ uống thì thực hiện như sau:

(1) Nhấn vào đồ uống — Khách hàng chọn đồ uống từ danh sách

(2) Gửi yêu cầu lấy thông tin đồ uống — App gửi yêu cầu đến Firebase Realtime Database

(3) Lấy thông tin đồ uống — Firebase truy vấn database lấy thông tin đồ uống theo ID

(4) Trả về thông tin đồ uống — Firebase trả về đối tượng Drink (id, name, description, price, banner, rating nested data, v.v.)

(5) Gửi yêu cầu lấy danh sách topping — App gửi yêu cầu lấy danh sách topping từ Firebase

(6) Lấy danh sách topping — Firebase truy vấn database lấy tất cả topping

(7) Trả về danh sách topping — Firebase trả về danh sách Topping (id, name, price)

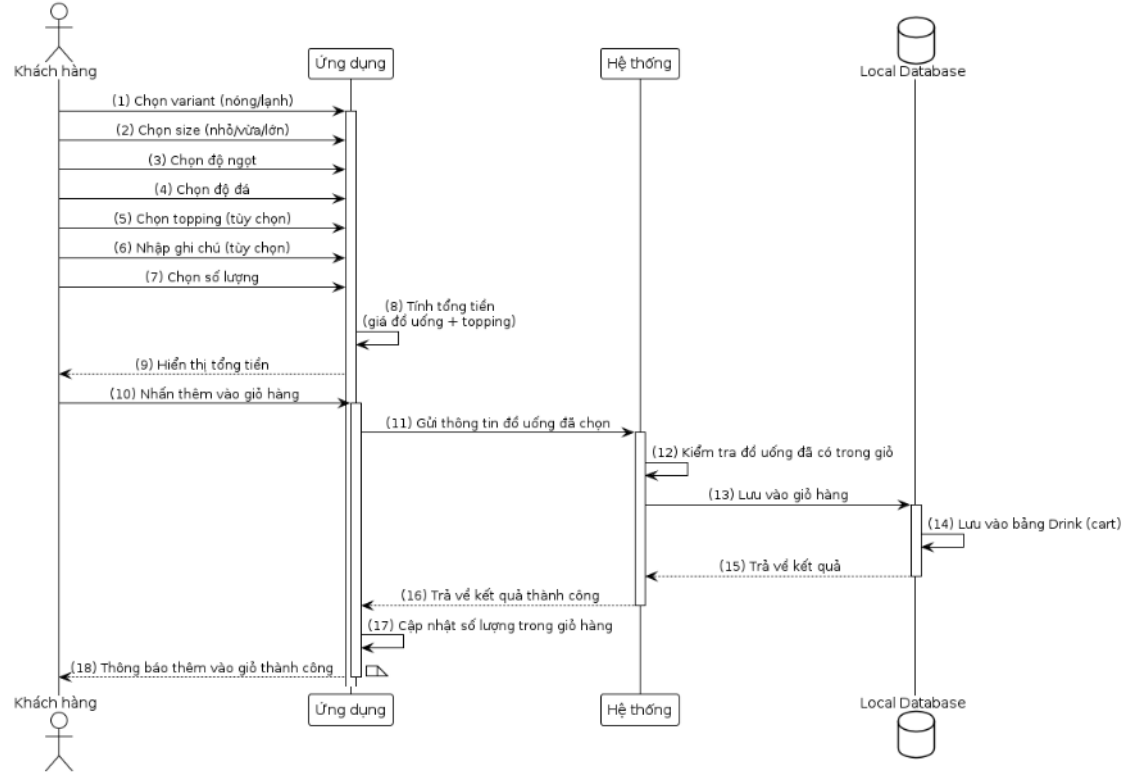
(8) Gửi yêu cầu lấy đánh giá — App xử lý dữ liệu rating từ nested data trong Drink object

(9) Lấy đánh giá và rating — App tính toán rating trung bình và số lượng đánh giá từ HashMap rating trong Drink

(10) Trả về đánh giá — App có đầy đủ thông tin rating

(11) Hiển thị thông tin đồ uống, topping, đánh giá — App hiển thị hình ảnh, tên, giá, mô tả, rating, và danh sách topping cho khách hàng

3.4.4. Biểu đồ tuần tự thêm vào giỏ hàng



Hình 3.26 Sơ đồ tuần tự chức năng thêm vào giỏ hàng

Người dùng muốn thực hiện chức năng thêm vào giỏ hàng thì thực hiện như sau:

(1) Chọn variant (nóng/lạnh) — Khách hàng chọn nóng hoặc lạnh

(2) Chọn size (nhỏ/vừa/lớn) — Khách hàng chọn kích thước

(3) Chọn độ ngọt — Khách hàng chọn độ ngọt (bình thường/ít đường)

(4) Chọn độ đá — Khách hàng chọn độ đá (bình thường/ít đá)

(5) Chọn topping (tùy chọn) — Khách hàng chọn topping từ danh sách

(6) Nhập ghi chú (tùy chọn) — Khách hàng nhập ghi chú thêm(7)

Chọn số lượng — Khách hàng chọn số lượng bằng nút +/-

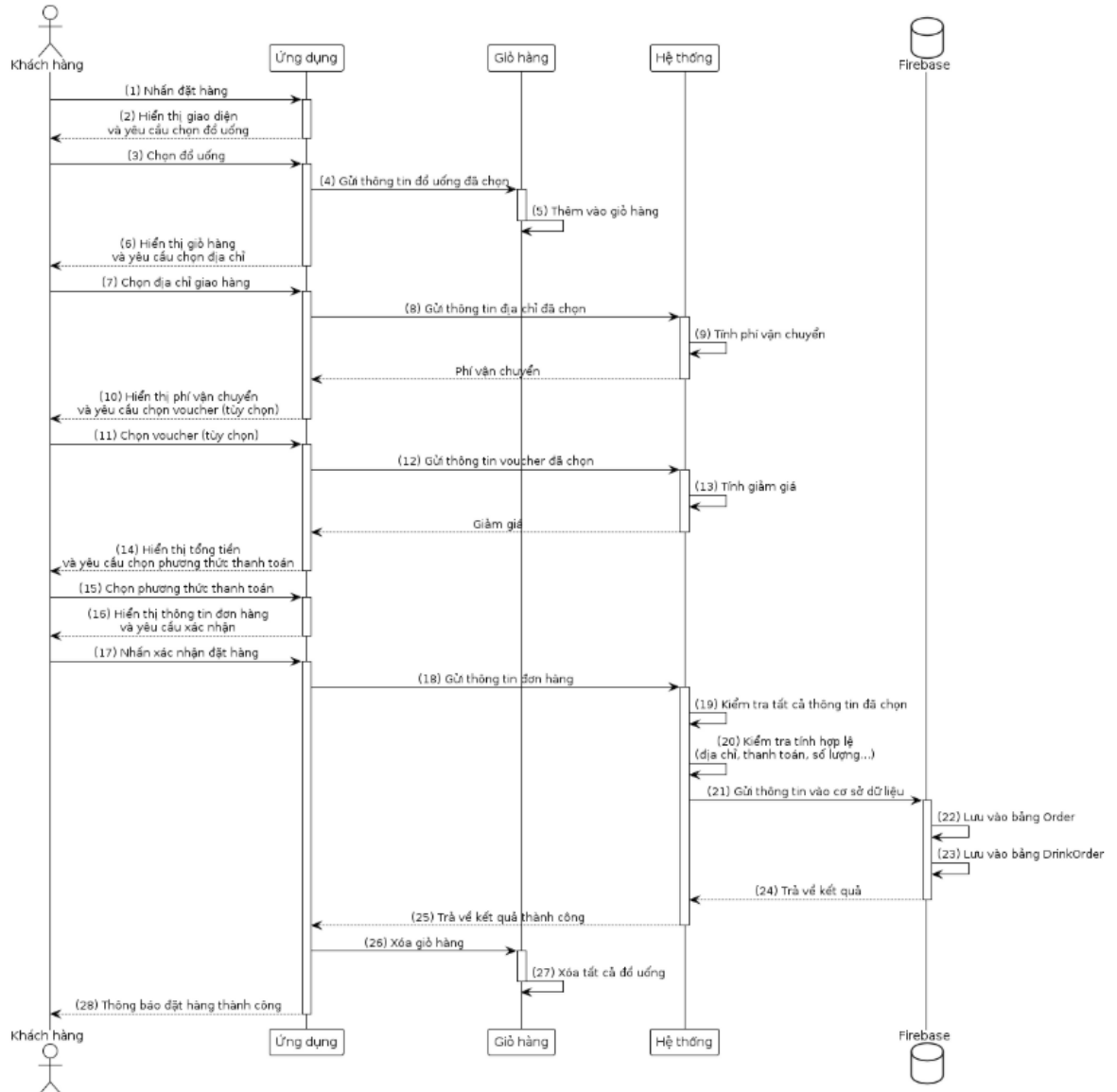
(8) Tính tổng tiền (giá đồ uống + topping) — App tính: $(\text{realPrice} + \text{tổng giá topping đã chọn}) \times \text{số lượng}$

(9) Hiển thị tổng tiền — App hiển thị tổng tiền trên giao diện

(10) Nhấn thêm vào giỏ hàng — Khách hàng nhấn nút "Thêm vào giỏ"

- (11) Gửi thông tin đồ uống đã chọn — App gửi đối tượng Drink (có đầy đủ variant, size, topping, số lượng) để xử lý
- (12) Kiểm tra đồ uống đã có trong giỏ — App kiểm tra trong Room Database xem đồ uống (theo ID) đã có chưa
- (13) Lưu vào giỏ hàng — App gọi phương thức lưu vào Room Database
- (14) Lưu vào bảng Drink (cart) — Room Database lưu đối tượng Drink vào bảng drink (local)
- (15) Trả về kết quả — Room Database trả về kết quả sau khi lưu thành công
- (16) Trả về kết quả thành công — App nhận kết quả lưu thành công
- (17) Cập nhật số lượng trong giỏ hàng — App gửi EventBus (DisplayCartEvent) để cập nhật badge số lượng trên icon giỏ hàng
- (18) Thông báo thêm vào giỏ thành công — App chuyển đến màn hình giỏ hàng (CartActivity) để khách hàng xem

3.4.5. Biểu đồ tuần tự đặt hàng



Hình 3.27 Sơ đồ tuần tự chức năng đặt hàng

Người dùng muốn thực hiện chức năng đặt hàng thì thực hiện như sau:

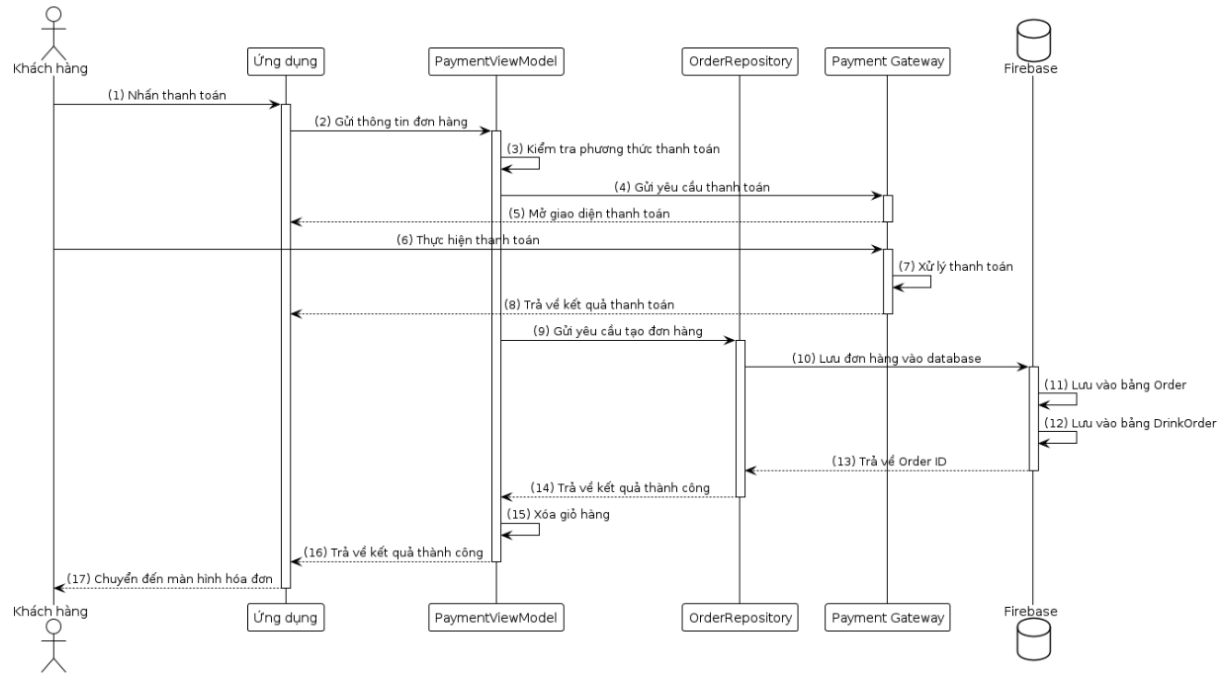
- (1) Nhấn đặt hàng — Khách hàng nhấn nút đặt hàng
- (2) Hiển thị giao diện và yêu cầu chọn đồ uống — App hiển thị màn hình giỏ hàng
- (3) Chọn đồ uống — Khách hàng xem và chọn đồ uống từ giỏ hàng (thường đã được thêm từ trước)
- (4) Gửi thông tin đồ uống đã chọn — App gửi thông tin đồ uống từ giỏ hàng để xử lý
- (5) Thêm vào giỏ hàng — Giỏ hàng xử lý thông tin đồ uống (thường đã có sẵn trong Room Database)
- (6) Hiển thị giỏ hàng và yêu cầu chọn địa chỉ — App hiển thị giỏ hàng và yêu cầu chọn địa chỉ giao hàng

- (7) Chọn địa chỉ giao hàng — Khách hàng chọn địa chỉ từ danh sách hoặc thêm mới
- (8) Gửi thông tin địa chỉ đã chọn — App gửi thông tin địa chỉ đến hệ thống để tính phí vận chuyển
- (9) Tính phí vận chuyển — Hệ thống tính phí dựa trên khoảng cách giữa cửa hàng và địa chỉ giao hàng (dùng Google Distance Matrix API)
- (10) Hiển thị phí vận chuyển và yêu cầu chọn voucher (tùy chọn) — App hiển thị phí vận chuyển và cho phép chọn voucher giảm giá
- (11) Chọn voucher (tùy chọn) — Khách hàng chọn voucher từ danh sách (nếu có)
- (12) Gửi thông tin voucher đã chọn — App gửi thông tin voucher đến hệ thống
- (13) Tính giảm giá — Hệ thống tính số tiền giảm dựa trên voucher đã chọn
- (14) Hiển thị tổng tiền và yêu cầu chọn phương thức thanh toán — App hiển thị tổng tiền (bao gồm phí vận chuyển và giảm giá) và yêu cầu chọn phương thức thanh toán
- (15) Chọn phương thức thanh toán — Khách hàng chọn phương thức (tiền mặt, chuyển khoản, VNPay, Blockchain, v.v.)
- (16) Hiển thị thông tin đơn hàng và yêu cầu xác nhận — App hiển thị thông tin đơn hàng đầy đủ để khách hàng xác nhận
- (17) Nhấn xác nhận đặt hàng — Khách hàng nhấn nút xác nhận
- (18) Gửi thông tin đơn hàng — App gửi toàn bộ thông tin đơn hàng đến hệ thống
- (19) Kiểm tra tất cả thông tin đã chọn — Hệ thống kiểm tra đầy đủ thông tin: đồ uống, địa chỉ, phương thức thanh toán, voucher
- (20) Kiểm tra tính hợp lệ (địa chỉ, thanh toán, số lượng...) — Hệ thống validate: địa chỉ hợp lệ, phương thức thanh toán, số lượng đồ uống > 0 , v.v.
- (21) Gửi thông tin vào cơ sở dữ liệu — Hệ thống gửi thông tin đơn hàng đến Firebase Realtime Database
- (22) Lưu vào bảng Order — Firebase lưu đối tượng Order chính vào node /orders/{orderId}
- (23) Lưu vào bảng DrinkOrder — Firebase lưu danh sách DrinkOrder (nested trong Order object) với thông tin từng đồ uống
- (24) Trả về kết quả — Firebase trả về kết quả sau khi lưu thành công
- (25) Trả về kết quả thành công — Hệ thống trả về kết quả thành công cho App
- (26) Xóa giỏ hàng — App yêu cầu xóa tất cả đồ uống trong giỏ hàng

(27) Xóa tất cả đồ uống — Giỏ hàng xóa tất cả dữ liệu trong Room Database (gọi deleteAllDrink())

(28) Thông báo đặt hàng thành công — App chuyển đến màn hình xác nhận đơn hàng (ReceiptOrderActivity) và thông báo thành công

3.4.6. Biểu đồ tuần tự thanh toán tổng quát



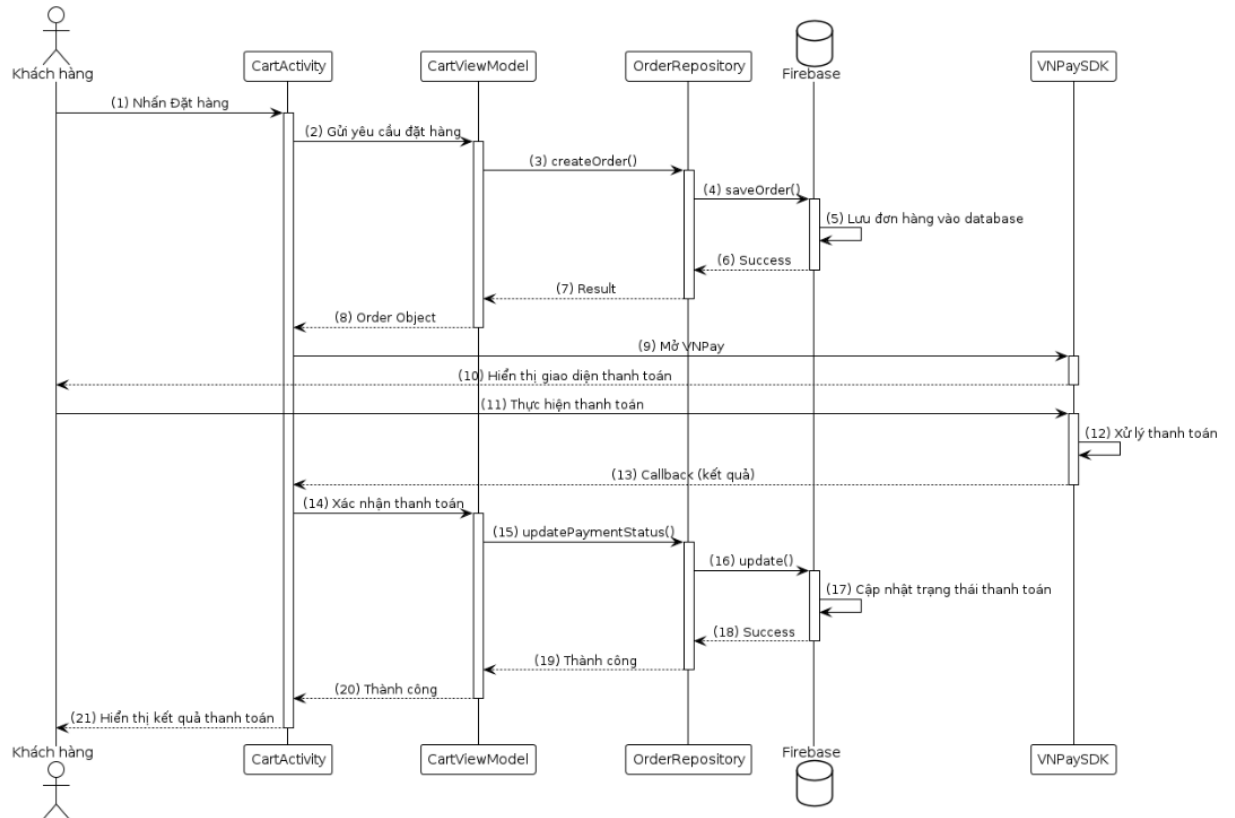
Hình 3.28 Sơ đồ tuần tự chức năng thanh toán tổng quát

Người dùng muốn thực hiện chức năng thanh toán tổng quát thì thực hiện như sau:

- (1) Nhấn thanh toán — Khách hàng nhấn nút thanh toán trong giỏ hàng
- (2) Gửi thông tin đơn hàng — App gửi thông tin đơn hàng đến PaymentViewModel
- (3) Kiểm tra phương thức thanh toán — ViewModel kiểm tra phương thức đã chọn (COD, VNPay, Blockchain)
- (4) Gửi yêu cầu thanh toán — ViewModel gửi yêu cầu đến Payment Gateway (tùy phương thức: VNPay/Blockchain/COD)
- (5) Mở giao diện thanh toán — Payment Gateway mở giao diện thanh toán (VNPay WebView, Blockchain Wallet, hoặc màn hình COD)
- (6) Thực hiện thanh toán — Khách hàng thực hiện thanh toán trên giao diện của Gateway (nhập thông tin, xác nhận, v.v.)
- (7) Xử lý thanh toán — Payment Gateway xử lý giao dịch (xác thực, xử lý tiền, tạo transaction)

- (8) Trả về kết quả thanh toán — Payment Gateway trả về kết quả (thành công/thất bại) và thông tin thanh toán
- (9) Gửi yêu cầu tạo đơn hàng — ViewModel gửi yêu cầu tạo đơn hàng đến OrderRepository
- (10) Lưu đơn hàng vào database — Repository gửi thông tin đơn hàng đến Firebase Realtime Database
- (11) Lưu vào bảng Order — Firebase lưu đối tượng Order chính vào node /orders/{orderId}
- (12) Lưu vào bảng DrinkOrder — Firebase lưu danh sách DrinkOrder (nested trong Order) với thông tin từng đồ uống
- (13) Trả về Order ID — Firebase trả về Order ID sau khi lưu thành công
- (14) Trả về kết quả thành công — Repository trả về kết quả thành công cho ViewModel
- (15) Xóa giỏ hàng — ViewModel xóa tất cả đồ uống trong giỏ hàng (gọi CartRepository.clearCart())
- (16) Trả về kết quả thành công — ViewModel thông báo kết quả thành công đến App (qua StateFlow)
- (17) Chuyển đến màn hình hóa đơn — App chuyển đến ReceiptOrderActivity để hiển thị hóa đơn đặt hàng

3.4.7. Biểu đồ tuần tự thanh toán VNPay



Hình 3.29 Sơ đồ tuần tự chức năng thanh toán VNPay

Người dùng muốn thực hiện chức năng thanh toán VNPay thì thực hiện như sau:

- (1) Nhấn Đặt hàng — Khách hàng nhấn nút đặt hàng trong giỏ hàng
- (2) Gửi yêu cầu đặt hàng — CartActivity gửi yêu cầu đặt hàng đến CartViewModel
- (3) createOrder() — ViewModel gọi Repository để tạo đơn hàng
- (4) saveOrder() — Repository gửi yêu cầu lưu đơn hàng đến Firebase
- (5) Lưu đơn hàng vào database — Firebase lưu đối tượng Order vào node /orders/{orderId}
- (6) Success — Firebase trả về kết quả thành công
- (7) Result — Repository trả về kết quả cho ViewModel
- (8) Order Object — ViewModel trả về đối tượng Order cho CartActivity
- (9) Mở VNPay — CartActivity mở VNPay SDK với payment URL (được build từ VnPayHelper)
- (10) Hiển thị giao diện thanh toán — VNPay SDK hiển thị giao diện thanh toán cho khách hàng

(11) Thực hiện thanh toán — Khách hàng nhập thông tin và xác nhận thanh toán trên giao diện VNPay

(12) Xử lý thanh toán — VNPay SDK xử lý giao dịch (xác thực, kết nối ngân hàng, xử lý tiền)

(13) Callback (kết quả) — VNPay SDK gọi callback qua deep link (shopfee://vnpay-result) với kết quả thanh toán

(14) Xác nhận thanh toán — PaymentResultActivity nhận callback và gửi yêu cầu xác nhận đến ViewModel

(15) updatePaymentStatus() — ViewModel gọi Repository để cập nhật trạng thái thanh toán

(16) update() — Repository gửi yêu cầu cập nhật đến Firebase

(17) Cập nhật trạng thái thanh toán — Firebase cập nhật payment status và transaction info trong Order

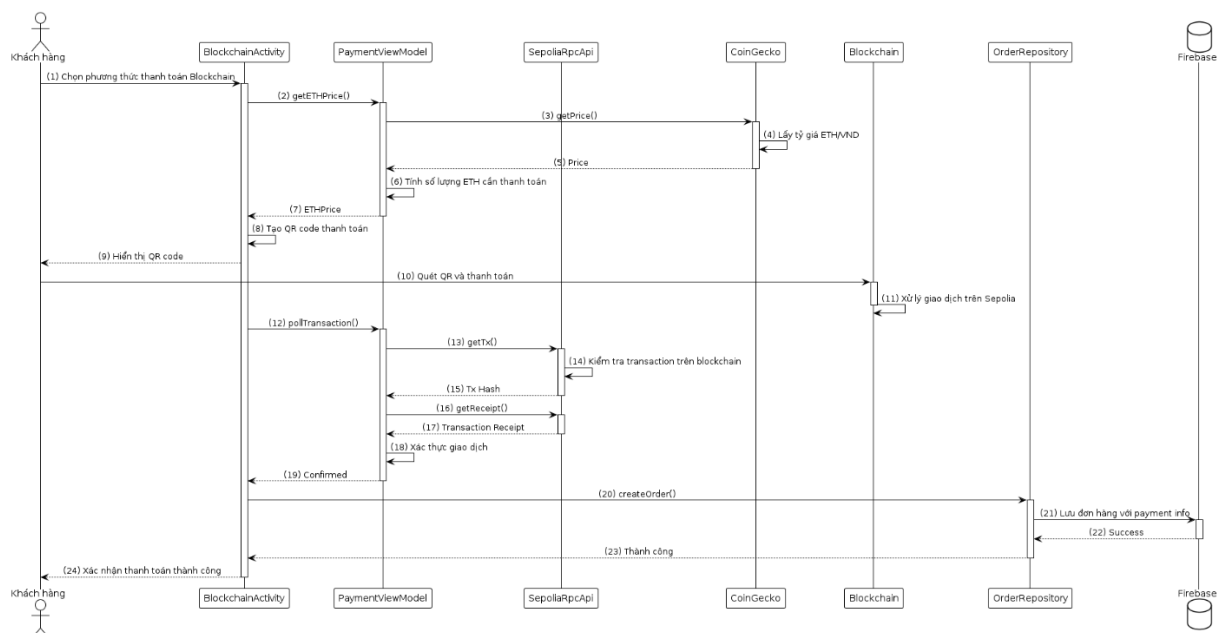
(18) Success — Firebase trả về kết quả thành công

(19) Thành công — Repository trả về kết quả thành công cho ViewModel

(20) Thành công — ViewModel thông báo kết quả thành công cho Activity

(21) Hiển thị kết quả thanh toán — Activity chuyển đến ReceiptOrderActivity để hiển thị hóa đơn và kết quả thanh toán

3.4.8. Biểu đồ tuần tự thanh toán blockchain



Hình 3.30 Sơ đồ tuần tự chức năng thanh toán blockchain

Người dùng muốn thực hiện chức năng thanh toán blockchain thì thực hiện như sau:

- (1) Chọn phương thức thanh toán Blockchain — Khách hàng chọn Blockchain từ các phương thức thanh toán
- (2) getETHPrice() — BlockchainPaymentActivity gọi API để lấy giá ETH (trong code gọi trực tiếp CoinGecko)
- (3) getPrice() — Activity gửi yêu cầu đến CoinGecko API
- (4) Lấy tỷ giá ETH/VND — CoinGecko tra cứu và tính tỷ giá hiện tại
- (5) Price — CoinGecko trả về tỷ giá ETH/VND
- (6) Tính số lượng ETH cần thanh toán — Activity tính: số tiền VND / tỷ giá ETH/VND = số lượng ETH, sau đó chuyển sang Wei (1 ETH = 10^{18} Wei)
- (7) ETHPrice — Activity có giá ETH và số lượng ETH/Wei cần thanh toán
- (8) Tạo QR code thanh toán — Activity tạo QR code từ payment URI
- (9) Hiển thị QR code — Activity hiển thị QR code cho khách hàng
- (10) Quét QR và thanh toán — Khách hàng mở ví (MetaMask, Trust Wallet, v.v.), quét QR và xác nhận giao dịch
- (11) Xử lý giao dịch trên Sepolia — Blockchain xử lý và thêm giao dịch vào Sepolia testnet
- (12) pollTransaction() — Activity bắt đầu auto-polling để kiểm tra giao dịch (quét các block mới)
- (13) getTx() — Activity gọi SepoliaRpcApi để lấy thông tin transaction theo hash
- (14) Kiểm tra transaction trên blockchain — RpcApi truy vấn Sepolia RPC node để kiểm tra transaction
- (15) Tx Hash — RpcApi trả về transaction hash và thông tin transaction
- (16) getReceipt() — Activity gọi SepoliaRpcApi để lấy transaction receipt
- (17) Transaction Receipt — RpcApi trả về receipt chứa trạng thái (status = 0x1 = success)
- (18) Xác thực giao dịch — Activity xác thực: transaction status = success, địa chỉ nhận đúng, số tiền đúng (cho phép tolerance)
- (19) Confirmed — Activity xác nhận giao dịch hợp lệ và sẵn sàng lưu đơn hàng
- (20) createOrder() — Activity gọi Repository để lưu đơn hàng (trong code gọi trực tiếp Firebase)

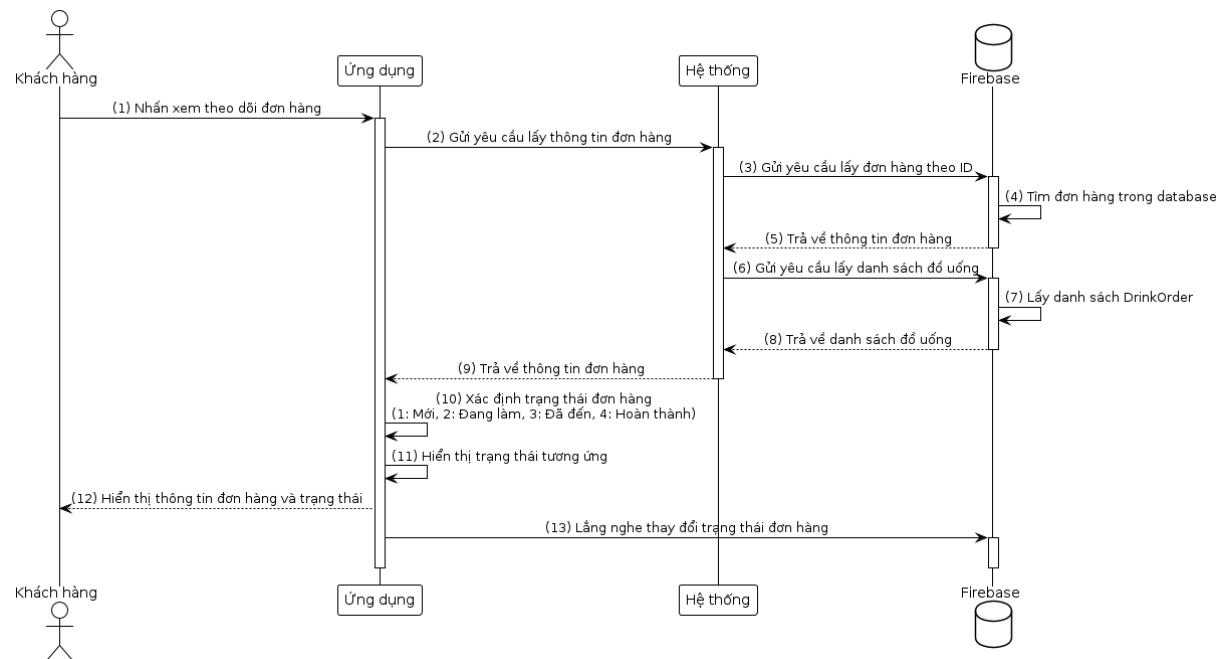
(21) Lưu đơn hàng với payment info — Repository lưu Order vào Firebase kèm payment info (paymentTxHash, paymentAmountWei, paymentChainId, paymentStatus = "paid")

(22) Success — Firebase trả về kết quả thành công

(23) Thành công — Repository trả về kết quả thành công cho Activity

(24) Xác nhận thanh toán thành công — Activity xóa giỏ hàng, cập nhật user rank, và chuyển đến ReceiptOrderActivity để hiển thị kết qu

3.4.9. Biểu đồ tuần tự theo dõi đơn hàng



Hình 3.31 Sơ đồ tuần tự chức năng theo dõi đơn hàng

Người dùng muốn thực hiện chức năng theo dõi đơn hàng thì thực hiện như sau:

(1) Nhấn xem theo dõi đơn hàng — Khách hàng nhấn vào đơn hàng để xem chi tiết

(2) Gửi yêu cầu lấy thông tin đơn hàng — App gửi yêu cầu đến Hệ thống

(3) Gửi yêu cầu lấy đơn hàng theo ID — Hệ thống gửi yêu cầu đến Firebase

(4) Tìm đơn hàng trong database — Firebase tìm kiếm đơn hàng

(5) Trả về thông tin đơn hàng — Firebase trả về thông tin đơn hàng (bao gồm danh sách đồ uống)

(6) Gửi yêu cầu lấy danh sách đồ uống — Hệ thống xử lý danh sách đồ uống từ Order object

(7) Lấy danh sách DrinkOrder — Hệ thống lấy danh sách đồ uống từ nested data

(8) Trả về danh sách đồ uống — Hệ thống có danh sách đồ uống

(9) Trả về thông tin đơn hàng — Hệ thống trả về thông tin đầy đủ cho App

(10) Xác định trạng thái đơn hàng (1: Mới, 2: Đang làm, 3: Đã đến, 4: Hoàn thành)
— App xác định trạng thái

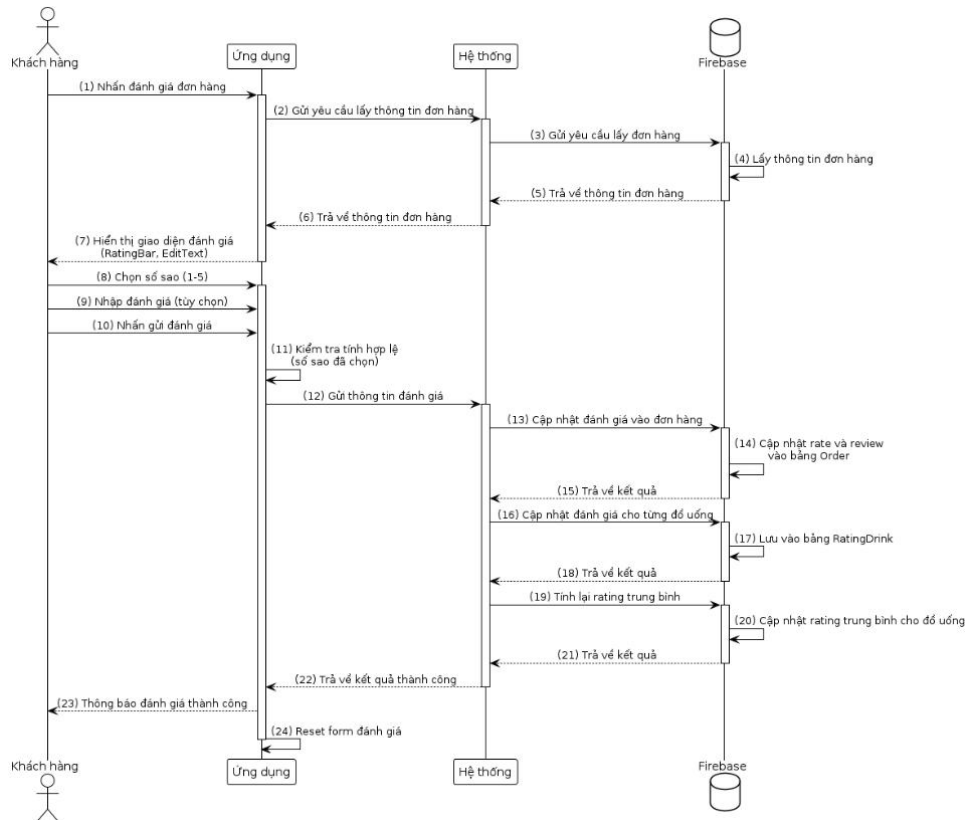
(11) Hiển thị trạng thái tương ứng — App cập nhật UI hiển thị trạng thái

(12) Hiển thị thông tin đơn hàng và trạng thái — App hiển thị cho khách hàng

(13) Lắng nghe thay đổi trạng thái đơn hàng — App đăng ký listener để theo dõi realtime

(Loop) Kiểm tra thay đổi → Cập nhật trạng thái mới → Cập nhật giao diện → Hiển thị trạng thái mới — Vòng lặp theo dõi realtime

3.4.10. Biểu đồ tuần tự đánh giá đơn hàng



Hình 3.32 Sơ đồ tuần tự chức năng đánh giá đơn hàng

Người dùng muốn thực hiện chức năng đồ đánh giá đơn hàng thì thực hiện như sau:

(1) Nhấn đánh giá đơn hàng — Khách hàng nhấn vào đơn hàng để đánh giá (thường sau khi đơn hàng hoàn thành)

(2) Gửi yêu cầu lấy thông tin đơn hàng — App gửi yêu cầu đến Hệ thống để lấy thông tin đơn hàng

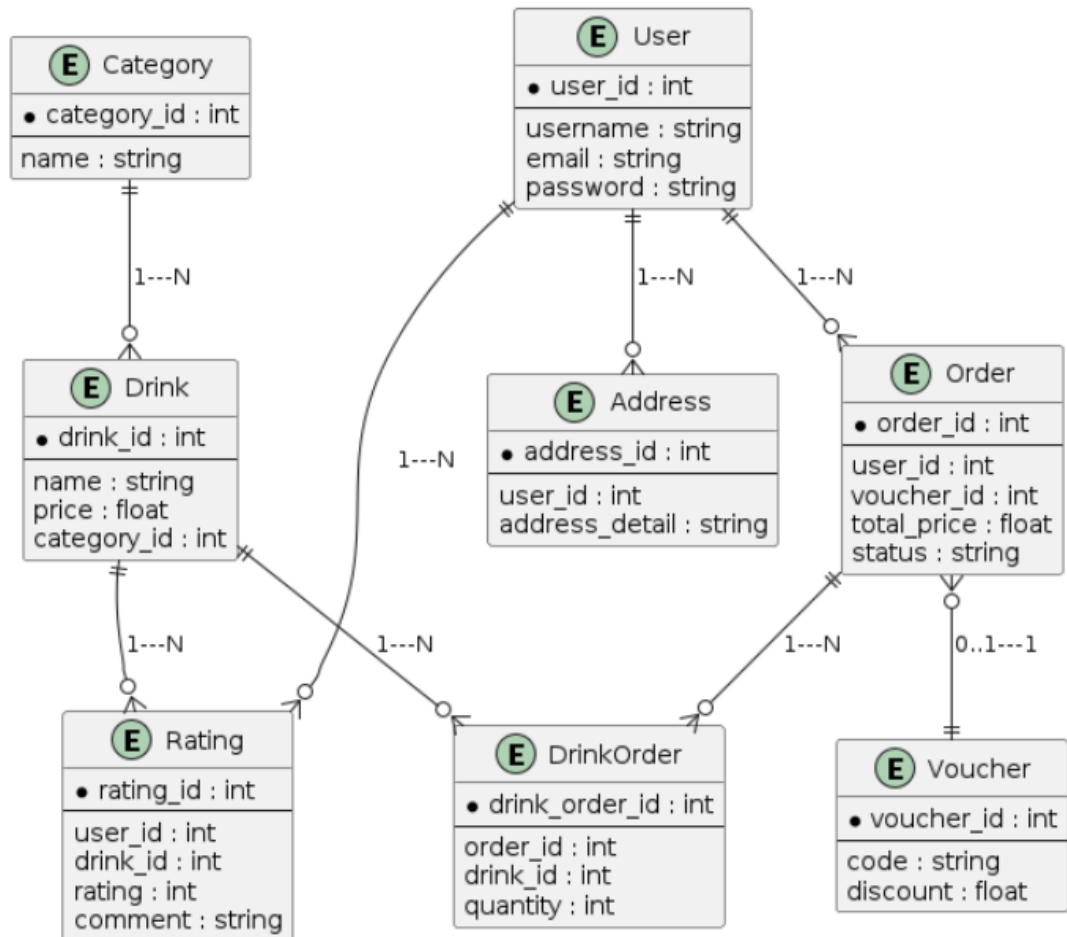
- (3) Gửi yêu cầu lấy đơn hàng — Hệ thống gửi yêu cầu đến Firebase để lấy thông tin đơn hàng theo ID
- (4) Lấy thông tin đơn hàng — Firebase truy vấn node /orders/{orderId} để lấy thông tin đơn hàng
- (5) Trả về thông tin đơn hàng — Firebase trả về đối tượng Order cho Hệ thống
- (6) Trả về thông tin đơn hàng — Hệ thống trả về thông tin đơn hàng cho App
- (7) Hiện thị giao diện đánh giá (RatingBar, EditText) — App hiện thị màn hình đánh giá với RatingBar (1-5 sao) và EditText (nhập đánh giá)
- (8) Chọn số sao (1-5) — Khách hàng chọn số sao đánh giá (1-5)
- (9) Nhập đánh giá (tùy chọn) — Khách hàng nhập nhận xét (có thể bỏ qua)
- (10) Nhấn gửi đánh giá — Khách hàng nhấn nút gửi đánh giá
- (11) Kiểm tra tính hợp lệ (số sao đã chọn) — App kiểm tra số sao đã được chọn (mặc định là 5 sao, luôn hợp lệ)
- (12) Gửi thông tin đánh giá — App gửi thông tin đánh giá (rate, review) đến Hệ thống
- (13) Cập nhật đánh giá vào đơn hàng — Hệ thống gửi yêu cầu cập nhật đánh giá vào Firebase
- (14) Cập nhật rate và review vào bảng Order — Firebase cập nhật trường rate và review vào node /orders/{orderId}
- (15) Trả về kết quả — Firebase trả về kết quả cập nhật thành công
- (16) Cập nhật đánh giá cho từng đồ uống — Hệ thống gửi yêu cầu cập nhật đánh giá cho từng đồ uống trong đơn hàng
- (17) Lưu vào bảng RatingDrink — Firebase lưu đánh giá vào node /drink/{drinkId}/rating/{userEmail} với Rating object (rate, review)
- (18) Trả về kết quả — Firebase trả về kết quả lưu thành công
- (19) Tính lại rating trung bình — Hệ thống yêu cầu tính lại rating trung bình cho đồ uống
- (20) Cập nhật rating trung bình cho đồ uống — Firebase tính lại rating trung bình từ tất cả đánh giá (trong code, rating trung bình được tính tự động qua getter trong Drink object)
- (21) Trả về kết quả — Firebase trả về kết quả
- (22) Trả về kết quả thành công — Hệ thống trả về kết quả thành công cho App

(23) Thông báo đánh giá thành công — App hiển thị toast message "Đánh giá thành công"

(24) Reset form đánh giá — App reset RatingBar về 5 sao và xóa text trong EditText,

3.5. Thiết kế cơ sở dữ liệu

3.5.1. Biểu đồ ERD



Hình 3.3.3 Biểu đồ ERD

3.5.2. Cấu trúc các bảng

Bảng 3.9. Bảng user

Tên cột	Kiểu dữ liệu	Ràng buộc	Mô tả
email	varchar(255)	primary key	Email đăng nhập
password	varchar(255)	not null	Mật khẩu (đã hash)
isAdmin	boolean	default false	Quyền quản trị
totalSpent	bigint	default 0	Tổng tiền đã chi tiêu
rankLevel	int	default 0	Hạng thành viên (0=Thường, 1=Bạc, 2=Vàng, 3=Kim cương)

Bảng 3.10. Bảng Category

Tên Cột	Kiểu dữ liệu	Ràng buộc	Mô tả
id	bigint	primary key	ID danh mục
name	varchar(255)	not null	Tên danh mục

Bảng 3.11. Bảng Drink

Tên cột	Kiểu dữ liệu	Ràng buộc	Mô tả
id	bigint	primary key	ID sản phẩm
name	varchar(255)	not null	Tên sản phẩm

Tên cột	Kiểu dữ liệu	Ràng buộc	Mô tả
description	text		Mô tả sản phẩm
price	int	not null	Giá gốc
image	text		URL hình ảnh
banner	text		URL banner
category_id	bigint	foreign key	ID danh mục
category_name	varchar(255)		Tên danh mục (denormalized)
sale	int	default 0	% giảm giá
isFeatured	boolean	default false	Sản phẩm nổi bật

Bảng 3.12. Bảng Order

Tên cột	Kiểu dữ liệu	Ràng buộc	Mô tả
id	bigint	primary key	ID đơn hàng
userEmail	varchar(255)	foreign key	Email khách hàng
dateTime	varchar(50)	not null	Thời gian đặt hàng
price	int	not null	Tổng tiền sản phẩm
voucher	int	default 0	Tiền giảm từ voucher
voucherId	bigint	foreign key	ID voucher

Tên cột	Kiểu dữ liệu	Ràng buộc	Mô tả
voucherCode	varchar(50)		Mã voucher
total	int	not null	Tổng tiền thanh toán
paymentMethod	varchar(50)		Phương thức thanh toán
status	int	default 1	Trạng thái (1=Mới, 2=Đang chuẩn bị, 3=Đang giao, 4=Hoàn thành)
rate	double		Đánh giá (1-5)
review	text		Nhận xét
latitude	double		Vĩ độ địa chỉ giao hàng
longitude	double		Kinh độ địa chỉ giao hàng
shippingFee	int	default 0	Phí vận chuyển
distanceKm	double		Khoảng cách (km)
paymentStatus	varchar(50)		Trạng thái thanh toán
paymentTxHash	varchar(255)		Transaction hash (Blockchain)
paymentAmountWei	varchar(255)		Số tiền thanh toán (Wei)
paymentChainId	bigint		Chain ID (Blockchain)

Bảng 3.13 Bảng DrinkOrder

Tên cột	Kiểu dữ liệu	Ràng buộc	Mô tả
id	bigint	primary key	ID chi tiết

Tên cột	Kiểu dữ liệu	Ràng buộc	Mô tả
orderId	bigint	foreign key	ID đơn hàng
drinkId	bigint	foreign key	ID sản phẩm
name	varchar(255)	not null	Tên sản phẩm
image	text		Hình ảnh
price	int	not null	Giá
count	int	not null	Số lượng
variant	varchar(20)		Biến thể (Nóng/Đá)
size	varchar(20)		Kích thước
sugar	varchar(20)		Độ ngọt
ice	varchar(20)		Độ đá
toppingIds	varchar(255)		Danh sách ID topping
note	text		Ghi chú

Bảng 3.14 Bảng Address

Tên cột	Kiểu dữ liệu	Ràng buộc	Mô tả
id	bigint	primary key	ID địa chỉ
userEmail	varchar(255)	foreign key	Email người dùng
name	varchar(255)		Tên người nhận
phone	varchar(20)		Số điện thoại

Tên cột	Kiểu dữ liệu	Ràng buộc	Mô tả
address	text	not null	Địa chỉ chi tiết
latitude	double		Vĩ độ
longitude	double		Kinh độ
isSelected	boolean	default false	Địa chỉ mặc định

Bảng 3.15 Bảng Voucher

Tên cột	Kiểu dữ liệu	Ràng buộc	Mô tả
id	bigint	primary key	ID voucher
code	Varchar(50)	unique	Mã voucher
discount	int	not null	% hoặc số tiền giảm
minOrder	int	default 0	Đơn hàng tối thiểu
maxDiscount	int		Giảm tối đa
startDate	Varchar(50)		Ngày bắt đầu
endDate	Varchar(50)		Ngày kết thúc
isActive	boolean	default true	Trạng thái
usedCount	int	default 0	Số lần đã sử dụng
maxUse	int		Số lần sử dụng tối đa

Bảng 3.16 Bảng Rating

Tên cột	Kiểu dữ liệu	Ràng buộc	Mô tả
userEmail	varchar(255)	primary key	Email người đánh giá
rate	double	not null	Điểm đánh giá (1-5)
comment	text		Nhận xét

3.6. Thiết kế bảo mật

3.6.1. Bảo mật xác thực người dùng

3.6.1.1. Firebase Authentication

- Mật khẩu được hash bằng bcrypt (Firebase xử lý)
- Session token quản lý tự động
- Token refresh để duy trì đăng nhập
- Đăng xuất xóa token

3.6.1.2. Phân quyền người dùng

- Kiểm tra quyền admin qua isAdmin trong User
- Admin: email có format @admin.com
- Middleware kiểm tra quyền trước khi thực hiện chức năng admin

3.6.2. Bảo mật dữ liệu

3.6.2.1. Firebase Realtime Database Rules

```
{
  "rules": {
    "user": {
      "$userEmail": {
        ".read": "$userEmail === auth.token.email",
        ".write": "$userEmail === auth.token.email"
      }
    },
    "order": {
```

```

"$orderId": {
  ".read": "auth != null && (data.child('userEmail').val() === auth.token.email ||
root.child('user').child(auth.token.email).child('isAdmin').val() === true)",
  ".write": "auth != null && (data.child('userEmail').val() === auth.token.email ||
root.child('user').child(auth.token.email).child('isAdmin').val() === true)"
},
},
"category": {
  ".read": "auth != null",
  ".write": "auth != null &&
root.child('user').child(auth.token.email).child('isAdmin').val() === true"
},
"drink": {
  ".read": "auth != null",
  ".write": "auth != null &&
root.child('user').child(auth.token.email).child('isAdmin').val() === true"
}
}
}
}

```

3.6.3. Bảo mật API và giao tiếp

3.6.3.1. VNPay Payment Security

- Hash Secret để tạo chữ ký
- Xác minh chữ ký từ callback
- HTTPS cho tất cả giao tiếp

3.6.3.2. Blockchain Payment Security

- Xác minh transaction hash trên blockchain
- Kiểm tra số tiền thanh toán khớp
- Kiểm tra địa chỉ ví nhận tiền
- Timeout cho polling transaction

3.6.4. Bảo mật ứng dụng

3.6.4.1. ProGuard/R8

- Obfuscate code khi build release
- Loại bỏ code không dùng
- Khó reverse engineering

Cấu hình:

- keep class com.pro.shopfee.model.** { *; }
- keep class com.google.firebase.** { *; }

3.6.4.2. API Keys Protection

- Không hardcode API keys trong code
- Sử dụng ApiKeys.kt (không commit)
- BuildConfig cho production
- Environment variables trong CI/CD

Cấu trúc:

app/src/main/java/com/pro/shopfee/utils/

└─ ApiKeys.kt.example (template)

└─ ApiKeys.kt (thực tế, không commit)

3.6.4.3. Certificate Pinning (Tùy chọn)

- Pin SSL certificate cho các API quan trọng
- Chống Man-in-the-Middle attacks

3.6.5. Bảo mật dữ liệu local

3.6.5.1. Room Database Encryption

- Có thể dùng SQLCipher để mã hóa database local
- Bảo vệ dữ liệu giỏ hàng khi thiết bị bị xâm nhập

3.6.5.2. SharedPreferences Security

- Không lưu thông tin nhạy cảm trong SharedPreferences
- Sử dụng EncryptedSharedPreferences nếu cần

3.6.6. Bảo mật giao tiếp mạng

- HTTPS cho tất cả API calls
- TLS 1.2 trở lên
- Certificate validation

- Không gửi thông tin nhạy cảm qua HTTP

3.6.7. Input Validation và Sanitization

- Validate input từ người dùng
- Chống SQL Injection (Room sử dụng parameterized queries)
- Chống XSS trong review/comment
- Giới hạn độ dài input

3.6.8. Rate Limiting và DDoS Protection

- Giới hạn số lần đăng nhập sai
- Giới hạn số request từ một IP
- Firebase có sẵn rate limiting

3.6.9. Logging và Monitoring

- Không log thông tin nhạy cảm
- Sử dụng Firebase Crashlytics để theo dõi lỗi
- Monitor các hoạt động đáng ngờ

3.6.10. Backup và Recovery

- Firebase tự động backup
- Có kế hoạch khôi phục dữ liệu
- Export dữ liệu định kỳ

CHƯƠNG 4. TRIỂN KHAI ỨNG DỤNG VÀ KẾT QUẢ ĐẠT ĐƯỢC

4.1. Môi trường phát triển và công cụ lập trình

Đề tài được xây dựng trên môi trường phát triển:

- Vi xử lý (Processor): Intel (R) Core (TM) i7-8565U CPU @ 1.80GHz 1.99GHz
- Ram: 8 GB
- Hệ điều hành: Windows 10 64bit
- Công cụ lập trình:
- Phần mềm viết code: Android Studio
- Cơ sở dữ liệu: Firebase (Cloud FireStore và Realtime Database)
- Ngôn ngữ lập trình: Kotlin

4.2. Triển khai ứng dụng

Bước 1: Mở dự án

- Mở Android Studio
- File → Open → Chọn thư mục CafeManagerProKotlin
- Đợi Gradle sync xong

Bước 2: Kiểm tra file cấu hình

- File google-services.json có trong app/
- File ApiKeys.kt đã được tạo (từ file .example)

Bước 3: Sync Gradle

- File → Sync Project with Gradle Files

Bước 4: Chọn thiết bị

- Option A: Tạo emulator (Tools → Device Manager → Create Device)
- Option B: Kết nối điện thoại (bật USB Debugging)

Bước 5: Chạy ứng dụng

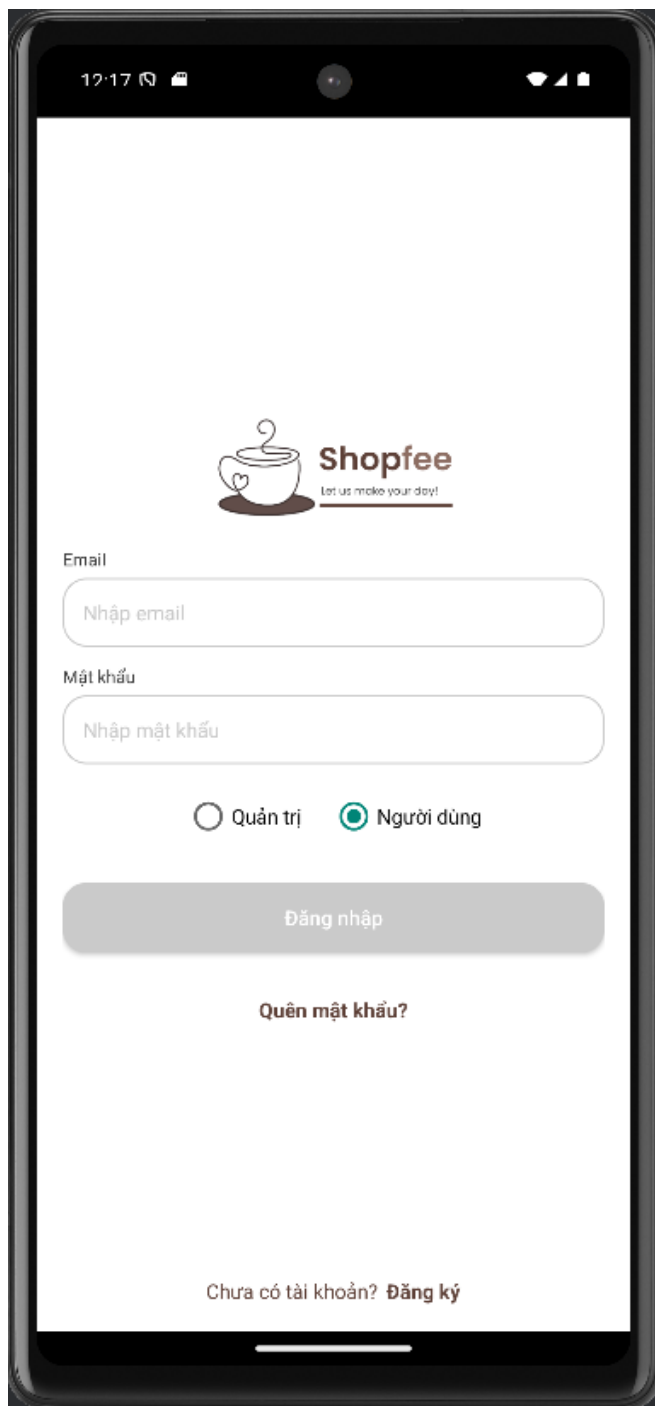
- Chọn thiết bị ở thanh toolbar
- Click Run (hoặc Shift+F10)
- Đợi build và cài đặt xong

4.3. Kết quả đạt được

4.3.1. Giao diện người dùng

4.3.1.1. Giao diện màn hình đăng nhập

Khách hàng và Người quản trị đều dùng chung màn hình này, giao diện như sau:

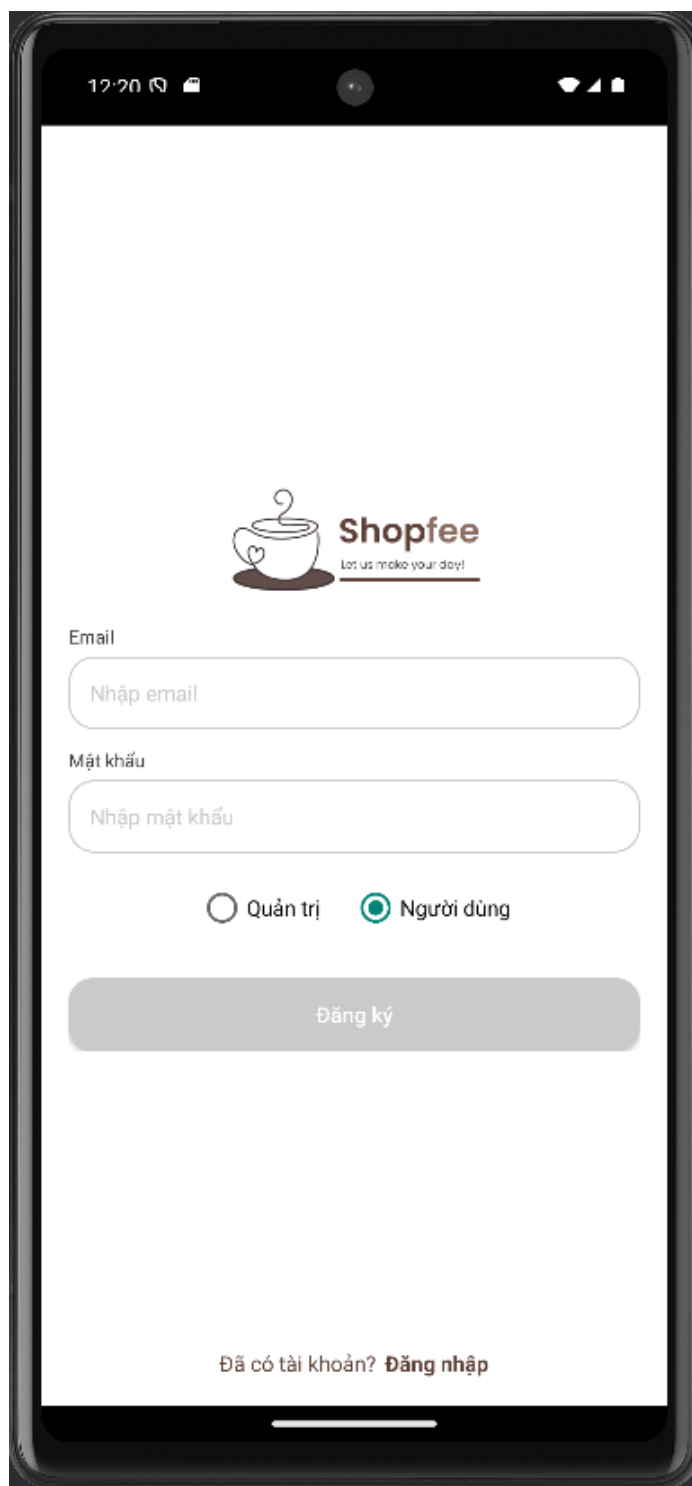


Hình 4.1 Giao diện màn hình đăng nhập

Người dùng cần nhập đầy đủ email và mật khẩu để đăng nhập vào hệ thống

4.3.1.2. Giao diện màn hình đăng ký

Khách hàng nhấn vào chức năng đăng ký. Giao diện đăng ký hiển thị như sau:

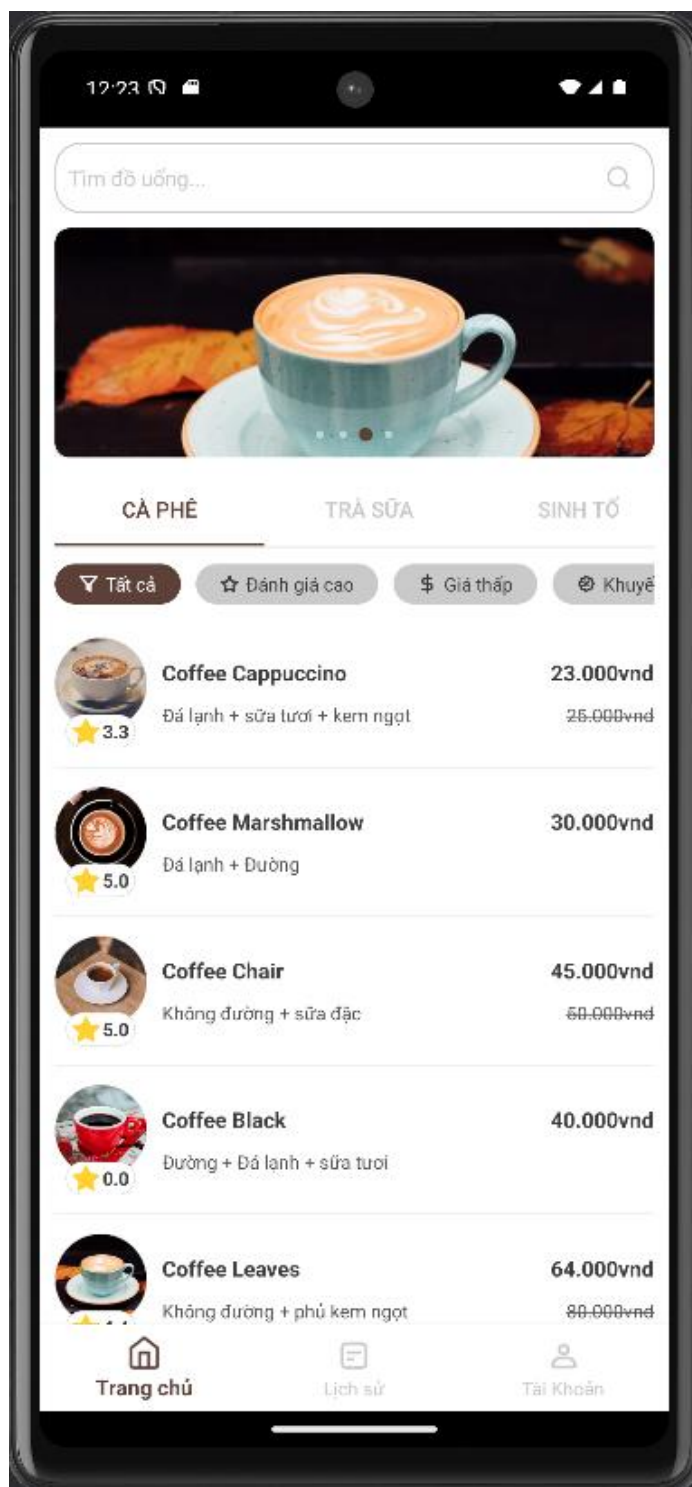


Hình 4.2 Giao diện màn hình đăng ký

Người dùng muốn đăng ký tài khoản cần chọn vào chức năng đăng ký ở màn hình đăng nhập. Giao diện đăng ký sẽ hiển thị. Sau khi nhập đầy đủ thông tin thì người dùng có thể tạo mới được tài khoản và đăng nhập vào hệ thống.

4.3.1.3. Giao diện màn hình trang chủ

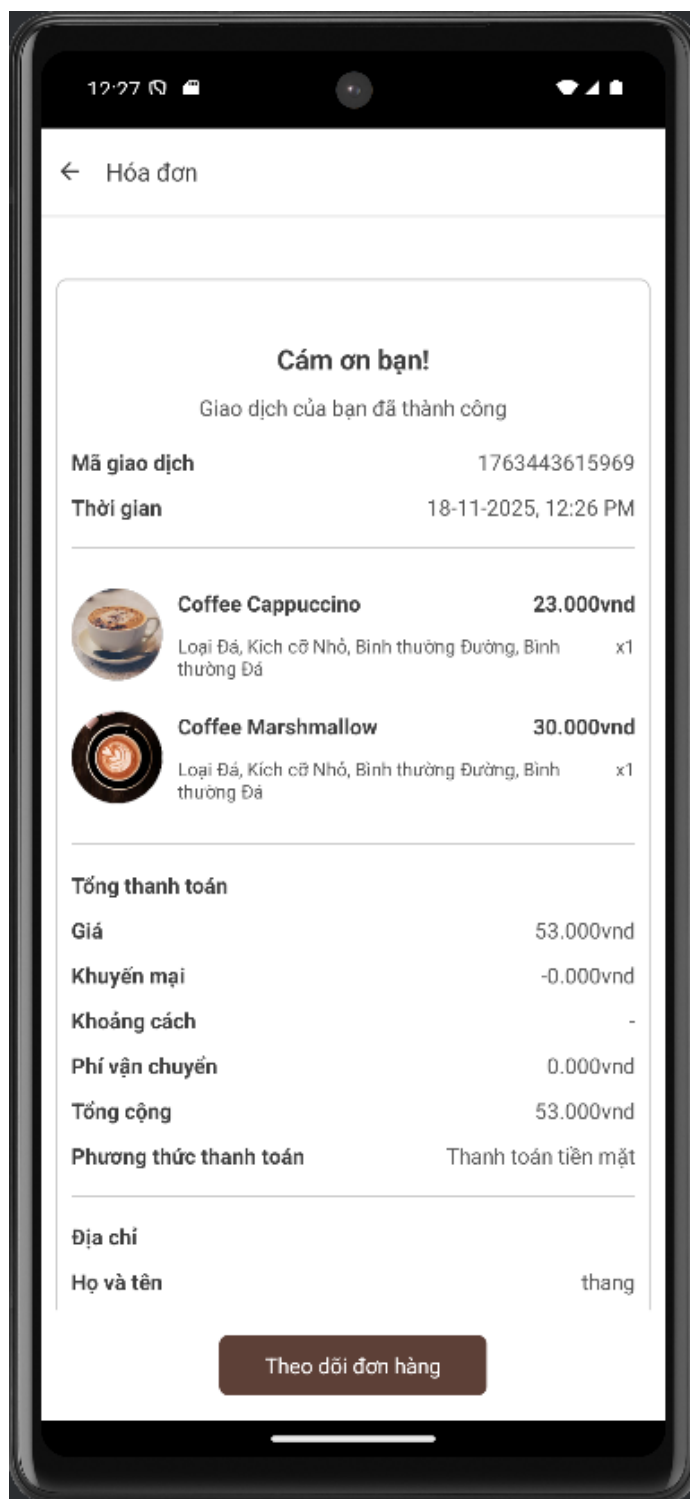
Sau khi đăng nhập thành công, hệ thống điều hướng đến trang chủ. Giao diện như sau:



Hình 4.3 Giao diện màn hình trang chủ

Màn hình trang chủ hiển thị danh sách các danh mục sản phẩm và các sản phẩm nổi bật. Khách hàng có thể duyệt và tìm kiếm các sản phẩm đồ uống từ quán cà phê.

4.3.1.4. Giao diện màn hình đơn hàng



Hình 4.4 Giao diện màn hình đơn hàng

Khi khách hàng đặt hàng, những đơn hàng đó được lưu lại dưới dạng danh sách và hiển thị như trên. Khách hàng có thể theo dõi trạng thái đơn hàng của mình.

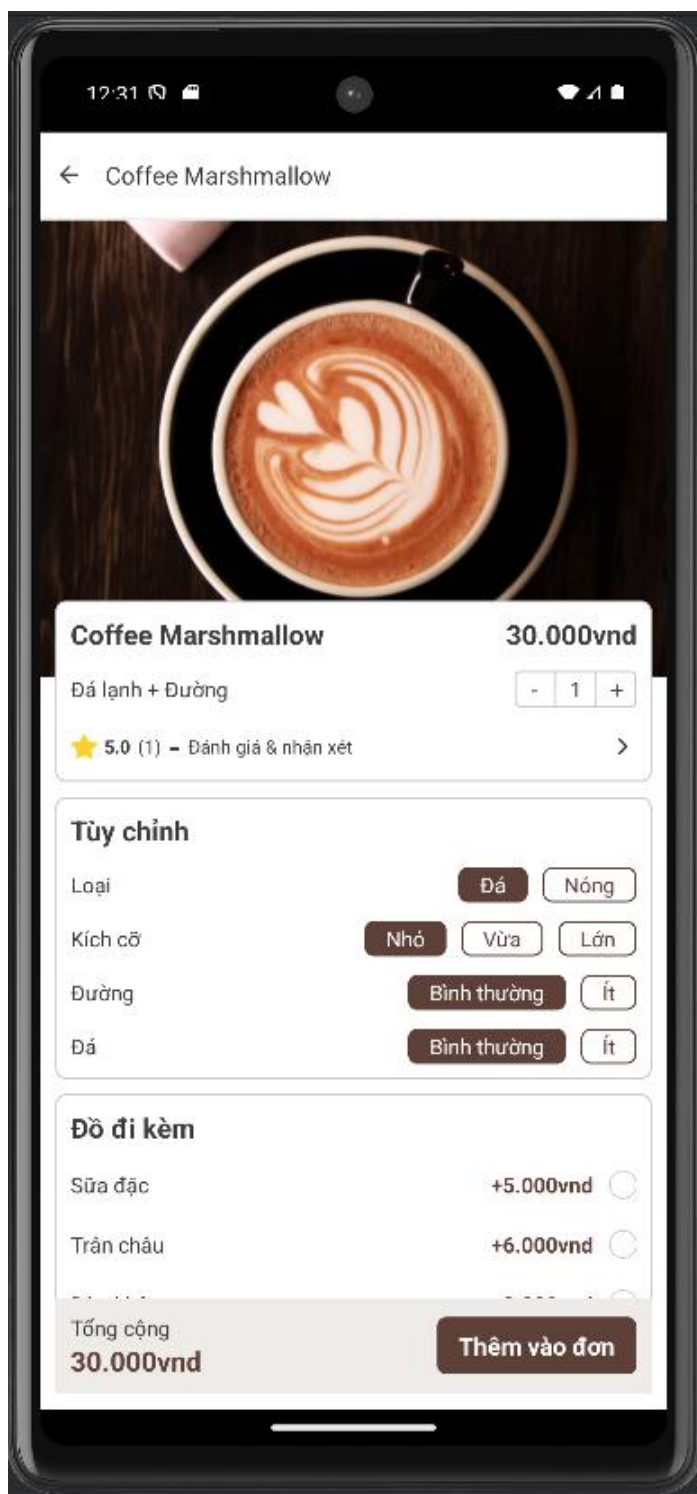
4.3.1.5. Giao diện màn hình danh mục sản phẩm



Hình 4.5 Giao diện màn hình danh mục sản phẩm

Sau khi khách hàng chọn một danh mục từ màn hình trang chủ. Hệ thống hiển thị danh sách sản phẩm thuộc danh mục đó cho khách hàng chọn.

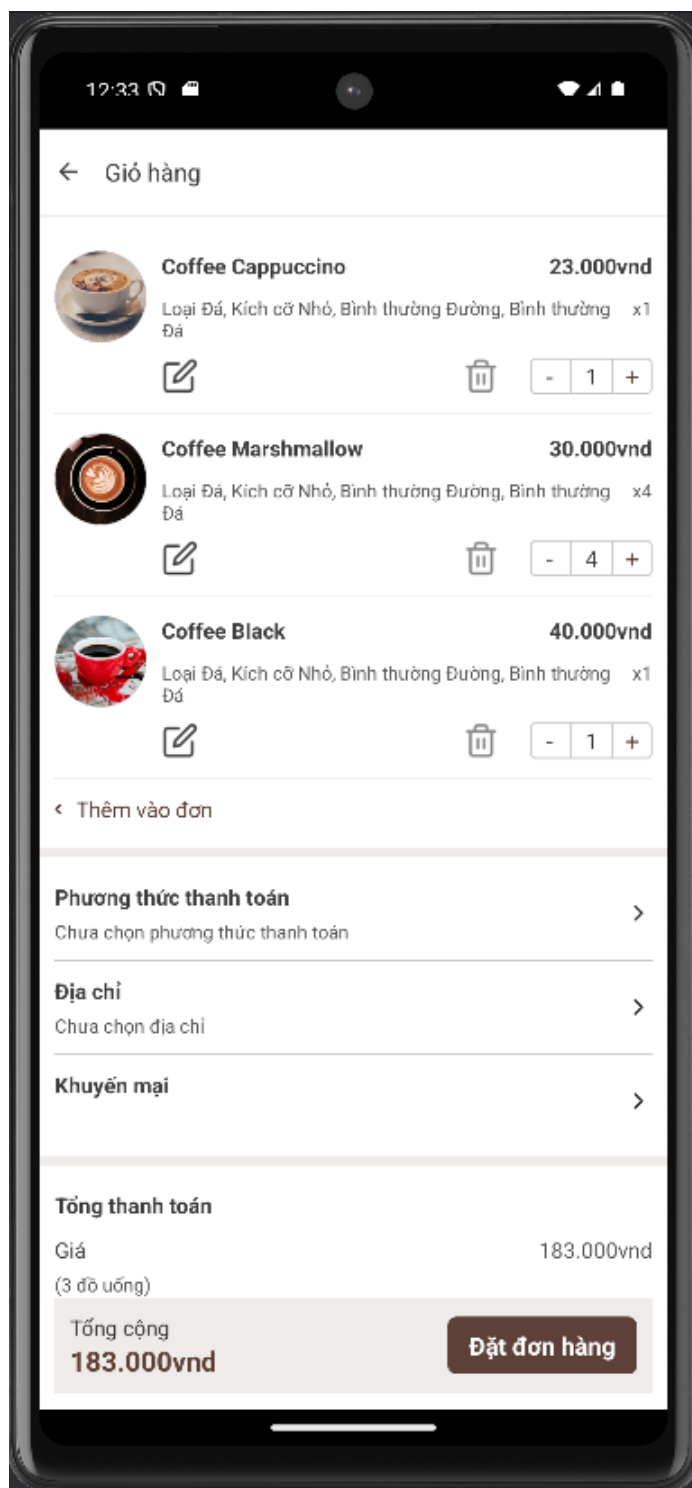
4.3.1.6. Giao diện màn hình chi tiết sản phẩm



Hình 4.6 Giao diện màn hình chi tiết sản phẩm

Sau khi khách hàng chọn một sản phẩm. Hệ thống hiển thị thông tin chi tiết về sản phẩm bao gồm: hình ảnh, tên, mô tả, giá, các tùy chọn (size, đường, đá), topping, đánh giá. Khách hàng có thể thêm sản phẩm vào giỏ hàng hoặc đặt hàng ngay.

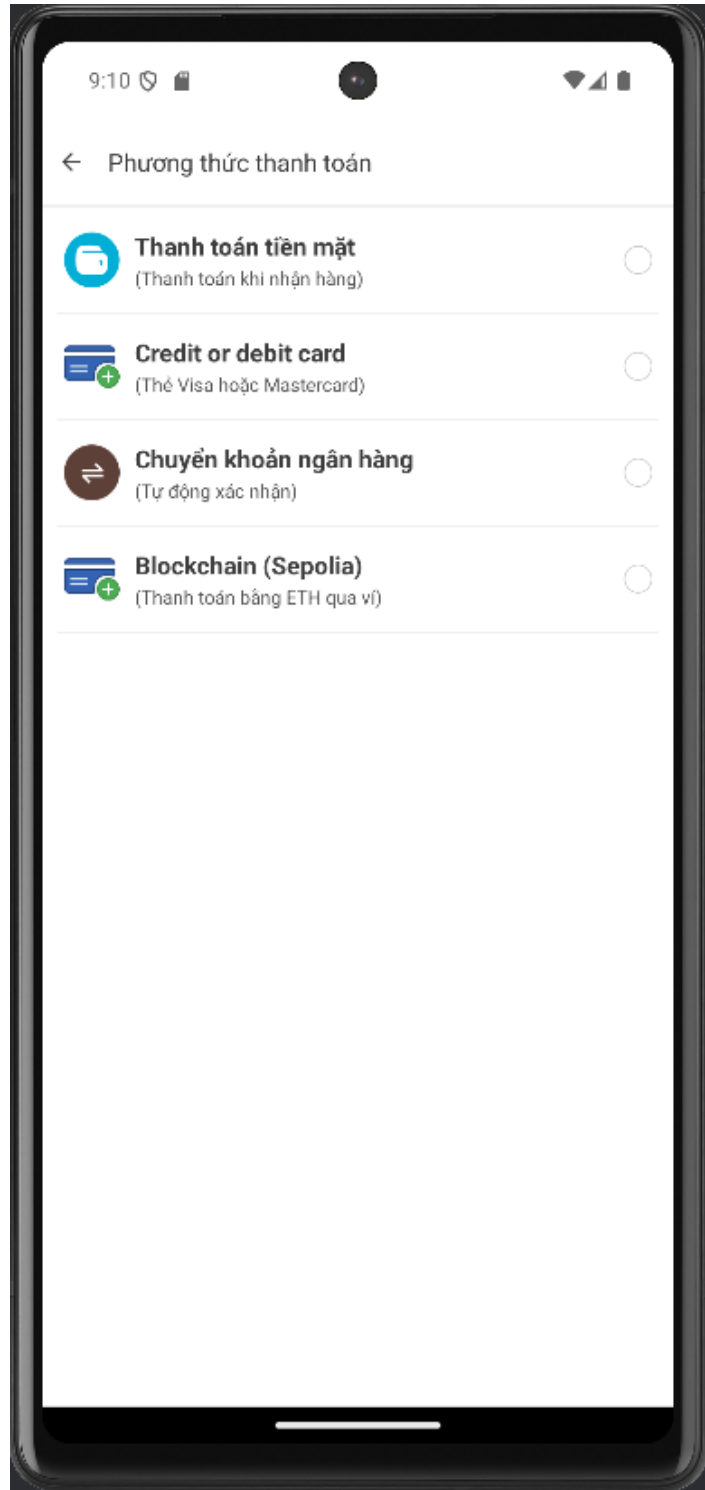
4.3.1.7. Giao diện màn hình giỏ hàng



Hình 4.7 Giao diện màn hình giỏ hàng

Màn hình giỏ hàng hiển thị tất cả sản phẩm mà khách hàng đã thêm vào. Khách hàng có thể chỉnh sửa số lượng, xóa sản phẩm, áp dụng voucher và tiến hành thanh toán.

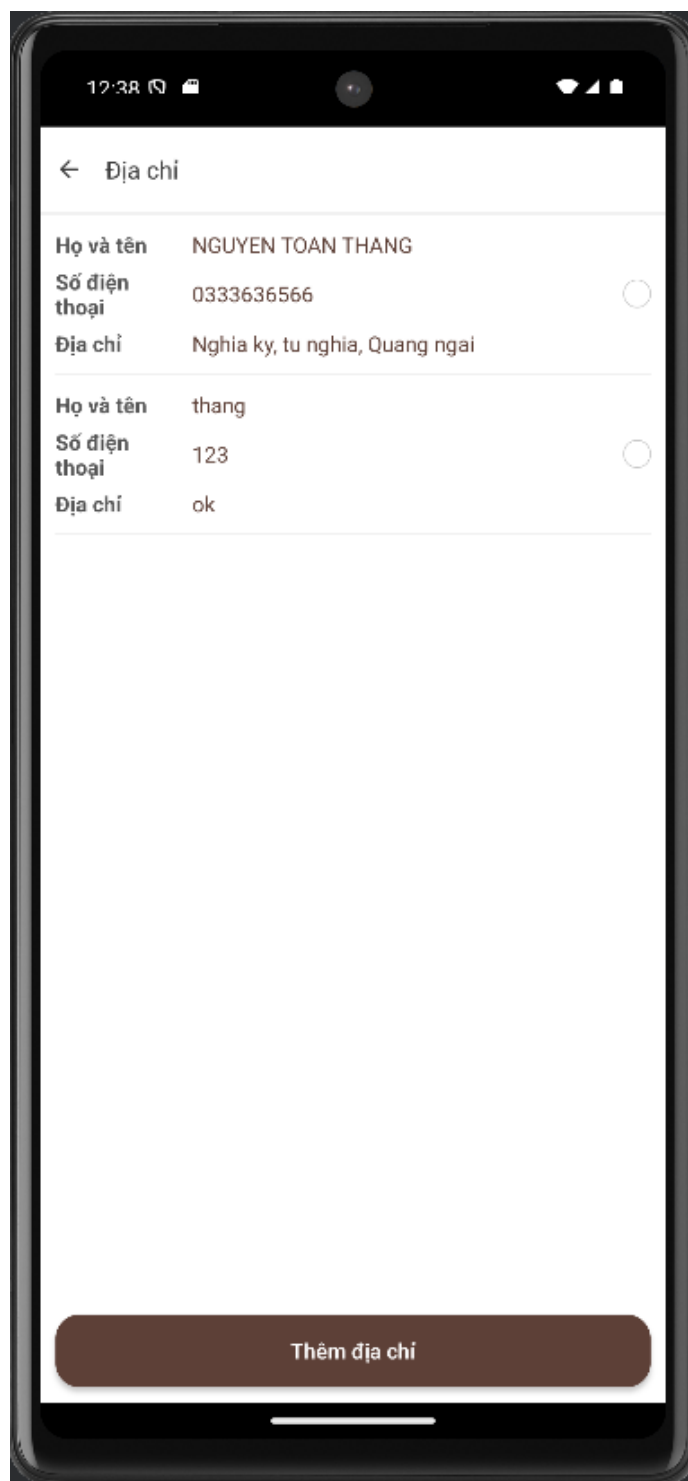
4.3.1.8. Giao diện màn hình chọn phương thức thanh toán



Hình 4.8 Giao diện màn hình thanh toán

Sau khi khách hàng ấn phương thức thanh toán. Hệ thống sẽ hiển thị 4 phương thức thanh toán: Thanh toán tiền mặt, credit, chuyển khoản ngân hàng, blockchain

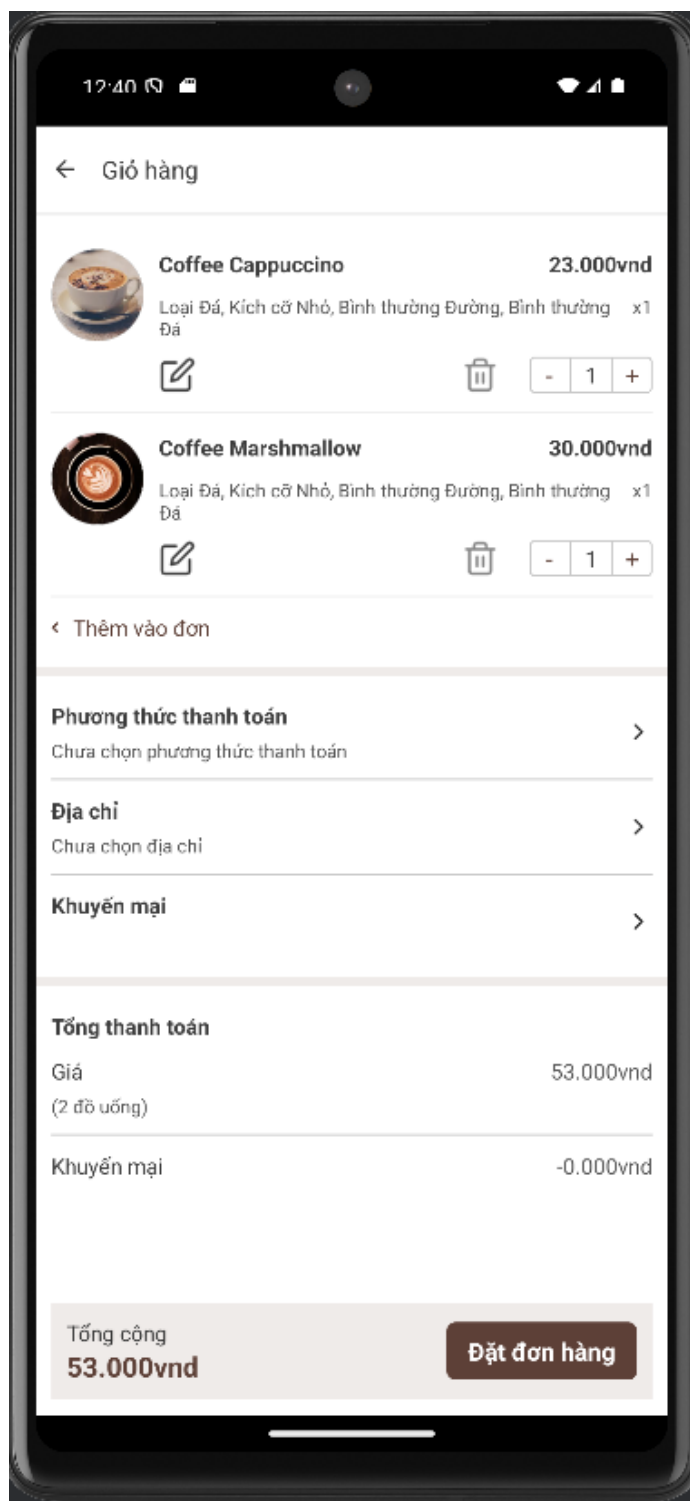
4.3.1.9. Giao diện màn hình chọn địa chỉ giao hàng



Hình 4.9 Giao diện màn hình chọn địa chỉ giao hàng

Sau khi khách hàng chọn phương thức thanh toán. Hệ thống yêu cầu khách hàng chọn hoặc thêm địa chỉ giao hàng. Ứng dụng tự động lấy vị trí GPS của người dùng và chuyển đổi thành địa chỉ.

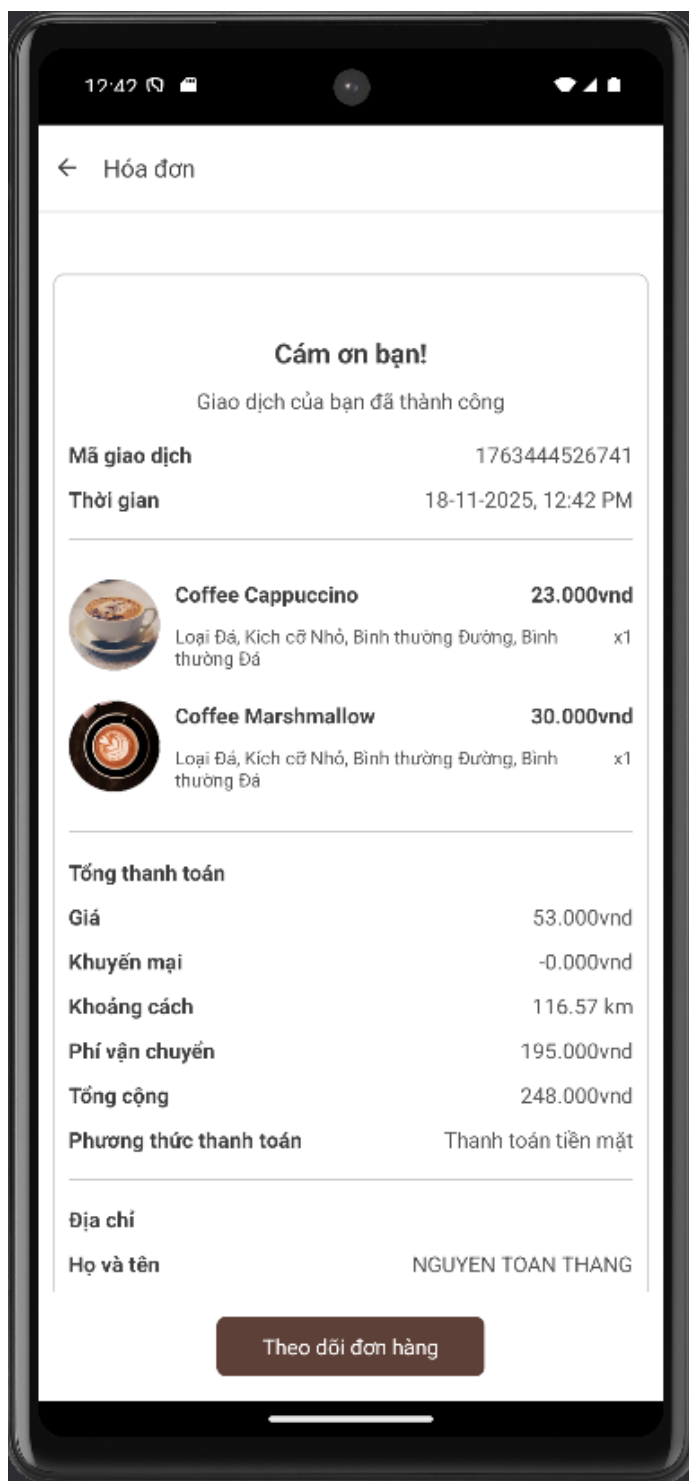
4.3.1.10. Giao diện màn hình xác nhận đơn hàng



Hình 4.10 Giao diện màn hình xác nhận đơn hàng

Sau khi khách hàng hoàn tất việc chọn địa chỉ và phương thức thanh toán. Hệ thống hiển thị thông tin đơn hàng để khách hàng xác nhận trước khi thanh toán.

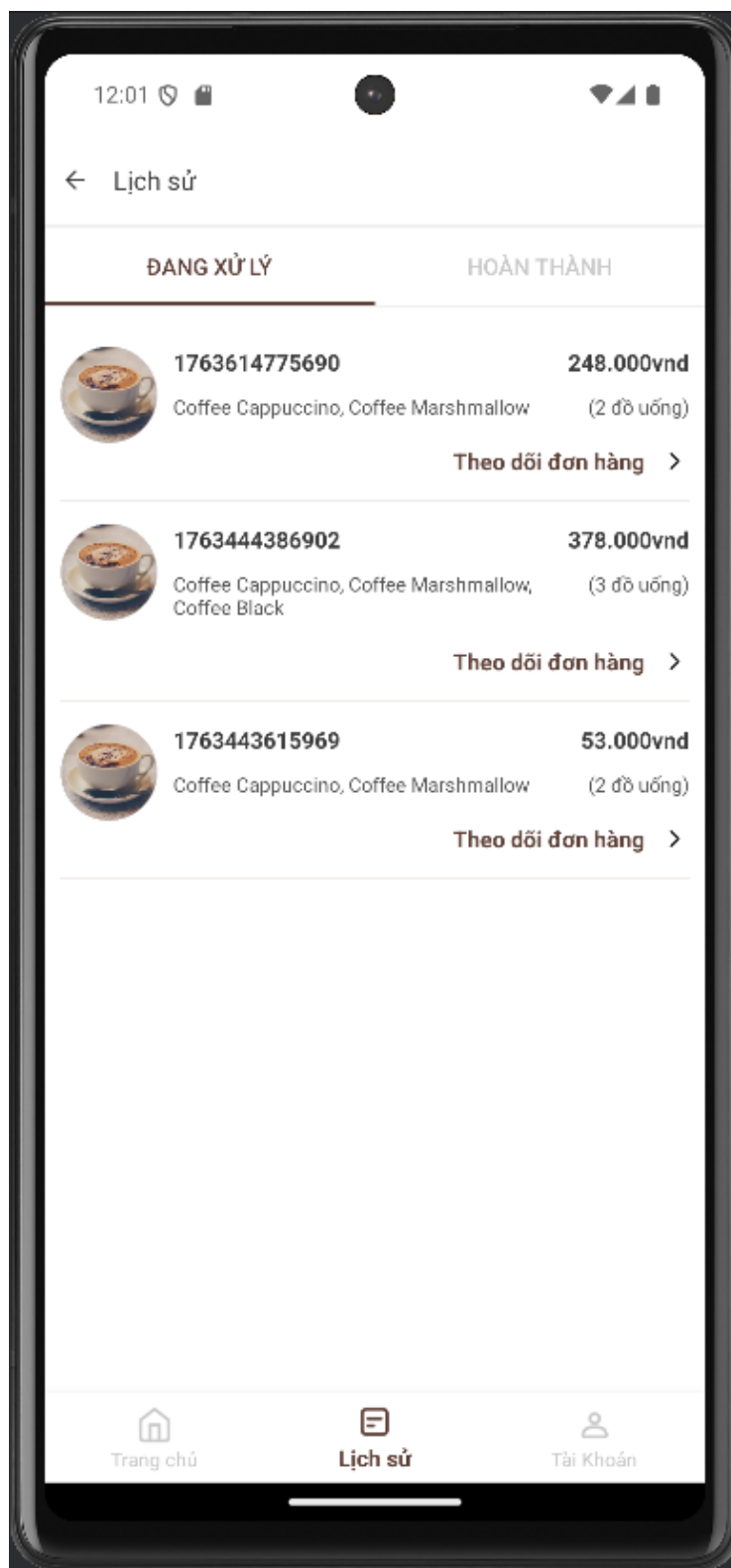
4.3.1.11. Giao diện màn hình đặt hàng thành công



Hình 4.11 Giao diện màn hình đặt hàng thành công

Khách hàng đặt hàng thành công hệ thống sẽ điều hướng đến màn hình này.

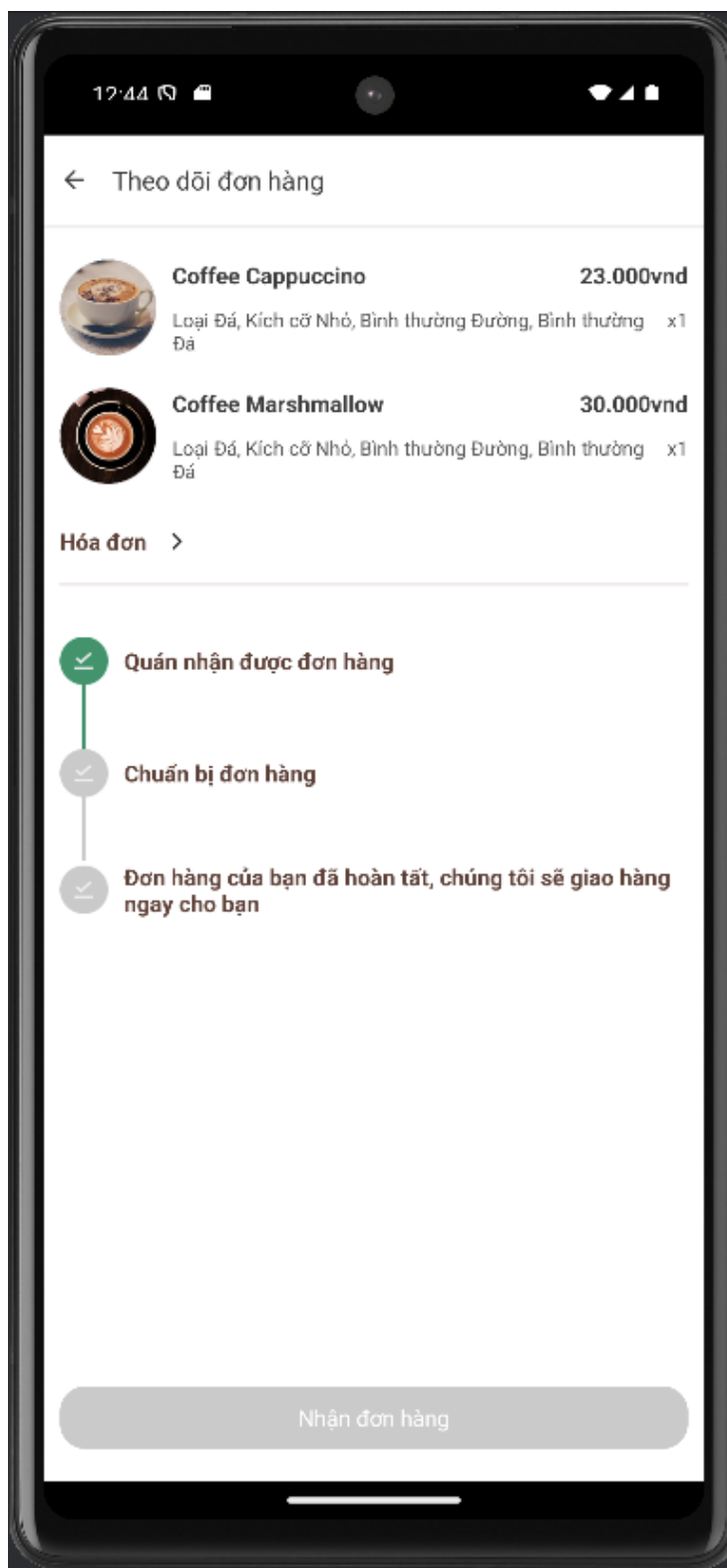
4.3.1.12. Giao diện màn hình lịch sử đơn hàng



Hình 4.12 Giao diện màn hình lịch sử đơn hàng

Khi khách hàng đặt hàng thành công, hệ thống tự động lưu đơn hàng. Khách hàng nhấn vào lịch sử đơn hàng sẽ hiển thị các đơn hàng.

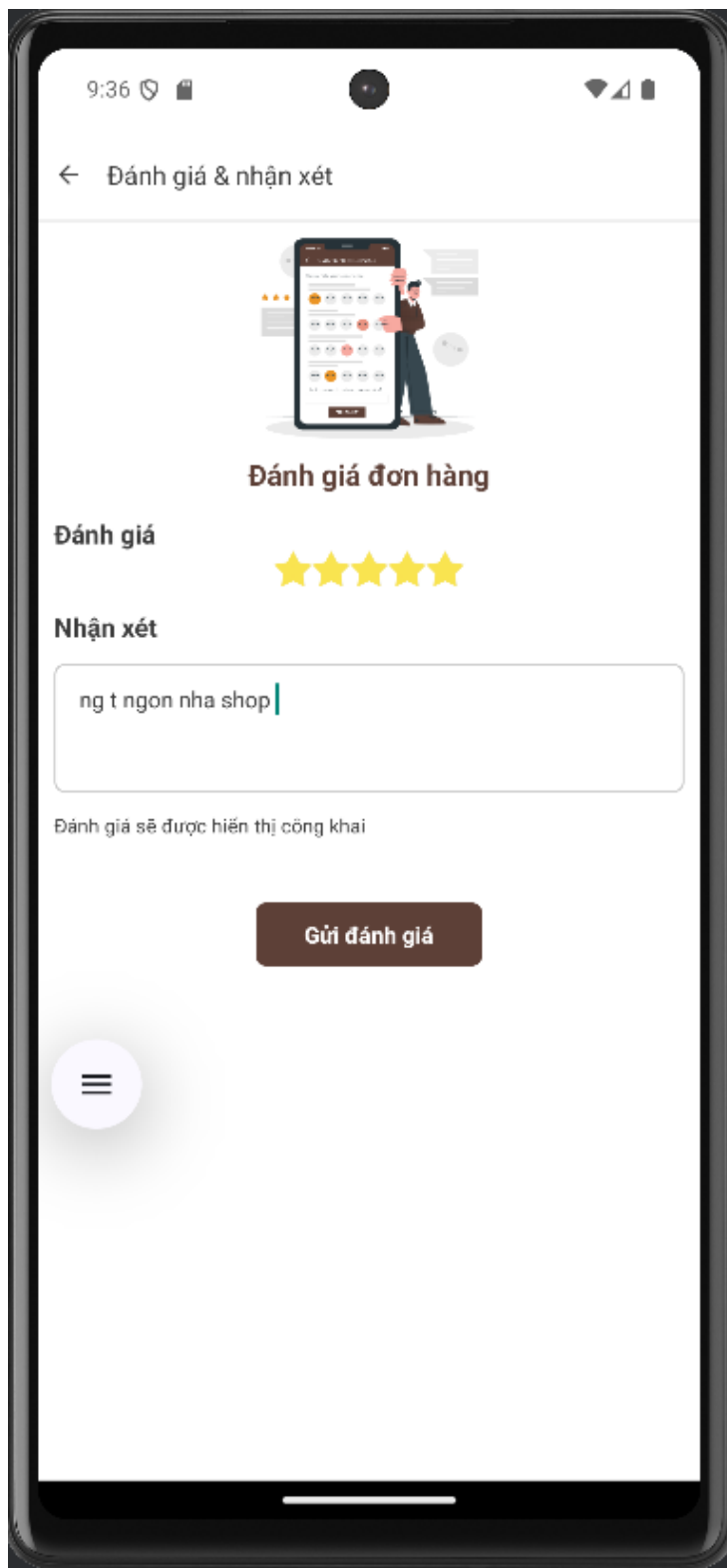
4.3.1.13. Giao diện màn hình theo dõi đơn hàng



Hình 4.13 Giao diện màn hình theo dõi đơn hàng

Khách hàng có thể theo dõi trạng thái đơn hàng của mình từ khi đặt hàng, xác nhận, đang chuẩn bị, đang giao hàng đến khi hoàn thành. Khi có thay đổi trạng thái, khách hàng sẽ nhận được thông báo từ hệ thống.

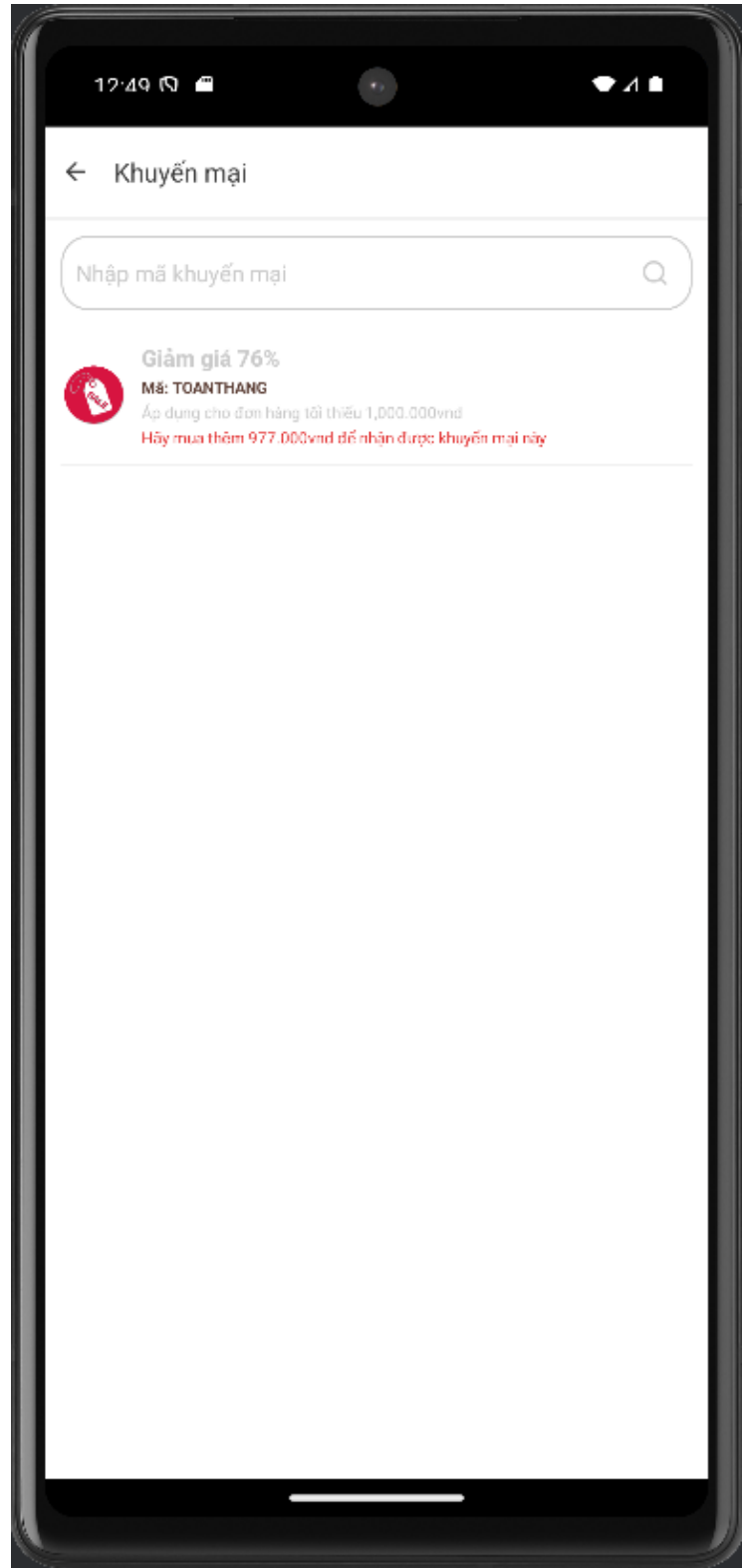
4.3.1.14. Giao diện màn hình đánh giá sản phẩm



Hình 4.14 Giao diện màn hình đánh giá sản phẩm

Sau khi nhận hàng, khách hàng có thể đánh giá và để lại nhận xét về sản phẩm đã mua. Đánh giá giúp khách hàng khác tham khảo trước khi quyết định mua.

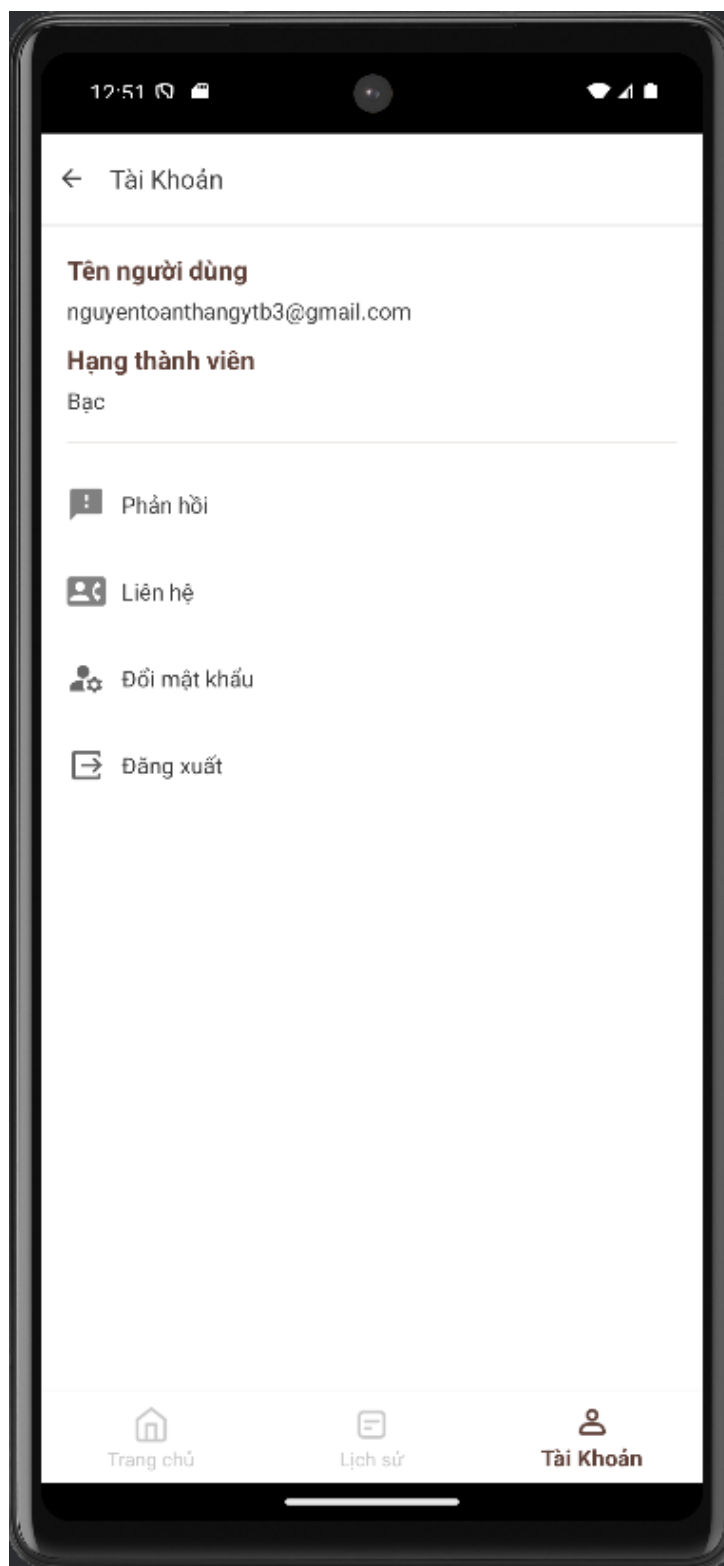
4.3.1.15. Giao diện màn hình voucher



Hình 4.15 Giao diện màn hình voucher

Màn hình này hiển thị danh sách các mã giảm giá và voucher mà khách hàng có thể sử dụng khi đặt hàng để tiết kiệm chi phí.

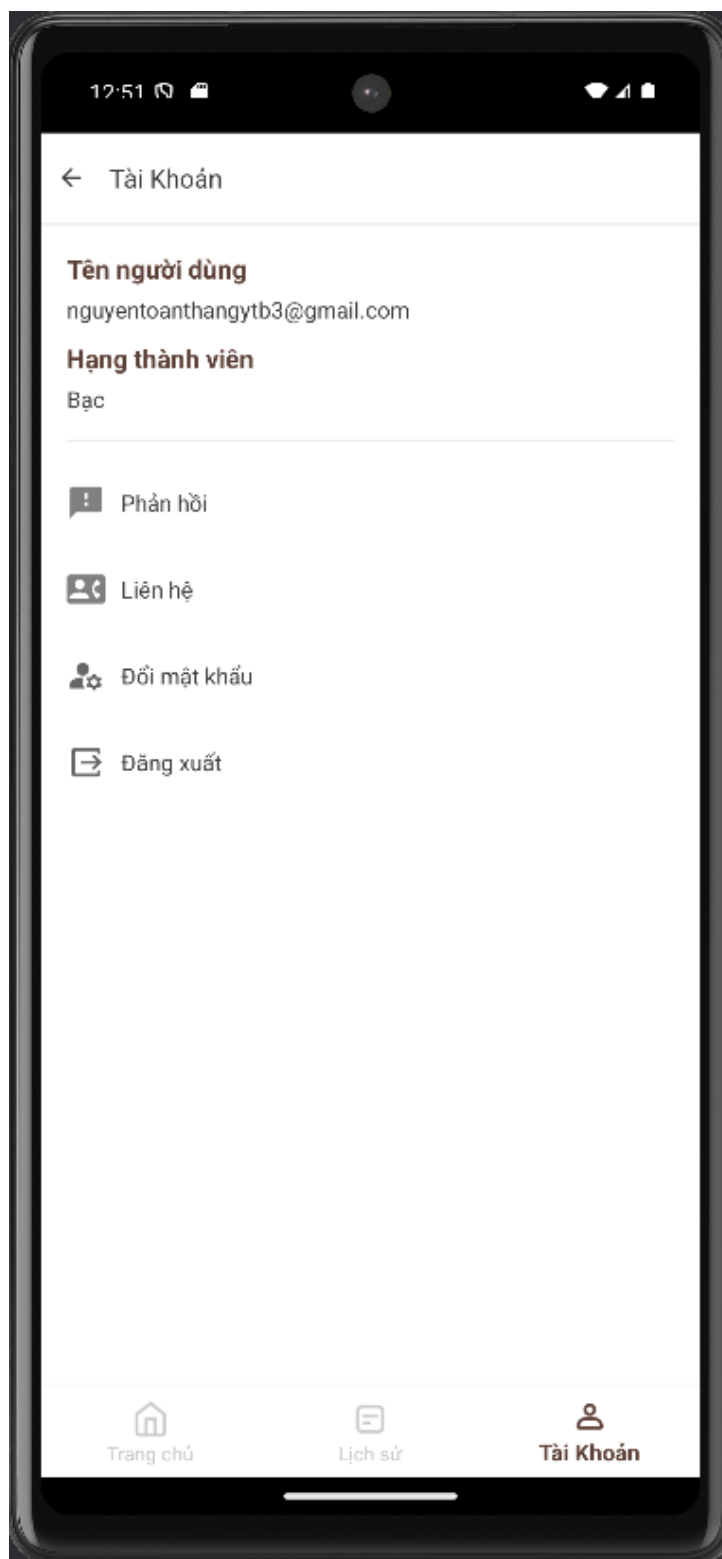
4.3.1.16. Giao diện màn hình tài khoản



Hình 4.16 Giao diện màn hình tài khoản

Màn hình này hiển thị những chức năng của tài khoản bao gồm: thông tin tài khoản, đơn hàng, địa chỉ, voucher, đổi mật khẩu, phản hồi, liên hệ, đăng xuất.

4.3.1.17. Giao diện màn hình thông tin tài khoản



Hình 4.17 Giao diện màn hình thông tin tài khoản

Màn hình này hiển thị thông tin chi tiết của tài khoản, khách hàng có thể chỉnh sửa thông tin cá nhân và xem hạng thành viên của mình (Thường, Bạc, Vàng, Kim cương) dựa trên tổng số tiền đã chi tiêu.

4.3.1.18. Giao diện màn hình đổi mật khẩu

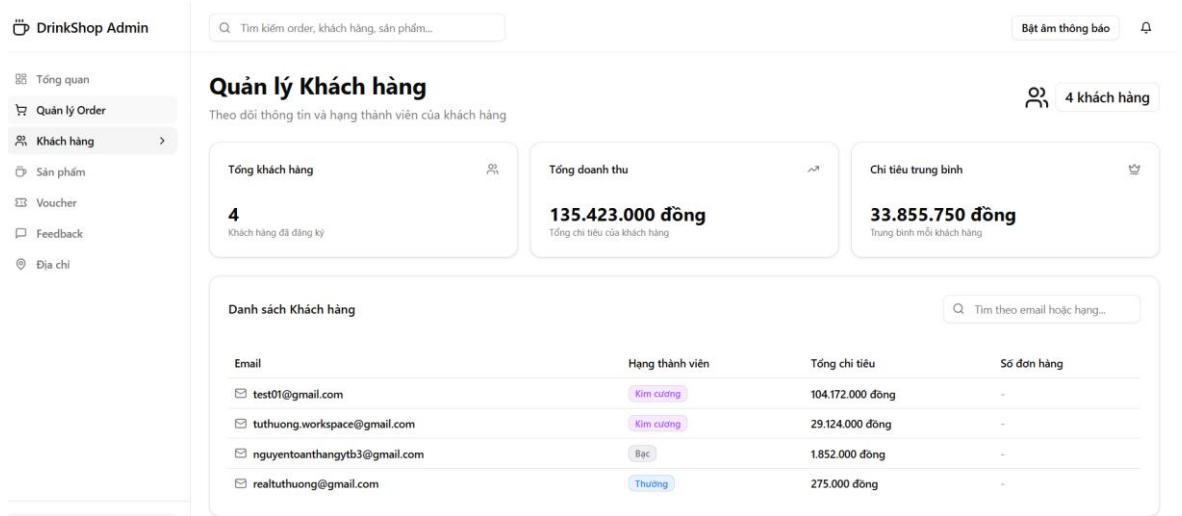


Hình 4.18 Giao diện màn hình đổi mật khẩu

Màn hình chỉ được phép hoạt động khi khách hàng đăng nhập vào hệ thống bằng tài khoản và mật khẩu.

4.3.2. Giao diện người quản trị

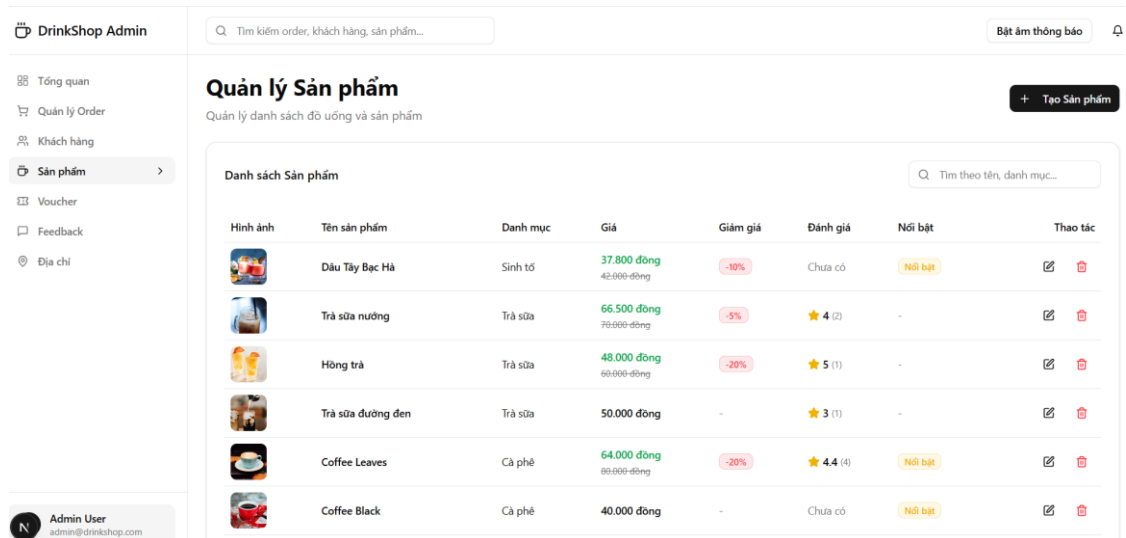
4.3.2.1. Giao diện màn hình trang chủ



Hình 4.19 Giao diện trang chủ người quản trị

Khi người quản trị đăng nhập thành công, hệ thống điều hướng đến màn hình trang chủ, màn hình này gồm thống kê doanh thu, số lượng đơn hàng, chi tiêu và các thông tin quan trọng khác.

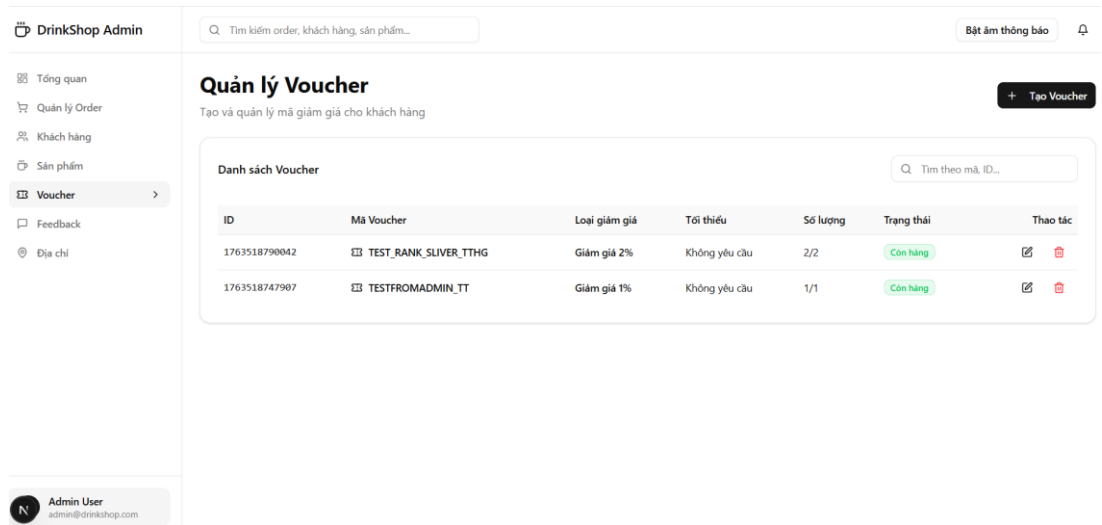
4.3.2.2. Giao diện màn hình quản lý sản phẩm



Hình 4.20 Giao diện trang quản lý sản phẩm

Khi người quản trị chọn tab sản phẩm, hệ thống điều hướng đến màn hình quản lý sản phẩm. Người quản trị có thể tạo mới, chỉnh sửa thông tin sản phẩm bao gồm: tên, mô tả, giá, hình ảnh, danh mục, giảm giá, sản phẩm nổi bật.

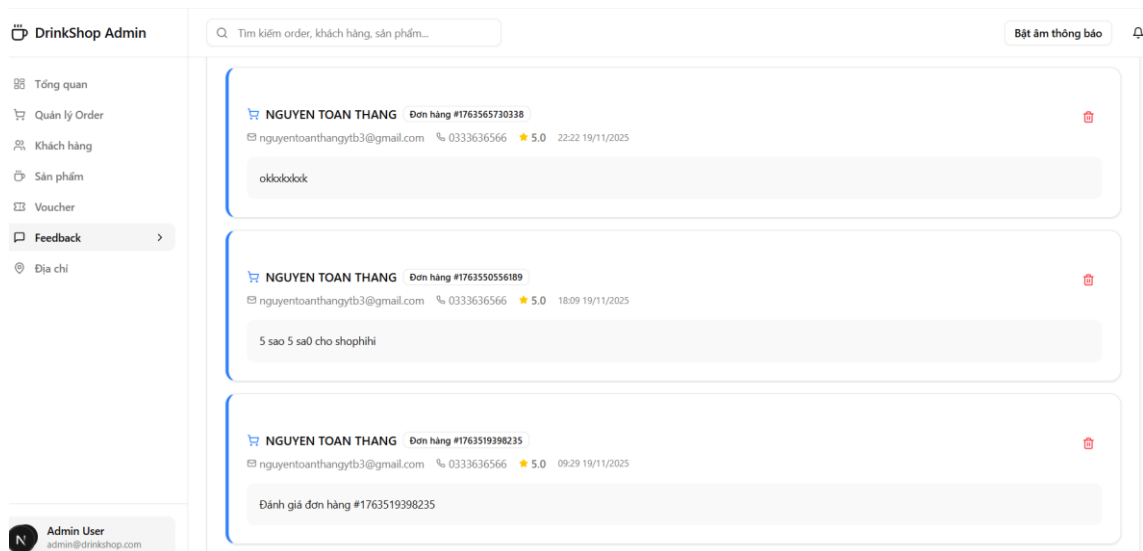
4.3.2.3. Giao diện màn hình quản lý voucher



Hình 4.21 Giao diện màn hình quản lý voucher

Khi người quản trị chọn tab voucher, hệ thống điều hướng đến màn hình quản lý voucher. Người quản trị có thể tạo mới mã giảm giá, chỉnh sửa điều kiện áp dụng, giá trị giảm giá và thời gian hiệu lực.

4.3.2.4. Giao diện màn hình quản lý phản hồi



Hình 4.22 Giao diện màn hình quản lý phản hồi

Khi người quản trị chọn tab phản hồi, hệ thống điều hướng đến màn hình quản lý phản hồi. Người quản trị có thể xem các phản hồi, góp ý từ khách hàng về dịch vụ và sản phẩm.

4.4. Kiểm thử hệ thống

4.4.1. Kiểm thử chức năng (Functional Testing)

4.4.1.1. Kiểm thử đăng nhập và đăng ký

Test case 1: Đăng ký tài khoản mới

- Input: Email hợp lệ (ví dụ: test@gmail.com), mật khẩu ≥ 6 ký tự
- Expected: Tạo tài khoản thành công, tự động đăng nhập và chuyển đến màn hình chính

- Result: Pass - Tài khoản được tạo thành công trong Firebase Authentication

Test case 2: Đăng ký với email đã tồn tại

- Input: Email đã có trong hệ thống
- Expected: Hiện thị thông báo lỗi "Email đã được sử dụng"
- Result: Pass - Firebase trả về lỗi và ứng dụng hiện thị thông báo đúng

Test case 3: Đăng nhập với thông tin đúng

- Input: Email và mật khẩu đúng
- Expected: Đăng nhập thành công, chuyển đến màn hình chính (hoặc màn hình Admin nếu là @admin.com)
- Result: Pass - Xác thực thành công, phân quyền hoạt động đúng

Test case 4: Đăng nhập với mật khẩu sai

- Input: Email đúng, mật khẩu sai
- Expected: Hiện thị thông báo lỗi "Mật khẩu không đúng"
- Result: Pass - Firebase Authentication trả về lỗi và ứng dụng xử lý đúng

Test case 5: Quên mật khẩu

- Input: Email hợp lệ đã đăng ký
- Expected: Gửi email reset mật khẩu thành công
- Result: Pass - Email được gửi qua Firebase

4.4.1.2. Kiểm thử quản lý sản phẩm

Test case 6: Xem danh sách sản phẩm

- Input: Mở màn hình trang chủ
- Expected: Hiện thị danh sách sản phẩm và danh mục từ Firebase

- Result: Pass - Dữ liệu được load và hiển thị đúng

Test case 7: Tìm kiếm sản phẩm

- Input: Nhập từ khóa "Cappuccino"

- Expected: Hiển thị các sản phẩm có tên chứa "Cappuccino"

- Result: Pass - Tìm kiếm hoạt động chính xác

Test case 8: Xem chi tiết sản phẩm

- Input: Nhấn vào một sản phẩm

- Expected: Hiển thị đầy đủ thông tin: tên, giá, mô tả, đánh giá, tùy chọn (size, đá, đường), topping

- Result: Pass - Màn hình chi tiết hiển thị đầy đủ thông tin

Test case 9: Admin thêm sản phẩm mới

- Input: Admin nhập thông tin sản phẩm và upload ảnh

- Expected: Sản phẩm được lưu vào Firebase, hiển thị trong danh sách

- Result: Pass - Sản phẩm được tạo và lưu thành công

Test case 10: Admin sửa/xóa sản phẩm

- Input: Admin chỉnh sửa thông tin hoặc xóa sản phẩm

- Expected: Dữ liệu được cập nhật/xóa trong Firebase, giao diện cập nhật realtime

- Result: Pass - Thao tác CRUD hoạt động đúng

4.4.1.3. Kiểm thử giỏ hàng

Test case 11: Thêm sản phẩm vào giỏ hàng

- Input: Chọn sản phẩm, tùy chọn (size: M, đá: 50%, đường: 50%), topping (Sữa tươi), số lượng: 2

- Expected: Sản phẩm được thêm vào giỏ hàng, số lượng giỏ hàng tăng, dữ liệu lưu vào Room Database

- Result: Pass - Giỏ hàng hoạt động offline, dữ liệu được lưu local

Test case 12: Cập nhật số lượng sản phẩm

- Input: Thay đổi số lượng từ 1 lên 3

- Expected: Tổng tiền giỏ hàng tăng gấp 3 lần, dữ liệu được cập nhật trong Room Database

- Result: Pass - Tính toán và cập nhật chính xác

Test case 13: Xóa sản phẩm khỏi giỏ hàng

- Input: Nhấn nút xóa sản phẩm
- Expected: Sản phẩm bị xóa khỏi giỏ hàng, tổng tiền giảm, dữ liệu được xóa khỏi Room Database

- Result: Pass - Xóa và cập nhật thành công

Test case 14: Giỏ hàng hoạt động offline

- Input: Tắt mạng, thêm sản phẩm vào giỏ hàng
- Expected: Sản phẩm vẫn được thêm vào giỏ hàng (lưu local)
- Result: Pass - Room Database hoạt động offline tốt

4.4.1.4. Kiểm thử đặt hàng

Test case 15: Tạo đơn hàng thành công

- Input: Chọn địa chỉ giao hàng, áp dụng voucher (nếu có), chọn phương thức thanh toán, xác nhận đơn hàng
- Expected: Đơn hàng được tạo trong Firebase, hiển thị màn hình thành công
- Result: Pass - Đơn hàng được lưu với đầy đủ thông tin

Test case 16: Áp dụng voucher

- Input: Nhập mã voucher hợp lệ (ví dụ: "GIAM20")
- Expected: Giảm giá được áp dụng, tổng tiền giảm đúng với giá trị voucher
- Result: Pass - Voucher được tính toán và áp dụng chính xác

Test case 17: Kiểm tra điều kiện voucher

- Input: Nhập voucher có điều kiện số tiền tối thiểu 100.000đ, đơn hàng chỉ 50.000đ
- Expected: Hiển thị thông báo "Đơn hàng chưa đạt số tiền tối thiểu"
- Result: Pass - Validation hoạt động đúng

Test case 18: Tính phí vận chuyển

- Input: Chọn địa chỉ cách quán 5km
- Expected: Phí vận chuyển được tính dựa trên khoảng cách (ví dụ: $5\text{km} \times 5.000\text{đ/km} = 25.000\text{đ}$)
- Result: Pass - Google Maps API tính khoảng cách chính xác, phí ship được tính đúng

Test case 19: Quản lý địa chỉ giao hàng

- Input: Thêm địa chỉ mới, đặt làm địa chỉ mặc định
- Expected: Địa chỉ được lưu với tọa độ GPS, địa chỉ mặc định được cập nhật
- Result: Pass - Địa chỉ được lưu và quản lý đúng

4.4.1.5. Kiểm thử thanh toán

Test case 20: Thanh toán VNPay thành công

- Input: Chọn VNPay, thực hiện thanh toán trong VNPay SDK
- Expected: Thanh toán thành công, callback được xử lý, đơn hàng được cập nhật trạng thái "Đã thanh toán"
- Result: Pass - VNPay SDK hoạt động, callback được xử lý đúng

Test case 21: Thanh toán Blockchain thành công

- Input: Chọn Blockchain, quét QR code, thanh toán bằng ví crypto (MetaMask)
- Expected: Giao dịch được xác nhận trên blockchain Sepolia, đơn hàng được cập nhật với transaction hash
- Result: Pass - Blockchain payment hoạt động, giao dịch được xác nhận

Test case 22: Quy đổi tỉ giá ETH/VND

- Input: Đơn hàng 500.000 VND, tỉ giá ETH = 50.000.000 VND
- Expected: Tính số ETH cần thanh toán = 0.01 ETH
- Result: Pass - CoinGecko API trả về tỉ giá, tính toán chính xác

4.4.1.6. Kiểm thử quản lý đơn hàng

Test case 23: Xem lịch sử đơn hàng

- Input: Vào màn hình đơn hàng
- Expected: Hiện thị danh sách đơn hàng theo tab (Mới, Đang xử lý, Đang giao, Hoàn thành)
- Result: Pass - Đơn hàng được phân loại và hiển thị đúng

Test case 24: Admin cập nhật trạng thái đơn hàng

- Input: Admin thay đổi trạng thái từ "Mới" sang "Đang xử lý"
- Expected: Trạng thái được cập nhật trong Firebase, khách hàng nhận thông báo
- Result: Pass - Cập nhật realtime hoạt động đúng

Test case 25: Đánh giá sản phẩm

- Input: Chọn đơn hàng đã hoàn thành, đánh giá 5 sao, viết review
- Expected: Đánh giá được lưu vào Firebase, hiển thị trong chi tiết sản phẩm
- Result: Pass - Đánh giá được lưu và hiển thị đúng

4.4.1.7. Kiểm thử quản lý hạng thành viên

Test case 26: Nâng hạng thành viên

- Input: Khách hàng có tổng chi tiêu 500.000đ (hạng Bạc), đặt thêm đơn hàng 300.000đ
- Expected: Tổng chi tiêu = 800.000đ, hạng được nâng lên Vàng (nếu điều kiện đủ)
- Result: Pass - Hệ thống tự động cập nhật hạng thành viên

4.4.2. Kiểm thử hiệu năng (Performance Testing)

Test case 27: Thời gian tải danh sách sản phẩm

- Kết quả: Với 100 sản phẩm, thời gian tải < 2 giây
- Đánh giá: Đạt yêu cầu - Sử dụng pagination và lazy loading

Test case 28: Thời gian tải hình ảnh

- Kết quả: Hình ảnh được cache bởi Glide, tải lại < 0.5 giây
- Đánh giá: Đạt yêu cầu - Cache hoạt động hiệu quả

Test case 29: Hoạt động offline

- Kết quả: Giao hàng hoạt động bình thường khi mất kết nối, dữ liệu được đồng bộ khi có mạng
- Đánh giá: Đạt yêu cầu - Room Database hỗ trợ offline tốt

Test case 30: Sử dụng bộ nhớ

- Kết quả: Ứng dụng sử dụng ~150MB RAM khi chạy
- Đánh giá: Chấp nhận được - Trong giới hạn cho phép

4.4.3. Kiểm thử bảo mật (Security Testing)

Test case 31: Mã hóa mật khẩu

- Kết quả: Mật khẩu được mã hóa bởi Firebase Auth, không lưu plain text
- Đánh giá: Đạt yêu cầu - Bảo mật tốt

Test case 32: Phân quyền Admin

- Kết quả: Chỉ email @admin.com mới truy cập được giao diện Admin
- Đánh giá: Đạt yêu cầu - Phân quyền hoạt động đúng

Test case 33: Bảo vệ API keys

- Kết quả: API keys được lưu trong file local, không commit lên Git
- Đánh giá: Đạt yêu cầu - Tuân thủ best practices

Test case 34: Xác minh chữ ký VNPay

- Kết quả: Chữ ký được xác minh trước khi cập nhật trạng thái đơn hàng
- Đánh giá: Đạt yêu cầu - Tránh gian lận thanh toán

4.4.4. Kết quả kiểm thử tổng hợp

Tổng số test case: 34 test cases

Kết quả:

- Pass: 34/34 (100%)
- Fail: 0/34 (0%)

Đánh giá: Hệ thống hoạt động ổn định, đáp ứng đầy đủ các yêu cầu chức năng, hiệu năng và bảo mật.

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

1. Kết luận đạt được

- Kiến thức lập trình Android
 - Nắm vững ngôn ngữ Kotlin và lập trình Android
 - Hiểu về Kotlin Coroutines cho xử lý bất đồng bộ
 - Sử dụng LiveData, StateFlow cho reactive programming
 - Làm việc với Android Architecture Components (ViewModel, Room Database)
- Kiến trúc và thiết kế
 - Nắm vững mô hình MVVM (Model-View-ViewModel)
 - Ứng dụng Dependency Injection với Hilt
 - Tổ chức code theo Repository pattern để tách biệt logic
- Tích hợp Firebase
 - Sử dụng Firebase Authentication để xác thực người dùng
 - Quản lý dữ liệu với Firebase Realtime Database
 - Đồng bộ realtime giữa nhiều thiết bị
 - Bảo mật dữ liệu với Firebase Security Rules
- Tích hợp dịch vụ bên thứ ba
 - Tích hợp VNPAY SDK để thanh toán qua ngân hàng
 - Tích hợp thanh toán blockchain (Ethereum Sepolia testnet)
 - Sử dụng Google Distance Matrix API để tính phí vận chuyển
 - Lấy tỷ giá cryptocurrency qua CoinGecko API
- Hoàn thiện ứng dụng
 - Đăng ký/đăng nhập, quản lý tài khoản
 - Xem sản phẩm, tìm kiếm, chi tiết sản phẩm
 - Quản lý giỏ hàng (Room Database)
 - Đặt hàng với voucher và tính phí vận chuyển tự động
 - Thanh toán đa dạng (COD, VNPAY, Blockchain)
 - Theo dõi đơn hàng realtime
 - Đánh giá sản phẩm và đơn hàng
 - Quản lý dành cho admin (sản phẩm, đơn hàng, voucher, topping)
- Bảo mật
 - Xác thực người dùng an toàn với Firebase Authentication
 - Xác minh chữ ký HMAC-SHA512 cho thanh toán VNPAY

- Xác minh giao dịch blockchain trên mạng
- Bảo vệ API keys khỏi lộ trên repository

2. Hướng phát triển

- Tích hợp Chat/Thông báo:
 - Thêm tính năng nhắn tin giữa khách hàng và admin
 - Push notification khi có đơn hàng mới, cập nhật trạng thái
 - Thông báo khuyến mãi, voucher mới
- Mở rộng sản phẩm:
 - Bán thêm phụ kiện cafe (cốc, ống hút, bánh kẹo)
 - Thêm combo, set đồ uống
 - Chương trình tích điểm, đổi quà
- Hệ thống thành viên:
 - Gói thành viên (VIP, Premium)
 - Nạp tiền vào ví điện tử trong app
 - Thanh toán bằng ví nội bộ kết hợp giảm giá
 - Lịch sử giao dịch ví
- Web Dashboard:
 - Xây dựng web admin để quản lý từ máy tính
 - Dashboard thống kê doanh thu, đơn hàng
 - Báo cáo và phân tích xu hướng bán hàng
 - Quản lý kho hàng và nhân viên
- Tối ưu và mở rộng:
 - Tích hợp bản đồ hiển thị vị trí giao hàng realtime
 - Đánh giá shipper, đánh giá cửa hàng
 - Đặt hàng trước (scheduled orders)
 - Tích hợp nhiều cửa hàng, đặt hàng từ nhiều cửa hàng
- Công nghệ mới:
 - Machine Learning để gợi ý sản phẩm
 - Tích hợp AI chatbot hỗ trợ khách hàng
 - Augmented Reality (AR) để xem trước sản phẩm

TÀI LIỆU THAM KHẢO

- [1] Nguyễn Mạnh Quang (2017), *"Cùng tìm hiểu kotlin(phần 1)"*.
- [2] JetBrains & Open-source Contributors , *"Concise. Cross-platform. Fun"*.
- [3] Lars vogel (2017), *"Android Architecture with MVP or MVVM - Tutorial"*.
- [4] Kyle Jablonski (2019), *"Reactive Programming with RxAndroid in Kotlin: An Introduction"*.
- [5] Janishar Ali (2017), *"Understanding Android Core: Looper, Handler, and HandlerThread"*.
- [6] firebase, *"<https://firebase.google.com/docs?hl=vi>"*.