

ĐẠI HỌC BÁCH KHOA HÀ NỘI

ĐỒ ÁN TỐT NGHIỆP

Xây dựng hệ thống thương mại điện tử trên mobile

LÊ CHÍ DŨNG

dung.lc210216@sis.hust.edu.vn

Chương trình đào tạo: Việt-Nhật (IT-E6)

Giảng viên hướng dẫn: ThS. Vũ Đức Vượng

Chữ ký GVHD

Khoa: Khoa học máy tính

Trường: Công nghệ Thông tin và Truyền thông

HÀ NỘI, 06/2025

LỜI CẢM ƠN

Đầu tiên, em xin được gửi làm cảm ơn đến gia đình của mình. Cảm ơn ba mẹ và chị gái đã luôn sát cánh, hỗ trợ, động viên tinh thần cho con trong suốt hành trình đi học của con từ lớp 1 đến khi tốt nghiệp đại học. Những tình cảm gia đình ấm áp, thiêng liêng ấy chính là nguồn động lực lớn nhất để con luôn cố gắng và vững tin trên chặng đường học vấn của mình.

Tiếp đến, em xin cảm ơn đến những người bạn của mình ở Bách Khoa. Đó là những người bạn cùng lớp quản lý ở đại học, bạn cùng lớp các môn đăng ký tín chỉ, những người bạn thân thiết nhất luôn cùng nhau học tập, vui chơi đồng hành cùng nhau suốt 4 năm tại Bách Khoa. Tất cả các bạn đã giúp cho em có được nhiều kỉ niệm đáng nhớ ở Bách Khoa và hỗ trợ, tư vấn góp ý cho em về mặt chuyên môn cũng như cổ vũ, động viên về tinh thần để giúp em có thể thực hiện xong được DATN.

Cuối cùng, em xin gửi lời cảm ơn đến toàn thể các thầy cô giáo ở trường Công nghệ thông tin và truyền thông nói riêng và Đại học Bách Khoa Hà Nội nói chung đã luôn hỗ trợ và giúp đỡ cho em trong suốt bốn năm học ở Bách Khoa. Những hướng dẫn của các thầy cô sẽ giúp ích cho em rất nhiều trên chặng đường sự nghiệp sau này. Đặc biệt, em cũng xin cảm ơn thầy Vũ Đức Vượng vì đã đưa ra những lời khuyên, những lời hướng dẫn tận tình và đúng đắn kể từ khi bắt đầu GR1 cho đến khi em hoàn thành xong DATN của mình.

Cảm ơn Bách Khoa - Mái nhà thứ hai của tôi.

One Love-One Future

LỜI CAM KẾT

Họ và tên sinh viên: Lê Chí Dũng

MSSV: 20210216

Điện thoại liên lạc: 0335620803

Email: dung.lc210216@sis.hust.edu.vn

Lớp: Việt Nhật 02 - K66

Chương trình đào tạo: Công nghệ thông tin Việt-Nhật 2021

Tôi – Lê Chí Dũng – cam kết Đồ án Tốt nghiệp (ĐATN) là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của Ths. Vũ Đức Vượng. Các kết quả nêu trong ĐATN là trung thực, là thành quả của riêng tôi, không sao chép theo bất kỳ công trình nào khác. Tất cả những tham khảo trong ĐATN – bao gồm hình ảnh, bảng biểu, số liệu, và các câu trích dẫn – đều được ghi rõ ràng và đầy đủ nguồn gốc trong danh mục tài liệu tham khảo. Tôi xin hoàn toàn chịu trách nhiệm với dù chỉ một sao chép vi phạm quy chế của nhà trường.

Hà Nội, ngày 16 tháng 6 năm 2025

Tác giả ĐATN

Lê Chí Dũng

TÓM TẮT NỘI DUNG ĐỒ ÁN

Trong thời đại số hóa hiện nay, thương mại điện tử đang phát triển mạnh mẽ và trở thành xu hướng phổ biến trong lĩnh vực kinh doanh. Tuy nhiên, phần lớn các giải pháp hiện tại tập trung vào các nền tảng web, trong khi nhu cầu mua sắm qua thiết bị di động ngày càng tăng cao. Bên cạnh đó, các hệ thống thương mại điện tử hiện có thường có cấu trúc phức tạp, khó mở rộng và bảo trì, đặc biệt là khi triển khai cả ứng dụng di động và hệ thống quản trị web song song. Một số hướng tiếp cận như xây dựng native app cho từng nền tảng hoặc sử dụng hệ thống quản lý riêng biệt cho web và mobile đều tồn tại những hạn chế như tốn kém chi phí phát triển, khó đồng bộ dữ liệu và khó kiểm soát hệ thống toàn diện.

Để giải quyết vấn đề trên, em lựa chọn hướng tiếp cận xây dựng hệ thống thương mại điện tử đa nền tảng sử dụng Flutter cho ứng dụng di động, React cho giao diện quản trị Web, Spring Boot cho Backend, và Firebase làm cơ sở dữ liệu chung. Hướng tiếp cận này giúp đảm bảo hiệu năng, khả năng mở rộng, dễ dàng bảo trì cũng như đồng bộ dữ liệu giữa các nền tảng. Flutter được lựa chọn nhờ khả năng phát triển ứng dụng đa nền tảng từ một codebase duy nhất, còn MVC giúp tách biệt logic, giao diện và xử lý dữ liệu rõ ràng. Tất cả dữ liệu được lưu trữ và đồng bộ qua Firebase, đảm bảo tính nhất quán và thời gian thực, cấu trúc NoSQL linh hoạt và tối ưu hóa hiệu năng cho mobile.

Đóng góp chính của đồ án là xây dựng một hệ sinh thái thương mại điện tử đầy đủ, có khả năng hoạt động hiệu quả trên nhiều nền tảng với chi phí phát triển tối ưu. Kết quả cuối cùng là một hệ thống hoàn chỉnh, có tính thực tiễn cao, dễ dàng triển khai và mở rộng trong môi trường kinh doanh thực tế.

MỤC LỤC

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI.....	1
1.1 Đặt vấn đề.....	1
1.2 Mục tiêu và phạm vi đề tài.....	1
1.3 Định hướng giải pháp.....	2
1.4 Bố cục đồ án	3
CHƯƠNG 2. PHÂN TÍCH YÊU CẦU.....	4
2.1 Khảo sát hiện trạng	4
2.1.1 Khảo sát hệ thống hiện có	4
2.1.2 Các chức năng cần có cho hệ thống.....	5
2.1.3 Tính năng đề xuất	5
2.2 Tổng quan chức năng	6
2.2.1 Biểu đồ use case tổng quát	7
2.2.2 Biểu đồ use case phân rã "Quản lý đơn hàng"	9
2.2.3 Biểu đồ use case phân rã "Quản lý người dùng"	9
2.2.4 Biểu đồ use case phân rã "Quản lý cửa hàng".....	10
2.2.5 Biểu đồ use case phân rã "Quản lý category"	11
2.2.6 Biểu đồ use case phân rã "Quản lý nhiệm vụ"	12
2.2.7 Biểu đồ use case phân rã "Xem thống kê".....	12
2.2.8 Biểu đồ use case phân rã "Quản lý voucher"	13
2.2.9 Biểu đồ use case phân rã "Quản lý sản phẩm"	14
2.2.10 Biểu đồ use case phân rã "Quản lý thương hiệu".....	15
2.2.11 Biểu đồ use case phân rã "Quản lý bạn bè"	15
2.2.12 Biểu đồ use case phân rã "Quản lý giỏ hàng"	16
2.2.13 Biểu đồ use case phân rã "Quản lý địa chỉ"	17

2.2.14 Biểu đồ use case phân rã "Tham gia nhóm săn sale"	18
2.2.15 Biểu đồ use case phân rã "Quản lý sản phẩm yêu thích"	18
2.2.16 Biểu đồ use case phân rã "Quản lý tài khoản.....	19
2.3 Đặc tả chức năng	20
2.3.1 Đặc tả use case "Xem sản phẩm"	20
2.3.2 Đặc tả use case "Thêm sản phẩm vào giỏ hàng"	21
2.3.3 Đặc tả use case "Đặt hàng"	23
2.3.4 Đặc tả use case "Thanh toán điện tử"	25
2.3.5 Đặc tả use case "Tham gia nhiệm vụ"	26
2.3.6 Đặc tả use case "Mời bạn bè tham gia nhóm".....	28
2.3.7 Đặc tả use case "Viết bình luận"	28
2.4 Yêu cầu phi chức năng	30
2.4.1 Yêu cầu về giao diện và trải nghiệm người dùng.....	30
2.4.2 Yêu cầu về tính dễ bảo trì.....	30
2.4.3 Yêu cầu về bảo mật	30
2.4.4 Yêu cầu về hiệu năng.....	31
CHƯƠNG 3. CÔNG NGHỆ SỬ DỤNG.....	32
3.1 Các công nghệ sử dụng bên phía mobile	32
3.1.1 Flutter[1].....	32
3.1.2 Dart[10]	33
3.1.3 Firebase[4]	34
3.2 Các công nghệ sử dụng bên phía server.....	36
3.2.1 ReactJS[2]	36
3.2.2 Spring Boot Framework[3]	36
3.3 Các công nghệ sử dụng để phát triển tính năng gợi ý.....	37
3.3.1 Python[11]	37

3.3.2 LightFM[6]	37
3.3.3 NodeJS[12]	38
3.3.4 Kaggle IDE[13]	38
3.4 Công nghệ sử dụng để xây dựng chatbot	38
CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG	40
4.1 Thiết kế kiến trúc.....	40
4.1.1 Lựa chọn kiến trúc phần mềm	40
4.1.2 Thiết kế tổng quan.....	43
4.1.3 Thiết kế chi tiết gói	46
4.2 Thiết kế chi tiết.....	49
4.2.1 Thiết kế giao diện	49
4.2.2 Thiết kế lớp	52
4.2.3 Thiết kế cơ sở dữ liệu	57
4.3 Xây dựng ứng dụng.....	67
4.3.1 Thư viện và công cụ sử dụng	67
4.3.2 Kết quả đạt được	68
4.3.3 Minh họa các chức năng chính	70
4.4 Kiểm thử.....	83
4.4.1 Kiểm thử tương thích.....	83
4.4.2 Kiểm thử hộp đen.....	83
CHƯƠNG 5. CÁC GIẢI PHÁP VÀ ĐÓNG GÓP NỔI BẬT	86
5.1 Tối ưu thời gian phản hồi khi xem danh sách sản phẩm	86
5.1.1 Đặt vấn đề	86
5.1.2 Giải pháp	86
5.1.3 Kết quả	87

5.2 Triển khai tính năng "Gợi ý sản phẩm cá nhân hóa theo người dùng"	88
5.2.1 Đặt vấn đề	88
5.2.2 Giải pháp	88
5.2.3 Kết quả	90
5.3 Triển khai tính năng "Chatbot tự động hóa quy trình mua sắm"	92
5.3.1 Đặt vấn đề	92
5.3.2 Giải pháp	93
5.3.3 Kết quả	94
5.4 Triển khai tính năng "Săn sale cùng bạn bè và chia sẻ nhanh qua mạng xã hội"	97
5.4.1 Đặt vấn đề	97
5.4.2 Giải pháp	98
5.4.3 Kết quả	100
5.5 Triển khai tính năng "Làm nhiệm vụ và đổi thưởng"	101
5.5.1 Đặt vấn đề	101
5.5.2 Giải pháp	102
5.5.3 Kết quả	105
5.6 Triển khai hệ thống "voucher, áp dụng và quản lý voucher"	106
5.6.1 Đặt vấn đề	106
5.6.2 Giải pháp	107
5.6.3 Kết quả	108
CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	110
6.1 Kết luận	110
6.2 Hướng phát triển.....	111
6.2.1 Tính năng Livestream kết hợp Đấu giá ngược	111
6.2.2 Tính năng Trợ lý ảo gợi ý phối đồ (AI Stylist).....	111

6.2.3 Tính năng Trợ lý mua sắm thông minh tích hợp AI.....	111
TÀI LIỆU THAM KHẢO.....	114

DANH MỤC HÌNH VẼ

Hình 2.1 Biểu đồ use case tổng quan	7
Hình 2.2 Biểu đồ use case phân rã "Quản lý đơn hàng"	9
Hình 2.3 Biểu đồ use case phân rã "Quản lý người dùng"	10
Hình 2.4 Biểu đồ use case phân rã "Quản lý cửa hàng"	10
Hình 2.5 Biểu đồ use case phân rã "Quản lý category"	11
Hình 2.6 Biểu đồ use case phân rã "Quản lý nhiệm vụ"	12
Hình 2.7 Biểu đồ use case phân rã "Xem thống kê"	13
Hình 2.8 Biểu đồ use case phân rã "Quản lý voucher"	13
Hình 2.9 Biểu đồ use case phân rã "Quản lý sản phẩm"	14
Hình 2.10 Biểu đồ use case phân rã "Quản lý thương hiệu"	15
Hình 2.11 Biểu đồ use case phân rã "Quản lý bạn bè"	16
Hình 2.12 Biểu đồ use case phân rã "Quản lý giỏ hàng"	16
Hình 2.13 Biểu đồ use case phân rã "Quản lý địa chỉ"	17
Hình 2.14 Biểu đồ use case phân rã "Tham gia nhóm săn sale"	18
Hình 2.15 Biểu đồ use case phân rã "Quản lý sản phẩm yêu thích"	19
Hình 2.16 Biểu đồ use case phân rã "Quản lý tài khoản"	19
 Hình 4.1 Kiến trúc Client-Server (nguồn: https://codelearn.io/sharing/tim-hieu-ve-mo-hinh-client-server)	40
Hình 4.2 Kiến trúc flutter mobile (nguồn: https://codewithan-drea.com/articles/comparison-flutter-app-architectures/)	41
Hình 4.3 Kiến trúc phân tầng của server Spring Boot (nguồn: https://shareprogramming.net/kien-truc-va-workflow-cua-spring-boot-framework/)	42
Hình 4.4 Biểu đồ package tổng quan phía Mobile	43
Hình 4.5 Biểu đồ package tổng quan Client Webadmin	44
Hình 4.6 Biểu đồ package tổng quan Server Webadmin	45
Hình 4.7 Biểu đồ package chi tiết cho tính năng "Xem thông báo" phía mobile	47
Hình 4.8 Biểu đồ package chi tiết cho tính năng "Duyệt yêu cầu tham gia nhóm" phía mobile	48
Hình 4.9 Biểu đồ package chi tiết cho tính năng "Xem chi tiết sản phẩm" phía mobile	49
Hình 4.10 Bố cục tổng quát của ứng dụng	50

Hình 4.11	Thiết kế giao diện cho màn hình cửa hàng	50
Hình 4.12	Thiết kế giao diện cho màn hình profile	51
Hình 4.13	Biểu đồ lớp cho chức năng "Mời bạn bè tham gia nhóm" . .	52
Hình 4.14	Biểu đồ tuần tự cho chức năng "Mời bạn bè tham gia nhóm"	53
Hình 4.15	Biểu đồ lớp cho chức năng "Duyệt lời mời tham gia nhóm" .	54
Hình 4.16	Biểu đồ tuần tự cho chức năng "Duyệt lời mời tham gia nhóm"	56
Hình 4.17	Biểu đồ thực thể liên kết	57
Hình 4.18	Màn hình thêm bạn bè	70
Hình 4.19	Màn hình chấp nhận lời mời tham gia nhóm	71
Hình 4.20	Màn hình tạo nhóm săn sale	72
Hình 4.21	Màn hình điểm danh hàng ngày	73
Hình 4.22	Màn hình chi tiết nhóm săn sale	74
Hình 4.23	Màn hình mời bạn bè tham gia nhóm săn sale	75
Hình 4.24	Màn hình nhiệm vụ chi tiết	76
Hình 4.25	Màn hình đơn hàng của tôi	77
Hình 4.26	Màn hình thông báo	78
Hình 4.27	Màn hình sản phẩm chi tiết	79
Hình 4.28	Màn hình xem bình luận	80
Hình 4.29	Màn hình gợi ý sản phẩm	81
Hình 4.30	Màn hình anh sách sản phẩm	82
Hình 5.1	So sánh khi sử dụng hàm Future.wait so với khi gọi API tuần tự	87
Hình 5.2	Minh họa thời gian phản hồi của hệ thống sau tối ưu	87
Hình 5.3	Số lượng người dùng đã ghi nhận tương tác	90
Hình 5.4	Tổng số lượng sản phẩm trong hệ thống	90
Hình 5.5	Tổng số tương tác được ghi nhận trong bảng user_events	90
Hình 5.6	Tỷ lệ các loại tương tác: view, click, add to cart, purchase .	90
Hình 5.7	Lên lịch job huấn luyện định kỳ bằng Cloud Scheduler . .	91
Hình 5.8	Bộ tham số tối ưu thu được từ Optuna	91
Hình 5.9	So sánh kết quả mô hình với và không sử dụng fine-tune .	91
Hình 5.10	Kết quả mô hình không sử dụng metadata	92
Hình 5.11	Kết quả mô hình sử dụng metadata	92
Hình 5.12	Người dùng bắt đầu cuộc hội thoại bằng lời chào, bot phản hồi ngay bằng các quick button gợi ý khám phá	95
Hình 5.13	Khi người dùng bấm khám phá 1 sản phẩm, bot hiển thị danh sách sản phẩm tìm thấy	95

Hình 5.14 Bot đưa ra các gợi ý là các quick button để người dùng có thể thêm nhanh sản phẩm vào giỏ hàng hoặc nhập câu lệnh tương tự theo mẫu hoặc xem giỏ hàng	95
Hình 5.15 Bot điều hướng đến giao diện giỏ hàng khi người dùng chọn thêm vào giỏ hàng	96
Hình 5.16 Bot điều hướng đến xem đơn hàng và đặt hàng nếu người dùng chọn xem đơn hàng	96
Hình 5.17 Bot hiển thị danh sách các mục để người dùng có thể quick button click và xem thêm các sản phẩm khác tiếp theo	96
Hình 5.18 Ngoài việc bấm quickly button, người dùng vẫn có thể tự nhập để tìm kiếm sản phẩm mình muốn ngoài những gợi ý của bot	97
Hình 5.19 Ngoài việc bấm quickly button, người dùng vẫn có thể tự nhập để thêm sản phẩm và số lượng mình thích vào giỏ hàng ngoài những gợi ý có sẵn của bot	97
Hình 5.20 Bot cảm ơn người dùng và kết thúc phiên trò chuyện.	97
Hình 5.21 Người dùng chọn chức năng chia sẻ qua mạng xã hội, có 2 lựa chọn là gửi link hoặc gửi QR	100
Hình 5.22 Khi người dùng chọn QR, hệ thống sẽ gen sẵn ảnh QR thông tin nhóm để người đó chia sẻ	100
Hình 5.23 Hình ảnh cả link và cả QR đã được chia sẻ qua messenger . .	100
Hình 5.24 Nếu bấm vào link hoặc quét QR mà người đó chưa cài app, sẽ yêu cầu họ nhập email để invite qua email	101
Hình 5.25 Nếu bấm vào link hoặc quét QR mà người đó đã cài app, sẽ có popup mời họ tham gia	101
Hình 5.26 Khi tập hợp đủ người, sẽ có thông báo nhóm tập hợp đủ người và thông báo gửi tới các thành viên đã nhận voucher thành công .	101
Hình 5.27 Biểu đồ class cho tính năng Mission Tracking	103
Hình 5.28 Màn hình danh sách các nhiệm vụ có sẵn trong hệ thống và trạng thái của người dùng đã hay chưa xử lý	105
Hình 5.29 Màn hình quy đổi điểm thưởng lấy voucher	105
Hình 5.30 Thông báo khi người dùng hoàn thành 1 nhiệm vụ và nhận được điểm thưởng tương ứng	105
Hình 5.31 Màn hình web admin CRUD nhiệm vụ	106
Hình 5.32 Màn hình thống kê theo type	106
Hình 5.33 Màn hình thống kê theo lịch sử thực hiện nhiệm vụ của người dùng	106

Hình 5.34 Màn hình danh lịch sử các voucher của người dùng có phân theo 3 nhóm khác nhau	108
Hình 5.35 Pop up gợi ý những voucher đủ điều kiện để người dùng sử dụng với điều kiện đơn hàng hiện tại	108
Hình 5.36 Màn hình checkout khi apply thành công 1 voucher	108
Hình 5.37 Màn hình web admin CRUD voucher	109
Hình 5.38 Màn hình thống kê voucher theo từng type	109
Hình 5.39 Màn hình thống kê voucher theo lịch sử áp dụng của người dùng	109

DANH MỤC BẢNG BIỂU

Bảng 2.1	Mô tả các tác nhân trong hệ thống	8
Bảng 2.2	Bảng đặc tả Use case "Xem sản phẩm"	20
Bảng 2.3	Dữ liệu đầu vào cho Use case "Xem sản phẩm"	21
Bảng 2.4	Bảng đặc tả Use case "Thêm sản phẩm vào giỏ hàng"	22
Bảng 2.5	Dữ liệu đầu vào cho Use case "Thêm sản phẩm vào giỏ hàng"	23
Bảng 2.6	Bảng đặc tả Use case "Đặt hàng"	24
Bảng 2.7	Bảng đặc tả Use case "Thanh toán điện tử"	25
Bảng 2.8	Dữ liệu đầu vào cho Use case "Thanh toán điện tử"	26
Bảng 2.9	Bảng đặc tả Use case Tham gia nhiệm vụ	27
Bảng 2.10	Bảng đặc tả Use case "Mời bạn bè tham gia nhóm"	28
Bảng 2.11	Bảng đặc tả Use case "Viết bình luận"	29
Bảng 2.12	Dữ liệu đầu vào cho Use case "Viết bình luận"	30
Bảng 4.1	Chi tiết bảng brands	58
Bảng 4.2	Chi tiết bảng categories	58
Bảng 4.3	Chi tiết bảng shops	59
Bảng 4.4	Chi tiết bảng products	59
Bảng 4.5	Chi tiết bảng reviews	60
Bảng 4.6	Chi tiết bảng users	60
Bảng 4.7	Chi tiết bảng daily_checkins	61
Bảng 4.8	Chi tiết bảng missions	61
Bảng 4.9	Chi tiết bảng sale_groups	62
Bảng 4.10	Chi tiết bảng vouchers	63
Bảng 4.11	Chi tiết bảng addresses	63
Bảng 4.12	Chi tiết bảng orders	64
Bảng 4.13	Chi tiết bảng claimed_vouchers	64
Bảng 4.14	Chi tiết bảng friend_requests	64
Bảng 4.15	Chi tiết bảng friends	65
Bảng 4.16	Chi tiết bảng group_requests	65
Bảng 4.17	Chi tiết bảng notifications	66
Bảng 4.18	Chi tiết bảng reward_items	66
Bảng 4.19	Chi tiết bảng user_missions	66
Bảng 4.20	Chi tiết bảng user_events	67
Bảng 4.21	Chi tiết bảng chatbot	67
Bảng 4.22	Danh sách thư viện và công cụ sử dụng	68

Bảng 4.23 Bảng thống kê các thông tin về ứng dụng	69
Bảng 4.24 Bảng kiểm thử sự tương thích của phần mềm trên các loại thiết bị	83
Bảng 4.25 Bảng kiểm thử nhóm chức năng Đơn hàng của tôi	83
Bảng 4.26 Bảng kiểm thử nhóm chức năng Nhóm săn sale	84
Bảng 4.27 Bảng kiểm thử nhóm chức năng voucher của tôi	84

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

1.1 Đặt vấn đề

Trong bối cảnh toàn cầu hóa và chuyển đổi số mạnh mẽ, thương mại điện tử (TMĐT) đã trở thành một kênh giao dịch chủ đạo, thay đổi thói quen tiêu dùng của hàng triệu người trên thế giới. Đặc biệt, sự phổ biến nhanh chóng của điện thoại thông minh cùng với mạng di động tốc độ cao đã tạo điều kiện thuận lợi để TMĐT trên nền tảng di động (mobile commerce) phát triển vượt bậc. Tại Việt Nam, theo các báo cáo gần đây, tỷ lệ người tiêu dùng sử dụng thiết bị di động để truy cập và thực hiện các giao dịch mua sắm trực tuyến chiếm đến hơn 70%, và con số này vẫn không ngừng gia tăng. Điều này phản ánh xu hướng tất yếu của hành vi tiêu dùng hiện đại: nhanh chóng, tiện lợi và linh hoạt.

Tuy nhiên, TMĐT trên mobile vẫn tồn tại nhiều vấn đề thực tiễn đáng quan ngại. Giao diện và trải nghiệm người dùng (UX) trên nhiều ứng dụng chưa được tối ưu hóa cho các thiết bị có kích thước màn hình nhỏ, gây khó khăn trong thao tác và dễ dẫn đến việc bỏ giỏ hàng. Bên cạnh đó, tốc độ tải chậm, lỗi hiển thị, gián đoạn kết nối hay thông báo quá tải cũng ảnh hưởng trực tiếp đến quyết định mua hàng. Vấn đề bảo mật và quyền riêng tư cũng là mối lo hàng đầu, khi ngày càng nhiều người dùng bị lộ thông tin cá nhân hoặc gặp phải các vụ lừa đảo qua nền tảng di động. Ngoài ra, các hệ thống thanh toán trực tuyến trên mobile vẫn chưa thực sự đồng bộ và thuận tiện, khiến nhiều người tiêu dùng, đặc biệt là nhóm người trung niên hoặc vùng nông thôn, e ngại khi mua sắm qua điện thoại.

Tình trạng này đặt ra yêu cầu cấp thiết đối với các doanh nghiệp và nhà phát triển hệ thống TMĐT: cần nhanh chóng cải tiến trải nghiệm người dùng trên mobile, đảm bảo tính ổn định và an toàn của hệ thống, đồng thời tạo ra các giải pháp thân thiện hơn với đa dạng đối tượng người dùng. Việc nâng cao chất lượng TMĐT trên di động không chỉ giúp tăng trưởng doanh số và mở rộng thị trường mà còn góp phần đẩy mạnh tiến trình số hóa nền kinh tế quốc gia. Do đó, vấn đề này có tính thời sự, cấp bách và mang ý nghĩa thực tiễn sâu sắc trong bối cảnh hiện nay.

1.2 Mục tiêu và phạm vi đề tài

Nhằm mục đích khảo sát và tìm giải pháp cho đề tài em vừa nêu trên, em đã thử trải nghiệm hai ứng dụng Shopee, TiktokShop. Từ đó em xác định những yếu tố, tính năng mà mình có thể học hỏi và có thể cải tiến cho hệ thống của mình.

Hiện nay, các nền tảng thương mại điện tử phổ biến như Shopee, Lazada, Tiki hay Amazon đều đã phát triển mạnh mẽ trên nền tảng di động, đáp ứng tương đối

đầy đủ các nhu cầu cơ bản của người tiêu dùng như tìm kiếm sản phẩm, mua hàng, thanh toán trực tuyến và theo dõi đơn hàng. Nhiều ứng dụng còn tích hợp ví điện tử, hỗ trợ giao hàng nhanh và thường xuyên tung ra các chương trình khuyến mãi, giảm giá hấp dẫn để thu hút người dùng. Tuy nhiên, hầu hết các ứng dụng này vẫn đang khai thác theo hướng đại trà, chưa cá nhân hóa sâu sát hành vi người dùng, và phần lớn trải nghiệm mua sắm vẫn mang tính đơn lẻ, thiếu tính kết nối cộng đồng và động lực sử dụng lâu dài.

Ứng dụng của em được phát triển với định hướng kế thừa những tính năng cơ bản của các nền tảng TMĐT hiện nay nhưng đồng thời tạo ra những điểm đột phá để nâng cao trải nghiệm người dùng.

- Thứ nhất, ứng dụng áp dụng mô hình học máy (Machine Learning) để phân tích hành vi người dùng và đưa ra gợi ý sản phẩm được cá nhân hóa, giúp người dùng tiếp cận đúng nhu cầu nhanh hơn, chính xác hơn.
- Thứ hai, tính năng săn sale theo nhóm cho phép người dùng kết bạn và cùng nhau tham gia vào các chương trình giảm giá, tạo ra sự tương tác xã hội và tối ưu hóa chi phí mua sắm.
- Thứ ba, hệ thống nhiệm vụ và thử thách theo ngày/tuần mang đến trải nghiệm mới mẻ, giúp tăng mức độ gắn bó với ứng dụng thông qua cơ chế tích điểm thưởng và đổi thưởng lấy voucher, giảm tiền hấp dẫn.
- Thứ tư, xây dựng chatbot tự động hóa quy trình mua sắm, hỗ trợ người dùng mới có những trải nghiệm đầy thú vị, đơn giản khi chỉ cần vài câu lệnh hay vài lần click các button gợi ý của bot là có thể hoàn thành quy trình mua sắm của riêng mình.

1.3 Định hướng giải pháp

Để xây dựng một hệ thống đáp ứng đầy đủ những chức năng mà em nêu ra ở trên, em đã quyết định xây dựng một mô hình MVC cho ứng dụng của mình, sử dụng framework Flutter [1] đơn giản hóa quá trình phát triển ứng dụng đa nền tảng, cung cấp bộ công cụ mạnh mẽ, linh hoạt giúp xây dựng ứng dụng với giao diện đẹp, mượt mà và hiệu suất cao.

Về phía Web Admin dành cho quản trị viên, em sử dụng thư viện ReactJs [2] để xây dựng phần giao diện và thực hiện các thao tác gửi và nhận dữ liệu với server backend sử dụng framework Spring Boot [3].

Cơ sở dữ liệu của ứng dụng được lưu trữ trên dịch vụ cơ sở dữ liệu trực tuyến Firebase [4], đồng bộ thời gian thực, dễ dàng tích hợp cho cả web và mobile, xác thực người dùng mạnh mẽ, cấu trúc NoSQL linh hoạt và có khả năng bảo mật, phân

quyền tốt.

Hệ thống thông báo em sử dụng Firebase Cloud Messaging [5], được thiết kế đặc biệt để hoạt động hiệu quả trên cả Android và IOS, có thể gửi thông báo cá nhân hóa theo từng người dùng hoặc nhóm người dùng, gửi theo thời gian thực.

Về thuật toán gợi ý, em sử dụng mô hình LightFM [6] vì kết hợp được cả Collaborative Filtering và Content-Based Filtering, hơn nữa giúp giải quyết hiệu quả vấn đề "cold-start".

Về xây dựng chatbot, em có sử dụng framework của RASA [7] hỗ trợ xây dựng chatbot nhanh, dễ dàng mở rộng và tùy chỉnh, linh hoạt xử lý các logic khác nhau.

Cuối cùng, em có tích hợp API của một số bên thứ ba:

- API của GHN để tự động tạo đơn, lấy thông tin giao hàng
- API của Stripe để tích hợp thanh toán điện tử

1.4 BỘ CỤC ĐỒ ÁN

Phần còn lại của báo cáo đồ án tốt nghiệp này được tổ chức như sau.

Chương 2 trình bày về phân tích yêu cầu của hệ thống thương mại điện tử trên mobile. Ở đây em sẽ sử dụng biểu đồ use case và đặc tả use case để miêu tả chi tiết các quy trình sử dụng hệ thống.

Trong chương 3, em sẽ trình bày về các công nghệ được sử dụng trong hệ thống. Trong chương này, em sẽ giới thiệu các công nghệ một cách cụ thể hơn và lý do chúng được sử dụng.

Chương 4 trình bày về kết quả thực nghiệm của hệ thống. Trong chương này, em sẽ trình bày về phân tích kiến trúc của hệ thống, phân tích chi tiết các thành phần của hệ thống như giao diện, lớp và cơ sở dữ liệu. Cuối cùng, em sẽ trình bày các công cụ và thư viện mà em đã sử dụng để xây dựng hệ thống của mình.

Chương 5 trình bày về giải pháp và đóng góp của em trong đồ án. Trong chương này, em sẽ trình bày về những khó khăn mà em đã gặp phải trong quá trình thực hiện đồ án và cách mà em đã giải quyết chúng.

Chương 6 đưa ra kết luận về ưu điểm, hạn chế của hệ thống so với những hệ thống tương tự đã có trên thị trường hiện nay và kiến thức kỹ năng mà em đã đạt được trong quá trình làm đồ án.

Vậy là trong chương 1, em đã trình bày lý do chọn đề tài, cơ sở xây dựng đồ án, mục tiêu đề tài và cấu trúc của đồ án. Sau đây, em sẽ trình bày nội dung của Chương 2 - Phân tích yêu cầu.

CHƯƠNG 2. PHÂN TÍCH YÊU CẦU

Trong chương này, em sẽ thực hiện khảo sát các hệ thống thương mại điện tử đã có trên thị trường, tìm ra ưu điểm và hạn chế còn tồn đọng của các hệ thống đã có, từ đó phân tích và đưa ra các chức năng chính và các yêu cầu phi chức năng của hệ thống

2.1 Khảo sát hiện trạng

2.1.1 Khảo sát hệ thống hiện có

Hai ứng dụng thương mại điện tử nổi bật trên nền tảng di động hiện nay là **Shopee** và **TikTok Shop**. Đây là những nền tảng lớn, có lượng người dùng đông đảo và nhiều chức năng phong phú, là cơ sở quan trọng để so sánh, phân tích và đề xuất cho hệ thống cần phát triển.

Shopee

• Ưu điểm:

- Giao diện trực quan, dễ sử dụng cho cả người bán và người mua.
- Hệ thống phân loại sản phẩm rõ ràng, nhiều bộ lọc thông minh.
- Tích hợp hệ thống thanh toán nội bộ (ShopeePay), ví điện tử và liên kết ngân hàng.
- Chức năng Flash Sale, Voucher hấp dẫn.
- Tính năng livestream bán hàng hỗ trợ người bán tiếp cận người dùng tốt hơn.
- Chính sách bảo vệ người mua, hoàn tiền rõ ràng.

• Nhược điểm:

- Giao diện đôi khi quá nhiều thông tin gây rối mắt.
- Tốc độ tải chậm ở một số thiết bị cấu hình yếu.

TikTok Shop

• Ưu điểm:

- Tích hợp trực tiếp với video ngắn giúp tăng trải nghiệm mua hàng giải trí.
- Khả năng tiếp cận sản phẩm thông qua KOL/influencer hiệu quả.
- Hệ thống gợi ý sản phẩm dựa trên AI theo hành vi người dùng.
- Liên kết bán hàng – video – thanh toán mượt mà trên cùng một nền tảng.

- **Nhược điểm:**

- Giao diện quản lý đơn hàng và sản phẩm của người bán còn thiếu chuyên nghiệp.
- Hệ thống phân loại và tìm kiếm sản phẩm chưa tối ưu.

2.1.2 Các chức năng cần có cho hệ thống

Dựa trên phân tích trên, hệ thống thương mại điện tử cần xây dựng nên có các chức năng cốt lõi sau:

- Đăng ký/đăng nhập, quản lý, cài đặt tài khoản
- Tìm kiếm, lọc, xem chi tiết sản phẩm.
- Giả hàng thanh toán đa phương thức (ví điện tử, ngân hàng, COD).
- Quản lý đơn hàng, trạng thái vận chuyển.
- Đánh giá, bình luận sản phẩm.
- Tạo và quản lý voucher, chương trình khuyến mãi.
- Quản trị viên quản lý người dùng, đơn hàng, sản phẩm.
- Tích hợp bên thứ ba: API thanh toán (Stripe, Momo), API giao hàng (GHN, GHTK).

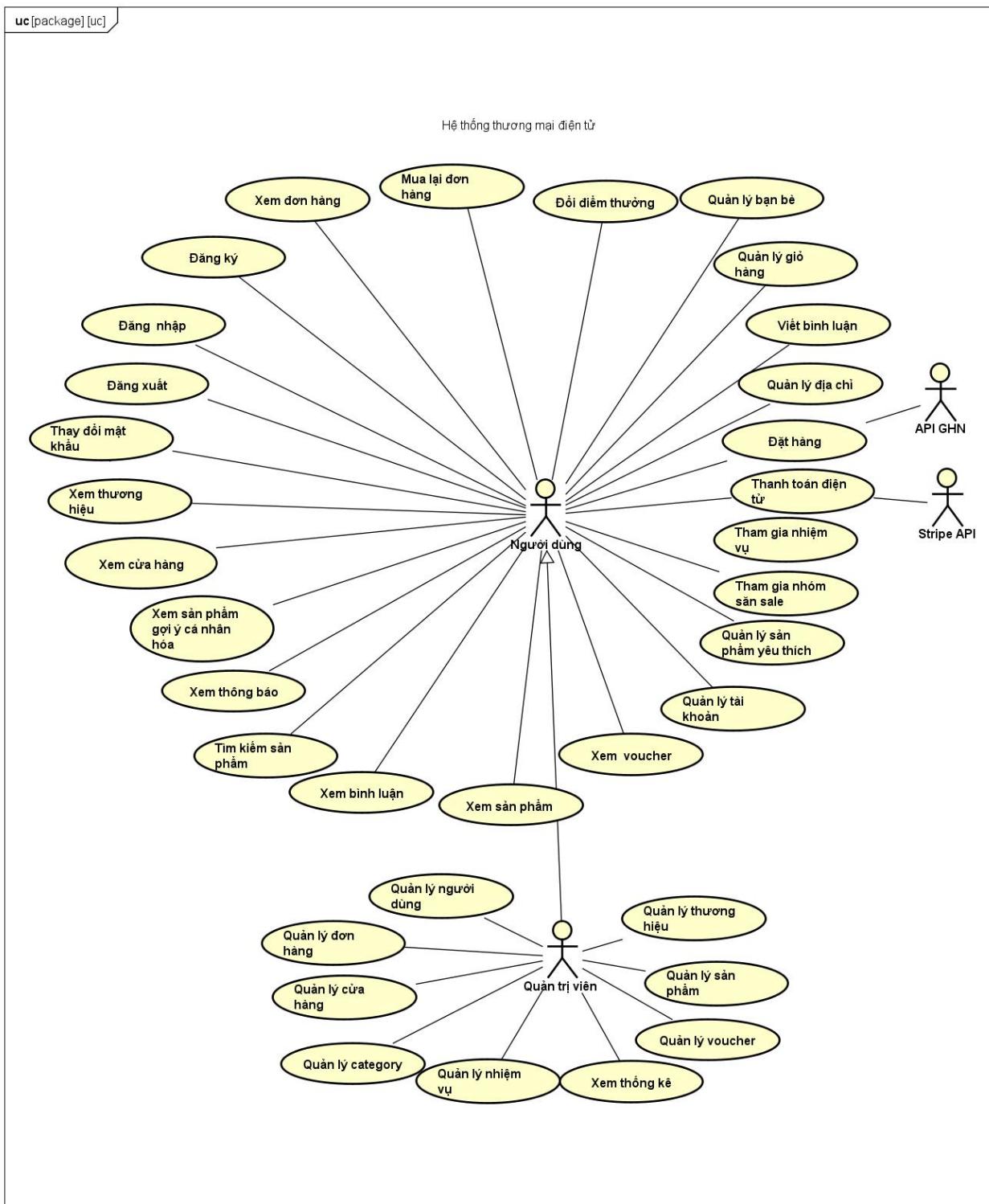
2.1.3 Tính năng đề xuất

Ngoài các tính năng chính cốt lõi trên, hệ thống của đồ án sẽ có thêm một số **tính năng** sau:

- **Gợi ý sản phẩm cá nhân hóa:** Sử dụng mô hình học máy để đưa ra các gợi ý cá nhân hóa cho mỗi người dùng
- **Chatbot tự động hóa quy trình mua sắm:** Chatbot tự động hóa quy trình mua sắm khi sẽ gợi ý cho người dùng những hành động người dùng chỉ cần nhập vài câu lệnh đơn giản hoặc click quick button là có thể tìm kiếm, xem sản phẩm, thêm vào giỏ hàng, xem giỏ hàng hay đặt hàng ngay lập tức, vô cùng nhanh chóng và tiện lợi.
- **Tạo nhóm săn sale:** Cho phép người dùng có thể kết bạn, tạo nhóm và mời bạn bè vào nhóm để cùng nhau săn sale và những voucher hấp dẫn của hệ thống
- **Tham gia các nhiệm vụ:** Hệ thống tạo các nhiệm vụ có thời gian hoàn thành đếm ngược và độ khó khác nhau khuyến khích người dùng tham gia trải nghiệm tương tác ứng dụng nhiều hơn để nhận các phần quà hấp dẫn.

2.2 Tổng quan chức năng

2.2.1 Biểu đồ use case tổng quát



Hình 2.1: Biểu đồ use case tổng quan

Mô tả tác nhân: Hệ thống gồm 4 tác nhân là: Người dùng, Quản trị viên, API thanh toán (Stripe) và API giao hàng (GHN)

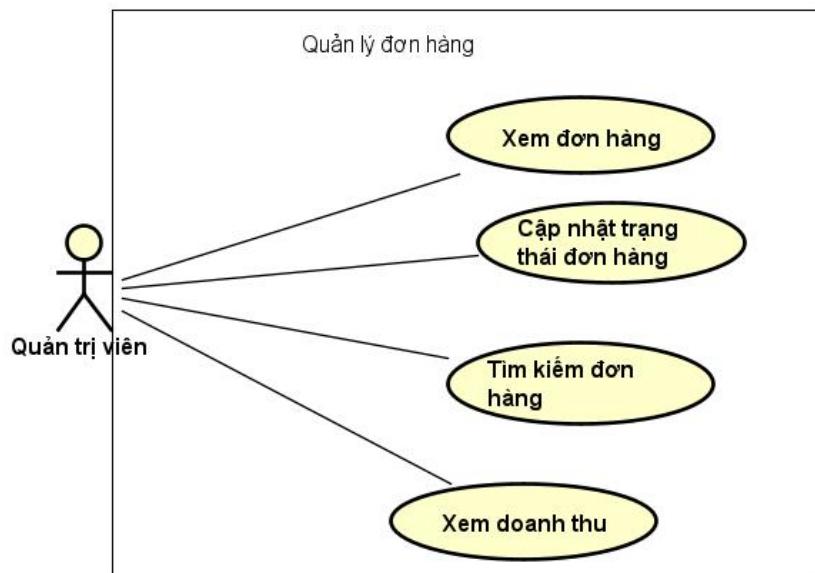
STT	Tên tác nhân	Mô tả
1	Người dùng	Là khách hàng sau khi đã đăng ký tài khoản và đăng nhập thành công. Người dùng có thể trải nghiệm đầy đủ các tính năng hấp dẫn của hệ thống thương mại điện tử bao gồm: xem và tìm kiếm sản phẩm, xem sản phẩm được cá nhân hóa, xem cửa hàng, category, thương hiệu nổi bật hiện có trên sàn, thêm vào giỏ hàng, đặt hàng, thanh toán, nhận thông báo trong hệ thống, xem và viết bình luận về sản phẩm, thu thập và sử dụng voucher, tham gia nhiệm vụ để tích điểm thưởng và đổi điểm thưởng lấy quà hấp dẫn. Ngoài ra người dùng có thể xem và theo dõi đơn hàng, mua lại đơn hàng, quản lý bạn bè, quản lý địa chỉ, quản lý cài đặt tài khoản, quản lý sản phẩm yêu thích. Và đặc biệt là tham gia hoạt động tạo nhóm săn sale cùng bạn bè vô cùng hấp dẫn và mang lại sự thích thú và tối ưu trong quá trình mua sắm.
2	Quản trị viên hệ thống	Có thể đăng nhập vào web Admin để thực hiện các công việc quản lý, xem thống kê sản phẩm, đơn hàng, người dùng, shops, thương hiệu, danh mục, voucher, nhiệm vụ trên hệ thống.
3	API thanh toán (Stripe)	Cung cấp chức năng thanh toán điện tử cho người dùng. Hệ thống sẽ gọi API này để xử lý thanh toán đơn hàng một cách an toàn, nhanh chóng và bảo mật.
4	API giao hàng (GHN)	Cung cấp dịch vụ giao hàng. Hệ thống gọi API này để tự động tạo đơn vận chuyển và lấy thông tin vận chuyển liên quan đến đơn hàng.

Bảng 2.1: Mô tả các tác nhân trong hệ thống

Hình 2.1 trình bày biểu đồ ca sử dụng tổng quan của hệ thống đối với các tác nhân đã được mô tả vắn tắt các ca sử dụng ở Bảng 2.1

2.2.2 Biểu đồ use case phân rã "Quản lý đơn hàng"

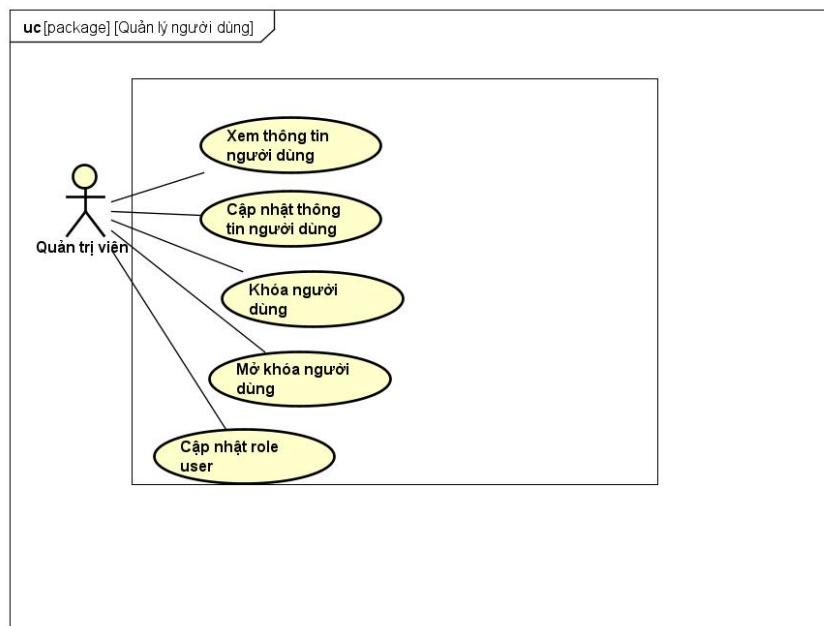
Hình 2.2 mô tả phân rã của use case "Quản lý đơn hàng". Với use case này, tác nhân là quản trị viên của hệ thống, có thể xem toàn bộ các đơn hàng của toàn bộ người dùng trong hệ thống, cập nhật trạng thái của đơn hàng, tìm kiếm và lọc đơn hàng, xem thống kê tổng doanh thu của hệ thống.



Hình 2.2: Biểu đồ use case phân rã "Quản lý đơn hàng"

2.2.3 Biểu đồ use case phân rã "Quản lý người dùng"

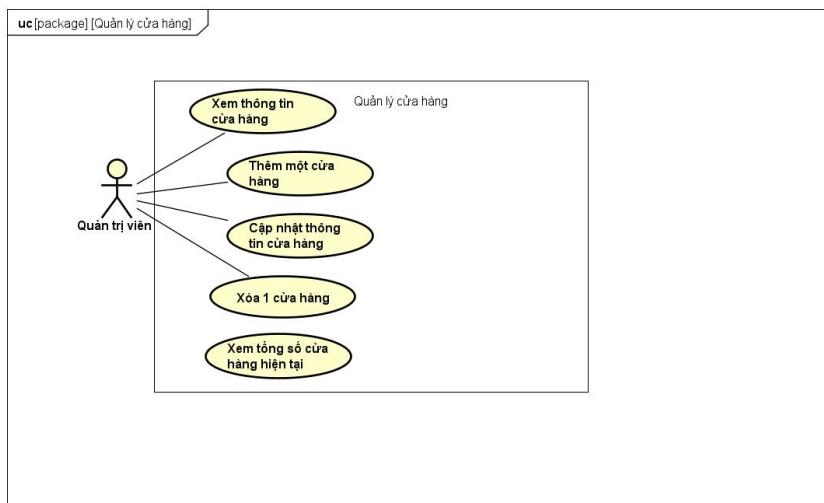
Hình 2.3 mô tả phân rã của use case "Quản lý người dùng". Với use case này, tác nhân là quản trị viên của hệ thống, có thể xem toàn bộ thông tin của tất cả người dùng trong hệ thống, cập nhật thông tin người dùng, cập nhật role của họ, tìm kiếm người dùng. Đặc biệt quản trị viên có thể khóa người dùng và mở khóa người dùng nếu phát hiện các hành vi gian lận mà không cần xóa tài khoản đó, có thể dễ dàng mở khóa lại người dùng sau khi hợp tác hoặc khi vấn đề đã được giải quyết.



Hình 2.3: Biểu đồ use case phân rã "Quản lý người dùng"

2.2.4 Biểu đồ use case phân rã "Quản lý cửa hàng"

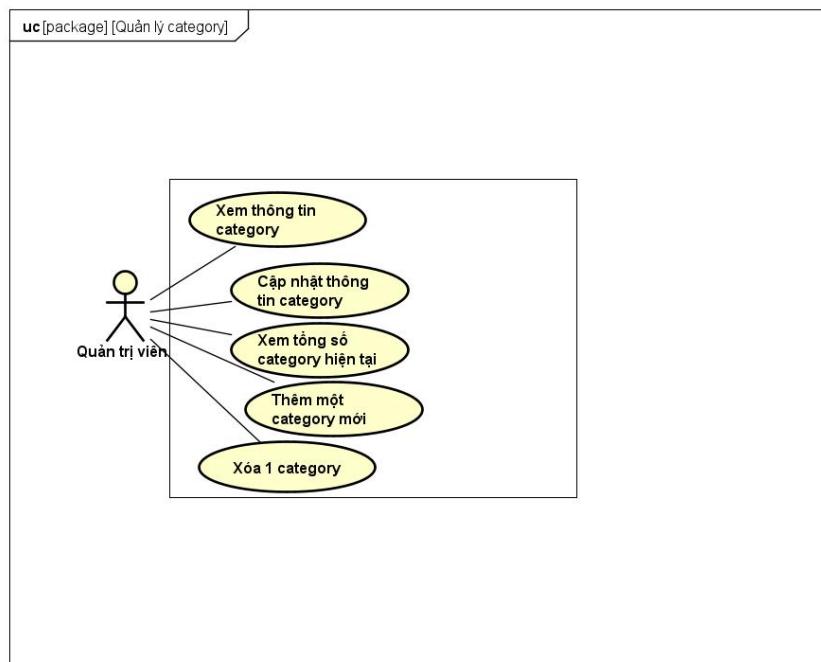
Hình 2.4 mô tả phân rã của use case "Quản lý cửa hàng". Với use case này, tác nhân là quản trị viên của hệ thống, có thể xem toàn bộ thông tin của tất cả các cửa hàng đã đăng ký trong hệ thống, tìm kiếm cửa hàng, thêm 1 cửa hàng mới, cập nhật thông tin cửa hàng, xóa cửa hàng và xem tổng số cửa hàng trong hệ thống.



Hình 2.4: Biểu đồ use case phân rã "Quản lý cửa hàng"

2.2.5 Biểu đồ use case phân rã "Quản lý category"

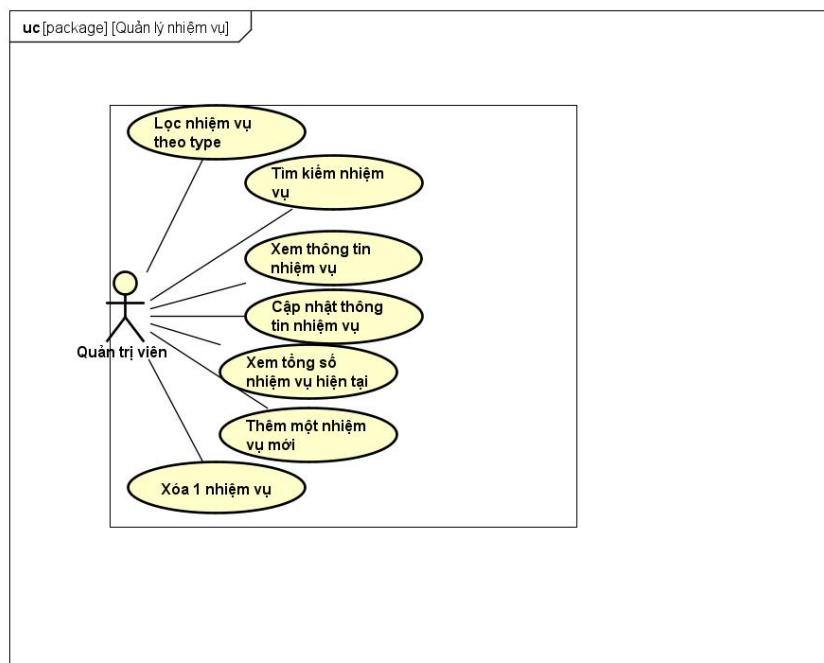
Hình 2.5 mô tả phân rã của use case "Quản lý category". Với use case này, tác nhân là quản trị viên của hệ thống, có thể xem toàn bộ thông tin của tất cả các category đang có trong hệ thống, tìm kiếm category, thêm 1 category mới, cập nhật thông tin category đã có, xóa category và xem tổng số category trong hệ thống.



Hình 2.5: Biểu đồ use case phân rã "Quản lý category"

2.2.6 Biểu đồ use case phân rã "Quản lý nhiệm vụ"

Hình 2.6 mô tả phân rã của use case "Quản lý nhiệm vụ". Với use case này, tác nhân là quản trị viên của hệ thống, có thể xem toàn bộ thông tin của tất cả các nhiệm vụ đang có trong hệ thống, tìm kiếm nhiệm vụ, lọc nhiệm vụ theo 1 type cụ thể, thêm 1 nhiệm vụ mới, cập nhật thông tin nhiệm vụ đã có, xóa nhiệm vụ và xem tổng số nhiệm vụ đã được tạo trong hệ thống.

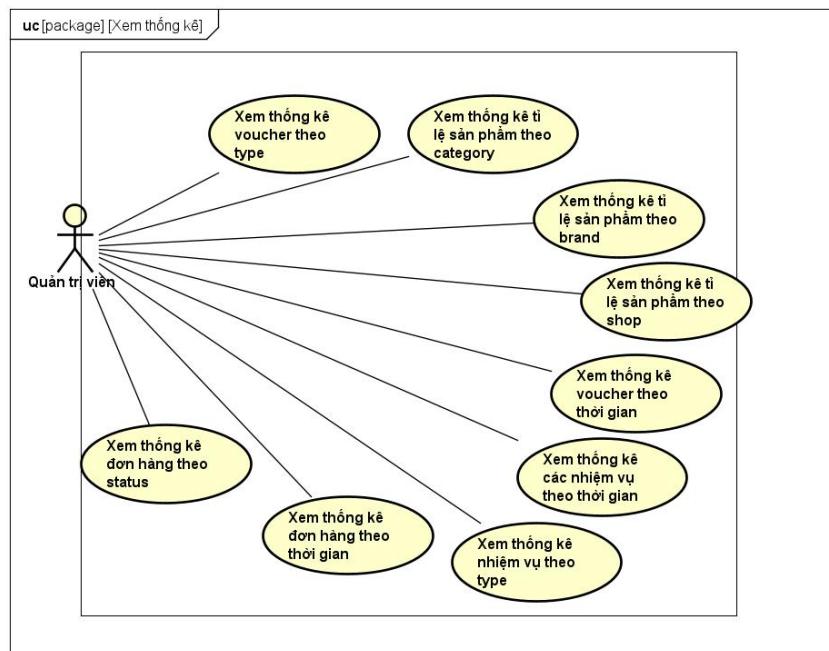


Hình 2.6: Biểu đồ use case phân rã "Quản lý nhiệm vụ"

2.2.7 Biểu đồ use case phân rã "Xem thống kê"

Hình 2.7 mô tả phân rã của use case "Xem thống kê". Với use case này, tác nhân là quản trị viên của hệ thống, có thể xem tất cả các biểu đồ thống kê trong hệ thống. Hệ thống có 2 loại biểu đồ chính là biểu đồ tròn: gồm các biểu đồ thống kê đơn hàng theo status, thống kê nhiệm vụ theo type, thống kê voucher theo type, thống kê tỉ lệ sản phẩm theo category hoặc brand hoặc shop. Loại thứ hai là biểu đồ đường, diễn tả sự biến động theo thời gian gồm: thống kê đơn hàng, thống kê nhiệm vụ, thống kê voucher theo thời gian.

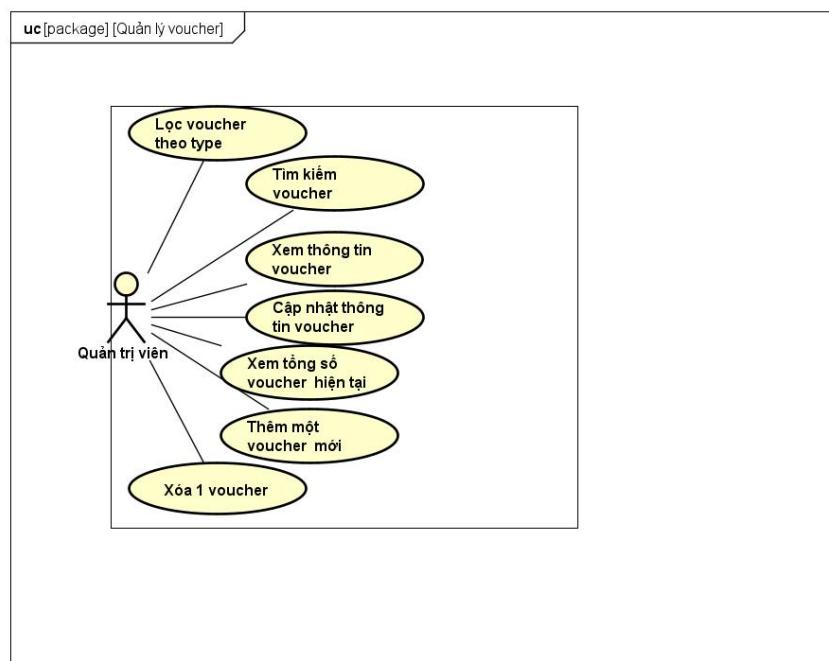
CHƯƠNG 2. PHÂN TÍCH YÊU CẦU



Hình 2.7: Biểu đồ use case phân rã "Xem thống kê"

2.2.8 Biểu đồ use case phân rã "Quản lý voucher"

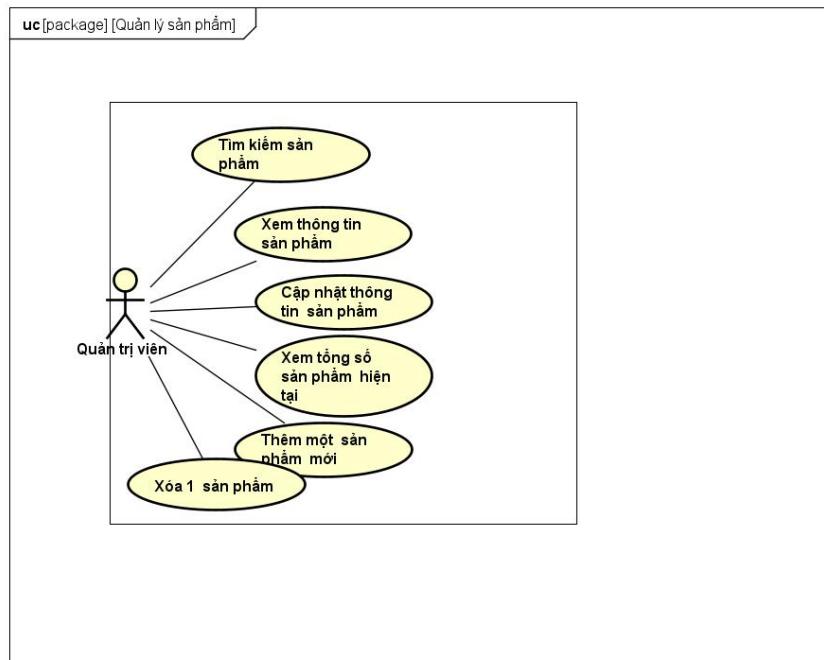
Hình 2.8 mô tả phân rã của use case "Quản lý voucher". Với use case này, tác nhân là quản trị viên của hệ thống, có thể xem toàn bộ thông tin của tất cả các voucher đang có trong hệ thống, tìm kiếm voucher, lọc voucher theo 1 type cụ thể, thêm 1 voucher mới, cập nhật thông tin voucher đã có, xóa voucher và xem tổng số voucher đã được tạo trong hệ thống.



Hình 2.8: Biểu đồ use case phân rã "Quản lý voucher"

2.2.9 Biểu đồ use case phân rã "Quản lý sản phẩm"

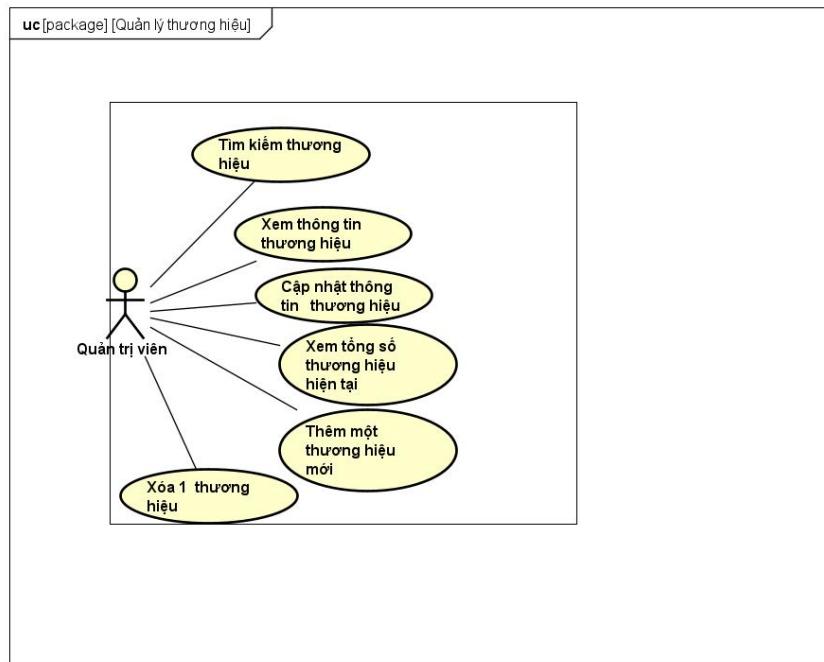
Hình 2.9 mô tả phân rã của use case "Quản lý sản phẩm". Với use case này, tác nhân là quản trị viên của hệ thống, có thể xem toàn bộ thông tin của tất cả các sản phẩm đang có trong hệ thống, tìm kiếm sản phẩm, thêm 1 sản phẩm mới, cập nhật thông tin sản phẩm đã có, xóa sản phẩm và xem tổng số sản phẩm đang có trong hệ thống.



Hình 2.9: Biểu đồ use case phân rã "Quản lý sản phẩm"

2.2.10 Biểu đồ use case phân rã "Quản lý thương hiệu"

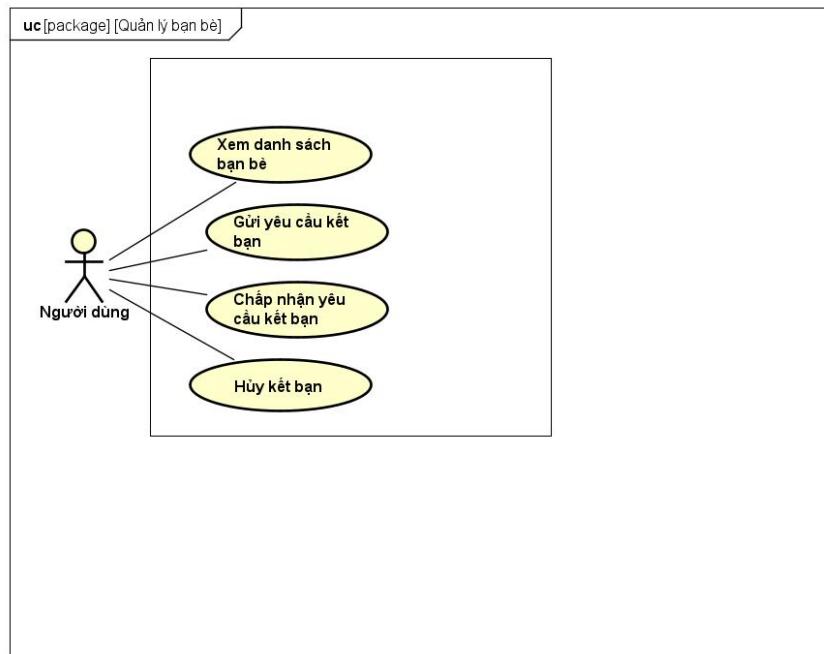
Hình 2.10 mô tả phân rã của use case "Quản lý thương hiệu". Với use case này, tác nhân là quản trị viên của hệ thống, có thể xem toàn bộ thông tin của tất cả các thương hiệu đang có trong hệ thống, tìm kiếm thương hiệu, thêm 1 thương hiệu mới, cập nhật thông tin thương hiệu đã có, xóa thương hiệu và xem tổng số thương hiệu đang có trong hệ thống.



Hình 2.10: Biểu đồ use case phân rã "Quản lý thương hiệu"

2.2.11 Biểu đồ use case phân rã "Quản lý bạn bè"

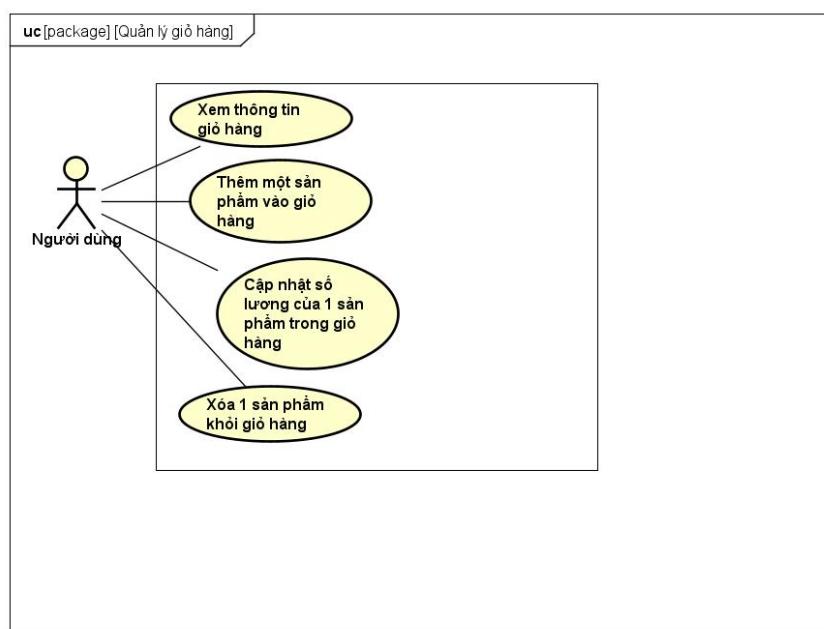
Hình 2.11 mô tả phân rã của use case "Quản lý bạn bè". Với use case này, tác nhân là người dùng hệ thống, có thể xem danh sách bạn bè hiện tại, gửi yêu cầu kết bạn tới những người khác cũng đang hoạt động trong hệ thống, chấp nhận yêu cầu kết bạn từ người khác gửi đến và hủy kết bạn với một người nào đó.



Hình 2.11: Biểu đồ use case phân rã "Quản lý bạn bè"

2.2.12 Biểu đồ use case phân rã "Quản lý giỏ hàng"

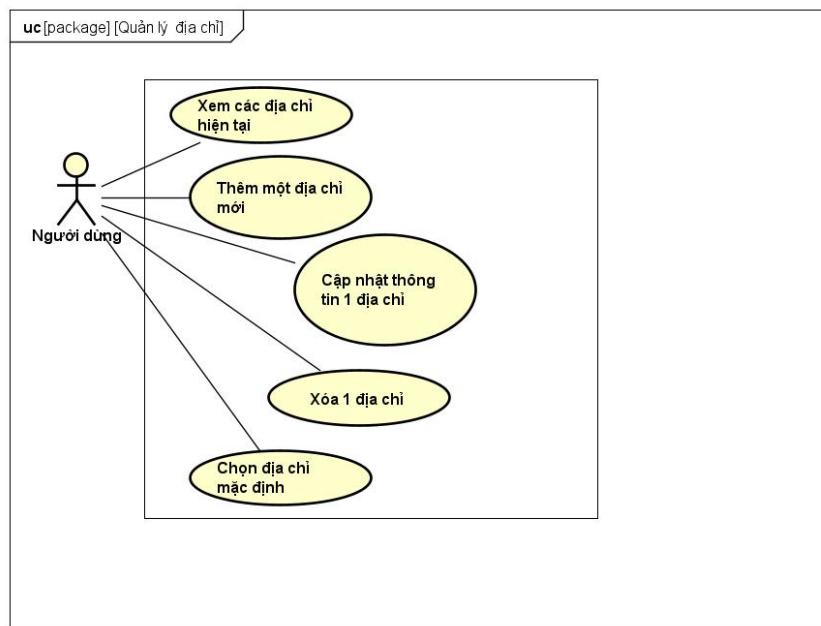
Hình 2.12 mô tả phân rã của use case "Quản lý giỏ hàng". Với use case này, tác nhân là người dùng hệ thống, có thể xem toàn bộ thông tin của tất cả các sản phẩm đang có trong giỏ hàng, thêm 1 sản phẩm mới vào giỏ hàng, cập nhật số lượng của các sản phẩm trong giỏ hàng, xóa sản phẩm khỏi giỏ hàng và xem giá trị của giỏ hàng hiện tại.



Hình 2.12: Biểu đồ use case phân rã "Quản lý giỏ hàng"

2.2.13 Biểu đồ use case phân rã "Quản lý địa chỉ"

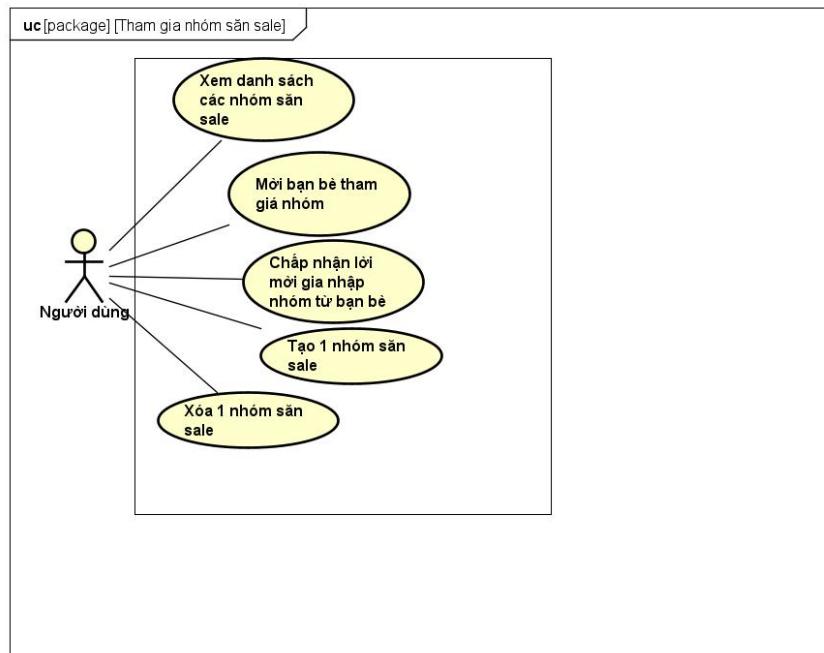
Hình 2.13 mô tả phân rã của use case "Quản lý địa chỉ". Với use case này, tác nhân là người dùng hệ thống, có thể xem toàn bộ thông tin của tất cả các địa chỉ mà họ đã tạo, thêm 1 địa chỉ mới, cập nhật thông tin của 1 địa chỉ cũ, xóa 1 địa chỉ và chọn một địa chỉ làm mặc định khi thanh toán đơn hàng



Hình 2.13: Biểu đồ use case phân rã "Quản lý địa chỉ"

2.2.14 Biểu đồ use case phân rã "Tham gia nhóm săn sale"

Hình 2.14 mô tả phân rã của use case "Tham gia nhóm săn sale". Với use case này, tác nhân là người dùng hệ thống, có thể xem toàn bộ thông tin của các nhóm săn sale mà họ đã tạo hoặc họ đã tham gia từ lời mời của bạn bè, tạo 1 nhóm săn sale mới, mời bạn bè tham gia nhóm săn sale mà mình vừa tạo, chấp nhận lời mời để tham gia nhóm săn sale cùng bạn bè, và xóa nhóm săn sale mà mình đã tạo.

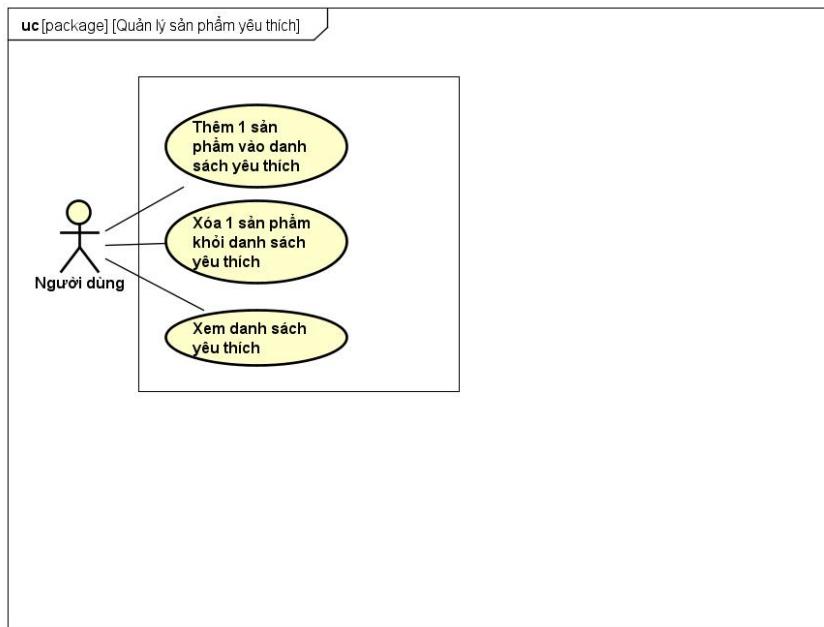


Hình 2.14: Biểu đồ use case phân rã "Tham gia nhóm săn sale"

2.2.15 Biểu đồ use case phân rã "Quản lý sản phẩm yêu thích"

Hình 2.15 mô tả phân rã của use case "Quản lý sản phẩm yêu thích". Với use case này, tác nhân là người dùng hệ thống, có thể xem toàn bộ thông tin của các sản phẩm mà họ đã thêm vào danh sách yêu thích, xóa một sản phẩm khỏi danh sách yêu thích hiện tại và thêm 1 sản phẩm mới vào danh sách yêu thích.

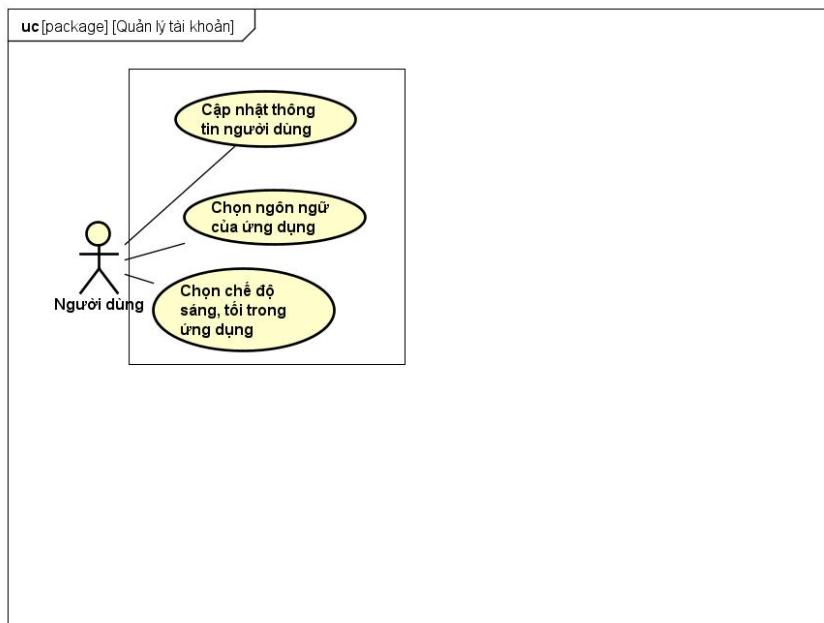
CHƯƠNG 2. PHÂN TÍCH YÊU CẦU



Hình 2.15: Biểu đồ use case phân rã "Quản lý sản phẩm yêu thích"

2.2.16 Biểu đồ use case phân rã "Quản lý tài khoản"

Hình 2.16 mô tả phân rã của use case "Quản lý tài khoản". Với use case này, tác nhân là người dùng hệ thống, có thể xem thông tin cá nhân của bản thân, bổ sung hoặc cập nhật thông tin cá nhân, chọn ngôn ngữ sẽ hiển thị trong ứng dụng: tiếng Anh hoặc tiếng Việt, chọn chế độ hiển thị trong ứng dụng là chế độ sáng(light mode) hoặc chế độ tối (dark mode) thì giao diện của ứng dụng sẽ cập nhật tương ứng.



Hình 2.16: Biểu đồ use case phân rã "Quản lý tài khoản"

2.3 Đặc tả chức năng

2.3.1 Đặc tả use case "Xem sản phẩm"

Mã Use case	1	Tên Use case	Xem sản phẩm
Tác nhân		Người dùng	
Tiền điều kiện		Người dùng đã đăng nhập vào ứng dụng	
Luồng sự kiện chính	STT	Thực hiện bởi	Hành động
	1	Người dùng	Truy cập danh sách sản phẩm
	2	Hệ thống	Lấy dữ liệu danh sách sản phẩm từ server
	3	Hệ thống	Hiển thị danh sách sản phẩm
	4	Người dùng	Nhấn vào sản phẩm để xem chi tiết
	5	Hệ thống	Truy vấn cơ sở dữ liệu để lấy ra thông tin chi tiết của sản phẩm thông qua productId
	6	Hệ thống	Chuyển hướng đến màn hình thông tin chi tiết của sản phẩm
	7	Hệ thống	Hiển thị thông tin chi tiết của product
Luồng sự kiện thay thế	2a	Hệ thống	Không thể tải danh sách sản phẩm, hiện thông báo lỗi
Xử lý ngoại lệ	5a	Hệ thống	Không thể hiển thị chi tiết sản phẩm, hiện thông báo lỗi
Hậu điều kiện	Không		

Bảng 2.2: Bảng đặc tả Use case "Xem sản phẩm"

CHƯƠNG 2. PHÂN TÍCH YÊU CẦU

* Dữ liệu đầu vào:

STT	Trường dữ liệu	Bắt buộc	Mô tả	Định dạng hiển thị	Ví dụ
1	title	Có	Tên sản phẩm	Chuỗi	Máy chạy bộ
2	description	Không	Mô tả chi tiết về sản phẩm	String	Công suất: 2.0HP
3	detail	Không	Mô tả một số thông tin khác như nơi sản xuất, năm sản xuất	String	Sản xuất tại Trung Quốc
4	price	Có	Giá của sản phẩm đó	Number	28,000,000
5	brandName	Có	Thương hiệu của sản phẩm	String	Samsung
6	shopName	Không	Tên cửa hàng đang bán sản phẩm đó	String	LoveShop
7	images	Không	Danh sách hình ảnh của sản phẩm	Danh sách	https://firebasestorage.googleapis.com/v0/b/da-gr1.appspot.com/o/Banners%2Forder?alt=media&token=..

Bảng 2.3: Dữ liệu đầu vào cho Use case "Xem sản phẩm"

2.3.2 Đặc tả use case "Thêm sản phẩm vào giỏ hàng"

Mã Use case	2	Tên Use case	Thêm sản phẩm vào giỏ hàng
Tác nhân		Người dùng	
Tiền điều kiện		Người dùng đang xem trang chi tiết sản phẩm hoặc ở màn hình danh sách sản phẩm	

Luồng sự kiện chính	STT	Thực hiện bởi	Hành động
	1	Người dùng	Chọn số lượng sản phẩm muốn mua (nếu có tùy chọn)
	2	Người dùng	Nhấn nút "Thêm vào giỏ hàng"
	3	Hệ thống	Kiểm tra tình trạng sản phẩm (còn hàng hay hết hàng)
	4	Hệ thống	Nếu còn hàng, thêm sản phẩm vào giỏ hàng
	5	Hệ thống	Hiển thị thông báo "Đã thêm sản phẩm vào giỏ hàng"
Luồng sự kiện thay thế	4a	Hệ thống	Sản phẩm đã hết hàng, hiển thị thông báo "Sản phẩm hiện đã hết hàng"
Xử lý ngoại lệ	4b	Hệ thống	Lỗi khi thêm sản phẩm vào giỏ hàng, hiển thị thông báo lỗi
Hậu điều kiện	Sản phẩm được thêm vào giỏ hàng của Người mua hàng. Hệ thống cập nhật số lượng sản phẩm trong giỏ hàng.		

Bảng 2.4: Bảng đặc tả Use case "Thêm sản phẩm vào giỏ hàng"

* Dữ liệu đầu vào:

STT	Trường dữ liệu	Bắt buộc	Mô tả	Định dạng hiển thị	Ví dụ
1	userId	Có	ID của người mua hàng thực hiện thao tác thêm vào giỏ	String	user123
2	productId	Có	ID của sản phẩm cần được thêm vào giỏ hàng	String	prod789
3	quantity	Có	Số lượng sản phẩm mà người dùng muốn mua	Number	2
4	addedAt	Không	Thời điểm sản phẩm được thêm vào giỏ hàng	DateTime	2025-05-23T14:32:00
5	selectedOptions	Không	Tùy chọn của sản phẩm như kích cỡ, màu sắc (nếu có)	JSON	{"size": "M", "color": "red"}

Bảng 2.5: Dữ liệu đầu vào cho Use case "Thêm sản phẩm vào giỏ hàng"

2.3.3 Đặc tả use case "Đặt hàng"

Mã Use case	3	Tên Use case	Đặt hàng
Tác nhân		Người dùng	
Tiền điều kiện		Người dùng đã đăng nhập vào hệ thống và giỏ hàng có ít nhất một sản phẩm	

Luồng sự kiện chính	STT	Thực hiện bởi	Hành động
	1	Người dùng	Xem giỏ hàng và click nút đặt hàng
	2	Hệ thống	Chuyển đến màn hình đặt hàng
	3	Hệ thống	Hiển thị tóm tắt thông tin của các sản phẩm trong đơn hàng hiển thị giá trị đơn hàng, tự động tính phí ship, VAT và tổng giá trị sau cùng
	4	Người dùng	Chọn nút áp dụng voucher
	5	Hệ thống	Tính toán và đưa ra danh sách các voucher mà người đó đủ điều kiện áp dụng
	6	Người dùng	Chọn 1 loại voucher bất kì trong danh sách
	7	Hệ thống	Tính toán và cập nhật lại giá trị đơn hàng, thêm vào list voucher được áp dụng và hiển thị list lên giao diện
	8	Người dùng	Chọn phương thức thanh toán
	9	Người dùng	Chọn địa chỉ giao hàng
	10	Hệ thống	Hiển thị đầy đủ thông tin phương thức thanh toán, địa chỉ giao hàng lên giao diện
	11	Người dùng	Chọn nút thanh toán
	12	Hệ thống	Kiểm tra thông tin người dùng đã hợp lệ chưa chuyển sang màn hình thanh toán
Luồng sự kiện thay thế	5a	Hệ thống	Nếu không có voucher nào có thể áp dụng, hiển thị thông báo "Không thể áp dụng voucher"
	7a	Hệ thống	Nếu 1 voucher sau khi áp dụng, giá trị đơn hàng đã về 0 mà người đó vẫn tiếp tục áp dụng thì đưa ra cảnh báo "Không cần áp dụng thêm voucher"
	10a	Hệ thống	Nếu người dùng chưa chọn thông tin địa chỉ hoặc phương thức thanh toán, hiển thị thông báo "Bạn cần chọn đầy đủ thông tin"
	12a	Hệ thống	Nếu người dùng không chọn nút thanh toán, mà back về trang trước thì hệ thống cũng xóa các thông tin đã chọn và điều hướng về màn hình đó
Xử lý ngoại lệ	9a	Hệ thống	Nếu chưa thêm địa chỉ nào, hiển thị thông báo lỗi và yêu cầu người dùng thêm một địa chỉ mới
Hậu điều kiện			Không

Bảng 2.6: Bảng đặc tả Use case "Đặt hàng"

2.3.4 Đặc tả use case "Thanh toán điện tử"

Mã Use case	4	Tên Use case	Thanh toán điện tử
Tác nhân		Người dùng	
Tiền điều kiện		Người dùng đã đăng nhập vào hệ thống Người dùng đã đặt hàng thành công	
Luồng sự kiện chính	STT	Thực hiện bởi	Hành động
	1	Người dùng	Chọn nút thanh toán
	2	Hệ thống	Chuyển đến màn hình thanh toán
	3	Hệ thống	Gọi đến API thanh toán Stripe và hiển thị popup thanh toán
	4	Người dùng	Nhập đầy đủ các thông tin như số thẻ, thời hạn, CVC, địa chỉ
	5	Người dùng	Kiểm tra lại thông tin đã nhập, xem số tiền phải trả và bấm thanh toán ngay
	6	Hệ thống	Gọi API, tạo giao dịch mới và gửi thông tin giao dịch sang cho Stripe xử lý
	7	Hệ thống	Nhận phản hồi từ Stripe, hiển thị kết quả lên giao diện
	8	Hệ thống	Gửi thông báo đơn hàng đã thanh toán thành công và điều hướng về màn hình home
Luồng sự kiện thay thế	6a	Hệ thống	Nếu người dùng nhập thiếu thông tin, hiển thị cảnh báo và yêu cầu người dùng phải nhập đủ
	7a	Hệ thống	Nếu có lỗi không thực hiện được giao dịch thì hiện thông báo lỗi, ẩn popup và trở về màn hình trước đó
Xử lý ngoại lệ	3a	Hệ thống	Không gọi dc API Stripe, hiện thông báo lỗi
Hậu điều kiện			Không

Bảng 2.7: Bảng đặc tả Use case "Thanh toán điện tử"

* Dữ liệu đầu vào:

STT	Trường dữ liệu	Bắt buộc	Mô tả	Định dạng hiển thị	Ví dụ
1	Số thẻ	Có	Số thẻ ngân hàng liên kết	String	4242 4242 4242 4242
2	Hiệu lực	Có	Thời gian còn hiệu lực của thẻ	Date	12/27
3	CVC	Có	Mật mã bí mật của thẻ	Number	222
4	Địa chỉ	Có	Quốc gia hoặc khu vực sử dụng	String	Việt Nam

Bảng 2.8: Dữ liệu đầu vào cho Use case "Thanh toán điện tử"

2.3.5 Đặc tả use case "Tham gia nhiệm vụ"

CHƯƠNG 2. PHÂN TÍCH YÊU CẦU

Mã Use case	5	Tên Use case	Tham gia nhiệm vụ
Tác nhân		Người dùng	
Tiền điều kiện		Người dùng đã đăng nhập vào hệ thống	
Luồng sự kiện chính	STT	Thực hiện bởi	Hành động
	1	Người dùng	Chọn màn hình danh sách nhiệm vụ
	2	Hệ thống	Truy vấn server để lấy danh sách nhiệm vụ và trạng thái thực hiện nhiệm vụ của người dùng đó
	3	Người dùng	Bấm vào 1 nhiệm vụ để xem thông tin chi tiết nhiệm vụ
	4	Hệ thống	Lấy id của nhiệm vụ gửi lên server và lấy thông tin chi tiết của nhiệm vụ, điều hướng đến màn hình chi tiết
	5	Người dùng	Xem thông tin chi tiết gồm tên, trạng thái nhiệm vụ, điểm nhận được và deadline hoàn thành
	6	Người dùng	Bấm nút bắt đầu nhiệm vụ
	7	Hệ thống	Hiển thị đồng hồ đếm ngược của nhiệm vụ, cập nhật trạng thái của nhiệm vụ, lưu thông tin nhiệm vụ của người dùng lên server
	8	Hệ thống	Hệ thống tạo tracking để theo dõi việc thực hiện nhiệm vụ của người đó, khi người dùng hoàn thành 1 step thì giao diện cũng sẽ tự động cập nhật và lưu lên server
	9	Hệ thống	Hệ thống theo dõi và thấy nếu người dùng hoàn thành nhiệm vụ trước khi hết thời gian đếm ngược thì sẽ tự động gửi thông báo nhiệm vụ thành công và cộng điểm cho người dùng, lưu lên server
Luồng sự kiện thay thế	7a	Hệ thống	Nếu 1 nhiệm vụ đã bắt đầu rồi thì nó sẽ hiển thị thời gian còn lại
	9a	Hệ thống	Nếu hết thời gian mà nhiệm vụ chưa hoàn thành thì nó tự động cập nhật trạng thái, ẩn đồng hồ đếm ngược và lưu lên server
Xử lý ngoại lệ	2a	Hệ thống	Không lấy được dữ liệu, hiện thông báo lỗi
Hậu điều kiện		Không	

Bảng 2.9: Bảng đặc tả Use case Tham gia nhiệm vụ

2.3.6 Đặc tả use case "Mời bạn bè tham gia nhóm"

Mã Use case	6	Tên Use case	Mời bạn bè tham gia nhóm
Tác nhân		Người dùng	
Tiền điều kiện		Người dùng đã đăng nhập vào hệ thống Người dùng đã tạo 1 nhóm săn sale	
Luồng sự kiện chính	STT	Thực hiện bởi	Hành động
	1	Người dùng	Chọn nút mời bạn bè tham gia nhóm
	2	Hệ thống	Điều hướng đến màn hình mời, hiển thị danh sách bạn bè kèm nút invite
	3	Người dùng	Chọn nút invite và mời một người bạn vào nhóm
	4	Hệ thống	Gửi request đến người bạn đó thông báo đã gửi yêu cầu thành công
	5	Hệ thống	Khi có phản hồi từ bạn bè, gửi thông báo đã chấp nhận hoặc đã từ chối lời mời
	6	Hệ thống	Cập nhật lại trạng thái, số thành viên của nhóm hiện tại
	7	Hệ thống	Nếu đã đủ thành viên, gửi thông báo săn sale thành công đến tất cả các thành viên trong nhóm, tự động thêm voucher săn sale vào kho của mỗi thành viên
Luồng sự kiện thay thế	2a	Hệ thống	Nếu chưa có người bạn nào sẽ thông báo và có nút để điều hướng đến màn hình kết bạn
Xử lý ngoại lệ	4a	Hệ thống	Không gửi được yêu cầu, hiện thông báo lỗi
Hậu điều kiện			Không

Bảng 2.10: Bảng đặc tả Use case "Mời bạn bè tham gia nhóm"

2.3.7 Đặc tả use case "Viết bình luận"

Mã Use case	7	Tên Use case	Viết bình luận
Tác nhân		Người dùng	
Tiền điều kiện		Người dùng đăng nhập vào hệ thống	
Luồng sự kiện chính	STT	Thực hiện bởi	Hành động
	1	Người dùng	Chọn nút viết bình luận
	2	Hệ thống	Điều hướng đến màn hình review 1 sản phẩm
	3	Hệ thống	Hiển thị form review sản phẩm gồm tên và ảnh của sản phẩm
	4	Người dùng	Chọn số sao vote cho sản phẩm
	5	Người dùng	Nhập bình luận về sản phẩm
	6	Hệ thống	Kiểm soát độ dài nội dung bình luận, hiển thị thông tin người dùng đã nhập
	7	Người dùng	Chọn nút upload ảnh hoặc video
	8	Hệ thống	Hiển thị popup để người dùng lựa chọn up ảnh hay up video up từ thư viện hay chụp mới
	9	Hệ thống	Điều hướng đến thư viện ảnh hoặc camera, lưu data người dùng chọn và quay lại form nhập bình luận thông báo đã up ảnh, video thành công
	10	Người dùng	Chọn chế độ up bình luận là công khai hay ẩn danh
	11	Hệ thống	Kiểm tra sự hợp lệ của form gửi dữ liệu và lưu lên server và quay lại màn hình xem bình luận về sản phẩm
Luồng sự kiện thay thế	6a	Hệ thống	Nếu người dùng nhập bình luận quá dài sẽ thông báo và không cho nhập thêm
Xử lý ngoại lệ	11a	Hệ thống	Không lưu được dữ liệu, hiện thông báo lỗi
Hậu điều kiện		Không	

Bảng 2.11: Bảng đặc tả Use case "Viết bình luận"

* Dữ liệu đầu vào:

STT	Trường dữ liệu	Bắt buộc	Mô tả	Định dạng hiển thị	Ví dụ
1	rating	Không	Số sao đánh giá cho sản phẩm	Number	5
2	review	Không	Nội dung bình luận	String	Sản phẩm tuyệt lẩm
3	image	Không	Các ảnh người dùng upload	Img	https://url/img
4	video	Không	Các video người dùng upload	Video	https://www.youtube.com/watch?v=5qt3enCkcwo&list=RDMMSASAH-gigmQ&index=27

Bảng 2.12: Dữ liệu đầu vào cho Use case "Viết bình luận"

2.4 Yêu cầu phi chức năng

Ngoài các chức năng đã được liệt kê ở phần trước, hệ thống cũng cần phải thêm một số yêu cầu phi chức năng để đáp ứng hơn nữa nhu cầu sử dụng của người dùng:

2.4.1 Yêu cầu về giao diện và trải nghiệm người dùng

Giao diện của ứng dụng cần phải được thiết kế bắt mắt, bố cục phù hợp, dễ sử dụng, màu sắc hài hòa, dễ nhìn và các nút bấm cần đủ to để người dùng có thể nhấn một cách dễ dàng trên thiết bị di động. Ngoài ra, khi người dùng gặp lỗi khi nhập liệu hay gửi yêu cầu, sẽ luôn có các thông báo lỗi được hiển thị để giúp người dùng biết được nội dung lỗi và chỉnh sửa lại sao cho phù hợp.

2.4.2 Yêu cầu về tính dễ bảo trì

Ứng dụng cần phải có tính dễ bảo trì để dễ dàng chỉnh sửa các thành phần riêng biệt mà không làm ảnh hưởng đến cả hệ thống. Phần client và server được phân tách riêng biệt với nhau bằng các thư mục riêng biệt. Ngoài ra phần giao diện được phân chia thành các component giúp dễ dàng trong việc quản lý code và chỉnh sửa sau này.

2.4.3 Yêu cầu về bảo mật

Ứng dụng cần đảm bảo tính bảo mật cao để bảo vệ thông tin cá nhân của người dùng cũng như dữ liệu liên quan đến đơn hàng, sản phẩm và thanh toán. Firebase cung cấp nhiều cơ chế bảo mật để hỗ trợ điều này:

- Xác thực người dùng (Firebase Authentication [8]): Tất cả người dùng phải đăng nhập thông qua các phương thức xác thực an toàn như Email/Password, Google, hoặc OTP qua điện thoại. Sau khi đăng nhập, người dùng sẽ được gán một mã định danh duy nhất (UID), giúp xác định danh tính trong các truy vấn

dữ liệu.

- Quy tắc bảo mật cho Firestore [9] (Security Rules): Ứng dụng sử dụng Firestore Security Rules để kiểm soát quyền đọc/ghi dữ liệu theo UID. Chỉ người dùng hợp lệ mới có thể truy cập dữ liệu của chính họ, ví dụ: đơn hàng, giỏ hàng, địa chỉ giao hàng,...
- Realtime Database / Firestore [9] đã mã hóa dữ liệu: Firebase tự động mã hóa dữ liệu ở trạng thái lưu trữ (at rest) và khi truyền tải (in transit), đảm bảo an toàn khi giao tiếp qua mạng.

2.4.4 Yêu cầu về hiệu năng

Ứng dụng cần phải hoạt động mượt mà, ổn định ít giật lag, thời gian chờ cho mỗi tác vụ gửi request và nhận response giữa client và server phải đủ nhanh kể cả khi lượng dữ liệu truyền đi hoặc nhận về lớn.

Như vậy, ở trong chương 2 này em đã đưa ra các chức năng của hệ thống thông qua biểu đồ use case với 2 tác nhân chính là người dùng và quản trị viên. Ngoài ra, em cũng đã nêu ra một số yêu cầu phi chức năng cần có của hệ thống. Tiếp đến, em sẽ giới thiệu một số công nghệ được sử dụng để thực hiện các chức năng và các yêu cầu phi chức năng trên.

CHƯƠNG 3. CÔNG NGHỆ SỬ DỤNG

Ứng dụng được chia thành hai thành phần riêng biệt là app mobile dành cho người dùng và web admin dành cho quản trị viên nên mỗi thành phần sẽ sử dụng các công nghệ khác nhau để thực hiện các nhiệm vụ. Về phần Mobile, các công nghệ chính được sử dụng là Flutter[1], Dart[10] và Firebase[4]. Về phía server, các công nghệ chính được sử dụng là Spring boot FrameworkSBF và ReactJS[2].

3.1 Các công nghệ sử dụng bên phía mobile

3.1.1 Flutter[1]

Flutter là một framework mã nguồn mở được phát triển bởi Google, cho phép lập trình viên xây dựng các ứng dụng đa nền tảng (cross-platform) từ một cơ sở mã (codebase) duy nhất. Flutter sử dụng ngôn ngữ Dart làm ngôn ngữ lập trình chính và cung cấp một bộ widget được xây dựng sẵn, tối ưu hóa cho hiệu suất và khả năng tùy biến cao. Ứng dụng được viết bằng Flutter có thể được biên dịch trực tiếp thành mã gốc (native code) cho Android, iOS, Web, Desktop mà không cần cầu nối (bridge) như các công nghệ hybrid khác như React Native hay Cordova.

Trong đồ án, em sử dụng Flutter để xây dựng toàn bộ giao diện và logic phía client của ứng dụng. Mục tiêu của việc sử dụng Flutter là nhằm đáp ứng các yêu cầu về hiệu suất, khả năng mở rộng và trải nghiệm người dùng nhất quán trên nhiều nền tảng.

Các ưu điểm nổi bật của Flutter:

- **Hiệu suất cao :** Flutter sử dụng công cụ đồ họa Skia để vẽ giao diện trực tiếp lên canvas, bỏ qua lớp trung gian như WebView hay Native Components. Do đó, tốc độ phản hồi và hiệu suất của ứng dụng rất cao, tương đương với ứng dụng native truyền thống.
- **Một codebase cho nhiều nền tảng:** Nhờ khả năng biên dịch sang mã native và hỗ trợ Android, iOS, web, desktop, lập trình viên chỉ cần viết mã một lần nhưng có thể triển khai trên nhiều nền tảng, giúp tiết kiệm thời gian phát triển và dễ bảo trì.
- **Hot Reload:** Flutter hỗ trợ tính năng Hot Reload, cho phép lập trình viên có thể cập nhật thay đổi trong code và xem kết quả ngay lập tức mà không cần biên dịch lại toàn bộ ứng dụng.
- **UI tùy biến mạnh mẽ:** Flutter cung cấp một bộ widget phong phú, đồng thời hỗ trợ tạo các widget tùy chỉnh dễ dàng, đáp ứng mọi yêu cầu thiết kế hiện đại.

như Material Design hay Cupertino.

- **Cộng đồng lớn và tài liệu đầy đủ:** Flutter có một cộng đồng phát triển sôi động, với nhiều thư viện mã nguồn mở hỗ trợ từ UI đến logic và API. Tài liệu chính thức từ Google được cập nhật liên tục, dễ tiếp cận cho người mới.

Lý do lựa chọn Flutter: So với các lựa chọn khác như React Native, Xamarin hay Kotlin Multiplatform, Flutter có lợi thế về hiệu suất do không sử dụng cầu nối JavaScript và có khả năng tùy biến UI sâu. Với yêu cầu của đồ án là xây dựng một ứng dụng có giao diện phong phú, tương tác cao, hiệu suất mượt mà và có thể triển khai trên nhiều nền tảng, Flutter là sự lựa chọn tối ưu hơn cả.

3.1.2 Dart[10]

Dart là ngôn ngữ lập trình do Google phát triển và được sử dụng làm ngôn ngữ chính cho Flutter. Dart là ngôn ngữ hướng đối tượng, có cú pháp tương tự như Java và JavaScript, hỗ trợ lập trình bất đồng bộ hiệu quả với các từ khóa `async`, `await` và `Future`. Trong dự án, Dart đóng vai trò trung tâm để xây dựng toàn bộ logic nghiệp vụ và giao tiếp dữ liệu với các dịch vụ backend. Tất cả các lớp liên quan đến hiển thị UI, quản lý trạng thái, xử lý logic, gọi API, lưu trữ tạm thời... đều được hiện thực bằng Dart.

Đặc điểm nổi bật của Dart:

- **Hỗ trợ lập trình bất đồng bộ mạnh mẽ:** Dart được thiết kế để tối ưu hóa cho các ứng dụng giao diện người dùng, nên có cơ chế xử lý bất đồng bộ rất hiệu quả, giúp ứng dụng luôn mượt mà trong quá trình tải dữ liệu từ mạng hoặc thực hiện tác vụ IO.
- **Ngôn ngữ đa năng:** Dart có thể chạy dưới dạng biên dịch AOT (Ahead-of-Time) để tạo mã native cho ứng dụng release, hoặc JIT (Just-In-Time) cho quá trình phát triển với Hot Reload. Điều này giúp cân bằng giữa hiệu suất và tốc độ phát triển.
- **Cú pháp dễ tiếp cận:** Với cú pháp tương tự JavaScript/Java và hỗ trợ tính năng hiện đại như null safety, Dart rất dễ tiếp cận đối với lập trình viên đã quen thuộc với các ngôn ngữ lập trình phổ biến.
- **Tích hợp sâu với Flutter:** Dart được thiết kế đi kèm với Flutter nên có khả năng tương thích tuyệt đối và tận dụng toàn bộ sức mạnh của framework, đặc biệt là trong việc quản lý widget và trạng thái.

So với JavaScript (React Native) hay Xamarin, Dart là ngôn ngữ hiện đại hơn, được thiết kế dành riêng cho xây dựng UI đa nền tảng, đồng thời có hiệu suất tốt hơn trong môi trường mobile nhờ khả năng biên dịch thành mã native thực sự. Vì thế,

Dart là lựa chọn tất yếu khi sử dụng Flutter.

3.1.3 Firebase[4]

Firebase là một nền tảng phát triển ứng dụng di động và web được cung cấp bởi Google. Firebase cung cấp nhiều dịch vụ backend như xác thực người dùng, cơ sở dữ liệu thời gian thực, lưu trữ file, thông báo đẩy, phân phối ứng dụng, v.v... giúp giảm tải công việc phát triển và quản lý backend cho lập trình viên, đặc biệt phù hợp với các ứng dụng nhỏ đến trung bình hoặc MVP (Minimum Viable Product).

Trong đồ án, em sử dụng Firebase làm nền tảng backend chính để hỗ trợ lưu trữ dữ liệu người dùng, xác thực tài khoản, gửi thông báo, và phân phối bản beta ứng dụng. Các dịch vụ Firebase được sử dụng bao gồm:

a, Firebase Authentication

Được sử dụng để xác thực người dùng trong ứng dụng. Firebase Authentication hỗ trợ nhiều hình thức đăng nhập như email/password, Google, Facebook, Apple, v.v. Em sử dụng xác thực bằng email/password và Google nhằm đơn giản hóa quá trình đăng ký và đăng nhập người dùng, đồng thời dễ tích hợp trong giao diện Flutter.

Ưu điểm:

- Dễ tích hợp với Flutter qua thư viện firebase auth
- Bảo mật cao, quản lý người dùng qua Firebase Console.
- Tự động xử lý email xác nhận, reset mật khẩu

b, Cloud Firestore

Đây là hệ quản trị cơ sở dữ liệu NoSQL thời gian thực do Firebase cung cấp. Firestore cho phép lưu trữ dữ liệu dạng document/collection với khả năng đồng bộ hóa giữa client và server theo thời gian thực. Trong đồ án, em dùng Firestore để lưu các dữ liệu liên quan đến người dùng như thông tin cá nhân, các lượt truy cập, thông tin đơn hàng, sản phẩm với nhiều trường phức tạp.

Ưu điểm:

- Hỗ trợ đọc/ghi theo thời gian thực.
- Có thể truy vấn dữ liệu phức tạp theo nhiều điều kiện.
- Tự động scale khi số lượng người dùng tăng.

c, Firebase Storage

Được sử dụng để lưu trữ các tệp tin như ảnh đại diện, ảnh sản phẩm, ảnh và video bình luận mà người dùng upload. Firebase Storage hoạt động dựa trên Google

Cloud Storage, hỗ trợ upload/download hình ảnh, với độ bảo mật cao.

Ưu điểm:

- Cho phép lưu trữ hình ảnh, video với số lượng và dung lượng lớn .
- Lưu trữ file lâu dài, được truy cập qua URL bảo mật.

d, Firebase App Distribution

Firebase App Distribution giúp phân phối phiên bản beta của ứng dụng đến người kiểm thử (tester) trước khi phát hành chính thức trên Play Store hoặc App Store. Trong dự án, em sử dụng App Distribution để gửi các bản build thử nghiệm đến nhóm bạn kiểm thử giao diện và tính năng.

Ưu điểm:

- Phân phối nhanh chóng không cần qua Store.
- Quản lý dễ dàng danh sách người kiểm thử.
- Tự động gửi email thông báo phiên bản mới.

e, Firebase Cloud Messaging (FCM)

Đây là dịch vụ gửi thông báo đẩy (push notification) miễn phí. Trong đồ án, FCM được dùng để gửi thông báo đến người dùng khi có sự kiện quan trọng như thông tin đơn hàng, khuyến mại, voucher, tích điểm thưởng, hoàn thành nhiệm vụ...

Ưu điểm:

- Cho phép gửi thông báo theo nhóm người dùng hoặc cá nhân.
- Có thể kết hợp logic phía server để tùy chỉnh nội dung gửi.
- Tăng tương tác với người dùng.

Lý do lựa chọn Firebase: so với các giải pháp backend như AWS Amplify, Supabase, hay tự triển khai server riêng, Firebase nổi bật nhờ:

- Tích hợp chặt chẽ với Flutter (có sẵn các thư viện).
- Miễn phí ở mức sử dụng cơ bản, phù hợp với dự án cá nhân.
- Không cần thiết lập server hay quản lý hạ tầng.
- Đáp ứng đầy đủ các chức năng backend cho ứng dụng di động.

Firebase giúp em tập trung vào phát triển tính năng và trải nghiệm người dùng mà không cần lo lắng về vấn đề triển khai hệ thống backend phức tạp.

3.2 Các công nghệ sử dụng bên phía server

3.2.1 ReactJS[2]

ReactJS là một thư viện JavaScript mã nguồn mở được phát triển bởi Facebook, chuyên dùng để xây dựng giao diện người dùng (UI), đặc biệt là cho các ứng dụng đơn trang (Single Page Applications - SPA). React cho phép xây dựng UI bằng cách chia nhỏ giao diện thành các thành phần (components) độc lập, tái sử dụng được.

Những đặc điểm nổi bật của ReactJS:

- **Virtual DOM:** React sử dụng Virtual DOM để tối ưu hóa hiệu suất. Khi dữ liệu thay đổi, React sẽ tính toán sự khác biệt giữa DOM hiện tại và DOM mới, sau đó chỉ cập nhật những phần thực sự thay đổi trên giao diện.
- **Component-Based Architecture:** Kiến trúc dựa trên component giúp tách biệt các phần giao diện, tăng khả năng tái sử dụng, dễ kiểm thử và dễ bảo trì mã nguồn.
- **JSX:** React sử dụng JSX (JavaScript XML) – một cú pháp mở rộng của JavaScript, cho phép viết mã giống HTML trong các file JavaScript, giúp việc viết và hiểu giao diện dễ dàng hơn.
- **Hệ sinh thái mạnh mẽ:** React có cộng đồng lớn, tài liệu đầy đủ và nhiều thư viện hỗ trợ (React Router, Redux, v.v.), giúp dễ dàng mở rộng và xây dựng các ứng dụng phức tạp.

Trong đồ án, em sử dụng ReactJS để xây dựng giao diện cho web Admin. React giúp giao diện hoạt động mượt mà, phản hồi nhanh với trải nghiệm người dùng tốt nhờ kiến trúc component hóa và khả năng cập nhật DOM hiệu quả.

3.2.2 Spring Boot Framework[3]

Spring Boot là một framework mã nguồn mở được xây dựng dựa trên nền tảng Spring Framework – một trong những nền tảng phổ biến nhất trong phát triển ứng dụng Java. Spring Boot giúp đơn giản hóa quá trình cấu hình và triển khai ứng dụng Java, đặc biệt là các ứng dụng web và microservices.

Một số đặc điểm nổi bật của Spring Boot:

- **Cấu hình tối thiểu:** Spring Boot giúp loại bỏ phần lớn cấu hình rườm rà bằng cách sử dụng các cấu hình mặc định hợp lý, giúp nhà phát triển bắt đầu nhanh chóng.
- **Tích hợp mạnh mẽ:** Tích hợp sẵn với các công nghệ như JPA/Hibernate, Spring Security, Spring Data, REST API, v.v., giúp dễ dàng xây dựng các ứng

dụng backend mạnh mẽ.

- **Đóng gói ứng dụng dễ dàng:** Spring Boot cho phép đóng gói ứng dụng dưới dạng file JAR hoặc WAR, có thể chạy độc lập mà không cần cài đặt thêm máy chủ.
- **Quản lý phụ thuộc linh hoạt:** Sử dụng Maven hoặc Gradle để quản lý thư viện một cách hiệu quả.
- **Kiến trúc RESTful:** Hỗ trợ xây dựng RESTful API dễ dàng nhờ các annotation như @RestController, @GetMapping, @PostMapping,...

Trong đồ án, em sử dụng Spring Boot để xây dựng backend server web admin, xử lý các yêu cầu từ client (ReactJS), thao tác với cơ sở dữ liệu, và cung cấp các RESTful API. Spring Boot đảm bảo hệ thống có khả năng mở rộng, bảo mật và dễ bảo trì.

3.3 Các công nghệ sử dụng để phát triển tính năng gợi ý

3.3.1 Python[11]

Python là một ngôn ngữ lập trình bậc cao, mạnh mẽ và dễ đọc, được sử dụng phổ biến trong các lĩnh vực như khoa học dữ liệu, trí tuệ nhân tạo và học máy. Với thư viện phong phú, cộng đồng lớn và khả năng viết mã ngắn gọn, Python rất phù hợp để phát triển các mô hình gợi ý dựa trên thuật toán học máy.

Trong đồ án, em sử dụng Python để xây dựng mô hình gợi ý. Các thư viện được sử dụng bao gồm:

- **Pandas, NumPy:** Dùng để xử lý dữ liệu người dùng, lịch sử tương tác.
- **LightFM:** Một thư viện mã nguồn mở được phát triển cho bài toán hệ thống gợi ý, hỗ trợ cả hai phương pháp: collaborative filtering và content-based.
- **Scikit-learn:** Hỗ trợ chuẩn hóa dữ liệu và đo lường độ chính xác của mô hình.

Python giúp quá trình xử lý dữ liệu và huấn luyện mô hình diễn ra linh hoạt, nhanh chóng và dễ triển khai.

3.3.2 LightFM[6]

LightFM là một thư viện Python dùng để xây dựng hệ thống gợi ý kết hợp (hybrid recommender system), hỗ trợ cả collaborative filtering và content-based filtering. Nó được tối ưu hóa cho dữ liệu thưa (sparse data) và phù hợp với các hệ thống có lượng lớn người dùng và item nhưng ít tương tác.

Các lý do em lựa chọn LightFM trong đồ án:

- Hỗ trợ gợi ý cá nhân hóa cho từng người dùng dựa trên lịch sử tương tác và

đặc trưng nội dung.

- Có thể điều chỉnh dễ dàng giữa các mô hình latent factor và mô hình sử dụng metadata.
- Cho phép huấn luyện nhanh và hiệu quả trên dữ liệu lớn, phù hợp cho môi trường thực tế.

Sau khi huấn luyện mô hình LightFM bằng Python, em đã lưu lại mô hình và danh sách gợi ý để triển khai phía server.

3.3.3 NodeJS[12]

NodeJS là một nền tảng chạy JavaScript phía server, được xây dựng trên engine V8. Với khả năng xử lý bất đồng bộ mạnh mẽ và tốc độ cao, NodeJS là lựa chọn phù hợp để xây dựng các API phục vụ cho mobile app.

Trong đồ án, em sử dụng NodeJS để:

- Triển khai API RESTful để mobile app có thể gọi và lấy danh sách gợi ý từ mô hình.
- Kết nối tới dữ liệu được xử lý từ Python thông qua file
- Đảm bảo khả năng mở rộng và phản hồi nhanh nhờ kiến trúc event-driven.

NodeJS giúp việc tích hợp giữa backend (API) và mobile app diễn ra hiệu quả và đồng bộ với toàn hệ thống.

3.3.4 Kaggle IDE[13]

Kaggle là một nền tảng nổi tiếng cho khoa học dữ liệu và học máy, cung cấp môi trường lập trình trực tuyến (Kaggle Notebooks) với tài nguyên sẵn có như GPU, thư viện học máy và quyền truy cập trực tiếp đến tập dữ liệu.

Trong đồ án, em sử dụng Kaggle để:

- Phát triển, thử nghiệm và huấn luyện mô hình LightFM trong môi trường trực tuyến.
- Tận dụng các gói thư viện được cài đặt sẵn và tài nguyên xử lý mạnh như GPU để tăng tốc quá trình huấn luyện.
- Xuất kết quả (danh sách gợi ý hoặc model đã huấn luyện) để sử dụng cho backend NodeJS.

3.4 Công nghệ sử dụng để xây dựng chatbot

RASA [7] là một nền tảng mã nguồn mở chuyên dùng để xây dựng chatbot có khả năng hội thoại tự nhiên. Với nhiều công cụ hỗ trợ mạnh mẽ, Rasa giúp lập trình

viên phát triển các hệ thống hội thoại thông minh, phù hợp với nhiều lĩnh vực như tự động hóa chăm sóc khách hàng, hỗ trợ tư vấn bán hàng, và các ứng dụng tương tác khác.

Rasa nổi bật nhờ khả năng xử lý ngôn ngữ tự nhiên hiệu quả, khả năng tùy biến cao, dễ tích hợp với nhiều dịch vụ và nền tảng như Facebook Messenger, Slack, Telegram, các hệ thống CRM hoặc API của các nền tảng web. Điểm mạnh của framework này là hoàn toàn miễn phí và được cộng đồng sử dụng rộng rãi.

Kiến trúc chính của Rasa bao gồm hai thành phần: Rasa NLU và Rasa Core. Trong đó, Rasa NLU đảm nhiệm vai trò phân tích ngôn ngữ đầu vào, giúp nhận biết ý định của người dùng và trích xuất các thông tin quan trọng (thực thể) từ văn bản. Nhờ đó, chatbot có thể hiểu rõ hơn nội dung và ngữ cảnh của cuộc trò chuyện, từ đó đưa ra phản hồi chính xác hơn.

Trong khi đó, Rasa Core đóng vai trò điều phối cuộc hội thoại. Thành phần này sử dụng các thuật toán học máy để xác định hành động tiếp theo dựa trên lịch sử tương tác và ngữ cảnh hiện tại. Điều này giúp chatbot có thể duy trì cuộc đối thoại một cách tự nhiên và linh hoạt hơn, xử lý được nhiều tình huống phức tạp.

Nhờ tính linh hoạt cao, dễ tiếp cận và khả năng hoạt động hiệu quả ngay cả với tập dữ liệu huấn luyện nhỏ, em đã lựa chọn sử dụng RASA làm framework chính để xây dựng chức năng chatbot cho hệ thống của mình.

Vậy là trong chương 3, em đã trình bày một số công nghệ được sử dụng trong ứng dụng và lý do vì sao sử dụng chúng. Sau đây, em sẽ trình bày nội dung của Chương 4 - Thiết kế, triển khai và đánh giá hệ thống.

CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG

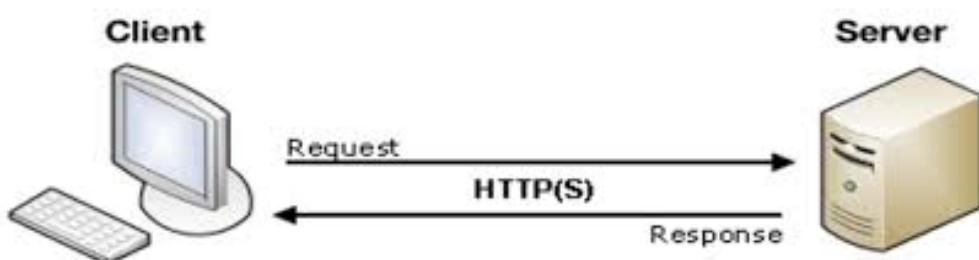
4.1 Thiết kế kiến trúc

4.1.1 Lựa chọn kiến trúc phần mềm

a, Kiến trúc tổng quan hệ thống

Hệ thống được xây dựng theo mô hình tổng thể Client - Server. Trong đó, ứng dụng chia thành hai phân hệ chính: ứng dụng mobile dành cho người dùng cuối và hệ thống web admin dành cho quản trị viên.

Hình 4.1 minh họa kiến trúc client-server



Hình 4.1: Kiến trúc Client-Server (nguồn: <https://codelearn.io/sharing/tim-hieu-ve-mo-hinh-client-server>)

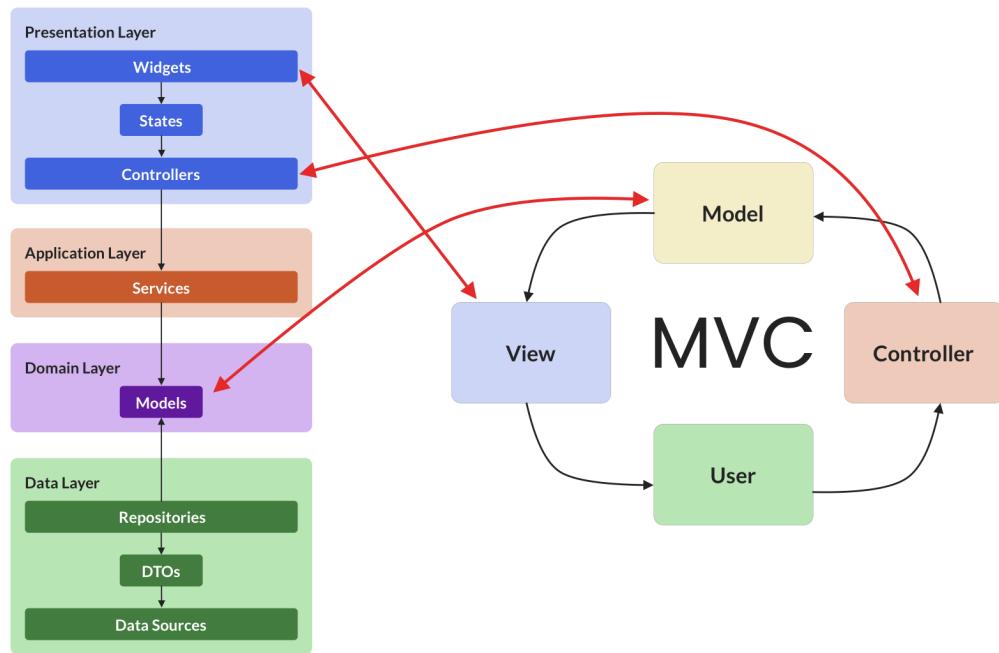
Phía mobile với client được phát triển bằng ngôn ngữ Dart và framework Flutter, server sử dụng Firebase. Phía web admin là một hệ thống web xây dựng theo mô hình client-server, trong đó phía client sử dụng ReactJS và phía server sử dụng Spring Boot. Cả hai phân hệ đều tuân theo nguyên tắc phân lớp rõ ràng nhằm đảm bảo khả năng mở rộng, bảo trì và phát triển lâu dài.

b, Kiến trúc phần mềm phía mobile (Flutter)

Ứng dụng mobile sử dụng kiến trúc dạng module với các package chính gồm:

- **Model:** chứa các lớp dữ liệu đại diện cho cấu trúc thông tin (ví dụ: User, Order, Product...).
- **Repository:** là lớp trung gian giữa dữ liệu và logic nghiệp vụ. Repository chịu trách nhiệm truy xuất, đồng bộ và cập nhật dữ liệu từ các nguồn khác nhau, trong đó có Firebase (Firestore và Storage).
- **Controller (hoặc Provider):** xử lý logic nghiệp vụ, nhận sự kiện từ người dùng và điều hướng dữ liệu giữa Model và giao diện.
- **Screen:** chứa giao diện người dùng của từng màn hình như HomeScreen, SearchScreen, SettingScreen, v.v.

Hình 4.2 minh họa kiến trúc bên mobile:



Hình 4.2: Kiến trúc flutter mobile (nguồn: <https://codewithandrea.com/articles/comparison-flutter-app-architectures/>)

Kiến trúc này tách biệt rõ ràng giữa giao diện, logic và dữ liệu nhằm đảm bảo tính dễ mở rộng và kiểm thử. Firebase đóng vai trò là backend chính cho ứng dụng mobile, cung cấp các dịch vụ như lưu trữ dữ liệu, xác thực người dùng, thông báo đẩy (Firebase Cloud Messaging).

c, Kiến trúc phần mềm phía web admin (ReactJS + Spring Boot)

Hệ thống web admin được phát triển theo mô hình Client - Server. Trong đó:

Phía client sử dụng ReactJS với mô hình component-based, bao gồm:

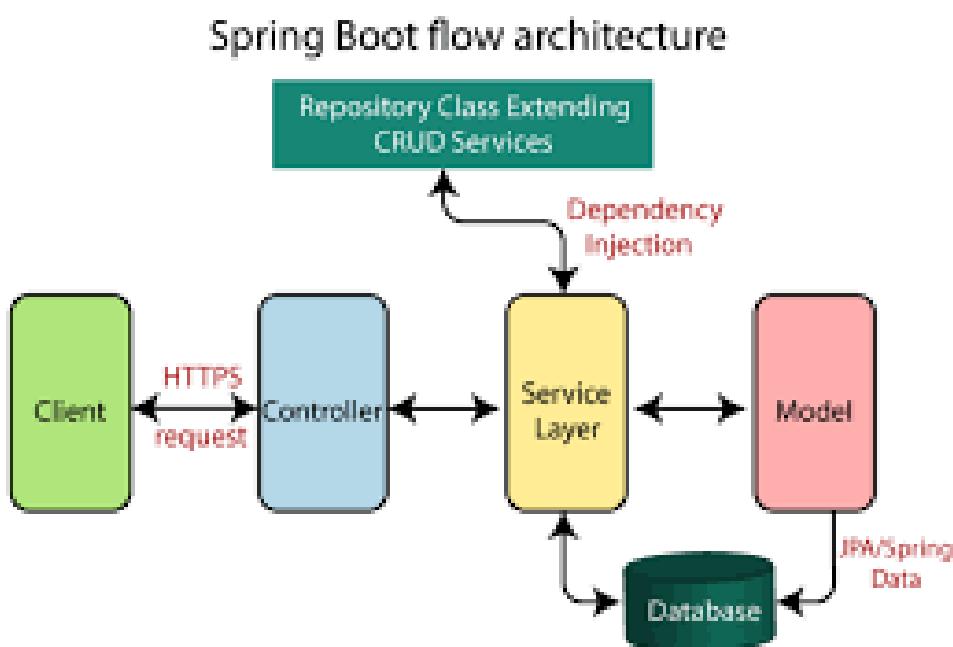
- **Component:** giao diện người dùng, chia nhỏ theo chức năng (quản lý người dùng, quản lý đơn hàng, thống kê)
- **Service/API layer:** gửi yêu cầu HTTP đến server Spring Boot và xử lý dữ liệu phản hồi (dùng thư viện axios).
- **State management:** sử dụng useEffect, useStateuseState để quản lý trạng thái ứng dụng.

Phía server sử dụng Spring Boot theo mô hình phân tầng rõ ràng:

- **Entity:** đại diện cho các bảng trong cơ sở dữ liệu (User, Address, Order, v.v.)
- **DTO (Data Transfer Object):** trung gian chuyển dữ liệu giữa các tầng, giúp giảm bớt dữ liệu không cần thiết.

- **Repository:** giao tiếp với cơ sở dữ liệu Firebase Firestore
- **Service:** chứa logic nghiệp vụ chính, xử lý dữ liệu nhận từ controller và repository.
- **Controller:** nhận yêu cầu từ client, gọi service xử lý và trả kết quả.
- **Config:** chứa các cấu hình hệ thống như bảo mật (Spring Security), CORS, kết nối cơ sở dữ liệu, v.v.

Hình 4.3 minh họa kiến trúc phân tầng của Spring Boot server:

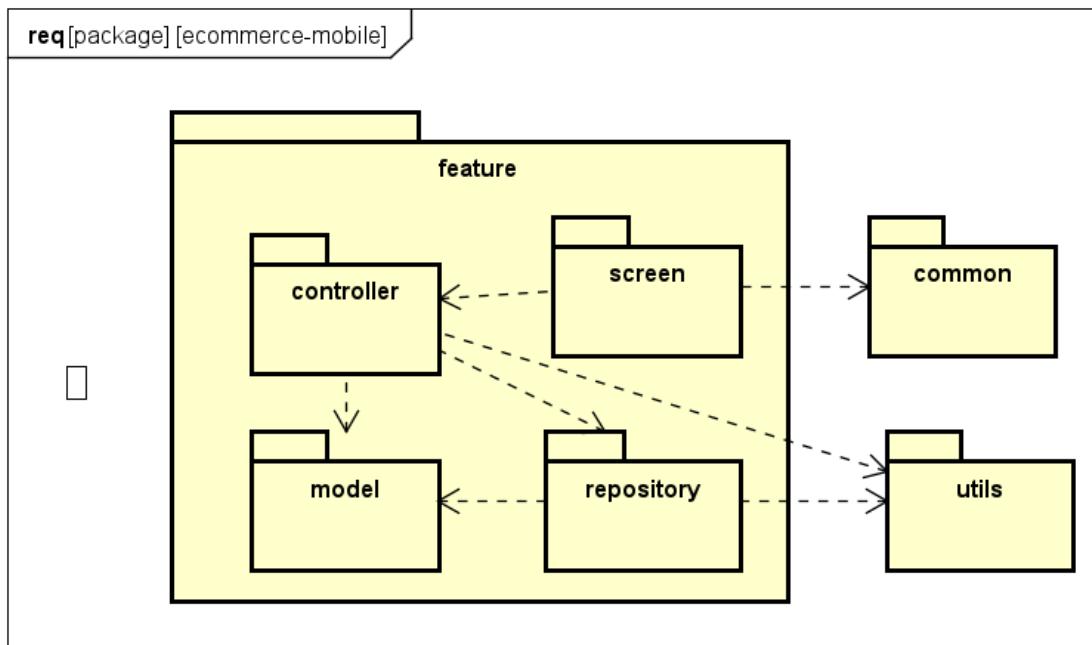


Hình 4.3: Kiến trúc phân tầng của server Spring Boot (nguồn: <https://shareprogramming.net/kien-truc-va-workflow-cua-spring-boot-framework/>)

Cách tổ chức phân tầng rõ ràng này giúp hệ thống dễ bảo trì, mở rộng, cũng như tái sử dụng các thành phần.

4.1.2 Thiết kế tổng quan

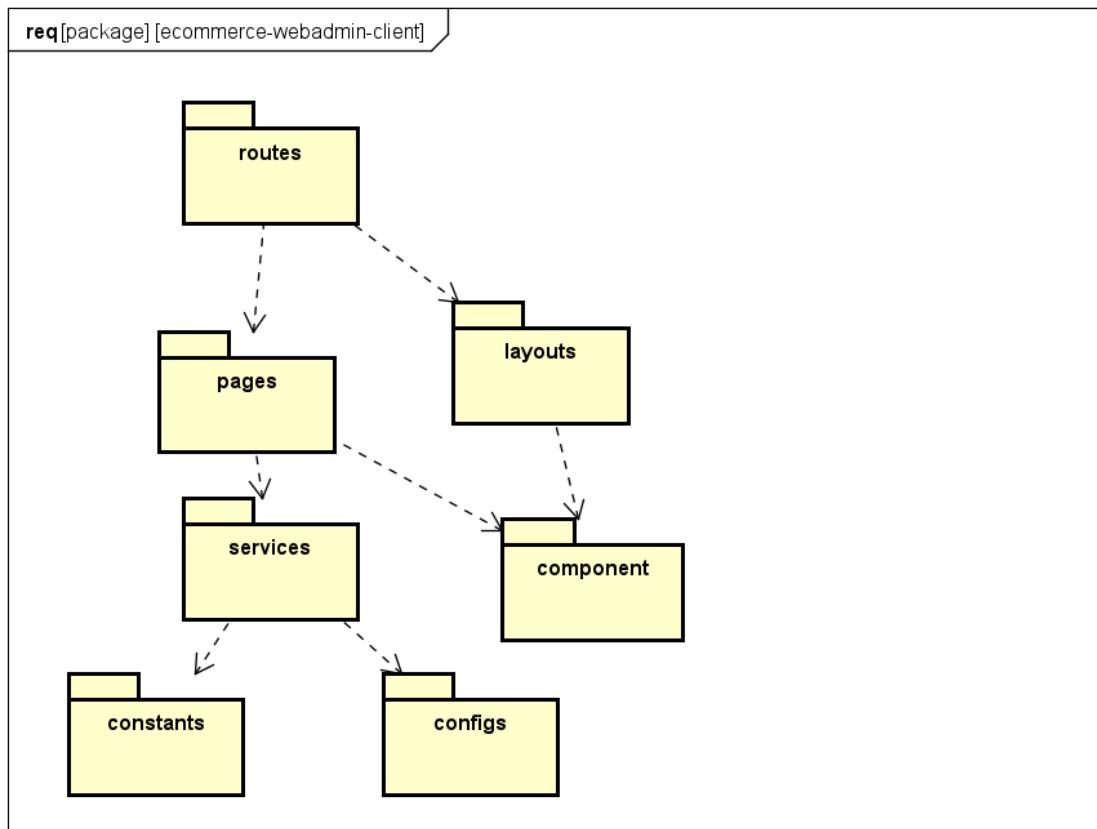
a, Thiết kế tổng quan phía Mobile



Hình 4.4: Biểu đồ package tổng quan phía Mobile

Hình 4.4 mô tả biểu đồ package tổng quan phía Mobile. Phần này bao gồm các gói: (i) gói controller là trung gian giữa dữ liệu và giao diện, phụ thuộc vào repository để truy vấn dữ liệu và model để parse thành model cho screen sử dụng, (ii) gói repository – chứa các logic kết nối với firebase và các truy vấn, trả về model cho controller sử dụng. (iii) gói screen - chứa giao diện chính của ứng dụng, gọi controller để cung cấp xử lý logic, gọi common để tái sử dụng lại các component. (iv) model mô hình hóa dữ liệu các đối tượng trong hệ thống, (v) common chứa các component giao diện để tái sử dụng trong các màn hình screen, (vi) gói utils chứa hằng số, các hàm phụ trợ như format thời gian, text, đơn vị tiền tệ,... Biểu đồ này minh họa rõ hơn cho kiến trúc phần mềm phía Mobile ở phần [4.2] trước đó.

b, Thiết kế tổng quan phía Client Webadmin



Hình 4.5: Biểu đồ package tổng quan Client Webadmin

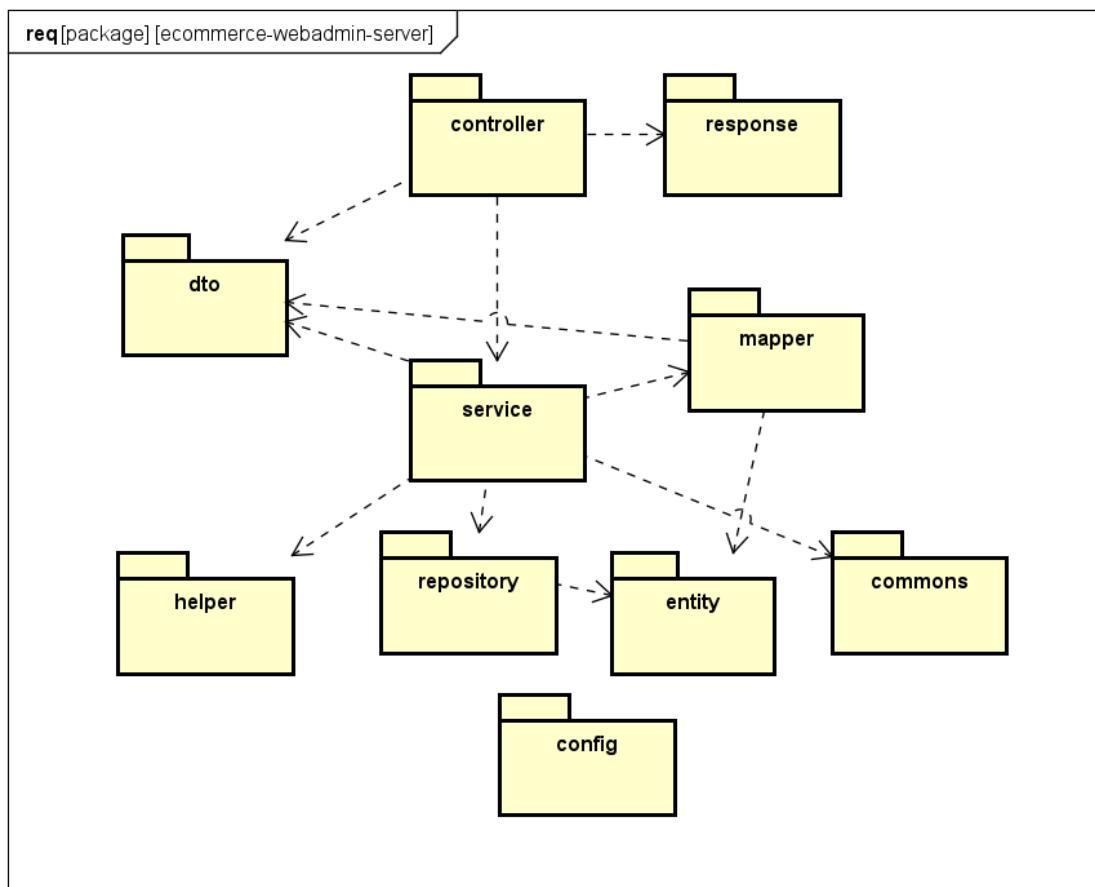
Hình 4.5 mô tả biểu đồ package tổng quan phía Client của web admin. Phần này bao gồm các gói:

- (i) **routes/** – chịu trách nhiệm định tuyến và điều hướng trong ứng dụng. Gói này khai báo các tuyến (route) tương ứng với từng màn hình trong hệ thống và ánh xạ đến các component giao diện tương ứng trong **pages/**.
- (ii) **pages/** – chứa các màn hình chính của ứng dụng (ví dụ: Dashboard, User-Management, Settings, ...). Mỗi màn hình thường bao gồm phần logic xử lý riêng và sử dụng các component từ **components/**, đồng thời gọi API từ **services/** để lấy dữ liệu cần thiết.
- (iii) **components/** – chứa các thành phần UI có thể tái sử dụng, chẳng hạn như Button, Modal, Table, Form,... Các component này được sử dụng trong nhiều **pages/** và **layouts/** để đảm bảo tính thống nhất trong giao diện người dùng.
- (iv) **services/** – chứa các hàm gửi yêu cầu HTTP đến server (thường sử dụng thư viện như Axios). Các hàm này giúp các **pages/** và **hooks/** lấy hoặc ghi dữ liệu thông qua các API backend. **services/** thường phụ thuộc vào các giá trị cấu hình được định nghĩa trong **config/**.

- (v) layouts/ – định nghĩa các bộ cục giao diện tổng thể như AdminLayout, AuthLayout,... Các layouts/ bao gồm các phần giao diện như header, sidebar, footer và sử dụng các thành phần từ components/.
- (vi) constants/ – chứa các hằng số dùng toàn hệ thống, như định danh quyền người dùng, trạng thái đơn hàng, định dạng ngày giờ,... constants/ có thể được sử dụng bởi mọi gói khác mà không gây ra phụ thuộc ngược.
- (vii) config/ – chứa các cấu hình của ứng dụng như địa chỉ API, khóa môi trường,... Gói này thường được services/ và một số phần logic trong pages/ sử dụng.

Các gói này được thiết kế với nguyên tắc phân tách trách nhiệm rõ ràng, đồng thời tuân theo cấu trúc phụ thuộc một chiều, giúp dễ bảo trì và mở rộng. Cụ thể, pages/ là trung tâm sử dụng, gọi đến services/, components/, constants/, trong khi routes/ là nơi khởi tạo điều hướng giữa các pages/. Gói config/ và constants/ là các gói phụ trợ dùng chung nhưng không phụ thuộc vào các gói còn lại.

c, Thiết kế tổng quan phía Server Webadmin



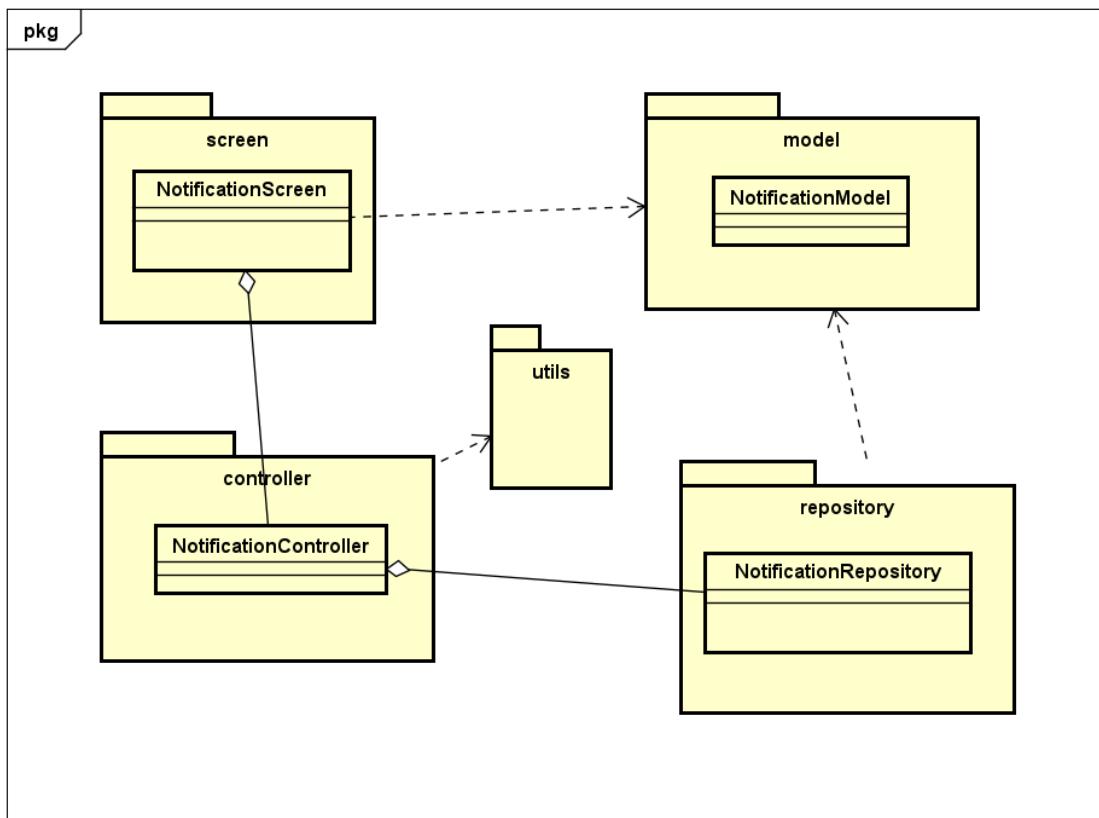
Hình 4.6: Biểu đồ package tổng quan Server Webadmin

Hình 4.6 mô tả biểu đồ package tổng quan phía Server của webadmin. Phần này bao gồm các gói:

- (i) controller – chứa các lớp điều khiển (REST controller) chịu trách nhiệm tiếp nhận và xử lý các yêu cầu HTTP từ phía client, gọi đến service để thực hiện logic nghiệp vụ và trả về kết quả dạng response class;
- (ii) service – chứa các lớp nghiệp vụ trung gian, thực hiện xử lý logic chính của ứng dụng. Gói này tương tác với repository để truy xuất hoặc ghi dữ liệu, đồng thời sử dụng các lớp dto, mapper và helper,commons để chuẩn hóa dữ liệu đầu vào/đầu ra.
- (iii) repository – chứa liên kết đến firebase được injection, dùng để truy xuất dữ liệu từ cơ sở dữ liệu
- (iv) dto (Data Transfer Object) – chứa các lớp trung gian đại diện cho dữ liệu trao đổi giữa client và server, giúp tách biệt mô hình dữ liệu nội bộ (entity) với mô hình dữ liệu dùng để giao tiếp bên ngoài, giảm các trường không dùng đến và bổ sung thêm 1 số trường cần thiết.
- (v) entity – chứa các lớp ánh xạ với bảng dữ liệu trong cơ sở dữ liệu, mỗi lớp đại diện cho một thực thể và được đánh dấu bằng các annotation như @Entity, @Table.
- (vi) mapper – chứa các interface dùng để chuyển đổi giữa entity và dto, sử dụng thư viện như MapStruct để tự động ánh xạ giữa dto và entity, rất thuận tiện khi các class này có nhiều trường phức tạp.
- (vii) response – chứa các lớp định nghĩa cấu trúc phản hồi gửi về phía client, thường bao gồm các trường như dto, statusCode, token... để chuẩn hóa định dạng phản hồi API.
- (viii) helper – chứa các lớp tiện ích và hàm hỗ trợ dùng chung trong toàn bộ ứng dụng, ví dụ như xử lý định dạng ngày giờ, mã hóa mật khẩu, gửi email, ghi log,...;
- (ix) config – chứa các lớp cấu hình như cấu hình bảo mật (Spring Security), cấu hình Swagger, cấu hình CORS, cấu hình bean,... để tùy chỉnh hoạt động của ứng dụng;
- (x) commons – chứa các hằng số (constants), enum, lớp lõi dùng chung và định nghĩa exception toàn cục giúp tiêu chuẩn hóa xử lý lỗi trong hệ thống.

4.1.3 Thiết kế chi tiết gói

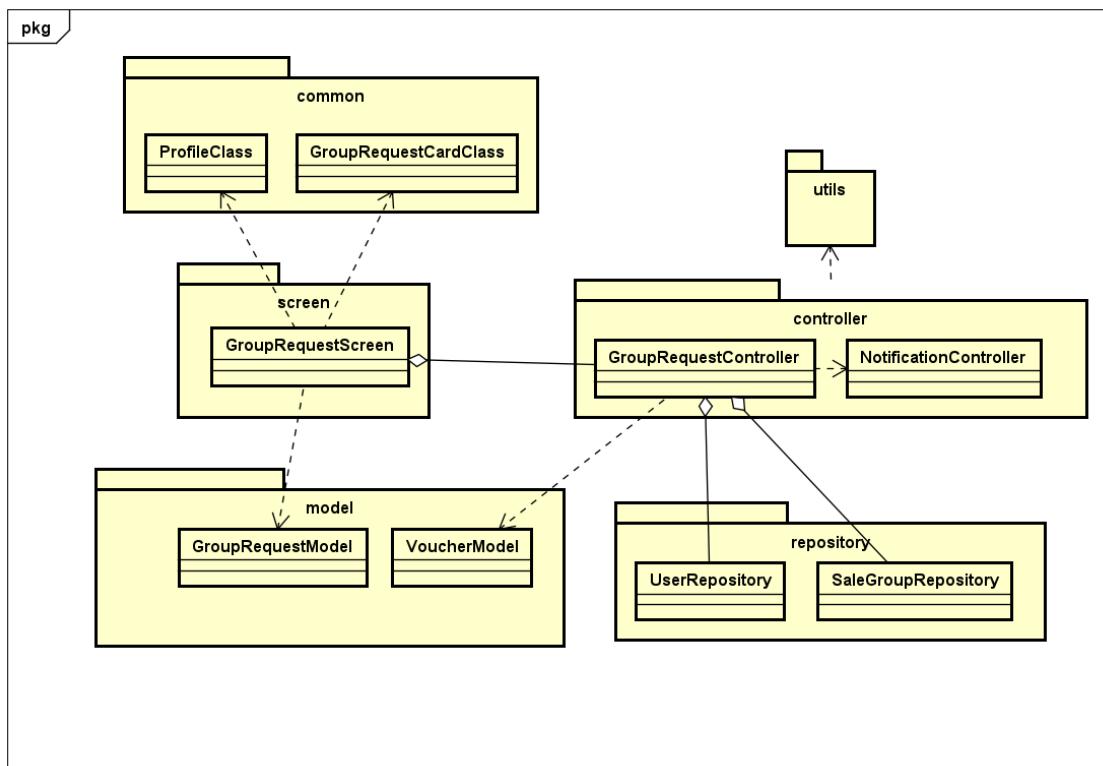
a, Thiết kế chi tiết gói cho tính năng "Xem thông báo" phía mobile



Hình 4.7: Biểu đồ package chi tiết cho tính năng "Xem thông báo" phía mobile

Hình 4.7 mô tả chi tiết gói cho tính năng "Xem thông báo" phía mobile. Lớp NotificationScreen tạo giao diện cho màn hình xem thông báo, nó gọi 1 đối tượng của Notification Controller để xử lý các logic nghiệp vụ, nó cũng gọi và sử dụng NotificationModel trong phương thức build của nó. Lớp controller chứa repository trong class của mình, để gọi phương thức truy vấn db và trả về dữ liệu, nó cũng gọi các class của gói utils để sử dụng các hàm hỗ trợ trong các phương thức của mình.

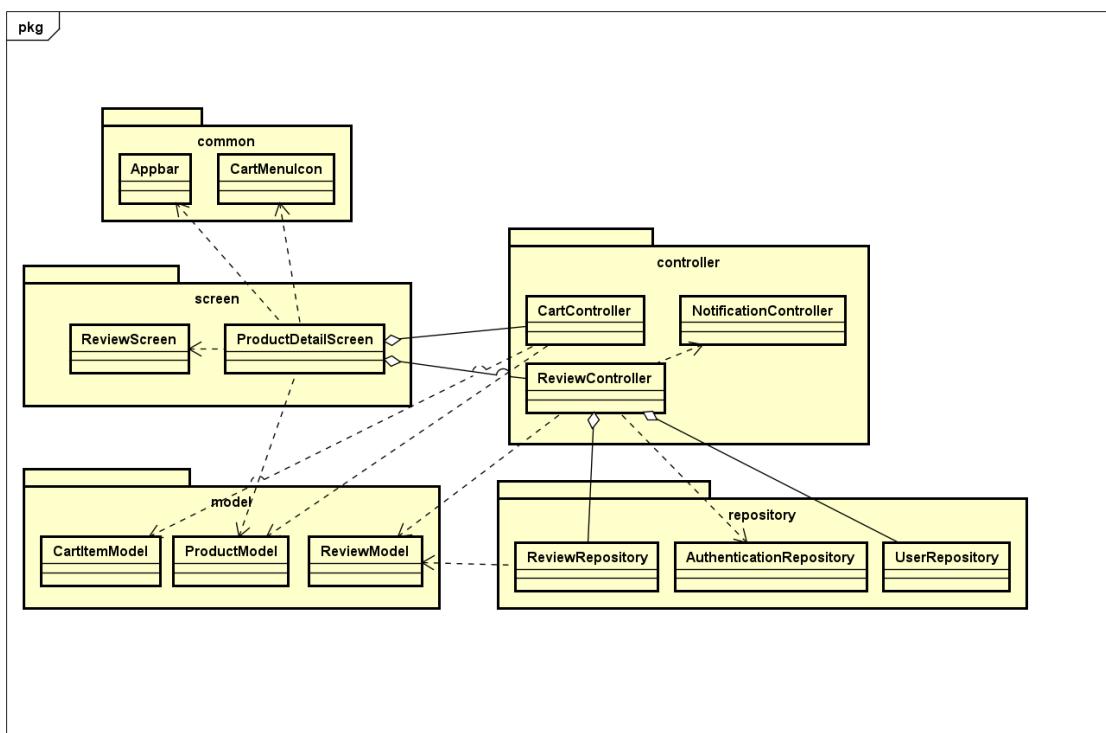
b, Thiết kế chi tiết gói cho tính năng "Duyệt yêu cầu tham gia nhóm" phía mobile



Hình 4.8: Biểu đồ package chi tiết cho tính năng "Duyệt yêu cầu tham gia nhóm" phía mobile

Hình 4.8 mô tả chi tiết gói cho tính năng "Duyệt yêu cầu tham gia nhóm" phía mobile. Lớp GroupRequestScreen tạo giao diện cho màn hình duyệt hoặc từ chối các lời mời tham gia nhóm, nó chứa group request controller trong object của mình để xử lý các logic nghiệp vụ, sử dụng GroupRequestModel trong phương thức build của nó. Lớp GroupRequestController chứa 2 repository là UserRepository và SaleGroupRepository để truy vấn db và trả về dữ liệu liên quan đến người mời, đến nhóm sẽ tham gia, ngoài ra nó cũng gọi notification controller trong 1 phương thức để gửi thông báo, có sử dụng voucher model trong 1 phương thức để tạo voucher cho các thành viên khi nhóm đạt đủ số lượng yêu cầu. Ngoài ra nó cũng gọi các class của gói utils để sử dụng các hàm hỗ trợ trong các phương thức của mình.

c, Thiết kế chi tiết gói cho tính năng "Xem chi tiết sản phẩm" phía mobile



Hình 4.9: Biểu đồ package chi tiết cho tính năng "Xem chi tiết sản phẩm" phía mobile

Hình 4.9 mô tả chi tiết gói cho tính năng "Xem chi tiết sản phẩm" phía mobile. Lớp ProductDetailScreen tạo giao diện cho màn hình xem chi tiết 1 sản phẩm, nó chứa cart controller và reviewcontroller trong object của mình để xử lý các logic nghiệp vụ liên quan đến giỏ hàng và viết bình luận của người dùng, sử dụng ProductModel trong phương thức build của nó. Nó gọi lớp Review screen, lớp Appbar và lớp Cart menu icon trong phương thức build của mình. Lớp ReviewController chứa 2 repository là UserRepository và ReviewRepository để truy vấn db và trả về dữ liệu liên quan đến người dùng hiện tại và các bình luận về sản phẩm, ngoài ra nó cũng gọi notification controller trong 1 phương thức để gửi thông báo và authentication repository trong 1 phương thức để xác thực, có sử dụng review model trong phương thức xử lý tạo bình luận mới. Lớp Cart Controller có gọi cart item model và product model trong các phương thức của mình.

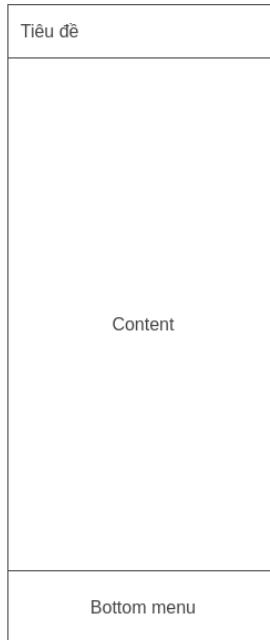
4.2 Thiết kế chi tiết

4.2.1 Thiết kế giao diện

Giao diện của ứng dụng thương mại điện tử được thiết kế phù hợp với kích thước của các loại điện thoại thông minh có kích thước từ 5 cho đến 7 inch. Các nút bấm khi thêm dữ liệu mới được thiết kế đồng bộ với phông chữ màu trắng và nền màu xanh lá cây. Các thông điệp phản hồi cũng được thống nhất một quy ước là nằm ở giữa bên dưới màn hình, thời gian xuất hiện là 4 giây. Dưới đây là hình ảnh minh

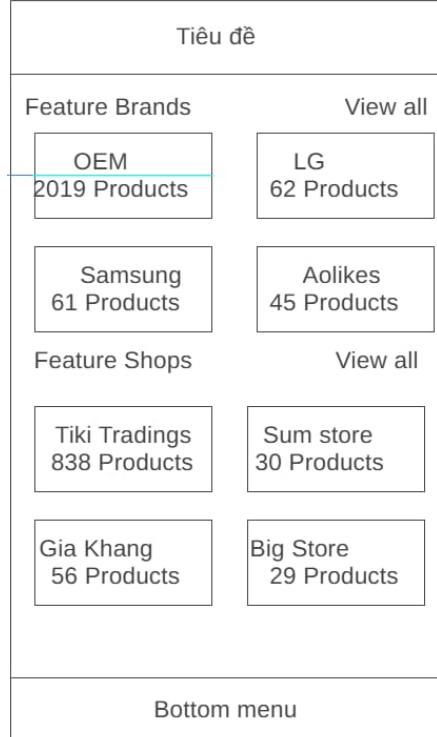
CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG

hoa thiết kế một số giao diện cho một vài chức năng của ứng dụng:



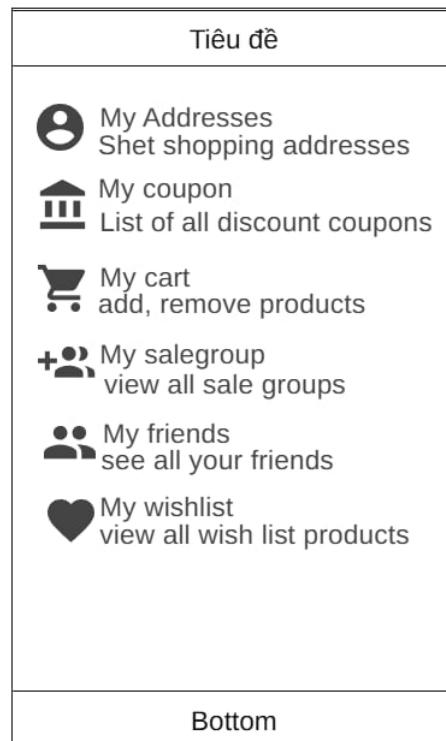
Hình 4.10: Bố cục tổng quát của ứng dụng

Hình 4.10 minh họa bố cục tổng quát của màn hình ứng dụng. Trong đó, một màn hình của ứng dụng sẽ gồm có ba thành phần chính là thanh tiêu đề ở trên cùng, ở giữa là thể hiện nội dung của màn hình và dưới cùng là thanh bottom menu để chuyển đến các màn hình khác.



Hình 4.11: Thiết kế giao diện cho màn hình cửa hàng

Hình 4.11 minh họa thiết kế giao diện cho màn hình cửa hàng. Màn hình này, sẽ bao gồm các thương hiệu trong hệ thống, các cửa hàng trong hệ thống và các sản phẩm đi kèm của mỗi thương hiệu, mỗi cửa hàng đó.

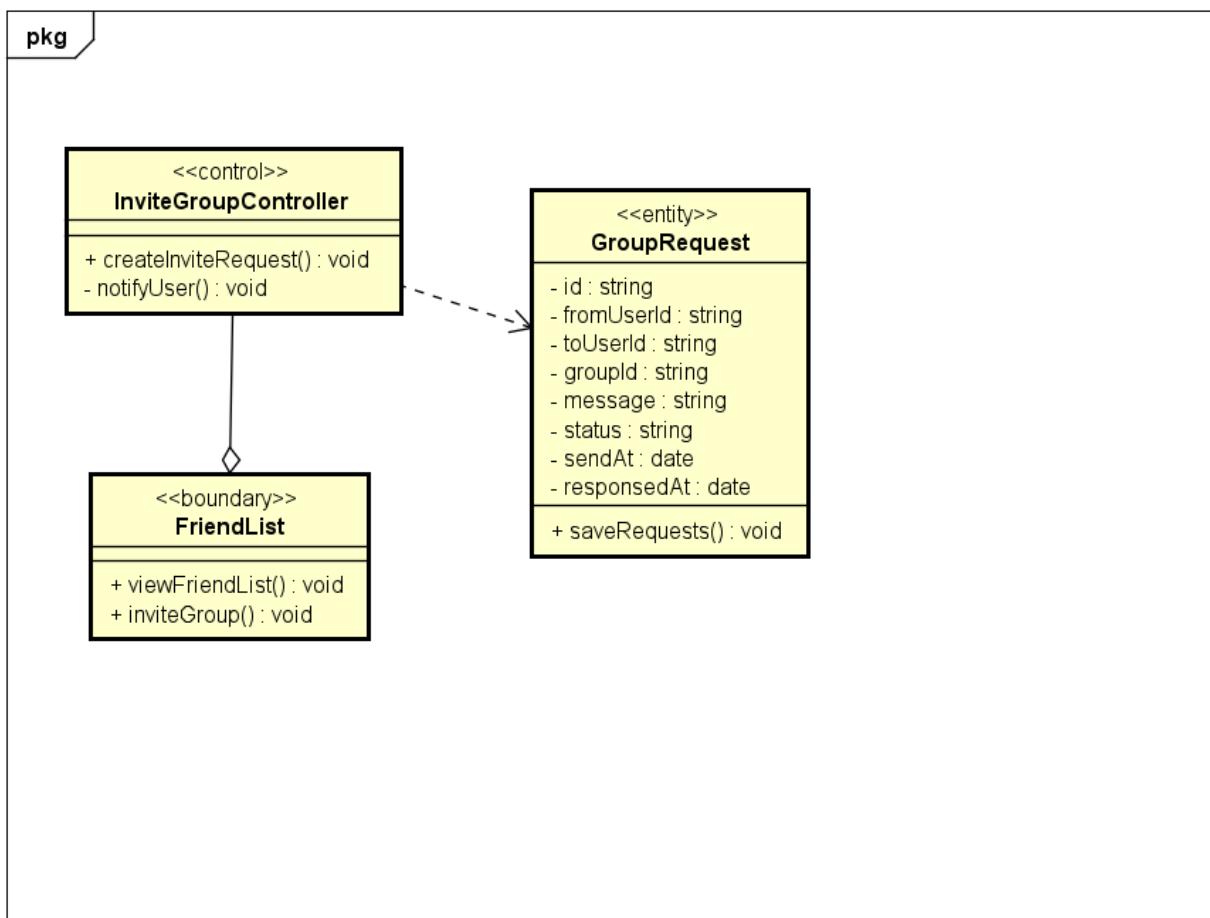


Hình 4.12: Thiết kế giao diện cho màn hình profile

Hình 4.12 minh họa thiết kế giao diện màn hình profile. Nơi đây người dùng có thể quản lý được mọi thông tin cũng như dữ liệu quan trọng, hay các tính năng hấp dẫn nhất. Mỗi mục sẽ có icon đại diện bên trái, tên tiêu đề mục ở trên và phần mô tả chi tiết mỗi mục ở phía dưới

4.2.2 Thiết kế lớp

a. Thiết kế lớp cho chức năng "Mời bạn bè tham gia nhóm"



Hình 4.13: Biểu đồ lớp cho chức năng "Mời bạn bè tham gia nhóm"

Hình 4.13 thể hiện biểu đồ lớp cho chức năng "Mời bạn bè tham gia nhóm". Dựa vào hình 4.13, em xin phép trình bày thiết kế của một số lớp như sau:

Lớp **GroupRequest**: biểu diễn và lưu trữ các thông tin của request mời bạn bè tham gia nhóm săn sale mà mình mới tạo

- Thuộc tính:
 - id: Định danh của 1 request cụ thể trong hệ thống
 - fromUserId: Định danh của người gửi request, tức là chủ sở hữu nhóm
 - toUserId: Định danh của người nhận request, tức bạn bè của chủ
 - groupId: Định danh của nhóm săn sale do chủ nhóm tạo
 - message: nội dung lời nhắn kèm theo yêu cầu
 - status: trạng thái của request: pending, accept, hoặc reject
 - sendAt: thời gian gửi request
 - respondedAt: thời gian người nhận phản hồi yêu cầu

- Phương thức:
 - saveRequests(): Lưu thông tin của yêu cầu lên database.

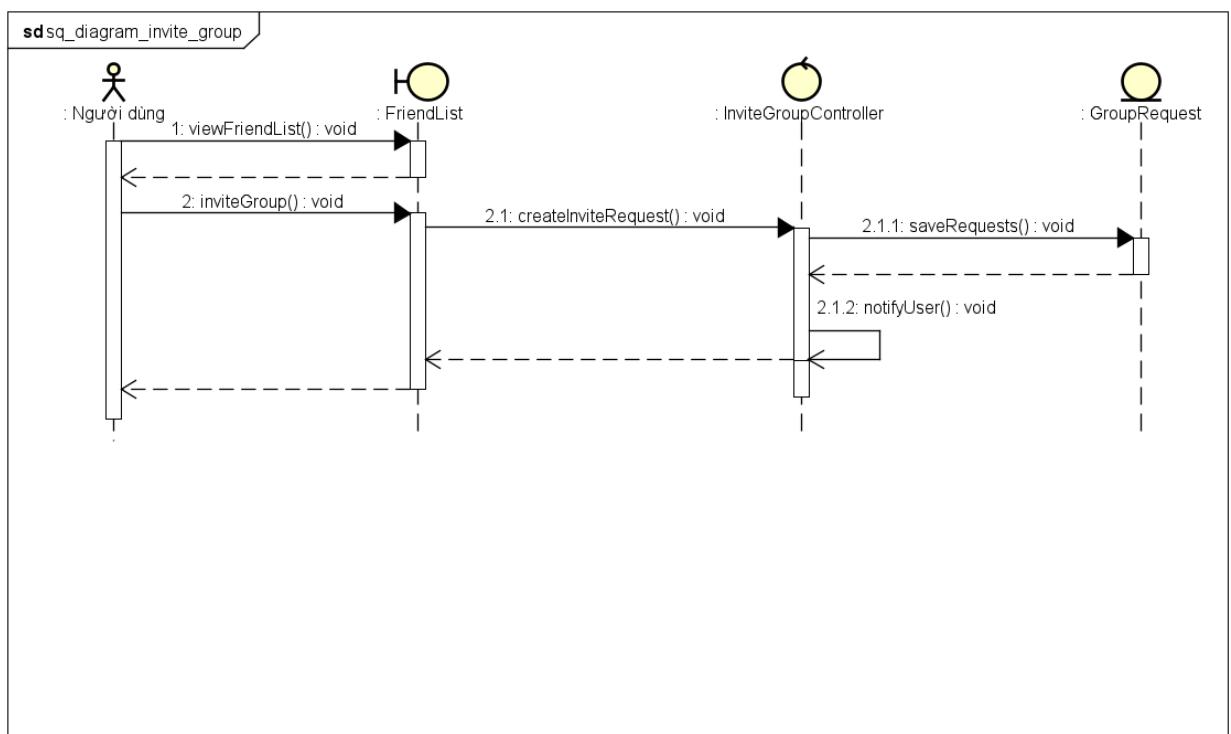
Lớp **InviteGroupController**: chứa các xử lý logic nghiệp vụ, là trung gian giữa lớp giao diện và lớp dữ liệu, gồm các phương thức:

- createInviteRequest(): Nhận vào thông tin và yêu cầu request từ UI, tạo model request và lưu lên database
- notifyUser(): Sau khi tạo và lưu request thành công thì gửi thông báo đến người gửi là đã gửi yêu cầu thành công và gửi đến người nhận là bạn có một lời mời vào nhóm mới.

Lớp **FriendList**: tạo ra giao diện cho màn hình mời bạn bè tham gia nhóm, gồm các phương thức

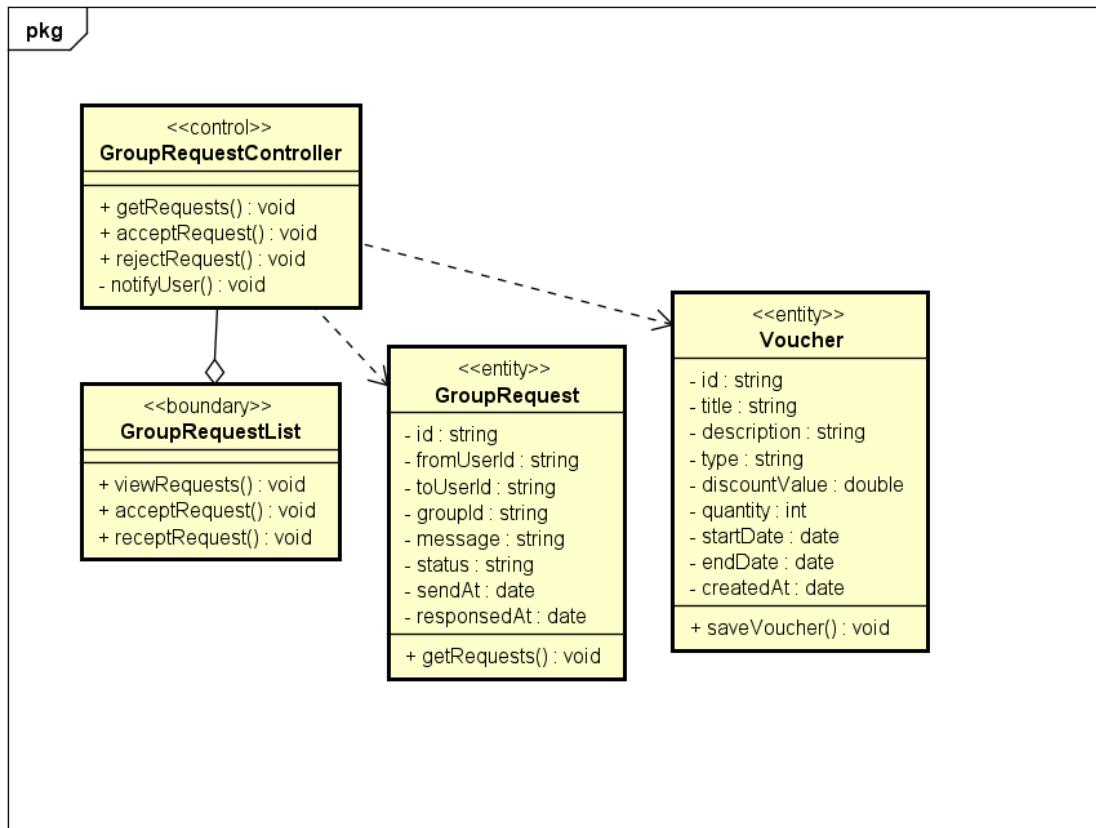
- viewFriendList(): Hiển thị danh sách các bạn bè của người dùng để họ có thể lựa chọn gửi lời mời
- inviteGroup(): Hiển thị button để mời 1 bạn bè vào group.

Để thể hiện rõ mối quan hệ giữa các lớp trên trong chức năng "Mời bạn bè tham gia nhóm", em có vẽ thêm biểu đồ tuần tự cho chức năng này như trong hình 4.14



Hình 4.14: Biểu đồ tuần tự cho chức năng "Mời bạn bè tham gia nhóm"

b. Thiết kế lớp cho chức năng "Duyệt lời mời tham gia nhóm"



Hình 4.15: Biểu đồ lớp cho chức năng "Duyệt lời mời tham gia nhóm"

Hình 4.15 thể hiện biểu đồ lớp cho chức năng "Duyệt lời mời tham gia nhóm".
Dựa vào hình 4.15, em xin phép trình bày thiết kế của một số lớp như sau:

Lớp **GroupRequest**: biểu diễn và lưu trữ các thông tin của request mời bạn bè tham gia nhóm săn sale mà mình mới tạo

- Thuộc tính:
 - id: Định danh của 1 request cụ thể trong hệ thống
 - fromUserId: Định danh của người gửi request, tức là chủ sở hữu nhóm
 - toUserId: Định danh của người nhận request, tức bạn bè của chủ
 - groupId: Định danh của nhóm săn sale do chủ nhóm tạo
 - message: nội dung lời nhắn kèm theo yêu cầu
 - status: trạng thái của request: pending, accept, hoặc reject
 - sendAt: thời gian gửi request
 - respondedAt: thời gian người nhận phản hồi yêu cầu
- Phương thức:
 - saveRequests(): Lưu thông tin của yêu cầu lên database.

Lớp **Voucher** biểu diễn và lưu trữ các thông tin của một voucher giảm giá được sử dụng trong hệ thống

- Thuộc tính:
 - id: Định danh của voucher trong hệ thống
 - title: Tên của voucher
 - description: Mô tả chi tiết thông tin của voucher
 - type: loại của voucher: thuộc 1 trong các loại: voucher freeship, voucher nhóm, voucher giảm giá cố định, voucher giảm theo category...
 - discountValue: giá trị mà voucher sẽ giảm: tùy loại voucher có thể là phần trăm đơn hàng hoặc một số tiền cố định
 - quantity: số lượng phát hành voucher đó
 - startDate: ngày voucher bắt đầu có hiệu lực
 - endDate: ngày voucher hết hiệu lực
 - createdAt: ngày tạo voucher
- Phương thức:
 - saveVoucher(): Lưu thông tin của voucher lên database.

Lớp **GroupRequestController**: Để xử lý logic chấp nhận hoặc từ chối các lời mời mà bạn bè mời mình tham gia nhóm, bao gồm các phương thức

- Phương thức:
 - getRequests: lấy tất cả các lời mời gửi tới người dùng hiện tại
 - acceptRequest: Chấp nhận lời mời tham gia nhóm của một người bạn
 - rejectRequest: từ chối lời mời tham gia nhóm của một người bạn
 - notifyUser: gửi thông báo đến người bạn rằng bạn đã từ chối lời mời tham gia nhóm của họ

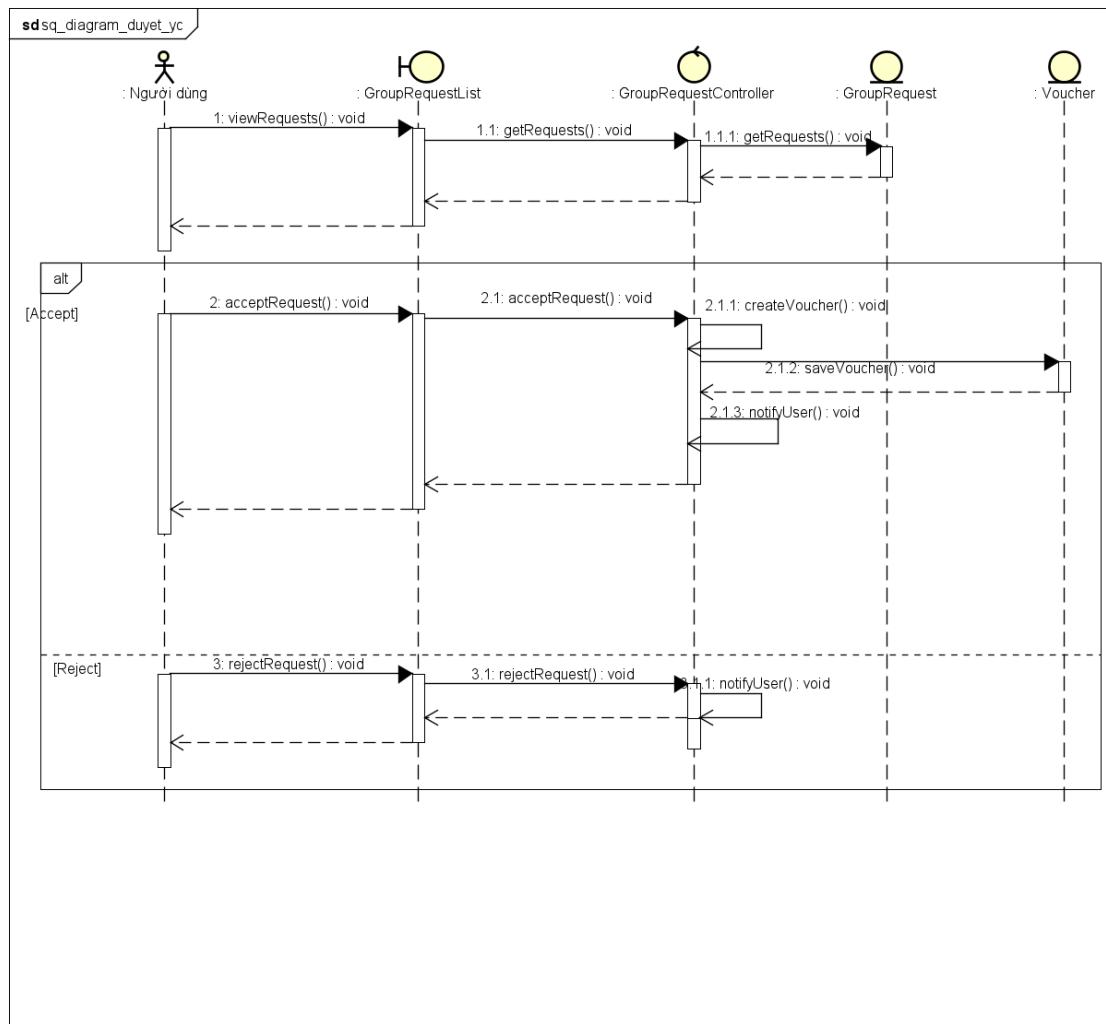
Lớp **GroupRequestList** tạo giao diện màn hình duyệt lời mời tham gia nhóm, gồm các phương thức tương tác với người dùng như:

- Phương thức:
 - viewRequests: Hiển thị danh sách các lời mời để người dùng xem xét và duyệt
 - acceptRequest: có button accept, người dùng click vào thì gọi phương thức chấp nhận lời mời

CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG

- rejectRequest: có button reject: người dùng click vào sẽ gọi các logic xử lý việc từ chối lời mời

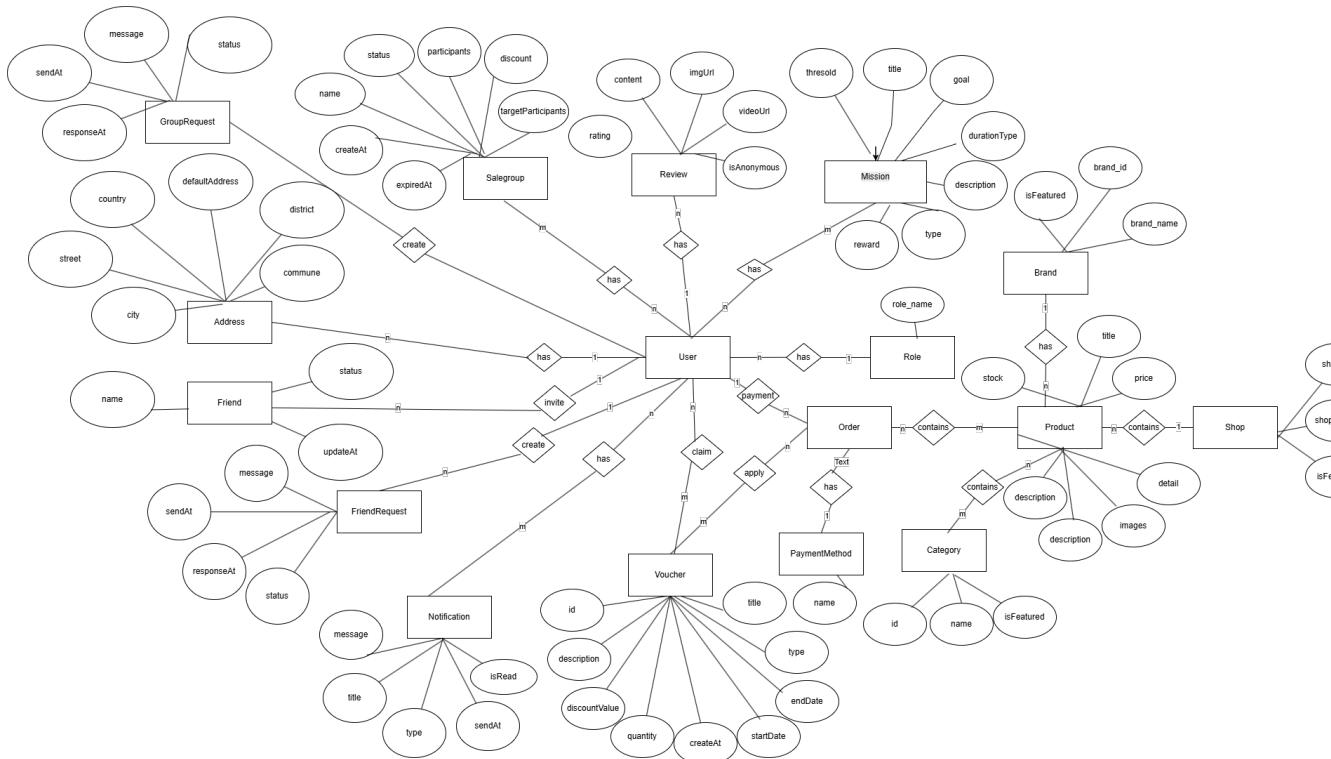
Để thể hiện rõ mối quan hệ giữa các lớp trên trong chức năng "Duyệt lời mời tham gia nhóm", em có vẽ thêm biểu đồ tuần tự cho chức năng này như trong hình 4.16



Hình 4.16: Biểu đồ tuần tự cho chức năng "Duyệt lời mời tham gia nhóm"

4.2.3 Thiết kế cơ sở dữ liệu

a, Biểu đồ thực thể liên kết



Hình 4.17: Biểu đồ thực thể liên kết

Hình 4.17 biểu diễn mô hình thực thể liên kết. Dựa vào mô hình thực thể liên kết này, em thiết kế cơ sở dữ liệu NoSql trên Firebase ở phần tiếp theo gồm 21 bảng.

b, Thiết kế cơ sở dữ liệu

- **Bảng brands:** Lưu thông tin các thương hiệu.
 - **Bảng categories:** Lưu thông tin các danh mục của các sản phẩm.
 - **Bảng shops:** Lưu thông tin các cửa hàng trong hệ thống
 - **Bảng products:** Lưu thông tin các sản phẩm.
 - **Bảng reviews:** Lưu thông tin các bình luận của người dùng về các sản phẩm.
 - **Bảng users:** Lưu thông tin về các người dùng trong hệ thống.
 - **Bảng daily_checkins:** Lưu lịch sử checkin hằng ngày của người dùng trong hệ thống.
 - **Bảng missions:** Lưu thông tin về các nhiệm vụ, thử thách cho người dùng để đổi thưởng.
 - **Bảng sale_groups:** Lưu thông tin về các nhóm săn sale mà người dùng tạo trong hệ thống.

CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG

- **Bảng vouchers:** Lưu thông tin về các voucher giảm giá trong hệ thống.
- **Bảng addresses:** Lưu các địa chỉ của người dùng trong hệ thống.
- **Bảng orders:** Lưu thông tin liên quan đến các đơn hàng.
- **Bảng user_vouchers:** Lưu thông tin người dùng nào đã thu thập và sử dụng voucher nào.
- **Bảng friend_requests:** Lưu thông tin về các lời mời kết bạn.
- **Bảng friends:** Lưu thông tin các mối quan hệ bạn bè của người dùng.
- **Bảng group_requests:** Lưu thông tin về lời mời tham gia nhóm săn sale.
- **Bảng notifications:** Lưu các thông báo khi có cập nhật mới.
- **Bảng reward_items:** Lưu thông tin người dùng quy đổi điểm thưởng lấy quà hoặc voucher.
- **Bảng user_missions:** Lưu thông tin người dùng đã thực hiện các nhiệm vụ nào trong hệ thống.
- **Bảng user_events:** Lưu thông tin các tương tác của người dùng với hệ thống.
- **Bảng chatbot:** Lưu thông tin các tương tác của người dùng với hệ thống.

Sau đây, em sẽ trình bày chi tiết thông tin của từng bảng:

Mô tả chi tiết bảng brands

Thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
id	string	NOT NULL	Định danh của thương hiệu
name	string	NOT NULL	Tên của thương hiệu
isFeatured	boolean	NULL	Đánh dấu một thương hiệu có là thương hiệu nổi bật hay không

Bảng 4.1: Chi tiết bảng brands

Mô tả chi tiết bảng categories

Thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
id	string	NOT NULL	Định danh của category
name	string	NOT NULL	Tên của category
isFeatured	boolean	NULL	Đánh dấu một category có là category nổi bật hay không

Bảng 4.2: Chi tiết bảng categories

Mô tả chi tiết bảng shops

CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG

Thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
id	string	NOT NULL	Định danh của shop
name	string	NOT NULL	Tên của shop
isFeatured	boolean	NONE	Đánh dấu một shop có là shop nổi bật hay không

Bảng 4.3: Chi tiết bảng shops

Mô tả chi tiết bảng products

Thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
id	string	NOT NULL	Định danh sản phẩm
title	string	NOT NULL	Tiêu đề của sản phẩm
description	string	NONE	Mô tả chi tiết sản phẩm, bao gồm đặc điểm, công dụng, màu sắc, chất liệu, trọng lượng, độ dài...
price	string	NOT NULL	Giá bán sản phẩm
stock	number	NOT NULL	Số lượng hàng tồn kho của sản phẩm
isFeatured	boolean	NONE	Xác định sản phẩm có được đánh dấu là nổi bật hay không
brandId	string	NONE	ID thương hiệu liên kết với sản phẩm
categoryIds	array of strings	NONE	Danh sách các ID danh mục mà sản phẩm thuộc về
shopId	string	NOT NULL	ID của cửa hàng bán sản phẩm
images	array of strings	NONE	Danh sách các URL hình ảnh của sản phẩm
details	map	NONE	Các chi tiết bổ sung về sản phẩm (ví dụ: chất liệu, nguồn gốc, thông số kỹ thuật...)
createdAt	timestamp	NOT NULL	Thời điểm tạo sản phẩm

Bảng 4.4: Chi tiết bảng products

Mô tả chi tiết bảng reviews

CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG

Thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
reviewId	string	NOT NULL	Định danh duy nhất của đánh giá
productId	string	NOT NULL	ID sản phẩm mà đánh giá này thuộc về
userId	string	NOT NULL	ID người dùng thực hiện đánh giá
comment	string	NULL	Nội dung bình luận của người dùng về sản phẩm
rating	number	NOT NULL	Điểm đánh giá của sản phẩm (ví dụ: từ 1 đến 5 sao)
imageUrls	array of strings	NULL	Danh sách các URL hình ảnh minh họa do người dùng đăng cùng đánh giá
videoUrls	array of strings	NULL	Danh sách các URL video minh họa (nếu có) đính kèm đánh giá
isAnonymous	boolean	NOT NULL	Xác định người dùng đánh giá có ẩn danh hay không
createdAt	timestamp	NOT NULL	Thời điểm đánh giá được tạo

Bảng 4.5: Chi tiết bảng reviews

Mô tả chi tiết bảng users

Thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
dateOfBirth	timestamp	NOT NULL	Ngày sinh của người dùng (theo múi giờ UTC+7)
email	string	NOT NULL	Địa chỉ email của người dùng
fcmToken	string	NULL	Mã token Firebase dùng để gửi thông báo đến thiết bị của người dùng
firstName	string	NOT NULL	Họ của người dùng
lastName	string	NOT NULL	Tên của người dùng
gender	string	NOT NULL	Giới tính của người dùng
phoneNumber	string	NOT NULL	Số điện thoại của người dùng
points	number	NOT NULL	Tổng điểm tích lũy của người dùng
profilePicture	string	NULL	Đường dẫn đến ảnh đại diện của người dùng (URL Firebase)
role	string	NOT NULL	Vai trò của người dùng trong hệ thống (ví dụ: admin, user)
status	string	NOT NULL	Trạng thái hoạt động của người dùng (ví dụ: Active, Blocked)
userName	string	NOT NULL	Tên người dùng (username) dùng để đăng nhập hoặc hiển thị

Bảng 4.6: Chi tiết bảng users

Mô tả chi tiết bảng daily_checkins

CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG

Thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
id	string	NOT NULL	Định danh duy nhất cho mỗi lần điểm danh hàng ngày
user_id	string	NOT NULL	ID của người dùng thực hiện điểm danh
check_in_date	timestamp	NOT NULL	Thời điểm người dùng thực hiện điểm danh (theo múi giờ UTC+7)
reward	number	NOT NULL	Số điểm thưởng mà người dùng nhận được khi điểm danh
streak_count	number	NOT NULL	Số ngày liên tiếp người dùng đã điểm danh

Bảng 4.7: Chi tiết bảng daily_checkins

Mô tả chi tiết bảng missions

Thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
id	string	NOT NULL	Định danh duy nhất cho nhiệm vụ
title	string	NOT NULL	Tiêu đề ngắn gọn mô tả nhiệm vụ
description	string	NOT NULL	Mô tả chi tiết về nhiệm vụ cần thực hiện
type	string	NOT NULL	Mô tả loại nhiệm vụ (ví dụ: write_review)
goal	number	NOT NULL	Số lượng hành động người dùng cần thực hiện để hoàn thành nhiệm vụ
threshold	number	NOT NULL	Yêu cầu về giá trị đơn hàng với một số nhiệm vụ liên quan đến đơn)
reward	number	NOT NULL	Số điểm thưởng người dùng nhận được sau khi hoàn thành nhiệm vụ
durationInHours	number	NOT NULL	Khoảng thời gian hiệu lực của nhiệm vụ (tính bằng giờ; 0 nếu không giới hạn)
durationType	string	NOT NULL	Kiểu giới hạn thời gian (ví dụ: daily, weekly)
startTime	timestamp	NULL	Thời gian bắt đầu hiệu lực của nhiệm vụ (nếu có)
endTime	timestamp	NULL	Thời gian kết thúc hiệu lực của nhiệm vụ (nếu có)
imgUrl	string	NULL	Đường dẫn ảnh minh họa cho nhiệm vụ

Bảng 4.8: Chi tiết bảng missions

Mô tả chi tiết bảng sale_groups

CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG

Thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
id	string	NOT NULL	Định danh duy nhất cho nhóm mua chung
name	string	NOT NULL	Tên nhóm mua chung
brandId	string	NOT NULL	ID thương hiệu mà các sản phẩm sẽ được áp dụng cho nhóm chung
categoryId	string	NULL	ID danh mục mà các sản phẩm sẽ được áp dụng cho nhóm chung
shopId	string	NULL	ID cửa hàng mà các sản phẩm sẽ được áp dụng cho nhóm chung
creatorId	string	NOT NULL	ID người tạo nhóm
participants	array of strings	NOT NULL	Danh sách ID người tham gia nhóm mua chung
currentParticipants	number	NOT NULL	Số lượng người đã tham gia hiện tại
targetParticipants	number	NOT NULL	Số lượng người tối thiểu để kích hoạt ưu đãi
discount	number	NOT NULL	Mức giảm giá được áp dụng khi nhóm đủ số lượng
selectedObjectName	string	NOT NULL	Tên sản phẩm hoặc thương hiệu được áp dụng mua chung
status	string	NOT NULL	Trạng thái của nhóm (ví dụ: pending, completed)
createdAt	string	NOT NULL	Thời điểm nhóm được tạo
expiresAt	string	NOT NULL	Thời điểm nhóm hết hạn nếu chưa đủ người tham gia

Bảng 4.9: Chi tiết bảng sale_groups

Mô tả chi tiết bảng vouchers

Thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
id	string	NOT NULL	Định danh duy nhất của voucher
title	string	NOT NULL	Tiêu đề của voucher
description	string	NULL	Mô tả chi tiết về voucher
type	string	NOT NULL	Loại voucher (ví dụ: fixed_discount)
discountValue	number	NOT NULL	Giá trị giảm giá của voucher (ví dụ: 50000 VNĐ)
maxDiscount	number	NULL	Giá trị giảm tối đa cho đơn hàng
minimumOrder	number	NULL	Giá trị đơn hàng tối thiểu để sử dụng voucher

CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG

Thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
quantity	number	NOT NULL	Tổng số lượng voucher được phát hành
remainingQuantity	number	NOT NULL	Số lượng voucher còn lại có thể sử dụng
requiredPoints	number	NULL	Số điểm cần để đổi voucher (nếu có)
isRedeemableByPoint	boolean	NOT NULL	Xác định voucher có thể đổi bằng điểm thường hay không
isActive	boolean	NOT NULL	Trạng thái hoạt động của voucher
startDate	timestamp	NOT NULL	Ngày bắt đầu hiệu lực của voucher
endDate	timestamp	NOT NULL	Ngày kết thúc hiệu lực của voucher
createdAt	timestamp	NOT NULL	Thời điểm tạo voucher
updatedAt	timestamp	NULL	Thời điểm cập nhật gần nhất của voucher

Bảng 4.10: Chi tiết bảng vouchers

Mô tả chi tiết bảng addresses

Thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
Id	string	NOT NULL	Định danh duy nhất của địa chỉ
Name	string	NOT NULL	Tên người nhận hàng
PhoneNumber	string	NOT NULL	Số điện thoại người nhận
Street	string	NOT NULL	Tên đường hoặc thông tin chi tiết về địa chỉ cụ thể
Commune	string	NOT NULL	Phường/xã nơi địa chỉ thuộc về
District	string	NOT NULL	Quận/huyện nơi địa chỉ thuộc về
City	string	NOT NULL	Thành phố nơi địa chỉ nằm trong
Country	string	NOT NULL	Quốc gia nơi địa chỉ thuộc về
SelectedAddress	boolean	NOT NULL	Địa chỉ này có đang được chọn làm địa chỉ giao hàng chính hay không
DateTime	timestamp	NOT NULL	Thời điểm địa chỉ được tạo hoặc cập nhật

Bảng 4.11: Chi tiết bảng addresses

Mô tả chi tiết bảng orders

CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG

Thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
id	string	NOT NULL	Mã định danh duy nhất của đơn hàng
orderDate	timestamp	NOT NULL	Thời điểm người dùng đặt hàng
deliveryDate	timestamp	NOT NULL	Ngày giao hàng dự kiến
updatedAt	timestamp	NOT NULL	Thời điểm cuối cùng đơn hàng được cập nhật
status	string	NOT NULL	Trạng thái đơn hàng (ví dụ: delivered, pending)
paymentMethod	string	NOT NULL	Phương thức thanh toán (ví dụ: Paypal, Momo, COD)
totalAmount	number	NOT NULL	Tổng giá trị đơn hàng
items	array of maps	NOT NULL	Danh sách các sản phẩm trong đơn hàng

Bảng 4.12: Chi tiết bảng orders

Mô tả chi tiết bảng claimed_vouchers

Thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
claimed_at	timestamp	NOT NULL	Thời điểm voucher được người dùng nhận
is_used	boolean	NOT NULL	Trạng thái voucher đã được sử dụng hay chưa
used_at	timestamp	NULLABLE	Thời điểm voucher được sử dụng
voucher_id	string	NOT NULL	Mã định danh của voucher tương ứng

Bảng 4.13: Chi tiết bảng claimed_vouchers

Mô tả chi tiết bảng friend_requests

Thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
fromUserId	string	NOT NULL	ID của người gửi lời mời kết bạn
toUserId	string	NOT NULL	ID của người nhận lời mời kết bạn
message	string	NULLABLE	Nội dung lời nhắn kèm theo lời mời kết bạn
sentAt	timestamp	NOT NULL	Thời điểm gửi lời mời kết bạn
respondedAt	timestamp	NULLABLE	Thời điểm người nhận phản hồi lời mời kết bạn
status	string	NOT NULL	Trạng thái của lời mời (ví dụ: pending, accepted, rejected)

Bảng 4.14: Chi tiết bảng friend_requests

CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG

Mô tả chi tiết bảng friends

Thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
id	string	NOT NULL	Mã định danh của mỗi quan hệ bạn bè
friendId	string	NOT NULL	ID của người bạn trong mỗi quan hệ
friendName	string	NOT NULL	Tên hiển thị của người bạn
status	string	NOT NULL	Trạng thái của mỗi quan hệ bạn bè (ví dụ: active, blocked)
createdAt	string	NOT NULL	Thời điểm mỗi quan hệ được tạo
updatedAt	string	NOT NULL	Thời điểm mỗi quan hệ được cập nhật gần nhất
acceptedAt	string	NOT NULL	Thời điểm lời mời kết bạn được chấp nhận

Bảng 4.15: Chi tiết bảng friends

Mô tả chi tiết bảng group_requests

Thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
fromUserId	string	NOT NULL	ID của người gửi lời mời tham gia nhóm
toUserId	string	NOT NULL	ID của người nhận lời mời tham gia nhóm
groupId	string	NOT NULL	ID của nhóm mà người nhận được mời tham gia
message	string	NULLABLE	Nội dung lời nhắn đi kèm lời mời vào nhóm
sentAt	timestamp	NOT NULL	Thời điểm gửi lời mời tham gia nhóm
respondedAt	timestamp	NULLABLE	Thời điểm người nhận phản hồi lời mời
status	string	NOT NULL	Trạng thái lời mời (ví dụ: pending, accepted, rejected)

Bảng 4.16: Chi tiết bảng group_requests

Mô tả chi tiết bảng notifications

CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG

Thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
id	string	NOT NULL	Mã định danh duy nhất của thông báo
title	string	NOT NULL	Tiêu đề của thông báo
message	string	NOT NULL	Nội dung chính của thông báo
type	string	NOT NULL	Loại thông báo (ví dụ: voucher, system, order)
read	boolean	NOT NULL	Trạng thái đã đọc của thông báo (true/false)
timestamp	timestamp	NOT NULL	Thời điểm thông báo được gửi

Bảng 4.17: Chi tiết bảng notifications

Mô tả chi tiết bảng reward_items

Thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
points_used	number	NOT NULL	Số điểm đã sử dụng để đổi phần thưởng
redeemed_at	timestamp	NOT NULL	Thời điểm người dùng đổi phần thưởng
voucher_id	string	NOT NULL	ID của voucher tương ứng với phần thưởng đã đổi

Bảng 4.18: Chi tiết bảng reward_items

Mô tả chi tiết bảng user_missions

Thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
missionId	string	NOT NULL	ID nhiệm vụ được gán cho người dùng
type	string	NOT NULL	Loại nhiệm vụ (ví dụ: quickOrder)
status	string	NOT NULL	Trạng thái nhiệm vụ (ví dụ: in-Progress, completed, expired)
progress	number	NOT NULL	Tiến độ hoàn thành nhiệm vụ
startedAt	string	NOT NULL	Thời điểm bắt đầu nhiệm vụ
expiredAt	string	NOT NULL	Thời điểm nhiệm vụ hết hạn
completedAt	timestamp	NULLABLE	Thời điểm hoàn thành nhiệm vụ (nếu đã hoàn thành)

Bảng 4.19: Chi tiết bảng user_missions

Mô tả chi tiết bảng user_events

CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG

Thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
event	string	NOT NULL	Tên sự kiện do người dùng thực hiện (ví dụ: favourite_product)
product_id	string	NOT NULL	ID của sản phẩm liên quan đến sự kiện
session_id	string	NOT NULL	Mã phiên người dùng để theo dõi hành vi theo từng phiên duyệt
timestamp	timestamp	NOT NULL	Thời điểm sự kiện diễn ra
user_id	string	NOT NULL	ID người dùng thực hiện sự kiện

Bảng 4.20: Chi tiết bảng user_events

Mô tả chi tiết bảng chatbot

Thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
createdAt	string (ISO 8601)	NOT NULL	Thời gian tin nhắn được tạo, định dạng chuỗi thời gian chuẩn ISO 8601 (ví dụ: "2025-06-12T23:44:26.821466")
isUser	boolean	NOT NULL	Cờ xác định xem tin nhắn có phải do người dùng gửi không. <code>true</code> nếu do người dùng gửi, <code>false</code> nếu do chatbot gửi.
text	string	NOT NULL	Nội dung tin nhắn được gửi trong đoạn hội thoại giữa người dùng và chatbot. Có thể chứa văn bản, biểu tượng hoặc đề xuất sản phẩm.

Bảng 4.21: Chi tiết bảng chatbot

4.3 Xây dựng ứng dụng

4.3.1 Thư viện và công cụ sử dụng

Bảng 4.22 liệt kê các công cụ, ngôn ngữ lập trình, API, thư viện, IDE, công cụ kiểm thử, v.v. mà em đã sử dụng để phát triển ứng dụng.

CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG

Mục đích	Công cụ	Địa chỉ URL
IDE lập trình web front-end	Visual Studio Code	https://code.visualstudio.com/
IDE lập trình web backend	IntelliJ IDEA	https://www.jetbrains.com/idea/#
IDE lập trình mobile	Android Studio	https://developer.android.com/studio
Framework Xây dựng app mobile	Flutter 3.24.4	https://flutter.dev/
Framework xây dựng web front-end	ReactJs	https://react.dev/
Framework xây dựng web backend	Springboot	https://spring.io/projects/spring-boot
Ngôn ngữ lập trình app mobile	Dart	https://dart.dev/
Kiểm thử API	Postman	https://www.postman.com/
Lưu trữ database	Firebase Firestore	https://firebase.google.com/docs/firestore?hl=en
Lưu trữ hình ảnh	Firebase Cloud Messaging	https://firebase.google.com/docs/storage?hl=en
Gửi thông báo real-time	Firebase Storage	https://firebase.google.com/docs/cloud-messaging?hl=vi
Xác thực người dùng	Firebase Authentication	https://firebase.google.com/docs/auth?hl=vi
Vẽ biểu đồ UML	Astah UML	https://astah.net/products/astah-uml/

Bảng 4.22: Danh sách thư viện và công cụ sử dụng

4.3.2 Kết quả đạt được

Sau quá trình nghiên cứu và phát triển, ứng dụng đã có thể đáp ứng được những yêu cầu cơ bản mà em đã đưa ra với đầy đủ các chức năng cần thiết dành cho các nhà quản lý khi thực hiện công việc quản lý một câu lạc bộ bóng chày. Với kiến trúc Client-Server, ứng dụng đã được phát triển với hai thành phần riêng biệt là Client và Server, và có thể trao đổi dữ liệu với nhau qua API.

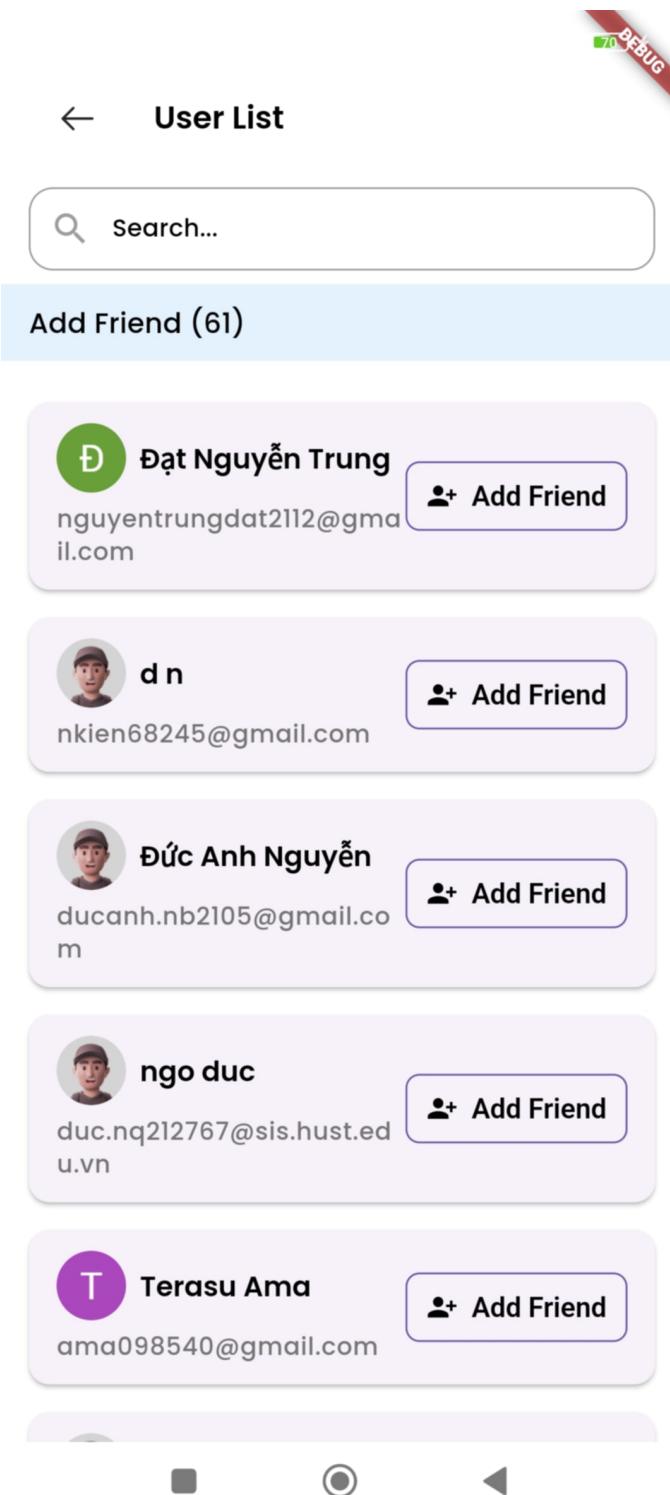
Thống kê các thông tin về ứng dụng

Thông tin	Thông kê	Mô tả
Dung lượng ứng dụng mobile	2.02 GB	Trong đó có 2 MB mã nguồn và 2 GB thư viện được cài đặt
Dung lượng ứng dụng phía web backend	1.07 MB	Bao gồm toàn bộ mã nguồn
Dung lượng file APK sau khi được đóng gói	229 MB	File APK được build bởi Android Studio
Dung lượng ứng dụng phía web frontend	316 KB	Bao gồm toàn bộ mã nguồn
Số dòng code web backend	4362	Bao gồm toàn bộ mã nguồn
Số dòng code web frontend	5294	Bao gồm toàn bộ mã nguồn

Bảng 4.23: Bảng thống kê các thông tin về ứng dụng

4.3.3 Minh họa các chức năng chính

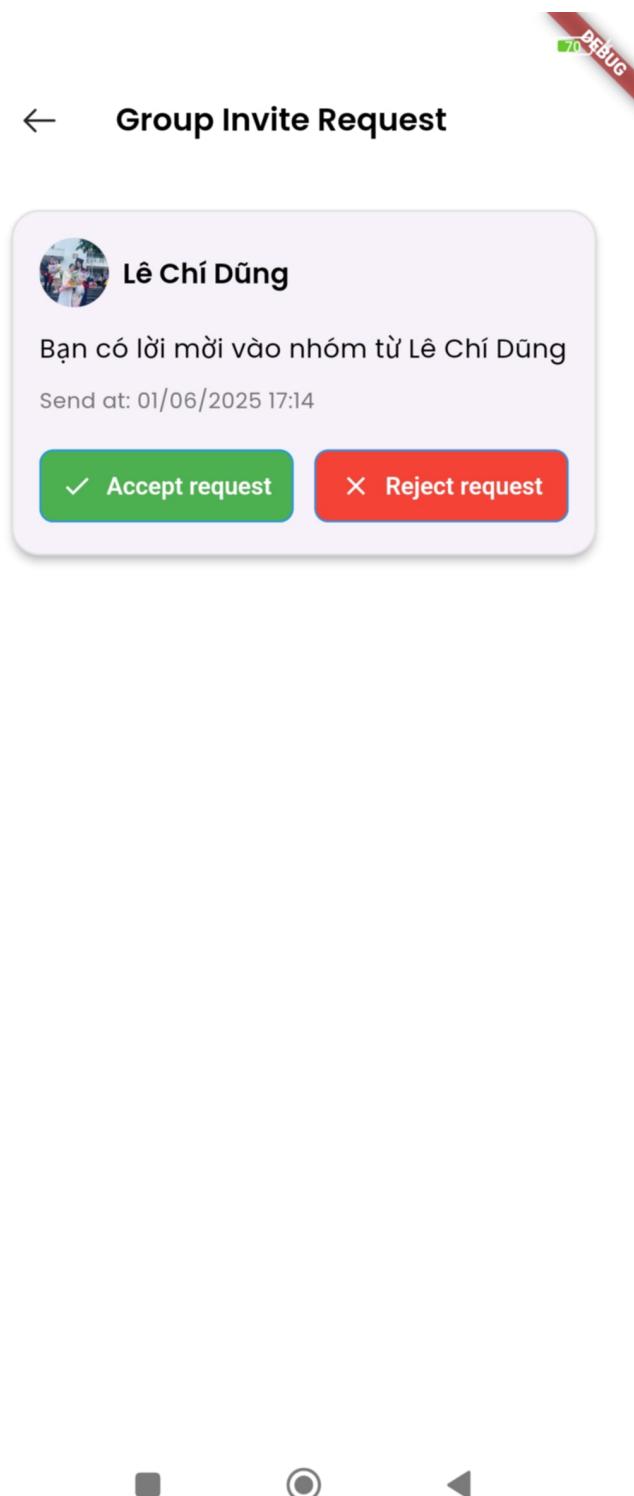
Màn hình thêm bạn bè



Hình 4.18: Màn hình thêm bạn bè

Hình 4.18 hiển thị màn hình thêm bạn bè. Màn hình sẽ hiển thị tổng số người dùng trong hệ thống (mà người đó chưa kết bạn), dưới là danh sách các người dùng đi kèm với nút gửi lời mời kết bạn. Ở phía trên có thanh search để tìm kiếm theo tên hoặc email người muốn kết bạn

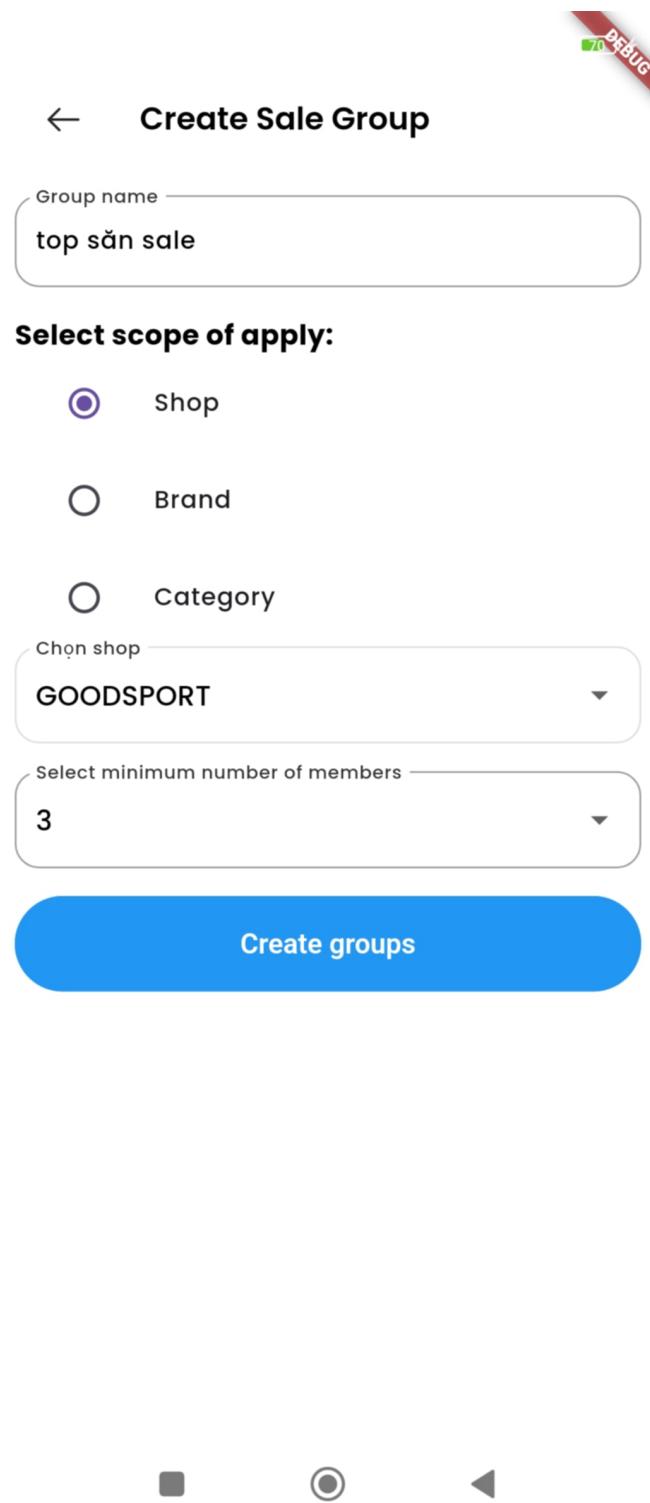
Màn hình chấp nhận lời mời tham gia nhóm



Hình 4.19: Màn hình chấp nhận lời mời tham gia nhóm

Hình 4.19 hiển thị màn hình chấp nhận lời mời tham gia nhóm từ bạn bè. Màn hình sẽ hiển thị toàn bộ các lời mời từ bạn bè người đó tham gia nhóm săn sale. Với mỗi lời mời người đó có thể chấp nhận hoặc từ chối lời mời.

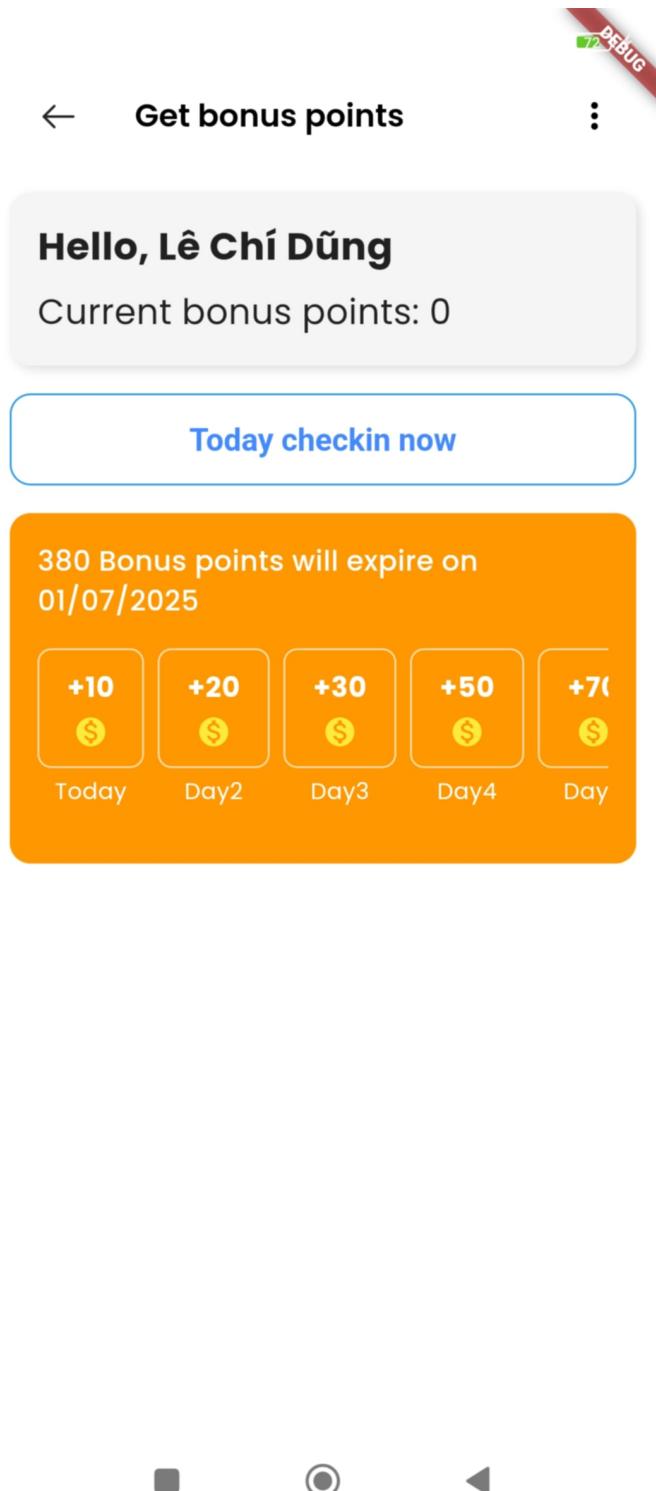
Màn hình tạo nhóm săn sale



Hình 4.20: Màn hình tạo nhóm săn sale

Hình 4.20 hiển thị form để người dùng tạo 1 nhóm săn sale mới. Người dùng cần nhập vào tên nhóm, chọn phạm vi mà khi nhóm hoàn thành thì các sản phẩm trong phạm vi sẽ được giảm giá, bao gồm các sản phẩm của 1 cửa hàng, 1 thương hiệu hoặc 1 danh mục cụ thể. Ngoài ra người dùng chọn giới hạn số người của nhóm, tùy thuộc vào phạm vi và số lượng khác nhau hệ thống sẽ tự quyết định mức giảm giá ưu đãi mà người dùng sẽ nhận được khi nhóm săn sale tập hợp đủ trước deadline.

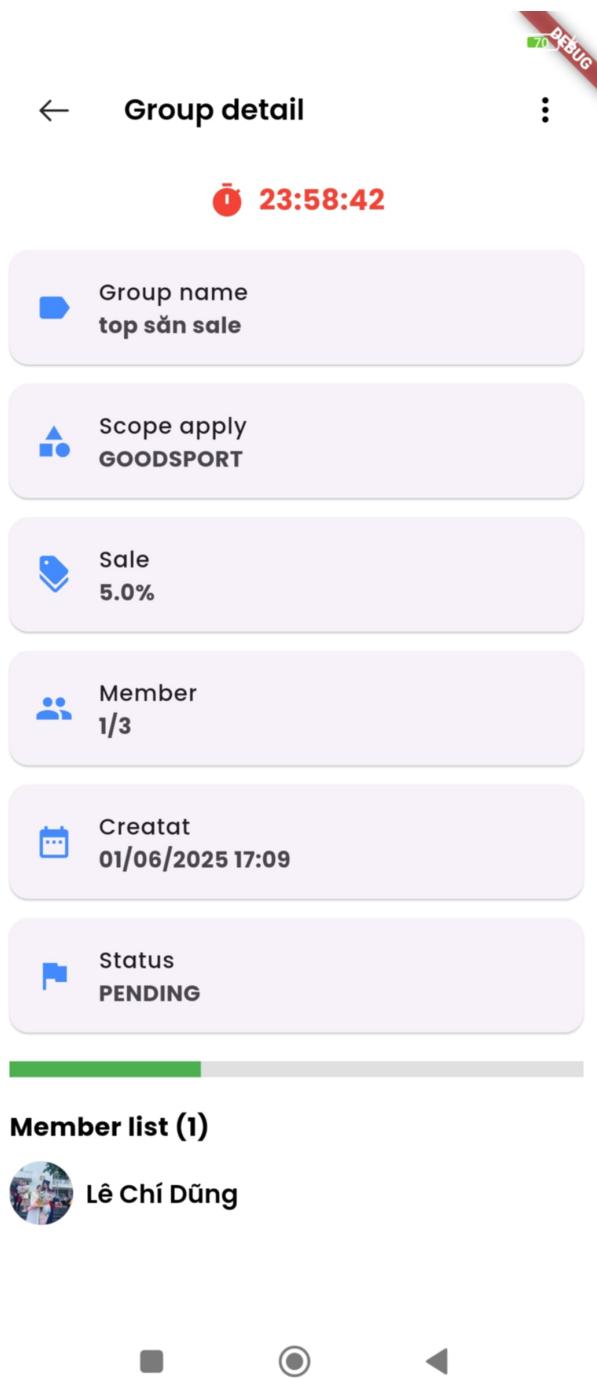
Màn hình điểm danh hàng ngày



Hình 4.21: Màn hình điểm danh hàng ngày

Hình 4.21 hiển thị màn hình điểm danh hàng ngày. Phía trên hiển thị tên người dùng, điểm thưởng hiện tại, ở dưới có nút để họ có thể checkin mỗi ngày một lần, và bảng theo dõi quá trình điểm danh của mình trong một tuần gần nhất và điểm thưởng nhận được với chuỗi ngày liên tiếp tương ứng

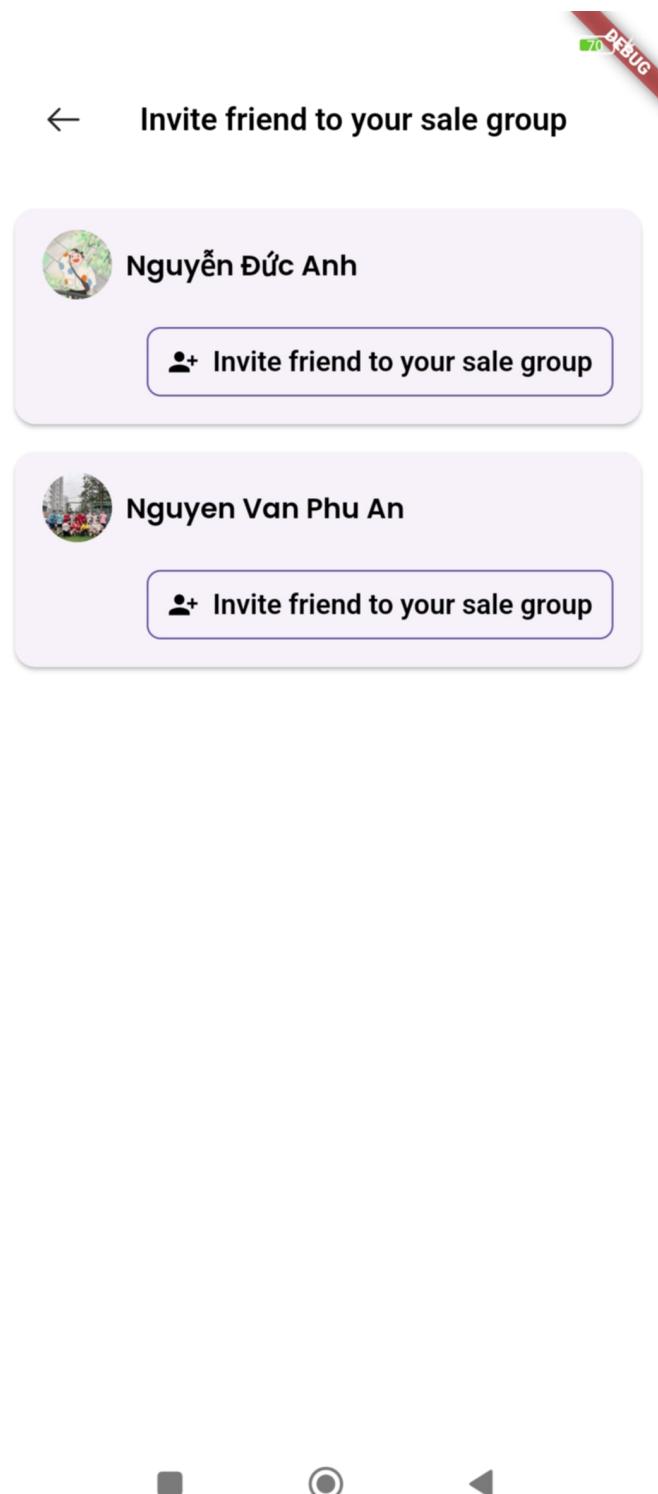
Màn hình chi tiết nhóm săn sale



Hình 4.22: Màn hình chi tiết nhóm săn sale

Hình 4.22 hiển thị màn hình chi tiết nhóm săn sale. Sau khi 1 nhóm săn sale được tạo thành công, màn hình này sẽ hiển thị đồng hồ đếm ngược thời gian còn lại cần phải tập hợp đủ người để kích hoạt nhóm nhận voucher. Ở phía góc phải của thanh navbar có nút điều hướng để có thể gọi đến màn hình mời thêm bạn bè của mình tham gia nhóm cho đủ thành viên. Trung tâm hiển thị các thông tin về nhóm gồm tên, phạm vi áp dụng, mức giảm giá, số thành viên hiện tại trên tổng thành viên tối thiểu cần đạt, thời gian tạo, trạng thái nhóm, thanh tiến trình và tên các thành viên đã tham gia.

Màn hình mời bạn bè tham gia nhóm săn sale



Hình 4.23: Màn hình mời bạn bè tham gia nhóm săn sale

Hình 4.23 hiển thị màn hình mời bạn bè tham gia nhóm. Nó hiển thị danh sách bạn bè của người dùng hiện tại kèm nút invite để có thể mời 1 hoặc nhiều người bạn vào nhóm săn sale của mình

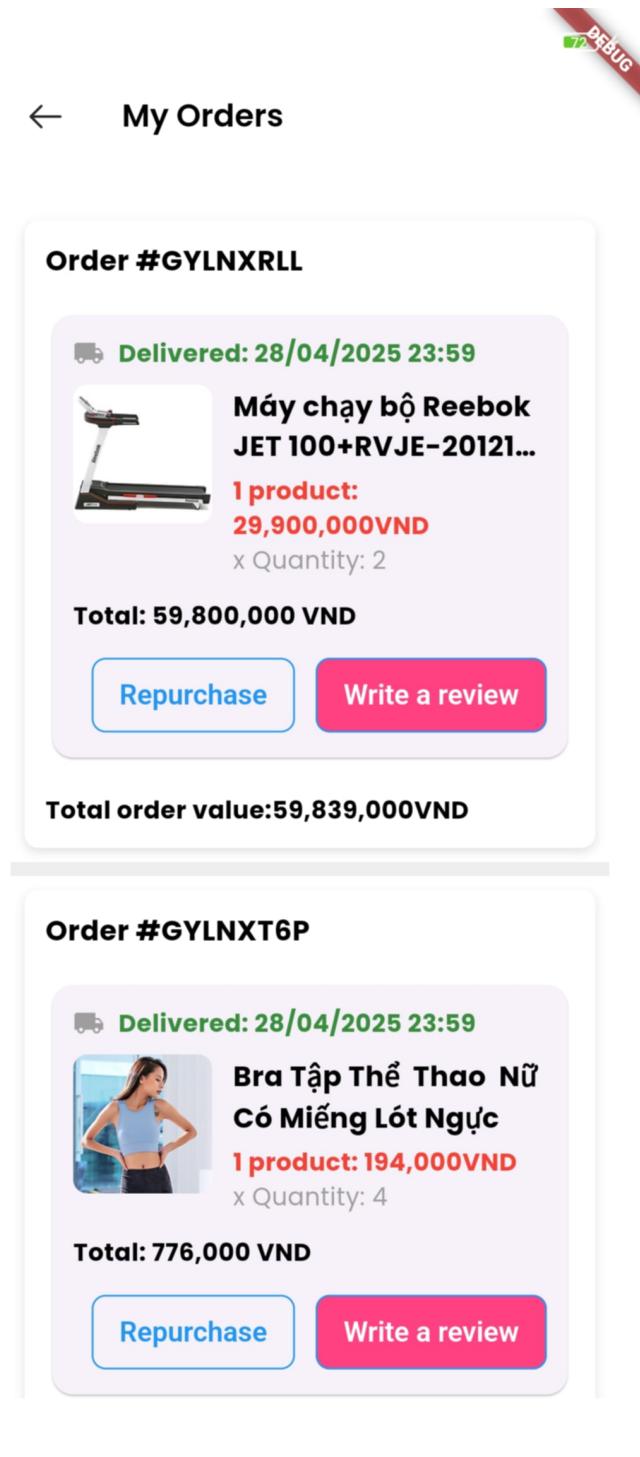
Màn hình nhiệm vụ chi tiết



Hình 4.24: Màn hình nhiệm vụ chi tiết

Hình 4.24 hiển thị màn hình nhiệm vụ chi tiết. Khi bấm vào 1 nhiệm vụ cụ thể sẽ điều hướng đến màn hình này. Màn hình hiển thị ảnh, tên nhiệm vụ, mô tả cụ thể về nhiệm vụ, mức điểm thưởng sẽ nhận được khi hoàn thành kèm thời hạn cần hoàn thành. Ở trung tâm là thanh trạng thái thể hiện tiến độ hoàn thành hiện tại và trạng thái của nhiệm vụ.

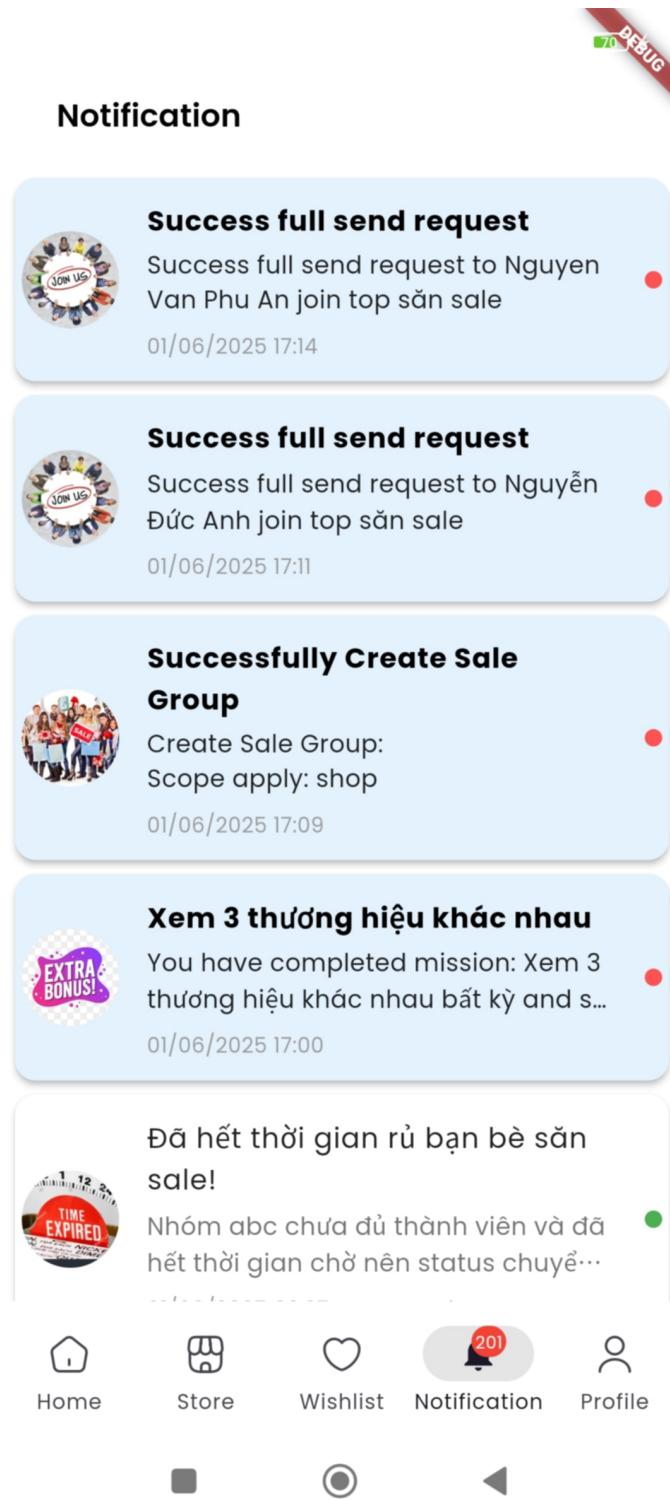
Màn hình đơn hàng của tôi



Hình 4.25: Màn hình đơn hàng của tôi

Hình 4.25 hiển thị màn hình đơn hàng của tôi. Màn hình này lưu lại lịch sử các đơn hàng mà người dùng đó đã mua, gồm mã đơn hàng, trạng thái, thời gian giao hàng, các sản phẩm đã mua trong đơn, tổng giá trị đơn trước và sau khi áp dụng các voucher. Ngoài ra ở mỗi đơn hàng có nút mua lại để thêm nhanh vào giỏ hàng và nút viết bình luận cho các sản phẩm đã mua.

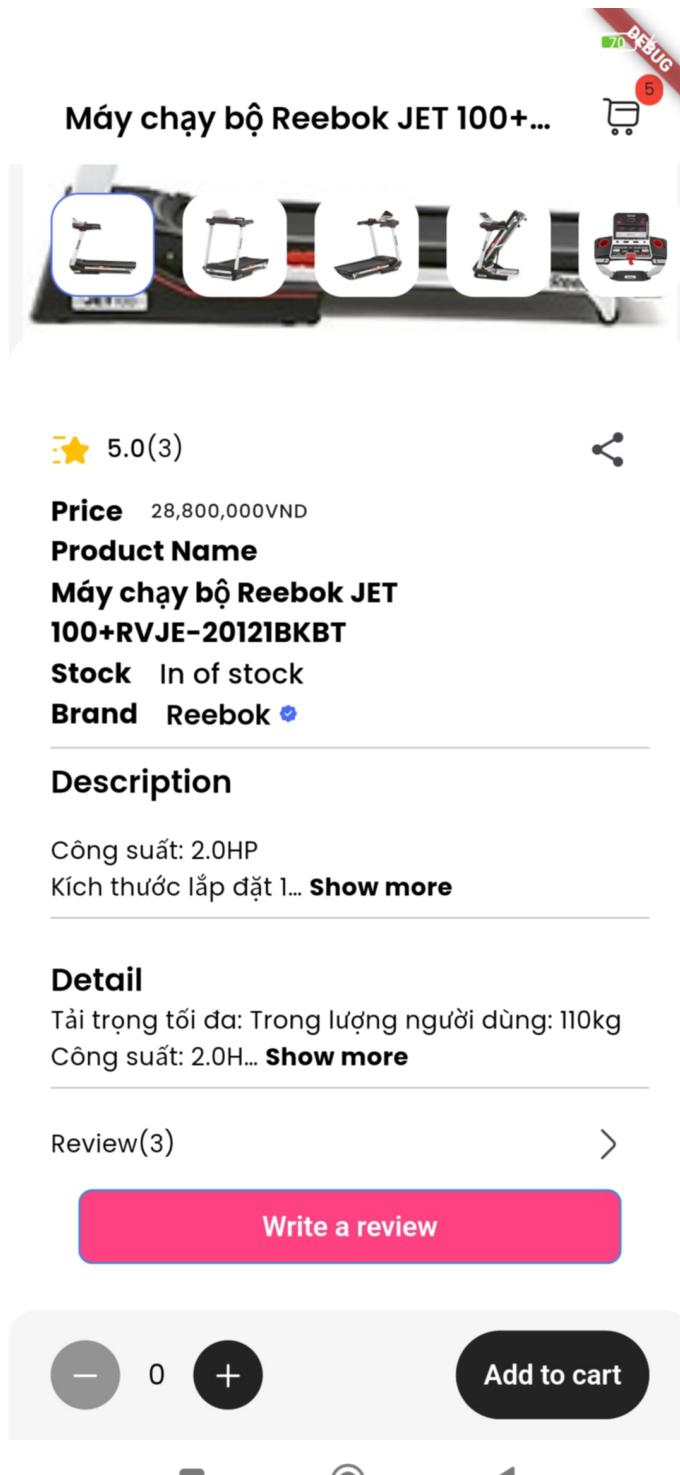
Màn hình thông báo



Hình 4.26: Màn hình thông báo

Hình 4.26 hiển thị màn hình thông báo. Màn hình này được điều hướng từ biểu tượng thông báo ở bottom nav bar, hiển thị tổng số thông báo chưa đọc. Ở màn hình này, hiển thị toàn bộ các thông báo được hệ thống gửi, gồm tiêu đề, nội dung, thời gian gửi và trạng thái đọc hay chưa đọc của người dùng.

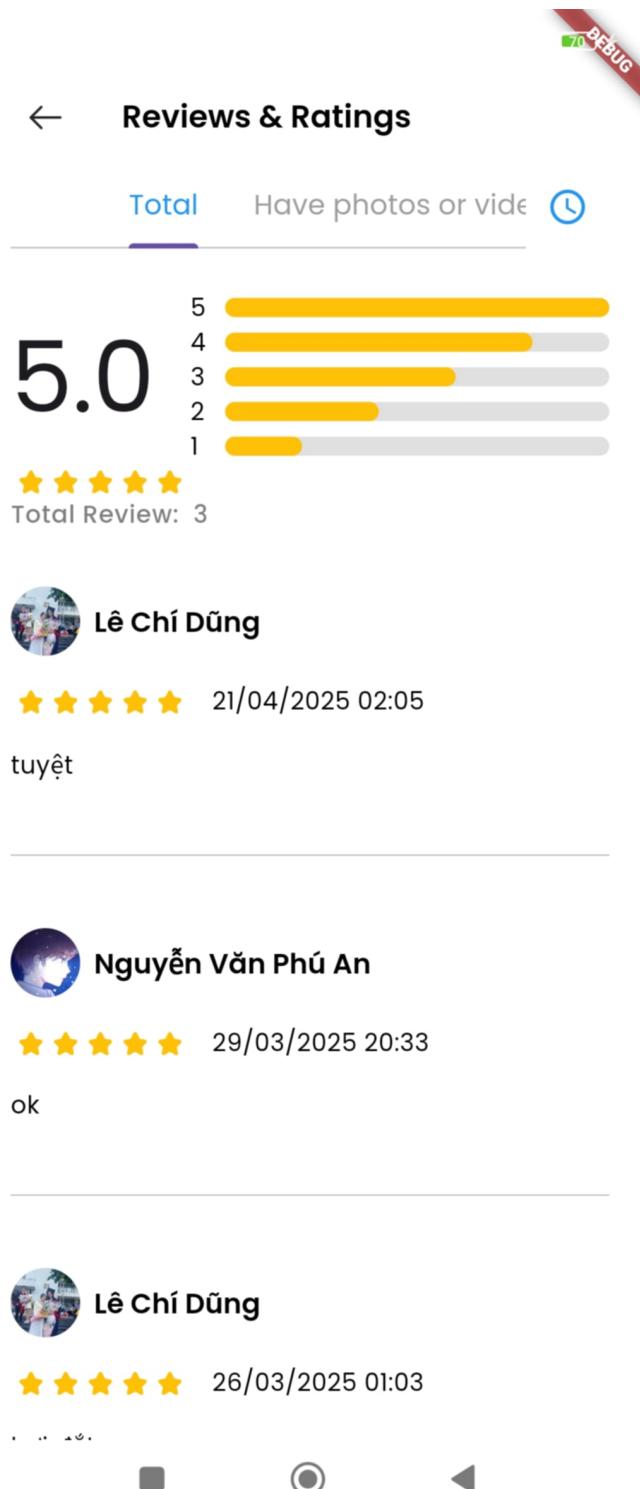
Màn hình sản phẩm chi tiết

**Hình 4.27:** Màn hình sản phẩm chi tiết

Hình 4.27 hiển thị màn hình sản phẩm chi tiết. Màn hình này hiển thị tên sản phẩm, các ảnh liên quan đến sản phẩm, có thể phóng to xem từng ảnh, số điểm đánh giá trung bình của người dùng, giá, thương hiệu, mô tả, chi tiết sản phẩm, tổng số bình luận về sản phẩm đó. Ngoài ra màn hình có các nút để điều hướng đến giỏ hàng, điều hướng đến màn hình xem chi tiết các review, màn hình viết bình luận mới, nút chọn số lượng muốn mua và thêm vào giỏ hàng

CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG

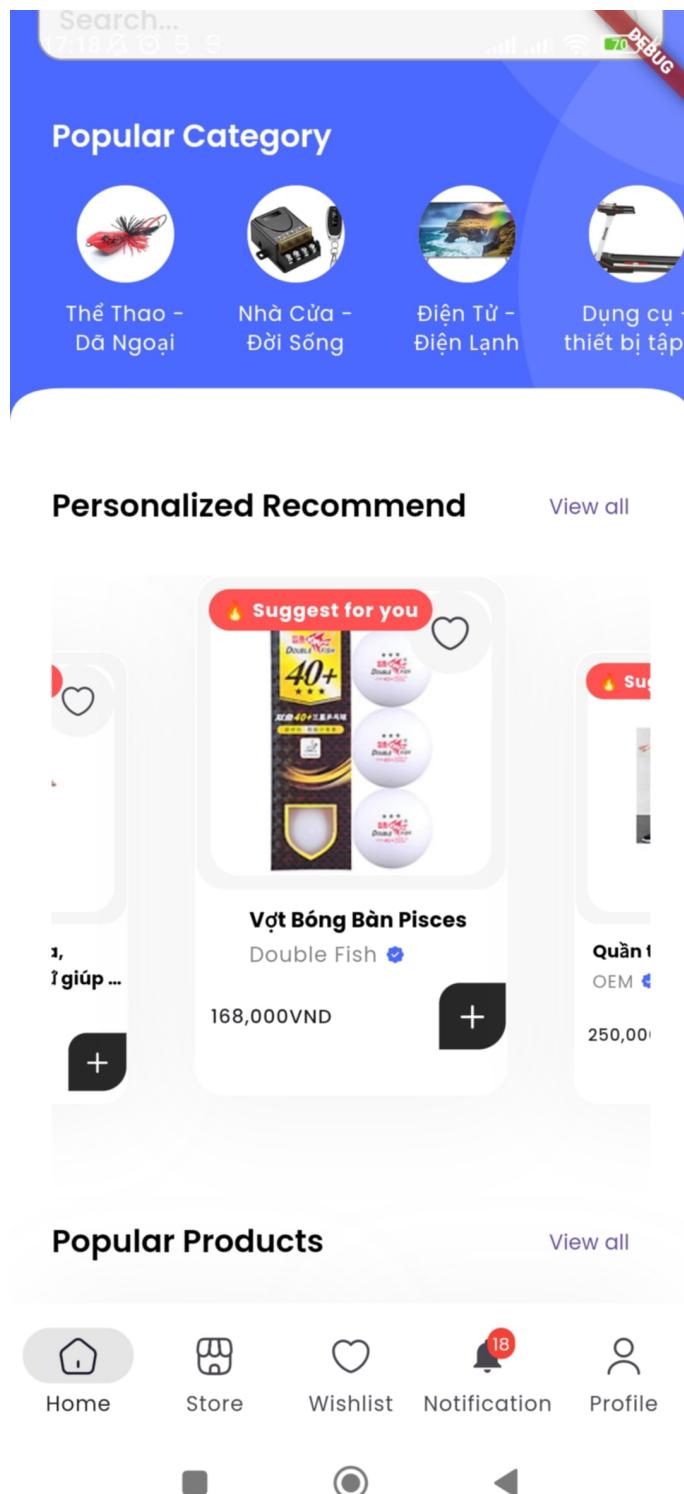
Màn hình xem bình luận



Hình 4.28: Màn hình xem bình luận

Hình 4.28 hiển thị màn hình xem bình luận. Màn hình này có chia ra các tab, lọc các bình luận theo các mục: xem tất cả, xem các bình luận có ảnh/video, và lọc theo các đánh giá từ 1 đến 5 sao. Với mỗi tab, giao diện sẽ hiển thị điểm trung bình, tổng bình luận, và chi tiết mỗi bình luận gồm người đăng, điểm, thời gian, bình luận, ảnh hoặc video đính kèm.

Màn hình gợi ý sản phẩm

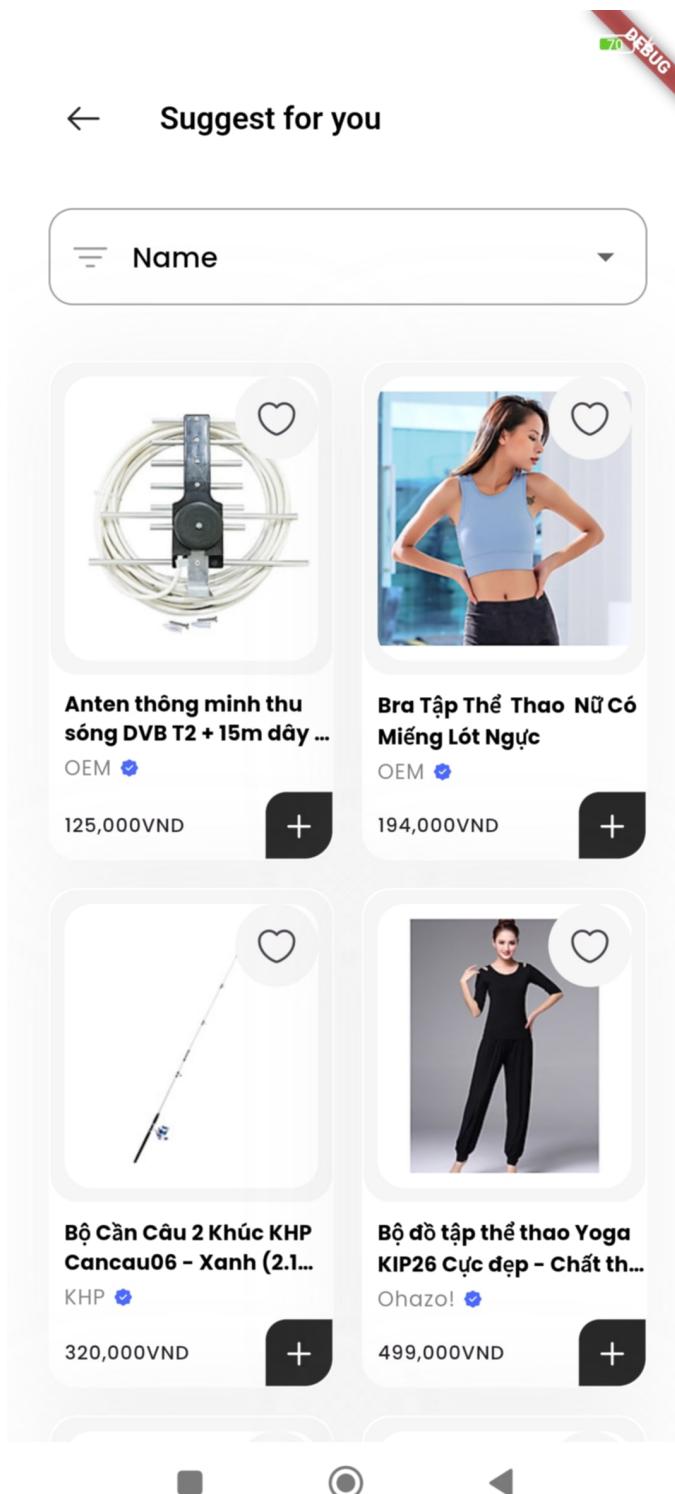


Hình 4.29: Màn hình gợi ý sản phẩm

Hình 4.29 hiển thị màn hình gợi ý sản phẩm. Phần gợi ý sản phẩm sẽ nằm ở chính giữa của màn hình home, dưới dạng sidebar sẽ tự động trượt ngang từ trái qua phải, hiển thị thông tin các sản phẩm được gợi ý từ hệ thống gợi ý dùng học máy được cá nhân hóa dành riêng cho mỗi người dùng. Bên cạnh có nút viewall để xem toàn bộ các sản phẩm được gợi ý riêng cho người dùng.

CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG

Màn hình danh sách sản phẩm



Hình 4.30: Màn hình anh sách sản phẩm

Hình 4.30 hiển thị màn hình danh sách sản phẩm . Ở phía trên có bộ lọc để sắp xếp sản phẩm theo thứ tự alphabet hoặc giá giảm dần, phía dưới hiển thị danh sách sản phẩm dạng lưới mỗi hàng 2 sản phẩm, cứ mỗi 20 sản phẩm ở cuối trang sẽ có nút loadmore để người dùng lựa chọn có xem thêm các sản phẩm hay không.

4.4 Kiểm thử

4.4.1 Kiểm thử tương thích

Để tiến hành kiểm thử độ tương thích của ứng dụng trên các thiết bị khác nhau, em đã tiến hành cài đặt ứng dụng trên ba thiết bị là Redmi Note 12 Pro, Samsung S20 FE và Iphone11. Trong đó, Redmi Note 12 Pro là thiết bị cá nhân của em, còn Samsung S20 FE và Iphone11 em deploy file apk lên firebase và chia sẻ để bạn bè dùng thử và trải nghiệm. Kết quả thu được được em trình bày trong bảng 4.24

Thiết bị	Thông số kỹ thuật	Giao diện	Chức năng
Redmi Note 12 Pro	Màn hình: 6.67" Hệ điều hành: Android 13	Đạt	Đạt
Samsung S20 FE	Màn hình: 6.5" Hệ điều hành: Android 13	Đạt	Đạt
Iphone11	Màn hình: 6.1" Hệ điều hành: IOS 17	Đạt	Đạt

Bảng 4.24: Bảng kiểm thử sự tương thích của phần mềm trên các loại thiết bị

4.4.2 Kiểm thử hộp đen

a, Kiểm thử nhóm chức năng Đơn hàng của tôi

Chức năng Giá trị đầu vào Đầu ra cần đạt Kết quả	X X X >m2cm
Xem thông tin đơn hàng Chọn đơn hàng cần xem thông tin Hiển thị chi tiết thông tin về đơn hàng đó	Đạt
Mua lại đơn hàng Chọn nút mua lại Các sản phẩm trong đơn hàng được tự động thêm lại vào giỏ hàng	Đạt
Viết bình luận về sản phẩm đã mua Chọn nút viết bình luận của sản phẩm trong các sản phẩm của đơn hàng Điều hướng đến màn hình viết bình luận với đầy đủ thông tin về sản phẩm vừa chọn	Đạt
Bảng 4.25: Bảng kiểm thử nhóm chức năng Đơn hàng của tôi	

b, Kiểm thử nhóm chức năng Nhóm săn sale

Chức năng	Giá trị đầu vào	Đầu ra cần đạt	Kết quả
Tạo nhóm săn sale mới	Các thông tin cần thiết để tạo 1 nhóm săn sale	Hiển thị thông báo tạo nhóm thành công và có 1 nhóm mới xuất hiện trong màn hình danh sách nhóm	Đạt
Xem danh sách nhóm	Vào phần Profile và My Sale Group	Hiển thị màn hình các nhóm đã tạo, thông tin tóm tắt	Đạt
Mời bạn bè vào nhóm	Chọn nút mời bạn bè trong thanh điều hướng ở appbar	Điều hướng đến màn hình mời bạn bè, hiển thị danh sách bạn bè muốn mời, sau khi ấn gửi thì có thông báo thành công ở cả người gửi và người nhận, đồng thời ở danh sách lời mời phía người nhận có 1 lời mời mới xuất hiện	Đạt
Phê duyệt lời mời tham gia nhóm	Chọn nút điều hướng đến màn hình danh sách lời mời từ thanh điều hướng trên appbar của màn hình danh sách nhóm	Hiển thị các lời mời từ bạn bè, khi ấn đồng ý thì có thông báo gửi cho cả hai, bản thân được thêm vào nhóm săn sale của bạn tạo, và số lượng cùng trạng thái của nhóm được cập nhật; còn nếu chọn từ chối thì chỉ gửi thông báo	Đạt

Bảng 4.26: Bảng kiểm thử nhóm chức năng Nhóm săn sale

c, Kiểm thử nhóm chức năng voucher của tôi

Bảng 4.27: Bảng kiểm thử nhóm chức năng voucher của tôi

Chức năng	Giá trị đầu vào	Đầu ra cần đạt	Kết quả
Xem lịch sử voucher	Chọn nút xem lịch sử trên thanh điều hướng của appbar	Hiển thị màn hình lịch sử gồm các tab: đã thu thập, đã sử dụng và đã hết hạn, giao diện mỗi tab tương tự màn hình xem voucher	Đạt
Xem danh sách voucher	Vào phần Profile và My Voucher	Hiển thị thông tin các voucher trong hệ thống, có thể tìm kiếm lọc các voucher theo type	Đạt

CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG

Thu thập voucher	Chọn nút thu thập ở một thẻ voucher cụ thể	Tự động thêm voucher vào danh sách các voucher đã thu thập, có thể xem trong phần lịch sử cũng như phần checkout khi chuẩn bị thanh toán	Đạt
Áp dụng voucher	Chọn nút áp dụng voucher khi chuẩn bị thanh toán đơn hàng	Hiển thị danh sách các voucher đã thu thập và thỏa mãn các điều kiện được áp dụng của đơn hàng, khi chọn 1 voucher cụ thể để áp dụng thì sẽ tự động giảm giá cho đơn hàng và hiển thị thông tin voucher vừa áp dụng lên giao diện	Đạt
Quy đổi voucher	Chọn nút đổi thưởng ở trên thanh điều hướng của màn hình voucher	Hiển thị điểm số hiện tại, danh sách các voucher người dùng có thể dùng điểm quy đổi, khi chọn nút quy đổi 1 voucher thì điểm tự động bị trừ và voucher đó tự động được thêm vào list voucher đã nhận	Đạt

Vậy là trong chương 4, em đã trình bày về phần thiết kế, triển khai và đánh giá hệ thống. Tiếp đến, em sẽ trình bày nội dung của Chương 5 - Các giải pháp và đóng góp nổi bật.

CHƯƠNG 5. CÁC GIẢI PHÁP VÀ ĐÓNG GÓP NỔI BẬT

5.1 Tối ưu thời gian phản hồi khi xem danh sách sản phẩm

5.1.1 Đặt vấn đề

Đây là tính năng rất quan trọng, có thể coi là được người dùng thường xuyên, liên tục với tần suất nhiều nhất trong hệ thống, nên tốc độ phản hồi ảnh hưởng rất lớn đến trải nghiệm người dùng với ứng dụng. Về dữ liệu sản phẩm thì trong hệ thống tôi đã thu thập được từ việc crawl dữ liệu trên mạng lại khá lớn, khoảng 4500 sản phẩm khác nhau. Dữ liệu được lưu trữ dưới dạng document với nhiều trường phức tạp, phải gọi nhiều API khác nhau để lấy được đầy đủ các thông tin về sản phẩm đó như thông tin shop, thông tin brand, thông tin categories. Vì có một số lượng lớn data lớn như vậy và mỗi sản phẩm lại phải call nhiều API nên khiến thời gian chờ người dùng khá lâu, ảnh hưởng không tốt đến trải nghiệm người dùng.

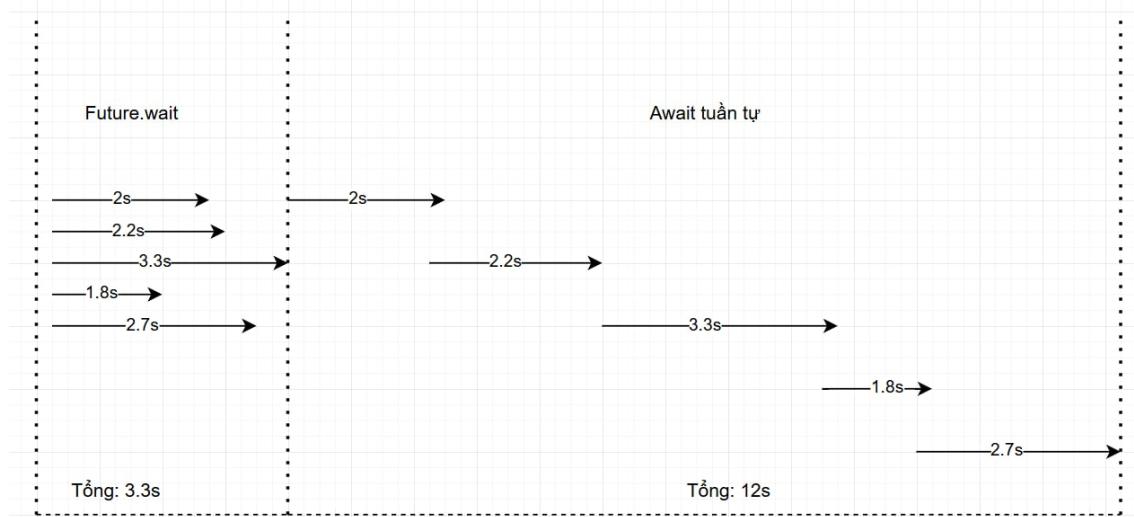
5.1.2 Giải pháp

Để cải thiện tốc độ phản hồi của hệ thống, em thực hiện đồng thời một số giải pháp sau:

Đầu tiên em không load và tải hết cả 4500 sản phẩm của hệ thống, hoặc vài trăm đến vài nghìn sản phẩm thuộc cùng 1 danh mục hay 1 brand lên ngay, vì làm như vậy trung bình hệ thống mất khoảng 10-15s để phản hồi. Thay vào đó em áp dụng chiến lược phân trang, mỗi lần load chỉ load 20 sản phẩm lên, nếu người dùng hứng thú sẽ click nút loadmore ở cuối trang, hệ thống sẽ load thêm 20 sản phẩm tiếp theo, và cứ như vậy cho tới khi hết sản phẩm thì sẽ thay nút loadmore bằng thông báo đã hết sản phẩm. Để thực thi được điều đó thì em sử dụng cơ chế phân trang dạng con trỏ (cursor-based pagination) của Firestore, cho phép truy vấn chính xác các tài liệu tiếp theo dựa trên document cuối cùng được lưu lại. Việc kết hợp với giới hạn truy vấn chỉ trả về 20 sản phẩm giúp hệ thống chỉ tải đúng 20 sản phẩm mỗi lần, tránh dư thừa dữ liệu và tăng tốc độ phản hồi đáng kể. Bằng cách này, hệ thống vừa đảm bảo hiệu năng, vừa mang lại trải nghiệm người dùng mượt mà hơn, đặc biệt trong các trường hợp người dùng duyệt nhiều sản phẩm trong cùng một danh mục hoặc thương hiệu.

Thứ hai, để tối ưu thời gian phản hồi khi hiển thị thông tin chi tiết của từng sản phẩm, em cũng thực hiện song song nhiều truy vấn không phụ thuộc nhau bằng cách sử dụng kỹ thuật bất đồng bộ với `Future.wait`. Thay vì truy xuất tuần tự dữ liệu của thương hiệu, cửa hàng và danh mục – mỗi bước đều phải chờ hoàn tất truy vấn trước đó – em tiến hành khởi tạo trước các `Future` tương ứng với mỗi

nguồn dữ liệu nếu có, và sau đó dùng `Future.wait` để thực hiện tất cả truy vấn một cách song song. Cách làm này giúp tiết kiệm đáng kể thời gian chờ đợi tổng thể, đặc biệt trong trường hợp người dùng duyệt nhiều sản phẩm liên tục, nhờ đó tăng hiệu suất xử lý và giảm cảm giác trễ khi hiển thị dữ liệu.



Hình 5.1: So sánh khi sử dụng hàm `Future.wait` so với khi gọi API tuần tự

Hình 5.1 biểu diễn sự so sánh giữa khi sử dụng hàm `Future.wait` so với khi gọi API tuần tự. Như trong hình ta có thể thấy, khi sử dụng `Future.wait` thời gian chạy của toàn bộ tất cả các tác vụ bằng thời gian chạy của tác vụ có thời gian lớn nhất. Trong khi đó, nếu chỉ thực hiện các tác vụ gọi API theo kiểu tuần tự thì thời gian là bằng tổng thời gian của tất cả các tác vụ cộng lại. Điều đó cho thấy sự hiệu quả khi áp dụng kỹ thuật lập trình bất đồng bộ sử dụng hàm `Future.wait` khi cần phải xử lý nhiều tác vụ gọi API cùng một lúc.

Thứ ba là với những dữ liệu nặng nhưng lại ít thay đổi, được gọi nhiều lần ở nhiều vị trí trong ứng dụng như top category, top shop hay top brand thì em chỉ gọi 1 lần duy nhất, và lưu vào cached, những lần sau sẽ check từ trong cached đã có rồi thì sẽ không truy vấn database nữa mà lấy ngay từ cached, giảm thời gian phản hồi của hệ thống.

5.1.3 Kết quả

```
I/flutter (10288): ⚡ Thời gian query lấy các product: 1452 ms
I/flutter (10288): ⚡ Thời gian load thông tin sản phẩm: 633 ms
I/flutter (10288): ⚡ Thời gian tải sản phẩm: 2086ms
I/flutter (10288): ⚡ Thời gian query lấy các product: 936 ms
I/flutter (10288): ⚡ Thời gian load thông tin sản phẩm: 397 ms
I/flutter (10288): ⚡ Thời gian tải 20 sản phẩm tiếp theo: 1335ms
```

Hình 5.2: Minh họa thời gian phản hồi của hệ thống sau tối ưu

Hình 5.2 cho thấy rất rõ, nhờ việc thực hiện đồng bộ các giải pháp trên mà thời gian phản hồi trung bình đã giảm 10 lần, từ khoảng 10-15s xuống còn từ 1-2s với lần load đầu tiên và các lần load sản phẩm tiếp theo, cải thiện rất nhiều về trải nghiệm

người dùng và tạo độ mượt mà ngang hệ thống lớn như shopee, tiktok shop.

5.2 Triển khai tính năng "Gợi ý sản phẩm cá nhân hóa theo người dùng"

5.2.1 Đặt vấn đề

Trong một hệ thống thương mại điện tử hoặc bán hàng trực tuyến, số lượng sản phẩm có thể lên tới hàng nghìn, thậm chí hàng chục nghìn mặt hàng thuộc nhiều danh mục và thương hiệu khác nhau. Điều này khiến người dùng, đặc biệt là những người mới, dễ rơi vào tình trạng “quá tải lựa chọn” (choice overload), khi phải mất nhiều thời gian để tìm được sản phẩm phù hợp với sở thích hoặc nhu cầu thực tế của bản thân. Việc để người dùng tự tìm kiếm, lọc và duyệt qua hàng loạt sản phẩm không chỉ làm giảm trải nghiệm sử dụng, mà còn ảnh hưởng tiêu cực tới tỷ lệ chuyển đổi đơn hàng của hệ thống.

Bên cạnh đó, dữ liệu hành vi của người dùng (như lịch sử mua hàng, lượt xem sản phẩm, lượt thêm vào giỏ hàng, v.v.) thường xuyên thay đổi và gia tăng theo thời gian. Do đó, nếu không có cơ chế cập nhật mô hình gợi ý một cách thường xuyên thì các đề xuất sẽ nhanh chóng trở nên lỗi thời, không còn phù hợp với sở thích hiện tại của người dùng. Việc huấn luyện lại mô hình toàn bộ từ đầu mỗi lần cập nhật cũng là giải pháp tốn kém về thời gian và tài nguyên tính toán.

Chính vì vậy, cần thiết phải xây dựng một tính năng gợi ý sản phẩm cá nhân hóa theo từng người dùng dựa trên mô hình học máy, lại có khả năng học gia tăng theo dữ liệu mới để không phải train lại toàn bộ data từ đầu.

5.2.2 Giải pháp

Để giải quyết vấn đề trên, em lựa chọn sử dụng mô hình LightFM – một mô hình hybrid kết hợp giữa collaborative filtering và content-based filtering – để khai thác cả hành vi người dùng và đặc điểm sản phẩm, từ đó tạo ra các đề xuất chính xác và có giá trị hơn. Đặc biệt, em thiết kế cơ chế tự động huấn luyện lại mô hình theo lịch định kỳ (dựa vào Cloud Scheduler), sử dụng phương pháp huấn luyện gia tăng (incremental learning) thông qua hàm `fit_partial` của LightFM. Điều này giúp hệ thống luôn cập nhật được dữ liệu người dùng mới mà không cần huấn luyện lại toàn bộ mô hình từ đầu, vừa tiết kiệm tài nguyên, vừa đảm bảo khả năng thích ứng linh hoạt theo thời gian.

Giải pháp được triển khai theo một quy trình tổng thể, gồm nhiều bước liên kết chặt chẽ từ frontend Flutter đến backend xử lý mô hình gợi ý. Ở phía ứng dụng Flutter, em xây dựng các hàm ghi log để ghi lại những tương tác thực tế của người dùng như xem sản phẩm, thêm vào giỏ, mua hàng, viết bình luận ... Các tương tác này được lưu vào bảng 4.20 trong Firestore, mỗi loại sự kiện sẽ có một trọng số

khác nhau phản ánh mức độ quan trọng của event khác nhau.

Tiếp theo, em sử dụng môi trường xử lý dữ liệu trên Kaggle để viết pipeline train mô hình. Pipeline bắt đầu bằng việc đọc dữ liệu từ Firestore, xây dựng ma trận tương tác giữa người dùng và sản phẩm với các trọng số tương ứng cho từng loại sự kiện. Đồng thời, em trích xuất thêm thông tin từ bảng 4.6 và bảng 4.4 để xây dựng các đặc trưng nội dung (user features và item features), giúp tăng độ chính xác của mô hình hybrid.

Để tối ưu hiệu quả huấn luyện, em tích hợp thư viện Optuna nhằm tự động tìm ra tổ hợp siêu tham số tốt nhất cho mô hình LightFM, bao gồm cả số lượng epoch, learning rate, item_alpha, user_alpha và loss function. Sau khi xác định được bộ tham số tối ưu, em tiến hành huấn luyện mô hình chính thức và lưu mô hình vào file định dạng .pkl, sau đó tải lên Firebase Storage để sử dụng sau này.

Quy trình huấn luyện lại định kỳ được triển khai thông qua một API sử dụng Flask, được deploy trên Google Cloud Run và thiết lập lịch gọi tự động mỗi 24 giờ nhờ Cloud Scheduler. Mỗi lần gọi, API sẽ truy vấn các tương tác mới phát sinh trong bảng 4.20 kể từ lần huấn luyện cuối cùng (dựa vào trường last_trained). Sau đó, hệ thống tạo lại một ma trận tương tác mới chỉ chứa các dữ liệu mới này, và tiến hành cập nhật mô hình hiện tại bằng phương pháp huấn luyện gia tăng thông qua hàm fit_partial. Mô hình sau khi được cập nhật sẽ lại được lưu vào Firebase Storage, đảm bảo hệ thống luôn có phiên bản mới nhất phục vụ cho việc gọi ý.

Bên cạnh đó, em xây dựng thêm một API thứ hai (cũng sử dụng Flask) để ứng dụng Flutter có thể gửi request đến và nhận danh sách sản phẩm được đề xuất dưới dạng JSON. Danh sách này sau đó sẽ được hiển thị trực tiếp trên giao diện người dùng, mang lại trải nghiệm gợi ý sản phẩm cá nhân hóa theo thời gian thực. Cả hai API đều được triển khai bằng Google Cloud Run, giúp hệ thống dễ mở rộng, dễ bảo trì, và tích hợp mượt mà với hạ tầng đám mây hiện tại.

Tổng thể, giải pháp này vừa tận dụng được các kỹ thuật học máy hiện đại để mang lại khả năng gợi ý chính xác, vừa đảm bảo hiệu suất và tính mở rộng nhờ thiết kế một pipeline hiện đại. Quá trình xử lý được tự động hóa từ khâu ghi nhận dữ liệu, huấn luyện mô hình, cập nhật định kỳ cho tới triển khai và tích hợp vào ứng dụng, mang lại một hệ thống gợi ý hoàn chỉnh, thông minh và thích nghi tốt với sự thay đổi hành vi người dùng theo thời gian.

5.2.3 Kết quả

Sau khi triển khai tính năng gợi ý sản phẩm cá nhân hóa theo người dùng, hệ thống đã ghi nhận được một lượng dữ liệu tương tác đáng kể từ phía người dùng thực tế, đồng thời các mô hình học máy cũng cho thấy hiệu quả rõ rệt khi so sánh giữa các phiên bản có và không fine-tune, có và không sử dụng meta-data. Kết quả cụ thể được trình bày như sau:

a, Thông kê dữ liệu thu thập được

```
print("Số lượng user metadata:", len(user_metadata))
```

Số lượng user metadata: 64

Hình 5.3: Số lượng người dùng đã ghi nhận tương tác

```
print("Số lượng item metadata:", len(item_features_list))
```

Total products: 4568
Số lượng item metadata: 4568

Hình 5.4: Tổng số lượng sản phẩm trong hệ thống

```
# In thống kê
print("Ma trận tương tác:")
print(f" Số người dùng: {interaction_matrix.shape[0]}")
print(f" Số sản phẩm: {interaction_matrix.shape[1]}")
print(f" Số lượng tương tác (non-zeros): {interaction_matrix.nnz}")
```

Tổng số tương tác: 3400
Sau khi gộp duplicate, tổng số tương tác: 2183
Ma trận tương tác:
Số người dùng: 53
Số sản phẩm: 451
Số lượng tương tác (non-zeros): 2183

Hình 5.5: Tổng số tương tác được ghi nhận trong bảng user_events

📊 Thống kê sự kiện:

- navigate_to_product_detail: 2348 lần
- add_to_cart: 361 lần
- favourite_product: 328 lần
- remove_one_in_cart: 105 lần
- increase_quantity: 58 lần
- write_review: 39 lần
- decrease_quantity: 39 lần
- navigate_to_review: 37 lần
- showEnlargedImage: 33 lần
- see_suggest_product: 21 lần
- add_one_in_cart: 19 lần
- select_review_tab: 11 lần
- repurchase: 1 lần

Hình 5.6: Tỷ lệ các loại tương tác: view, click, add to cart, purchase

Các dữ liệu về người dùng như ở hình 5.3 gồm 64 người dùng và dữ liệu mua sắm, tương tác với hệ thống của họ ở hình 5.6 đều là các người dùng và tương tác thật từ những người dùng đã thực tế trải nghiệm ứng dụng của em, đảm bảo tính chính xác, khách quan và gợi ý chính xác cho từng cá nhân.

b, Cơ chế tự động huấn luyện định kỳ

Mô hình được huấn luyện định kỳ mỗi 24 giờ để cập nhật các dữ liệu tương tác mới thông qua cơ chế incremental learning nhờ mô hình có hàm fit_partial cho phép train tiếp liên tục với data bổ sung, đây cũng là 1 điểm mạnh của mô hình này so với các mô hình học máy khác. Việc này được thực hiện nhờ sự kết hợp giữa Flask API, Google Cloud Run và Cloud Scheduler lặp lịch tự động.



Hình 5.7: Lên lịch job huấn luyện định kỳ bằng Cloud Scheduler

c, Kết quả tìm kiếm tham số tối ưu

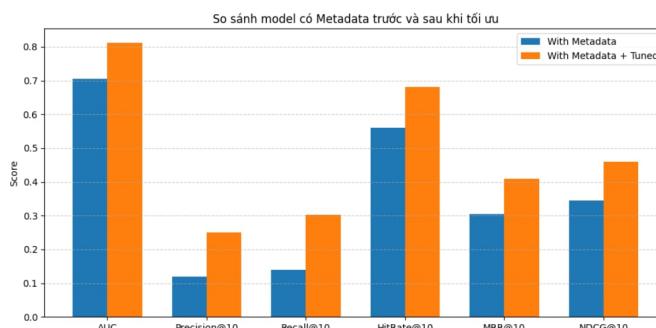
Để nâng cao độ chính xác của mô hình, em sử dụng thư viện Optuna để thử nghiệm và chọn ra bộ tham số tối ưu nhất.

```
Best hyperparameters (with metadata):
no_components: 29
learning_rate: 0.0005920184503895527
item_alpha: 8.107228770840291e-10
user_alpha: 5.169932550844715e-09
max_sampled: 20
learning_schedule: adadelta
loss: warp
```

Hình 5.8: Bộ tham số tối ưu thu được từ Optuna

d, So sánh hiệu quả mô hình có và không fine-tune

Dưới đây là kết quả so sánh giữa hai phiên bản mô hình: một phiên bản chỉ train ban đầu, và một phiên bản được train bằng bộ tham số tối ưu mà optuna tìm ra, kết quả bộ tham số tối ưu cho ra kết quả vượt trội hơn ở tất cả các chỉ số đánh giá.



Hình 5.9: So sánh kết quả mô hình với và không sử dụng fine-tune

e, Hiệu quả của mô hình khi không sử dụng metadata

Ngoài ra, để kiểm chứng mức độ đóng góp của metadata (đặc trưng sản phẩm và người dùng), em cũng thực hiện huấn luyện mô hình LightFM chỉ sử dụng ma trận tương tác. Kết quả bất ngờ cho thấy mô hình không sử dụng metadata lại cho kết quả tốt hơn về mọi mặt:

```
◆ Model without Metadata Evaluation:  
AUC: 0.8610  
Precision@10: 0.3800  
Recall@10: 0.4571  
HitRate@10: 0.8400  
MRR@10: 0.6026  
NDCG@10: 0.6377
```

Hình 5.10: Kết quả mô hình không sử dụng metadata

```
◆ Model with Metadata + Tuned Evaluation:  
AUC: 0.8127  
Precision@10: 0.2500  
Recall@10: 0.3027  
HitRate@10: 0.6800  
MRR@10: 0.4101  
NDCG@10: 0.4591
```

Hình 5.11: Kết quả mô hình sử dụng metadata

Giải thích: Trong một số trường hợp, việc sử dụng metadata (user/item features) có thể khiến mô hình hoạt động kém hơn vì:

- Dữ liệu metadata chưa đủ đặc trưng hoặc bị nhiễu: Ví dụ như đặc trưng của sản phẩm quá chung chung (như category phổ biến) hoặc user profile không phân biệt rõ nhóm người dùng.
- Mô hình bị overfitting do số lượng đặc trưng lớn nhưng phân bố không đồng đều, đặc biệt là khi số lượng user/item nhỏ so với feature dimension.
- Ma trận tương tác vốn đã rất đầy đủ và mạnh, nên việc thêm metadata đôi khi lại làm "loãng" tín hiệu gốc từ hành vi người dùng.

Do đó, dù về lý thuyết mô hình hybrid có thể cải thiện kết quả, nhưng thực tế tùy vào chất lượng feature, metadata không phải lúc nào cũng nâng cao hiệu suất.

Đây là một phát hiện thực tiễn quan trọng, cho thấy không phải lúc nào thêm nhiều đặc trưng cũng cải thiện kết quả. Việc đánh giá hiệu quả các nguồn dữ liệu đầu vào là rất cần thiết trong các hệ thống gợi ý. Hình 4.29 đã nêu ở chương 4 minh họa cho kết quả triển khai mô hình gợi ý này trên giao diện mobile của người dùng

5.3 Triển khai tính năng "Chatbot tự động hóa quy trình mua sắm"

5.3.1 Đặt vấn đề

Trong các hệ thống thương mại điện tử hiện nay, người dùng thường phải thực hiện nhiều thao tác thủ công lặp đi lặp lại để hoàn tất một quy trình mua sắm cơ bản: từ việc tìm kiếm sản phẩm, lọc lựa chọn phù hợp, thêm vào giỏ hàng, đến kiểm tra và xác nhận đặt hàng. Điều này khiến trải nghiệm mua sắm trở nên rườm rà, đặc biệt là trên thiết bị di động hoặc đối với người dùng không có nhiều thời gian hay không rành công nghệ.

Ngoài ra, một số người dùng có xu hướng cần trợ giúp trực tiếp hoặc thích tương tác bằng ngôn ngữ tự nhiên thay vì thao tác qua nhiều bước trên giao diện truyền thống. Trong khi đó, nhân sự hỗ trợ khách hàng lại không thể làm việc 24/7, dẫn đến tình trạng nghẽn tương tác, mất đơn hàng, hoặc bỏ lỡ nhu cầu tiềm năng.

Từ phía doanh nghiệp, việc không thể tự động hóa các quy trình cơ bản này làm tăng chi phí vận hành, chậm trễ trong việc phục vụ người dùng, và giảm hiệu quả trong việc chuyển đổi hành vi mua sắm. Đặc biệt, trong bối cảnh số lượng người dùng tăng nhanh, dữ liệu giao dịch ngày càng nhiều thì việc tối ưu hóa hành trình người dùng và giảm tải cho hệ thống là một yêu cầu cấp thiết.

Vì vậy, cần có một giải pháp thông minh, có khả năng giao tiếp bằng ngôn ngữ tự nhiên, hiểu được mục đích người dùng, và tự động thực hiện các thao tác như tìm kiếm, thêm sản phẩm, đặt hàng... chỉ thông qua hội thoại – giúp đơn giản hóa quy trình, cá nhân hóa trải nghiệm và nâng cao hiệu quả vận hành toàn hệ thống.

5.3.2 Giải pháp

Với những lợi ích rõ rệt như tiết kiệm thời gian thao tác, nâng cao trải nghiệm người dùng và tối ưu hiệu suất vận hành, em lựa chọn triển khai chatbot sử dụng nền tảng mã nguồn mở **Rasa** – một framework mạnh mẽ hỗ trợ xử lý ngôn ngữ tự nhiên (NLP) và xây dựng chatbot có khả năng thực hiện hành động cụ thể theo ngữ cảnh.

Kiến trúc hệ thống chatbot gồm các thành phần chính sau:

- **NLU (Natural Language Understanding):** Phân tích câu nhập của người dùng để xác định *intent* (mục đích) và *entities* (các thực thể liên quan).
- **Core:** Quản lý trạng thái hội thoại, dựa vào intent và lịch sử trò chuyện để quyết định hành động tiếp theo.
- **Actions Server:** Xử lý các hành động tùy chỉnh phức tạp, ví dụ như gọi API để tìm sản phẩm, thêm vào giỏ hàng hoặc hiển thị giỏ hàng, được viết trong tệp `actions.py`.
- **Front-end Flutter App:** Gửi câu thoại của người dùng lên server Rasa và hiển thị các phản hồi (bao gồm cả dạng button) từ chatbot.

Cấu trúc thư mục chính của chatbot Rasa:

- `nlu.yml`: Chứa tập dữ liệu huấn luyện NLU, định nghĩa các intent như `greet`, `search_product`, `add_to_cart`, `place_order`, v.v.
- `domain.yml`: Định nghĩa toàn bộ intents, entities, actions, slots, responses (bao gồm cả trả về button), và các session rule.

- `stories.yml` và `rules.yml`: Mô tả luồng hội thoại bằng cách ghi lại các kịch bản mẫu và quy tắc hội thoại.
- `actions.py`: Chứa các hàm xử lý logic như gọi API sản phẩm, quản lý giỏ hàng, xử lý đặt hàng,... Các hàm này có thể trả về tin nhắn dạng JSON (custom payload) để hiện các quick reply buttons trên UI.

Flow hoạt động tổng thể:

- Người dùng nhập một câu như: "Mình muốn mua găng tay".
- Rasa NLU phân tích và gán intent là `search_product`, đồng thời trích xuất entity là `găng tay`.
- Rasa Core xác định hành động tương ứng là `action_search_product`.
- Tại `actions.py`, action này gọi đến API tìm kiếm sản phẩm với từ khóa "`găng tay`", sau đó trả về danh sách kết quả dưới dạng json, sau đó sẽ trích xuất trường `title`, giới hạn trả về 50 sản phẩm, đánh số thứ tự sản phẩm để cho bước tiếp theo như thêm vào giỏ hàng dễ dàng nhập "thêm 2 cái số 3 vào giỏ hàng" thay vì phải nhập cả tên sản phẩm rất dài. Ngoài ra thì response cũng bao gồm trường button với `title` và `payload` phục vụ cho việc hiển thị quick button ở trên giao diện, tối ưu trải nghiệm người dùng, giúp họ không cần nhập thủ công mà chỉ cần click button gợi ý trên giao diện là bot sẽ tự động thực thi hành động tương ứng, vô cùng nhanh và tiện lợi
- Trên giao diện Flutter, hệ thống gửi câu lệnh mà người dùng nhập vào lên rasa server, nhận response trả về, bóc tách các trường để xử lý logic và hiển thị text, hiển thị quick button lên giao diện
- Tương tự, các flow như hiển thị giỏ hàng, đặt hàng, hỏi tiếp sản phẩm khác, xin chào/tạm biệt... đều được triển khai dưới dạng intent riêng với action tùy chỉnh tương ứng.

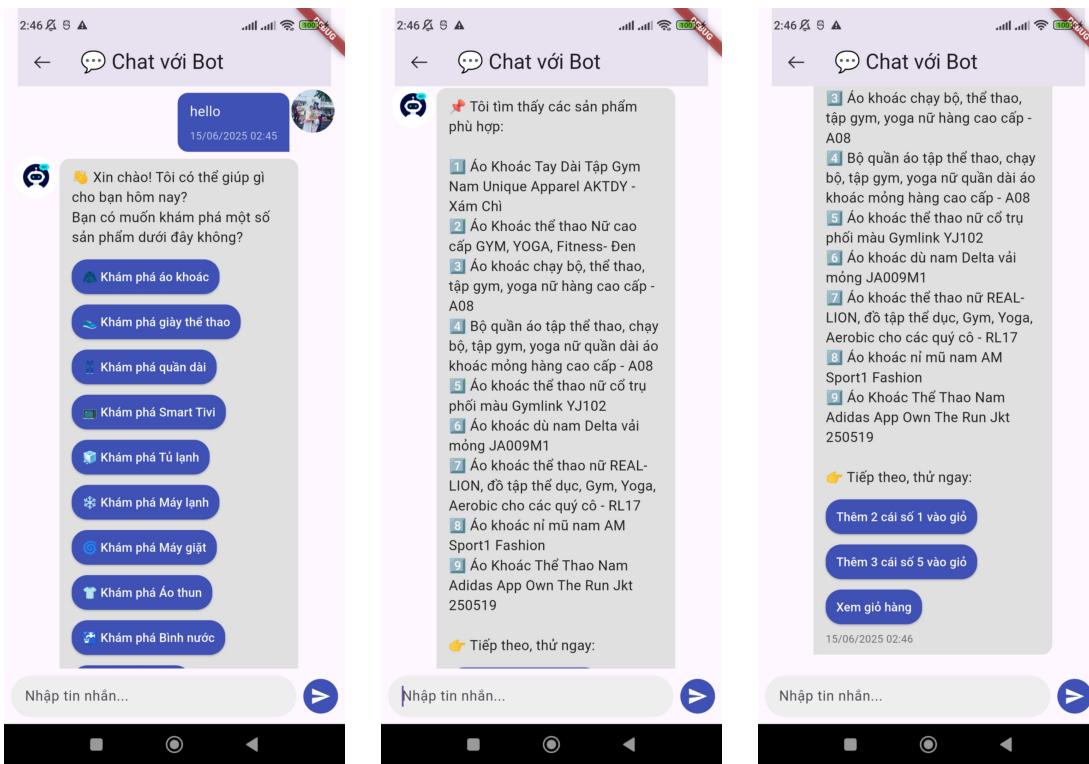
Sau khi đã khai báo và hoàn thiện các file trong rasa, chỉ cần gõ lệnh `rasa run` để build model, mất khoảng 2-3 phút, sau đó chạy lệnh `rasa run actions` để run actions server và `rasa run` để chạy rasa server, rasa server sẽ tự động gọi đến action-server qua port 5055. Sau khi đã build rasa server thành công, flutter mobile sẽ gọi đến server bằng https request để gửi tin nhắn và nhận phản hồi

5.3.3 Kết quả

Người dùng có giao diện chat bot đẹp mắt để trò chuyện với bot, tự động hóa quy trình mua sắm ngay trong chat page, có thể nhập hoặc đơn giản hơn là chỉ với vài click là có thể ra lệnh cho chatbot thực thi hành động ngay lập tức. Luồng trò

CHƯƠNG 5. CÁC GIẢI PHÁP VÀ ĐÓNG GÓP NỔI BẬT

chuyện hoàn chỉnh với chatbot được tôi demo qua các hình ảnh ở dưới

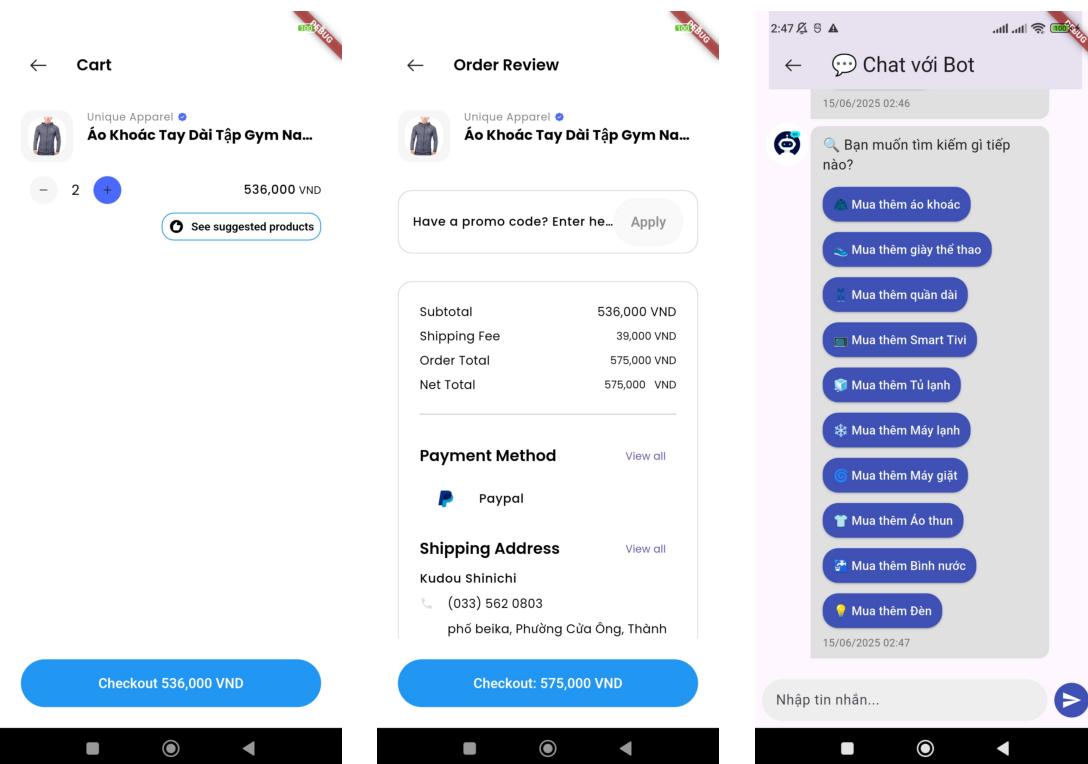


Hình 5.12: Người dùng bắt đầu cuộc hội thoại bằng lời chào, bot phản hồi ngay bằng các quick button gợi ý khám phá

Hình 5.13: Khi người dùng bấm khám phá 1 sản phẩm, bot hiển thị danh sách sản phẩm tìm thấy

Hình 5.14: Bot đưa ra các gợi ý là các quick button để người dùng có thể thêm nhanh sản phẩm vào giỏ hàng hoặc nhập câu lệnh tương tự theo mẫu hoặc xem giỏ hàng

CHƯƠNG 5. CÁC GIẢI PHÁP VÀ ĐÓNG GÓP NỔI BẬT

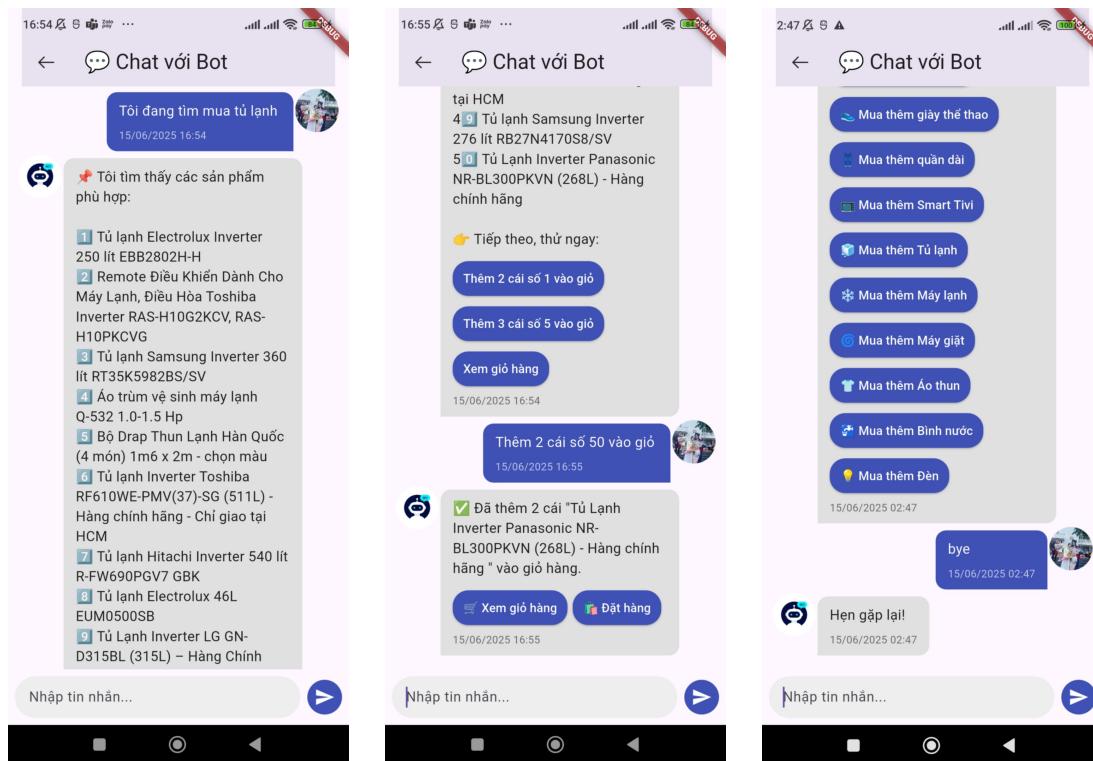


Hình 5.15: Bot điều hướng đến giao diện giỏ hàng khi người dùng chọn thêm vào giỏ hàng

Hình 5.16: Bot điều hướng đến xem đơn hàng và đặt hàng nếu người dùng chọn xem đơn hàng

Hình 5.17: Bot hiển thị danh sách các mục để người dùng có thể quick button click và xem thêm các sản phẩm khác tiếp theo

CHƯƠNG 5. CÁC GIẢI PHÁP VÀ ĐÓNG GÓP NỔI BẬT



Hình 5.18: Ngoài việc bấm quickly button, người dùng vẫn có thể tự nhập để tìm kiếm sản phẩm mình muốn ngoài những gợi ý của bot

Hình 5.19: Ngoài việc bấm quickly button, người dùng vẫn có thể tự nhập để thêm sản phẩm và số lượng mình thích vào giỏ hàng ngoài những gợi ý có sẵn của bot

Hình 5.20: Bot cảm ơn người dùng và kết thúc phiên trò chuyện.

5.4 Triển khai tính năng "Săn sale cùng bạn bè và chia sẻ nhanh qua mạng xã hội"

5.4.1 Đặt vấn đề

Trong thời đại số hiện nay, hành vi tiêu dùng của người dùng đang dần thay đổi theo hướng cộng đồng và tương tác cao. Người dùng không còn đơn thuần mua sắm một cách cá nhân, mà có xu hướng tham khảo ý kiến bạn bè, chia sẻ khuyến mãi hấp dẫn và cùng nhau tìm kiếm các cơ hội mua sắm tiết kiệm.

Tuy nhiên, phần lớn các hệ thống hiện nay vẫn chưa thực sự tập trung phát triển tính năng mua sắm theo nhóm kết hợp chia sẻ qua mạng xã hội. Dù người dùng có nhu cầu rõ rệt trong việc kết nối bạn bè và cùng nhau săn sale, đa số nền tảng vẫn chỉ dừng lại ở các hình thức chia sẻ thủ công hoặc thiếu sự tích hợp trực tiếp với các kênh mạng xã hội quen thuộc. Điều này khiến người dùng thường bỏ lỡ các cơ hội mua sắm theo nhóm do không kịp mời bạn bè tham gia, hoặc gặp khó khăn trong việc tìm và kết nối với những người có nhu cầu tương tự trong khoảng thời gian giới hạn.

Đây chính là khoảng trống lớn mà ứng dụng của em muốn khai thác: một nền

tảng nơi người dùng có thể dễ dàng lập nhóm, mời bạn bè và chia sẻ thông tin săn sale chỉ với vài thao tác đơn giản qua các kênh mạng xã hội quen thuộc như Messenger, Zalo, Discord,... Việc mua sắm không chỉ còn là hành động cá nhân, mà trở thành một trải nghiệm kết nối, tiết kiệm và đầy hứng khởi.

Nhu cầu chia sẻ, kết nối và mua sắm thông minh đang trở thành xu hướng tất yếu. Nếu ứng dụng không kịp thời bắt nhịp với xu thế này, rất có thể sẽ đánh mất nhóm người dùng trẻ năng động – những người luôn muốn vừa mua sắm vừa kết nối xã hội. Vì vậy, một tính năng săn sale nhóm kết hợp chia sẻ nhanh qua mạng xã hội là không chỉ cần thiết mà còn là một điểm khác biệt chiến lược, tạo lợi thế cạnh tranh cho sản phẩm.

5.4.2 Giải pháp

Đầu tiên, trong mục My Sale Groups của mục Profile trong ứng dụng, khi người dùng click vào sẽ xem danh sách các nhóm họ đã tạo hoặc đã tham gia theo lời mời của bạn bè. Nếu chưa có nhóm nào thì họ sẽ điều hướng đến màn hình tạo group để tạo group săn sale mới. Màn hình này sẽ có form để người dùng nhập vào các thông tin như tên nhóm, số lượng người tối thiểu của nhóm để kích hoạt voucher, và phạm vi voucher áp dụng: các sản phẩm thuộc cùng 1 shop hoặc 1 brand hoặc 1 category nhất định. Sau khi đã chọn đầy đủ thành công và click nút tạo nhóm, hệ thống sẽ tạo và lưu nhóm mới tạo lên database firestore, tự tính toán voucher với mức giảm giá dựa trên số người và phạm vi được lựa chọn. Sau khi tạo thành công hệ thống gửi thông báo đã tạo nhóm thành công đến cho người dùng, kèm các thông tin cơ bản của nhóm, và cập nhật lại danh sách nhóm. Với mỗi nhóm người dùng có thể xem thông tin chi tiết của nhóm đó, gồm tên, phạm vi, mức sale, số thành viên hiện tại và số thành viên cần đạt, thời gian tạo, thời gian hết hạn, trạng thái nhóm, thanh tiến độ và đồng hồ đếm ngược 24h.

Để mời thêm bạn bè của mình tham gia nhóm để nhóm đạt đủ KPI thành viên trước deadline, em có xây dựng 2 cách để mời khác nhau:

- Thứ nhất là mời với những bạn bè đã kết bạn với nhau trong hệ thống. Với cách mời này, người dùng sẽ chọn và được điều hướng đến màn hình mời bạn bè tham gia nhóm. Màn hình sẽ hiển thị danh sách bạn bè hiện tại của người dùng, với mỗi người bạn sẽ có kèm một nút mời. Khi người dùng click vào nút này, sẽ gửi request đến bạn của mình, nếu request được gửi thành công sẽ có thông báo từ hệ thống gửi đến người dùng đã gửi thành công và đến người bạn đó là có 1 lời mời tham gia nhóm mới. Người bạn đó khi nhận được thông báo sẽ vào màn hình danh sách request để kiểm tra, Ở đây người đó có thể chấp nhận hoặc từ chối lời mời. Nếu chấp nhận thì người đó sẽ được hệ thống thêm

vào nhóm, các thông tin về số thành viên hiện tại, tiến độ, trạng thái của nhóm cũng sẽ được tự động cập nhật tức thời. Nếu số lượng người tham gia đạt đến yêu cầu và vẫn trước deadline thì tất cả các thành viên trong nhóm sẽ đều nhận được thông báo nhóm đã tạo thành công, đã nhận được voucher săn sale của nhóm và tự động thêm vào mục các voucher người đó đã thu thập. Còn nếu người đó chọn từ chối thì cũng sẽ có thông báo gửi đến người dùng là lời mời của bạn đã bị bạn bè từ chối.

- Thứ hai là gửi link hoặc QR chia sẻ qua mạng xã hội. Với giải pháp này thì có thể mời bất kì ai, kể cả đã kết bạn hay chưa. Người dùng chọn mục chia sẻ qua mạng xã hội, sẽ có 1 pop up hiện lên với hai lựa chọn chia sẻ link liên kết hoặc chia sẻ QR. Sau đó sẽ có popup hiện lên cho người đó lựa chọn nền tảng muốn chia sẻ, chính là các nền tảng mạng xã hội mà người đó đã cài đặt như: messenger, zalo, facebook, instagram... Khi bạn bè nhận được link hoặc QR, bấm vào link hoặc quét QR, tùy thuộc vào việc người đó chưa tải và đăng nhập ứng dụng thì hệ thống sẽ điều hướng người đó đến màn hình cho phép nhập email của họ và sẽ gửi link tải ứng dụng qua email. Người đó có thể mở email, tải ứng dụng về máy, tạo tài khoản và đăng nhập vào hệ thống. Việc chia sẻ link tải và invite qua email này là nhờ em đã tích hợp dịch vụ [14]. Trường hợp người đó đã tải và đã đăng nhập hệ thống rồi, thì khi mở link và quét QR, hệ thống tự điều hướng đến màn hình home và popup hiển thị thông tin nhóm, nút chọn tham gia nhóm. Khi người dùng chọn tham gia nhóm thì không cần chờ chủ nhóm phê duyệt mà có thể tham gia nhóm ngay lập tức.

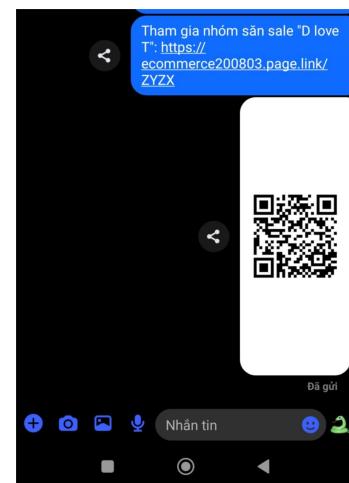
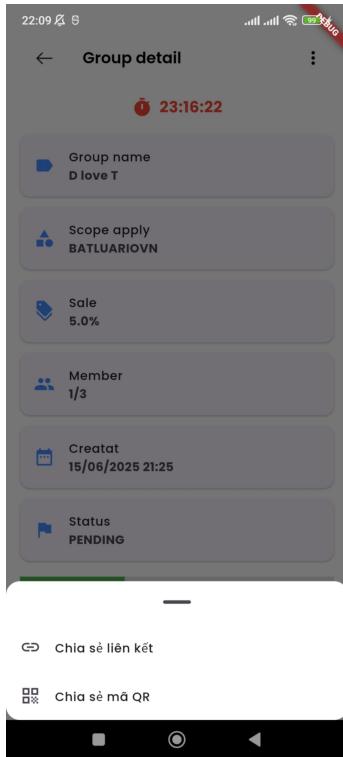
Để phục vụ cho tính năng mời bạn bè đã kết bạn trong hệ thống, em cũng đã phát triển **tính năng kết bạn trong nhóm**. Ở mục My Friends trong mục Profile, người dùng có thể xem danh sách bạn bè hiện tại của mình trong ứng dụng. Nếu chưa có hoặc muốn kết thêm bạn, người đó sẽ vào màn hình kết bạn, ở đây sẽ hiển thị danh sách các người dùng trong hệ thống, có nút tìm kiếm để tìm một người mà họ quen theo email hoặc username của họ. Nếu muốn kết bạn, người đó sẽ click vào nút add friend, hệ thống sẽ tự động gửi request đến người kia, và gửi thông báo là bạn có một lời mời kết bạn mới. Người nhận sẽ vào check danh sách lời mời kết bạn, cũng có thể chấp nhận hoặc từ chối. Nếu chấp nhận thì cả hai sẽ trở thành bạn bè, và gửi thông báo cho người gửi lời mời. Còn nếu người nhận từ chối thì cũng sẽ có thông báo đã bị từ chối và có thể gửi lời mời khác.

Khi các nhóm đã qua deadline mà chưa tập hợp đủ người, thì nhóm đó sẽ tự động chuyển sang trạng thái expired, và chủ nhóm sẽ không thể mời thêm người vào nhóm nữa.

5.4.3 Kết quả

Ở chương 4, em đã có minh họa một số kết quả của tính năng này, đó là màn hình thêm bạn bè 4.18, màn hình chấp nhận tham gia nhóm 4.19, màn hình tạo nhóm săn sale 4.20, màn hình chi tiết nhóm săn sale 4.22, màn hình mời bạn bè tham gia nhóm săn sale 4.23.

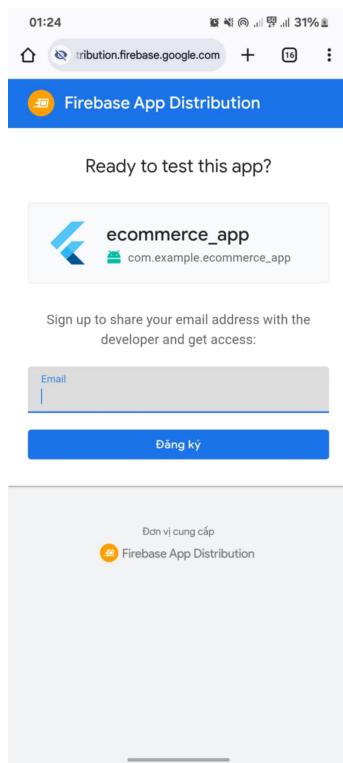
Ngoài các màn hình đó, em sẽ minh họa tính năng chia sẻ qua mạng xã hội:



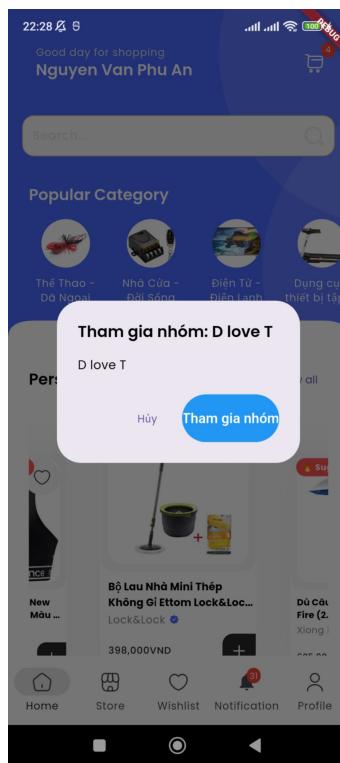
Hình 5.21: Người dùng chọn chức năng chia sẻ qua mạng xã hội, có 2 lựa chọn là gửi link hoặc gửi QR

Hình 5.22: Khi người dùng chọn QR, hệ thống sẽ gen sẵn ảnh QR thông tin nhóm để người đó chia sẻ

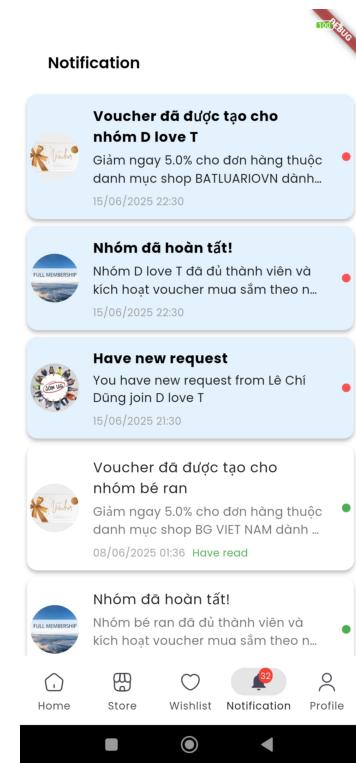
Hình 5.23: Hình ảnh cả link và cả QR đã được chia sẻ qua messenger



Hình 5.24: Nếu bấm vào link hoặc quét QR mà người đó chưa cài app, sẽ yêu cầu họ nhập email để invite qua email



Hình 5.25: Nếu bấm vào link hoặc quét QR mà người đó đã cài app, sẽ có popup mời họ tham gia



Hình 5.26: Khi tập hợp đủ người, sẽ có thông báo nhóm tập hợp đủ người và thông báo gửi tới các thành viên đã nhận voucher thành công

5.5 Triển khai tính năng "Làm nhiệm vụ và đổi thưởng"

5.5.1 Đặt vấn đề

Trong bối cảnh thị trường thương mại điện tử ngày càng cạnh tranh, việc thu hút người dùng mới đã trở nên khó khăn hơn bao giờ hết — nhưng giữ chân người dùng cũ và tăng mức độ tương tác với nền tảng còn là một thách thức lớn hơn. Người dùng ngày nay không chỉ kỳ vọng vào trải nghiệm mua sắm tiện lợi, mà còn mong muốn được tương tác, tham gia và nhận lại giá trị khi họ chủ động sử dụng ứng dụng thường xuyên.

Tuy nhiên, thực tế cho thấy phần lớn người dùng chỉ quay lại ứng dụng khi có nhu cầu mua sắm cụ thể, sau đó rời đi mà không duy trì sự hiện diện thường xuyên. Điều này dẫn đến tỷ lệ người dùng hoạt động hàng ngày (DAU) thấp, thời gian phiên sử dụng ngắn và khả năng tạo ra doanh thu bị hạn chế. Các nền tảng thiếu cơ chế duy trì thói quen sử dụng, khuyến khích hành vi tích cực, hoặc tạo cảm giác “được thưởng” cho những người dùng trung thành.

Trong xu hướng game hoá (gamification) ngày càng được áp dụng rộng rãi, việc tích hợp các yếu tố như nhiệm vụ hàng ngày, thử thách theo thời gian, hệ thống điểm thưởng và quy đổi quà tặng không chỉ giúp tăng tính hấp dẫn mà còn tạo

động lực để người dùng quay lại và tương tác thường xuyên. Đây không chỉ là một cách để làm cho trải nghiệm mua sắm trở nên thú vị hơn, mà còn là chiến lược hiệu quả để tăng mức độ trung thành, kéo dài vòng đời người dùng và thúc đẩy doanh thu.

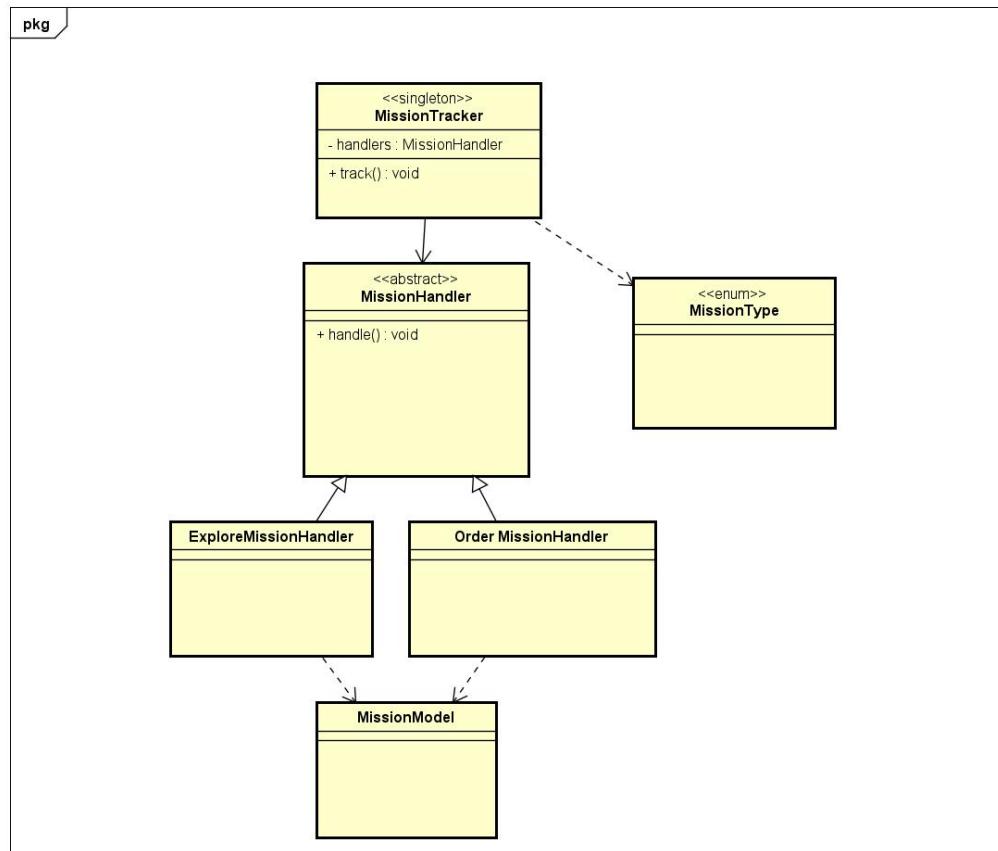
Chính vì vậy, việc triển khai một hệ thống nhiệm vụ kết hợp đổi thưởng là một giải pháp cấp thiết để nâng cao sự gắn bó của người dùng, gia tăng tần suất sử dụng ứng dụng, đồng thời tạo ra một lợi thế cạnh tranh bền vững trên thị trường.

5.5.2 Giải pháp

Từ vấn đề trên, em đã xây dựng tính năng làm nhiệm vụ tích điểm thường và tính năng đổi điểm lấy voucher.

Về các nhiệm vụ, thử thách cho người dùng, thì đầu tiên là nhiệm vụ điểm danh hàng ngày. Em sẽ tạo chuỗi thử thách 7 ngày điểm danh liên tục, chuỗi điểm danh liên tiếp càng dài thì ngày tiếp theo sẽ được càng nhiều điểm hơn. Nếu người đó bị miss điểm danh ngày nào thì chuỗi tự reset và quay về từ ngày 1. Nếu đã hoàn thành điểm danh liên tục cả 7 ngày cũng sẽ tự reset để bắt đầu một chuỗi điểm danh mới.

Nhóm nhiệm vụ thứ hai là thử thách thực sự với người dùng, khi phải thực hiện đủ số lượng nhiệm vụ yêu cầu trong một thời gian quy định. Người dùng vào phần My Missions của mục Profile trong app, sẽ thấy các nhiệm vụ được hệ thống tạo sẵn, và trạng thái nhiệm vụ nào mình chưa làm, nhiệm vụ nào đang làm, đã làm hoặc đã hết hạn. Khi xem chi tiết một nhiệm vụ sẽ có ảnh minh họa nhiệm vụ, tên, mô tả chi tiết nhiệm vụ, điểm thường, deadline, và đồng hồ đếm ngược thời gian còn lại của nhiệm vụ, cũng như thanh tiến trình hiển thị nhiệm vụ đã hoàn thành được bao nhiêu %. Khi người dùng bắt đầu một nhiệm vụ, hệ thống sẽ tự tracker toàn ứng dụng để theo dõi khi nào người dùng thực hiện hành động nào, nếu hành động đó trùng với hành động mà nhiệm vụ đang được tính thời gian thì nó sẽ ghi nhận và tự động cập nhật tiến độ của nhiệm vụ. Để theo dõi tiến độ hoàn thành nhiệm vụ một cách tự động và chính xác, hệ thống được thiết kế theo hướng linh hoạt và mở rộng dễ dàng. Khi người dùng bắt đầu một nhiệm vụ, hệ thống sẽ tự động theo dõi toàn bộ ứng dụng để nhận diện khi người dùng thực hiện các hành động có liên quan. Nếu hành động đó trùng với loại nhiệm vụ đang được kích hoạt và tính thời gian, hệ thống sẽ tự động ghi nhận và cập nhật tiến độ.



Hình 5.27: Biểu đồ class cho tính năng Mission Tracking

Hình 5.27 minh họa biểu đồ lớp của hệ thống theo dõi nhiệm vụ. Các lớp chính trong hệ thống được thiết kế rõ ràng, tuân thủ nguyên tắc thiết kế hướng đối tượng và đảm bảo khả năng mở rộng linh hoạt:

- **MissionTracker:**

- Là một **service trung tâm** của hệ thống, đóng vai trò như bộ điều phối logic xử lý nhiệm vụ khi người dùng thực hiện hành động tương ứng.
- Được triển khai theo **Singleton pattern**, đảm bảo chỉ có một thể hiện duy nhất trong toàn bộ ứng dụng. Điều này giúp dễ dàng gọi hàm track ở bất kỳ đâu trong ứng dụng cần tracking như ở màn hình xem sản phẩm, xem thương hiệu, hay khi người dùng đặt đơn hàng.
- Chứa một **Map các handler** – mỗi handler xử lý một loại nhiệm vụ nhất định, ánh xạ theo enum ‘MissionType’. Điều này giúp hệ thống xử lý linh hoạt đa dạng nhiệm vụ mà không cần thay đổi cấu trúc lõi.
- Phương thức track truy vấn Firestore để tìm các nhiệm vụ đang hoạt động thuộc type tương ứng, sau đó gọi đến handler tương ứng để xử lý logic cập nhật nhiệm vụ.

- **MissionHandler (abstract class):**

- Là lớp trừu tượng, định nghĩa phương thức ‘handle(...)' như một hợp đồng chung mà tất cả các lớp con phải hiện thực.
- Phụ thuộc UserMissionModel – đây là model đại diện cho một nhiệm vụ cụ thể mà người dùng đang tham gia. Handler sử dụng model này để lấy thông tin chi tiết và cập nhật tiến độ nhiệm vụ.
- **ExploreMissionHandler, OrderMissionHandler:** Là các lớp cụ thể kế thừa từ MissionHandler, hiện thực phương thức handle(...) tương ứng với từng loại hành động: xem nội dung, đặt hàng, v.v.

Hệ thống trên sở hữu nhiều điểm mạnh nổi bật về thiết kế:

- **Tính mở rộng cao và tuân thủ nguyên lý Open-Closed:**
 - Khi cần thêm một loại nhiệm vụ mới (ví dụ: "chia sẻ sản phẩm lên mạng xã hội"), em chỉ cần:
 1. Định nghĩa một giá trị mới trong enum MissionType.
 2. Tạo một lớp mới kế thừa từ MissionHandler, ví dụ ShareMissionHandler.
 3. Đăng ký handler mới vào map _handlers trong MissionTracker.
 - Toàn bộ quy trình xử lý tracking vẫn hoạt động như cũ, không cần chỉnh sửa lại các class hiện tại. Đây là minh chứng rõ ràng cho việc hệ thống mở với mở rộng, đóng với chỉnh sửa.
- **Áp dụng mô hình Strategy Pattern** thông qua việc phân phối nhiệm vụ đến handler phù hợp với chiến lược xử lý riêng biệt.
- **Tự động hóa toàn diện:** Người dùng không cần thao tác thủ công để cập nhật tiến độ nhiệm vụ. Mỗi hành động được hệ thống tự động ghi nhận và xử lý đúng với logic tương ứng.
- **Tuân thủ nguyên lý Single Responsibility (SRP):** MissionTracker chỉ điều phối nhiệm vụ, còn các MissionHandler xử lý logic nhiệm vụ cụ thể, đảm bảo sự rõ ràng và dễ bảo trì.

Nhờ cách tổ chức module hợp lý, hệ thống có thể dễ dàng mở rộng để hỗ trợ hàng chục, hàng trăm loại nhiệm vụ khác nhau trong tương lai, mà vẫn đảm bảo hiệu năng, tính ổn định và khả năng kiểm soát logic một cách rõ ràng.

Khi người dùng hoàn thành một nhiệm vụ, thì sẽ có thông báo nhiệm vụ đã hoàn thành, kèm số điểm nhận được, và điểm thưởng sẽ tự động được tăng lên.

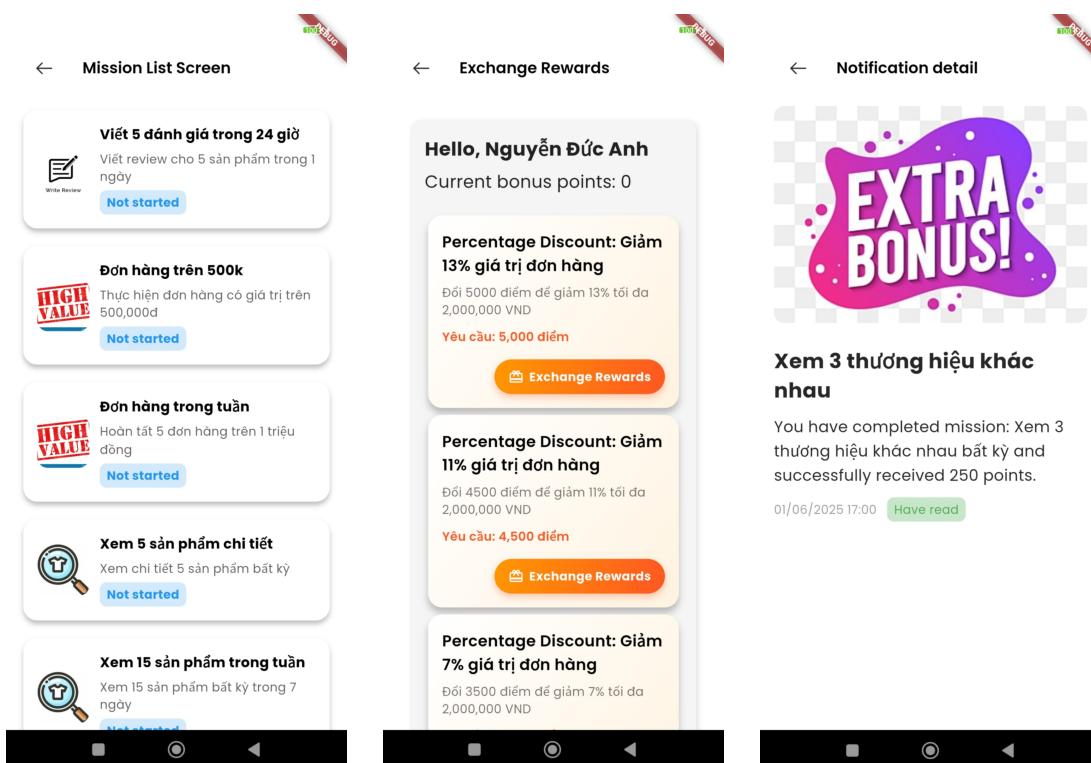
Hệ thống cũng có tích hợp tính năng đổi thưởng, khi hệ thống cũng sẽ quy định

sẵn một số loại voucher có thể quy đổi bằng điểm thưởng, gồm thông tin voucher, mức điểm thưởng yêu cầu và nút quy đổi. Khi người dùng click nút quy đổi thì tự động voucher đó sẽ được nhận và điểm thưởng của người dùng sẽ bị trừ tương ứng.

Ngoài ra, ở phần web admin quản trị viên có thể quản lý CRUD các nhiệm vụ trong hệ thống, cũng như xem thống kê các nhiệm vụ theo type, theo lịch sử người dùng đã thực hiện.

5.5.3 Kết quả

Ở chương 4, em đã minh họa các tính năng điểm danh hàng ngày 4.21 và màn hình nhiệm vụ chi tiết 4.24. Ở chương này, em sẽ minh họa tiếp giao diện của các màn hình khác liên quan đến nhóm chức năng này



Hình 5.28: Màn hình danh sách các nhiệm vụ có sẵn trong hệ thống và trạng thái của người dùng đã hay chưa xử lý

Hình 5.29: Màn hình quy đổi điểm thưởng lấy voucher

Hình 5.30: Thông báo khi người dùng hoàn thành 1 nhiệm vụ và nhận được điểm thưởng tương ứng

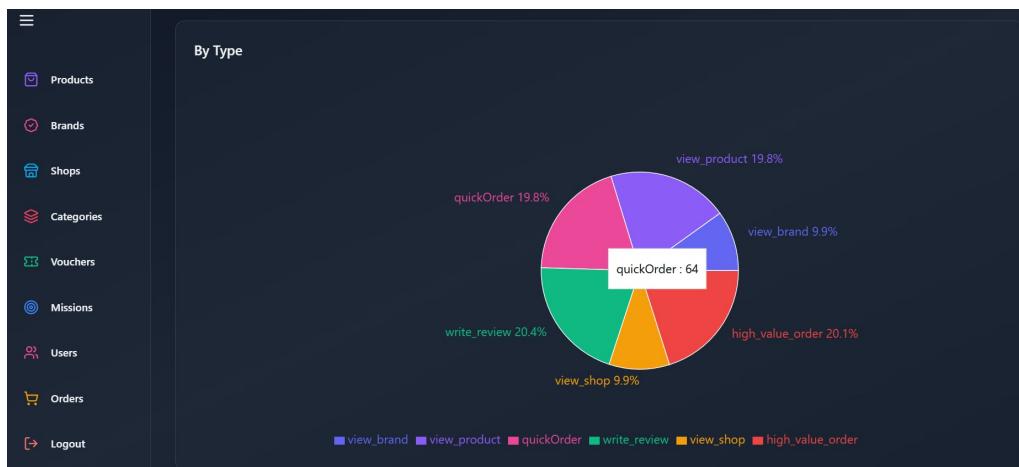
CHƯƠNG 5. CÁC GIẢI PHÁP VÀ ĐÓNG GÓP NỔI BẬT

The screenshot shows a dark-themed admin dashboard for mission management. On the left is a sidebar with icons for Brands, Shops, Categories, Vouchers, Missions, Users, Orders, and Logout. The main area is titled 'Mission' and displays a summary box for 'Total Missions' (21). Below is a 'Voucher List' table with the following data:

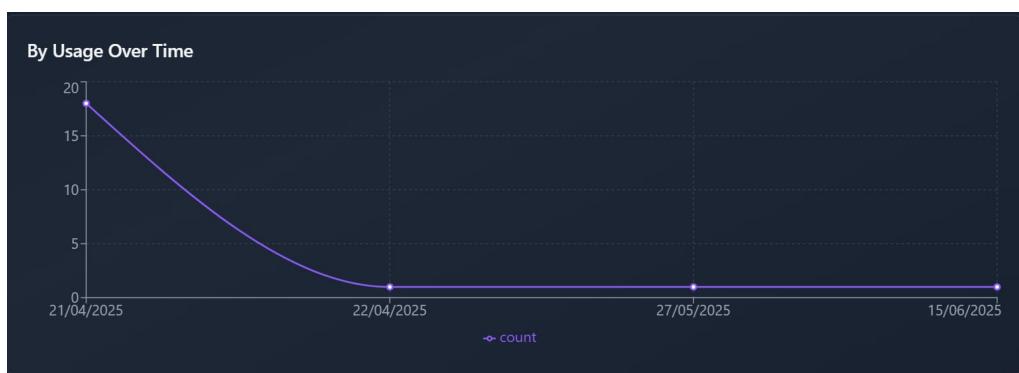
Image	Title	Type	Duration	Description	Goal	Reward	Created At	Actions
	Xem 20 thương hiệu trong tuần	View Brand	Weekly	Xem 20 thương hiệu khác nhau trong vòng 7 ngày	20	+1.800 điểm	—	
	Viết 5 đánh giá trong 24 giờ	Write Review	Daily	Viết review cho 5 sản phẩm trong 1 ngày	5	+250 điểm	—	
	Đơn hàng trên 500k	High Value Order	None	Thực hiện đơn hàng có giá trị trên 500.000đ	1	+400 điểm	—	

Hình 5.31: Màn hình web admin CRUD nhiệm vụ

Hình 5.31 hiển thị màn hình quản trị viên CRUD nhiệm vụ, ngoài ra có thể search kết hợp lọc nhiệm vụ theo type



Hình 5.32: Màn hình thống kê theo type



Hình 5.33: Màn hình thống kê theo lịch sử thực hiện nhiệm vụ của người dùng

5.6 Triển khai hệ thống "voucher, áp dụng và quản lý voucher"

5.6.1 Đặt vấn đề

Trong bối cảnh thị trường thương mại điện tử ngày càng cạnh tranh, việc tạo ra các hình thức khuyến mãi thông minh và cá nhân hóa là một trong những yếu tố

then chốt để thu hút và giữ chân người dùng. Một trong những công cụ quan trọng nhất hỗ trợ cho chiến lược này chính là hệ thống voucher – đóng vai trò như một cầu nối giữa ưu đãi từ nhà bán và hành vi mua sắm của người dùng.

Tuy nhiên, ở nhiều nền tảng hiện nay, hệ thống voucher vẫn còn tồn tại nhiều hạn chế. Việc tìm kiếm và áp dụng mã giảm giá thường yêu cầu người dùng phải nhớ mã, tự điền thủ công, hoặc phụ thuộc vào thông báo rời rạc từ hệ thống. Điều này gây bất tiện, đặc biệt trong các tình huống cần thao tác nhanh như trong giai đoạn thanh toán. Ngoài ra, người dùng cũng gặp khó khăn trong việc theo dõi các voucher đã thu thập, thời hạn sử dụng, và điều kiện áp dụng cụ thể cho từng đơn hàng. Tình trạng "quên dùng" voucher vẫn diễn ra thường xuyên, làm giảm hiệu quả của chương trình khuyến mãi, đồng thời ảnh hưởng đến trải nghiệm người dùng.

Về phía nhà quản trị hệ thống, việc không có một nền tảng quản lý voucher tập trung và trực quan khiến cho quá trình tạo mới, kiểm soát, phân tích hiệu quả, và đo lường hành vi người dùng trở nên thủ công, thiếu chính xác và khó mở rộng. Điều này làm giảm khả năng triển khai các chiến dịch khuyến mãi linh hoạt và mang tính cá nhân hóa theo từng đối tượng mục tiêu.

Chính vì vậy, nhu cầu về một hệ thống voucher toàn diện – từ việc thu thập, theo dõi, áp dụng cho đến quản lý và thống kê – là vô cùng cần thiết. Việc thiết kế một hệ thống như vậy không chỉ giúp tối ưu trải nghiệm người dùng khi mua sắm, mà còn mang lại công cụ mạnh mẽ để các nhà phát triển và quản trị viên dễ dàng điều phối chiến lược khuyến mãi một cách linh hoạt, minh bạch và hiệu quả.

5.6.2 Giải pháp

Để giải quyết vấn đề trên, đầu tiên trong ứng dụng của em khi người dùng mở mục My Vouchers trong phần Profile, người dùng sẽ xem được danh sách các voucher mà người đó chưa thu thập và vẫn còn hiệu lực trong hệ thống. Màn hình có giao diện thân thiện, đẹp mắt, có nút search kết hợp với lọc theo từng loại voucher giúp người dùng nhanh chóng tìm thấy voucher mong muốn. Để tối ưu thời gian phản hồi, hệ thống cũng không hiển thị toàn bộ voucher mà áp dụng chiến lược phân trang, mỗi lần chỉ hiển thị 20 voucher đầu tiên, người dùng muốn xem thêm sẽ click nút loadmore sẽ lại load thêm 20 voucher tiếp theo. Với mỗi voucher sẽ có nút claim để người đó có thể thu thập và chuẩn bị sử dụng.

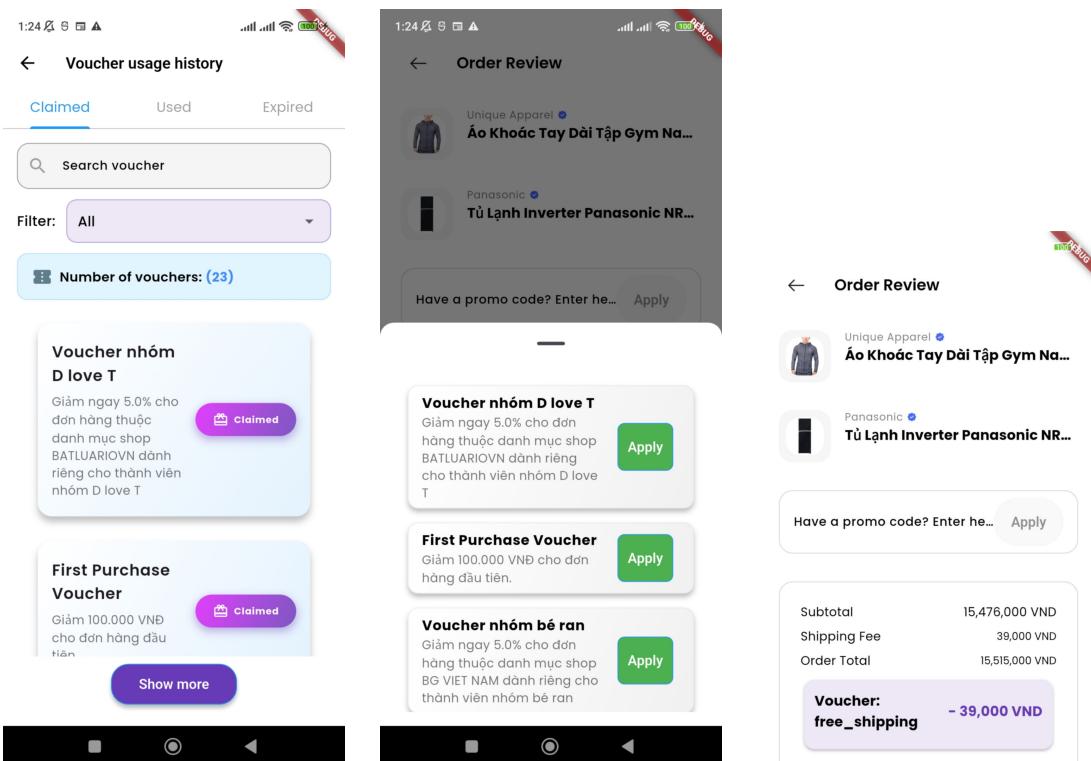
Thứ hai, từ màn hình danh sách voucher có thể điều hướng sang màn hình lịch sử voucher của người dùng. Màn hình này có giao diện gần tương tự màn hình danh sách ở trên, nhưng có chia làm 3 tab là voucher đã thu thập, voucher đã sử dụng và voucher đã hết hạn để người dùng dễ dàng quản lý việc sử dụng của mình.

Thứ ba, ở màn hình checkout, trước khi click thanh toán có nút apply voucher, khi người dùng ấn nút này, thì sẽ hiển thị popup các loại voucher người đó đã thu thập, chưa sử dụng, còn thời hạn hiệu lực và đủ điều kiện áp dụng với điều kiện đơn hàng hiện tại. Khi người dùng chọn một voucher cụ thể và click button apply, hệ thống sẽ tự động tính toán được trừ của đơn hàng sau khi áp dụng, và hiển thị thông tin của voucher vừa áp dụng lên trên giao diện màn hình checkout ngay lập tức. Hệ thống cũng tự động kiểm tra nếu giá trị đơn hàng đã về 0 hoặc đã áp dụng 1 voucher 2 lần(ví dụ voucher freeship) cũng sẽ có cảnh báo và không tiến hành áp dụng để không lãng phí voucher

Thứ tư, ở phần web admin, quản trị viên có thể quản lý CRUD các voucher trong hệ thống, cũng như xem thống kê các voucher theo type, theo lịch sử người dùng đã áp dụng.

5.6.3 Kết quả

Dưới đây em sẽ minh họa các giao diện liên quan đến nhóm chức năng này: màn hình lịch sử voucher, popup apply voucher và màn hình chekout khi áp dụng thành công 1 voucher



Hình 5.34: Màn hình danh sách lịch sử các voucher của người dùng có phân theo 3 nhóm khác nhau

Hình 5.35: Pop up gợi ý những voucher đủ điều kiện để người dùng sử dụng với điều kiện đơn hàng hiện tại

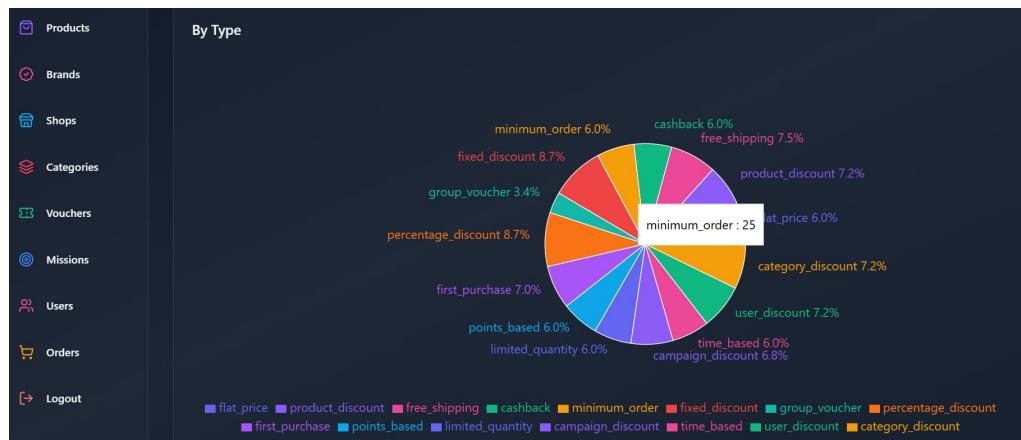
Hình 5.36: Màn hình checkout khi apply thành công 1 voucher

CHƯƠNG 5. CÁC GIẢI PHÁP VÀ ĐÓNG GÓP NỔI BẬT

Title	Description	Discount	Condition	Quantity	Status	Validity	Created At	Actions
Fixed Discount	Giảm 50.000 VND cho đơn hàng.	Giảm 50.002 VND	Min Order: 10,000 VND	1497 / 1500	Active	Từ 22:13:12 21/01/2025 đến 22:13:12 27/01/2025	22:13:12 21/01/2025	
First Purchase Voucher	Giảm 100.000 VND cho đơn hàng đầu tiên.	Giảm 100.008 VND	Min Order: 40,000 VND	2998 / 3000	Active	Từ 22:13:25 21/01/2025 đến 22:13:25 25/01/2025	22:13:25 21/01/2025	
Category-specific Voucher	Giảm 20% cho các sản phẩm thuộc danh mục Thời Trang Nam.	20% 0	Không yêu cầu	998 / 1000	Active	Từ 22:13:16 21/01/2025 đến 22:13:16 21/01/2026	22:13:16 21/01/2025	

Hình 5.37: Màn hình web admin CRUD voucher

Hình 5.37 hiển thị màn hình quản trị viên CRUD voucher, ngoài ra có thể search kết hợp lọc voucher theo type



Hình 5.38: Màn hình thống kê voucher theo từng type



Hình 5.39: Màn hình thống kê voucher theo lịch sử áp dụng của người dùng

Như vậy là ở chương 5 này, em đã trình bày những giải pháp và đóng góp nổi bật của mình trong ứng dụng. Sau đây, em xin phép được trình bày chương 6 - Kết luận và hướng phát triển.

CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

6.1 Kết luận

Trong khi tiến hành thiết kế, xây dựng và triển khai hệ thống ecommerce cho cả mobile và web admin này, vì để thực hiện các chức năng đáp ứng các nhu cầu của hệ thống, bản thân em cũng đã được tiếp xúc với nhiều loại công nghệ mới mà em chưa bao giờ được biết trước đây. Nhờ vậy, mà em đã xây dựng thành công một hệ thống có những đặc điểm sau:

- Xây dựng thành công ứng dụng flutter mobile thương mại điện tử với giao diện đẹp mắt, thân thiện, đầy đủ các tính năng cơ bản của một hệ thống thương mại điện tử, ngoài ra đã phát triển được một số tính năng, đóng góp nổi bật, có sự khác biệt với các sản phẩm trên thị trường như tính năng chatbot tự động hóa quy trình mua sắm, gợi ý cá nhân hóa theo sở thích, lịch sử người dùng, tính năng săn sale theo nhóm, hệ thống nhiệm vụ đầy thử thách.
- Deploy được một số backend API như backend cho train tự động mô hình gợi ý, API cho chatbot [7] gọi đến lên [15], có lịch tự động chạy các API.
- Sử dụng các dịch vụ của nền tảng Firebase từ Firestore để lưu trữ database [9], Firebase Authentication để xác thực đăng nhập bằng Google [8], Firebase Storage để lưu trữ ảnh, video [16], Firebase Cloud Messaging để gửi thông báo realtime [5], và Firebase App Distribution để người dùng tải ứng dụng và trải nghiệm [14]. Đây đều là những kiến thức và trải nghiệm mới đầy thú vị.
- Xây dựng được trang web dành cho quản trị viên với giao diện đẹp mắt sử dụng [2], backend sử dụng [3]
- Khả năng tự nghiên cứu và khai thác các mô hình AI hiện đại, đặc biệt là mô hình gợi ý LightFM [6], thông qua nền tảng Kaggle [13] và ngôn ngữ lập trình Python [11], từ đó từng bước xây dựng quy trình pipeline để xử lý dữ liệu, huấn luyện mô hình và đánh giá hiệu quả gợi ý. Việc áp dụng thư viện [6] không chỉ giúp tăng độ chính xác trong gợi ý mà còn mở rộng kiến thức về hybrid recommendation systems kết hợp collaborative filtering và content-based filtering.

Tuy nhiên, song song với những điểm mạnh ấy, ứng dụng vẫn còn một số những khuyết điểm như:

- Một số request, truy vấn database còn khá chậm ảnh hưởng đến trải nghiệm người dùng, cần tối ưu nhất có thể để mọi response của hệ thống chỉ phản hồi tối đa 1-2s.

- Ứng dụng hiện chỉ mới được xây dựng dành riêng cho thiết bị điện thoại thông thường, giao diện chưa tối ưu và thích ứng tốt trên các phiên bản màn hình lớn như tablet.
- Chatbot hiện tại vẫn hiểu được khá ít các mẫu câu tự nhiên của người dùng, cần huấn luyện thêm để phản hồi tự nhiên, chính xác hơn với nhiều câu nhập khác nhau của người dùng hơn
- Cần thu thập nhiều data hơn, so sánh thêm nhiều mô hình gợi ý khác ngoài[6] để càng ngày càng cải thiện tính năng gợi ý tốt hơn nữa

6.2 Hướng phát triển

6.2.1 Tính năng Livestream kết hợp Đấu giá ngược

Trong tương lai, em mong muốn tích hợp cơ chế **đấu giá ngược** vào tính năng livestream để nâng cao mức độ tương tác giữa người bán và người mua. Cụ thể, trong quá trình livestream, người bán có thể kích hoạt chế độ giảm giá dần theo thời gian cho một sản phẩm cụ thể. Người mua có thể theo dõi giá giảm theo thời gian thực và quyết định “chốt đơn” tại thời điểm phù hợp. Càng chờ lâu, giá càng rẻ – nhưng nguy cơ sản phẩm bị người khác mua trước cũng cao hơn. Đây là một cơ chế giải trí, kích thích FOMO (sợ bỏ lỡ) và tạo trải nghiệm mua sắm thú vị, khác biệt hoàn toàn so với hình thức flash sale truyền thống. Việc triển khai đấu giá ngược cần kết hợp giữa cập nhật real-time, UI động và xử lý đồng thời nhiều phiên livestream khác nhau.

6.2.2 Tính năng Trợ lý ảo gợi ý phối đồ (AI Stylist)

Để nâng cao trải nghiệm cá nhân hóa trong lĩnh vực thời trang, em hướng tới phát triển một **trợ lý ảo gợi ý phối đồ** sử dụng AI. Tính năng này sẽ học từ lịch sử mua hàng, hành vi xem sản phẩm và xu hướng thời trang để đưa ra gợi ý phối đồ phù hợp theo mùa, sự kiện hoặc phong cách riêng của người dùng. Kết hợp với công nghệ thực tế tăng cường (AR), người dùng còn có thể thử đồ trực tiếp trên camera thông qua “phòng thử đồ ảo”. Đây là bước đột phá giúp cá nhân hóa trải nghiệm người dùng ở mức cao, đồng thời góp phần giảm tỷ lệ hoàn trả sản phẩm vì không vừa ý. Việc xây dựng hệ thống cần tích hợp recommendation system, xử lý hình ảnh thời gian thực và UI AR hiện đại.

6.2.3 Tính năng Trợ lý mua sắm thông minh tích hợp AI

Bên cạnh các chức năng cơ bản, một hướng đi mới đầy tiềm năng là xây dựng **trợ lý mua sắm thông minh** (Smart Shopping Assistant) tích hợp AI để đồng hành cùng người dùng trong toàn bộ hành trình mua sắm. Trợ lý này có thể:

- Phân tích thói quen mua sắm và đưa ra gợi ý phù hợp theo nhu cầu.

- Gợi ý thời điểm mua tốt nhất dựa trên lịch sử giá sản phẩm.
- Nhắc người dùng nếu sản phẩm yêu thích sắp hết hàng hoặc đang có khuyến mãi.
- Hỗ trợ người dùng tìm sản phẩm thông qua giọng nói hoặc hình ảnh.

Việc triển khai tính năng này cần kết hợp nhiều thành phần công nghệ như NLP (Natural Language Processing) để xử lý ngôn ngữ tự nhiên, computer vision để nhận diện hình ảnh sản phẩm, và mô hình học máy để dự đoán hành vi người dùng. Đây sẽ là yếu tố then chốt nâng tầm trải nghiệm cá nhân hóa và đưa nền tảng tới gần hơn với xu thế thương mại điện tử thông minh (Smart e-Commerce).

TÀI LIỆU THAM KHẢO

- [1] Google, *Flutter*, Google. [Online]. Available: <https://flutter.dev> (visited on 06/16/2025).
- [2] Meta Platforms, Inc., *React – a javascript library for building user interfaces*, Meta Platforms, Inc. [Online]. Available: <https://reactjs.org> (visited on 06/16/2025).
- [3] VMware, Inc., *Spring boot - spring framework*, VMware, Inc. [Online]. Available: <https://spring.io/projects/spring-boot> (visited on 06/16/2025).
- [4] Google, *Firebase*, Google. [Online]. Available: <https://firebase.google.com> (visited on 06/16/2025).
- [5] Google, *Firebase cloud messaging*, Google. [Online]. Available: <https://firebase.google.com/messaging> (visited on 06/16/2025).
- [6] M. Kula, “Metadata embeddings for user and item cold-start recommendations,” in *Proceedings of the 2nd Workshop on New Trends on Content-Based Recommender Systems co-located with 9th ACM Conference on Recommender Systems (RecSys 2015), Vienna, Austria, September 16-20, 2015.*, T. Bogers and M. Koolen, Eds., ser. CEUR Workshop Proceedings, vol. 1448, CEUR-WS.org, 2015, pp. 14–21. [Online]. Available: <http://ceur-ws.org/Vol-1448/paper4.pdf>.
- [7] Rasa Technologies Inc., *Rasa: Open source conversational ai*, Rasa Technologies Inc. [Online]. Available: <https://rasa.com> (visited on 06/16/2025).
- [8] Google, *Firebase authentication*, Google. [Online]. Available: <https://firebase.google.com/auth> (visited on 06/16/2025).
- [9] Google, *Firebase firestore*, Google. [Online]. Available: <https://firebase.google.com/firestore> (visited on 06/16/2025).
- [10] Google, *Dart programming language*, Google. [Online]. Available: <https://dart.dev> (visited on 06/16/2025).
- [11] P. S. Foundation, *Python programming language*, Python Software Foundation. [Online]. Available: <https://www.python.org> (visited on 06/16/2025).
- [12] O. Foundation, *Node.js*, OpenJS Foundation. [Online]. Available: <https://nodejs.org> (visited on 06/16/2025).
- [13] Kaggle, *Kaggle: Your machine learning and data science community*, Kaggle. [Online]. Available: <https://www.kaggle.com> (visited on 06/16/2025).
- [14] Google, *Firebase app distribution*, Google. [Online]. Available: <https://firebase.google.com/distribution> (visited on 06/16/2025).

- [15] Google, *Google cloud run*, Google. [Online]. Available: <https://cloud.google.com/run> (visited on 06/16/2025).
- [16] Google, *Firebase storage*, Google. [Online]. Available: <https://firebase.google.com/products/storage> (visited on 06/16/2025).

PHỤ LỤC