

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH



ĐỒ ÁN TỐT NGHIỆP
PHÁT TRIỂN ỨNG DỤNG MUA
BÁN THỰC PHẨM DƯ THỪA

Ngành : Khoa học máy tính

HỘI ĐỒNG:

GVPB:

TKHD:

GVHD: ThS. Lê Đình Thuận

—oOo—


SVTH: Lương Đức Dũng (1912954)

Nguyễn Đăng Tú (1912384)

TP. Hồ Chí Minh, Tháng 07/2023

Lời cam đoan

Nhóm xin cam đoan đây là công trình nghiên cứu của riêng nhóm dưới sự hướng dẫn của ThS. Lê Đình Thuận. Nội dung phân tích và thiết kế hệ thống của nhóm đều là trung thực. Các lý thuyết được sử dụng cho quá trình phân tích, xây dựng được chính nhóm thu thập từ nhiều nguồn khác nhau và sẽ được ghi rõ trong phần tài liệu tham khảo. Ngoài ra, nhóm cũng có sử dụng một số nhận xét, đánh giá và số liệu của các tác giả khác, cơ quan tổ chức khác. Tất cả đều có trích dẫn và chú thích nguồn gốc. Nếu phát hiện có bất kì sự gian lận nào, nhóm xin hoàn toàn chịu trách nhiệm về nội dung đồ án chuyên ngành của mình. Trường đại học Bách Khoa thành phố Hồ Chí Minh không liên quan đến những vi phạm tác quyền, bản quyền do nhóm gây ra trong quá trình thực hiện.

KÝ TÊN: 

NGÀY: _____

KÝ TÊN: 

NGÀY: _____

Lời cảm ơn

Nhóm xin gửi lời cảm ơn đến giáo viên hướng dẫn - Thầy Lê Đình Thuận, người đã tận tâm hướng dẫn và hỗ trợ các vấn đề của nhóm trong suốt quá trình thực hiện đề tài. Nhờ những hướng dẫn và kinh nghiệm thầy chia sẻ mà nhóm đã hoàn thành đồ án tốt nghiệp này tốt nhất có thể.

Cuối cùng, nhóm xin gửi lời cảm ơn đến quý thầy cô trong Khoa Khoa học và Kỹ thuật máy tính đã dạy dỗ và truyền đạt những kiến thức quý báu cùng với những kinh nghiệm làm việc trong suốt những năm qua. Những điều ấy là không chỉ giúp nhóm ở trong đồ án này mà còn là hành trang vô giá đi theo mỗi thành viên suốt cả sự nghiệp sau này.

Trân trọng,
Nhóm thực hiện đề tài.

Tóm tắt đề tài

Nhóm hướng đến mục đích là phát triển một ứng dụng mua bán lương thực thực phẩm dư thừa bằng cách kết nối giữa các cửa hàng và những người dùng có nhu cầu mua thực phẩm giá rẻ, thực phẩm “cận date”. Giúp các cửa hàng xử lý những thực phẩm có hạn sử dụng ngắn hạn, chất lượng giảm đi nhanh chóng và vẫn có thể thu về một khoảng chi phí đáng kể thay vì vứt bỏ chúng và lãng phí, và giúp người tiêu dùng muốn giảm khoảng chi phí cho thực phẩm tiết kiệm một khoản tiền không hề nhỏ. Qua đó, nhóm hy vọng sẽ đạt được mục đích to lớn hơn là góp phần giảm việc sử dụng nguồn nguyên liệu tươi (nguyên liệu chưa qua chế biến), giảm tiêu tốn năng lượng, phát thải khí độc ra môi trường (thông qua đốt chất thải) và cuối cùng giúp giảm đáng kể việc ô nhiễm nước rỉ rác từ việc chôn lấp rác thải.

Project summary

Our goal is to develop a mobile application for buying and selling surplus food by connecting grocer's shop or small business and consumers who are looking for affordable, "near-expiry" food. This helps stores manage items with short shelf life that quickly perish and allows them to recover a significant portion of the cost instead of wasting or discarding them. At the same time, it helps consumers save a considerable amount of money on groceries. Through this initiative, we hope to achieve a greater goal of reducing the use of fresh raw materials (unprocessed ingredients), minimizing energy consumption, reducing emissions into the environment (through waste disposal), and ultimately significantly reducing water pollution from landfilling waste.

Mục lục

Danh sách hình vẽ	7
1 GIỚI THIỆU	9
1.1 Về đề tài	9
1.2 Mục tiêu đề tài	10
1.3 Phạm vi đề tài	10
1.4 Ý nghĩa đề tài	10
1.5 Cấu trúc của đồ án	11
2 GIẢI THUẬT	12
2.1 Thuật toán gợi ý sản phẩm NBCF	12
2.1.1 Ý tưởng	12
3 PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG	15
3.1 Giới thiệu hệ thống	15
3.2 Hệ thống tương tự	16
3.3 Use-case	17
3.3.1 Use-case diagram	17
3.3.2 Use-case description	18
3.4 Activity diagram	44
3.5 Kiến trúc phần mềm	57
3.6 Class diagram	58
3.6.1 Overview	58
3.6.2 Presentation layer	59
3.6.3 Business layer	61
3.6.4 Persistence layer	62
3.7 Database diagram	62
4 HIỆN THỰC HỆ THỐNG	65
4.1 Công nghệ sử dụng	65
4.1.1 Front-end	65
4.1.1.1 React Native	65
4.1.1.2 Redux	66
4.1.2 Back-end	66
4.1.2.1 GraphQL	66
4.1.2.2 NodeJS	67

4.2	Hiện thực giao diện	68
4.2.1	Xác thực người dùng	68
4.2.1.1	Đăng ký	68
4.2.1.2	Đăng nhập	70
4.2.2	Tính năng người dùng	71
4.2.3	Tính năng chủ cửa hàng	76
4.2.4	Tính năng admin	79
4.3	Triển khai hệ thống	81
5	KIỂM THỬ VÀ ĐÁNH GIÁ HỆ THỐNG	85
5.1	Tổng quát	85
5.1.1	Người dùng	85
5.1.1.1	Đăng nhập	85
5.1.1.2	Đăng ký	85
5.1.1.3	Cập nhật thông tin	85
5.1.1.4	Sản phẩm	86
5.1.1.5	Giỏ hàng	86
5.1.1.6	Đơn hàng	86
5.1.1.7	Đánh giá đơn hàng	86
5.1.1.8	Đăng xuất	86
5.1.2	Cửa hàng	86
5.1.2.1	Đăng nhập	86
5.1.2.2	Đăng ký	86
5.1.2.3	Cập nhật thông tin	87
5.1.2.4	Sản phẩm	87
5.1.2.5	Đơn hàng	87
5.1.2.6	Đăng xuất	87
5.2	Chi tiết test case	87
5.2.1	Người dùng	87
5.2.1.1	DNSN01 – Đăng nhập thành công	87
5.2.1.2	DNSN02 – Đăng nhập thất bại	88
5.2.1.3	DKSN01 – Đăng ký thành công	88
5.2.1.4	DKSN02 – Đăng ký thất bại	88
5.2.1.5	CNTTSN01 – Cập nhật thông tin cá nhân	89
5.2.1.6	CNTTSN02 – Cập nhật mật khẩu	89
5.2.1.7	SPSN01 – Xem danh sách sản phẩm	90
5.2.1.8	SPSN02 – Xem chi tiết sản phẩm	90
5.2.1.9	GDSN01 – Thêm vào giỏ hàng	90
5.2.1.10	GHSN02 – Xóa sản phẩm khỏi giỏ hàng	91
5.2.1.11	GHSN03 – Cập nhật số lượng sản phẩm trong giỏ hàng	91
5.2.1.12	GHSN04 – Đặt hàng	91
5.2.1.13	DHSN01 – Xem danh sách các đơn hàng đang xử lý	92
5.2.1.14	DHSN02 – Xem chi tiết đơn hàng đang xử lý	92
5.2.1.15	DHSN03 – Hủy đơn hàng	92
5.2.1.16	DHSN01 – Xem danh sách lịch sử đơn hàng	92

5.2.1.17	DHSN02 - Xem chi tiết lịch sử đơn hàng	93
5.2.1.18	DGDHSN01 – Đánh giá đơn hàng	93
5.2.1.19	CHDXSN01 – Đăng xuất thành công	93
5.2.2	Cửa hàng	93
5.2.2.1	CHDNSN01 – Đăng nhập thành công	93
5.2.2.2	CHDNSN02 – Đăng nhập thất bại	94
5.2.2.3	CHDKSN01 – Đăng ký thành công	94
5.2.2.4	CHDKSN02 – Đăng ký thất bại	95
5.2.2.5	CHCNTTSN01 – Cập nhật thông tin cá nhân . . .	95
5.2.2.6	CHCNTTSN02 – Cập nhật mật khẩu	96
5.2.2.7	CHSPSN01 – Xem danh sách sản phẩm	96
5.2.2.8	CHSPSN02 – Thêm sản phẩm	97
5.2.2.9	CHSPSN03 – Sửa thông tin sản phẩm	97
5.2.2.10	CHSPSN04 – Xóa sản phẩm	98
5.2.2.11	CHDHSN01 – Xem danh sách các đơn hàng đang xử lý	98
5.2.2.12	CHDHSN02 - Xem chi tiết đơn hàng đang xử lý .	98
5.2.2.13	CHDHSN03 – Hoàn thành đơn hàng	98
5.2.2.14	CHDHSN04 – Hủy đơn hàng	99
5.2.2.15	CHDHSN05 – Xem danh sách lịch sử đơn hàng . .	99
5.2.2.16	DHSN06 - Xem chi tiết lịch sử đơn hàng	99
5.2.2.17	CHCHDXSN01 – Đăng xuất thành công	100
6	TỔNG KẾT	102
6.1	Tổng kết kết quả đạt được và những hạn chế của đề tài	102
6.1.1	Kết quả đạt được	102
6.1.2	Các điểm hạn chế	102
6.2	Hướng phát triển	102

Danh sách hình vẽ

3.1	Use-case diagram cho toàn bộ hệ thống	17
3.2	Activity cho tính năng Đặt hàng	44
3.3	Activity cho tính năng Đăng ký người dùng	45
3.4	Activity cho tính năng Đăng nhập người dùng	46
3.5	Activity cho tính năng Đăng xuất	47
3.6	Activity cho tính năng Nhấn tin	48
3.7	Activity cho tính năng Đánh giá đơn hàng	49
3.8	Activity cho tính năng Đăng ký chủ cửa hàng	50
3.9	Activity cho tính năng Thêm sản phẩm	51
3.10	Activity cho tính năng Cập nhật sản phẩm	52
3.11	Activity cho tính năng Xóa sản phẩm	53
3.12	Activity cho tính năng Cập nhật thông tin	54
3.13	Activity cho tính năng Thay đổi mật khẩu	55
3.14	Activity cho tính năng Hoàn thành đơn hàng	56
3.15	Activity cho tính năng Hủy đơn hàng	57
3.16	Các lớp tiêu chuẩn trong kiến trúc n-tier	58
3.17	Class diagram cho toàn bộ hệ thống	59
3.18	Presentation layer	59
3.19	Một số tính năng xác thực người dùng	59
3.20	Một số tính năng của người dùng	60
3.21	Một số tính năng quản lý sản phẩm của chủ cửa hàng	60
3.22	Một số tính năng quản lý đơn hàng của chủ cửa hàng	60
3.23	Các tính năng của admin	60
3.24	Business layer	61
3.25	User business	61
3.26	User feature business	61
3.27	Shop owner business	62
3.28	Admin business	62
3.29	Persistence layer	62
3.30	Schema design của database	64
4.1	Giao diện trang đăng ký người dùng	68
4.2	Giao diện trang đăng ký chủ cửa hàng	68
4.3	Giao diện trang đăng ký	69
4.4	Giao diện trang đăng ký	69
4.5	Giao diện trang đăng ký	69

4.6	Giao diện trang đăng ký	69
4.7	Giao diện trang đăng nhập	70
4.8	Giao diện trang đăng nhập	70
4.9	Giao diện trang đăng nhập	70
4.10	Giao diện trang nhập OTP	71
4.11	Giao diện trang quên mật khẩu	71
4.12	Giao diện trang chủ	71
4.13	Giao diện trang sản phẩm	71
4.14	Giao diện trang sản phẩm	72
4.15	Giao diện trang sản phẩm	72
4.16	Giao diện trang sản phẩm - chi tiết	72
4.17	Giao diện trang sản phẩm - thông tin cửa hàng	73
4.18	Giao diện trang đặt hàng	73
4.19	Giao diện trang giỏ hàng	73
4.20	Giao diện trang đặt hàng	73
4.21	Giao diện trang giỏ hàng	74
4.22	Giao diện trang giỏ hàng	74
4.23	Giao diện trang thông tin người dùng	74
4.24	Giao diện trang chỉnh sửa thông tin người dùng	74
4.25	Giao diện trang đổi mật khẩu người dùng	75
4.26	Giao diện trang tin nhắn	75
4.27	Giao diện trang tin nhắn	75
4.28	Giao diện trang thông tin chủ cửa hàng	76
4.29	Giao diện trang chỉnh sửa thông tin chủ cửa hàng	76
4.30	Giao diện trang đổi mật khẩu chủ cửa hàng	77
4.31	Giao diện trang sản phẩm chủ cửa hàng	77
4.32	Giao diện trang chủ chủ cửa hàng	77
4.33	Giao diện trang thêm sản phẩm	77
4.34	Giao diện trang thêm sản phẩm	78
4.35	Giao diện trang quản lý đơn hàng	78
4.36	Giao diện trang quản lý đơn hàng	78
4.37	Giao diện trang quản lý đơn hàng	78
4.38	Giao diện trang quản lý đơn hàng	79
4.39	Giao diện trang thống kê doanh thu	79
4.40	Giao diện trang đăng nhập admin	79
4.41	Giao diện trang phê duyệt tài khoản	80
4.42	Giao diện trang quản lý cửa hàng	80
4.43	Giao diện trang quản lý doanh thu	81
4.44	Cấu trúc thư mục mã nguồn	83

Chương 1

GIỚI THIỆU

1.1 Về đề tài

Lãng phí thực phẩm vẫn luôn là vấn đề chung của toàn thế giới, khi mà nó đang dần có xu hướng gia tăng theo từng ngày. Theo số liệu của Tổ chức Lương thực và Nông nghiệp Liên hợp quốc (FAO), 14% thực phẩm bị thất thoát trong chuỗi cung ứng trước khi bán lẻ và 17%¹ thực phẩm khác bị lãng phí trong bán lẻ và tiêu dùng, những con số này tương đương với 400 tỷ đô la - một con số khủng khiếp. Nếu lượng thức ăn lãng phí này được sử dụng hay tiết kiệm thì nó có thể nuôi sống 2 tỷ người mỗi ngày và nạn đói ở các quốc gia nghèo đói có thể sẽ bị xóa bỏ. Cụ thể, số lương thực trên đã bị hết hạn hoặc bị vứt bỏ trước khi đến tay người tiêu dùng. Lucie Basch, người đồng sáng lập ứng dụng chống lãng phí thực phẩm Too Good To Go, thậm chí đã đánh giá rằng tình trạng này có xu hướng xấu đi trong những năm gần đây, khi tỷ lệ này lên tới 40%². Thêm vào đó, quá trình sản xuất thực phẩm chưa từng được tiêu thụ này đã tạo ra khoảng 10% lượng phát thải khí nhà kính toàn cầu.

Lãng phí thực phẩm là vấn đề không của riêng ai, nhất là khi nước ta cũng đóng góp một phần không nhỏ vào những hệ quả nêu trên khi mà chúng ta đứng thứ 2 về chỉ số lãng phí thực phẩm trong khu vực. Thêm vào đó, theo một khảo sát của Bộ Nông nghiệp và Phát triển nông thôn năm 2020, tỷ lệ thất thoát thực phẩm, nông sản trước chế biến trung bình của trái cây Việt Nam là 10%, rau củ là 20-50%, thủy, hải sản là 30-35%. Tổn thất lương thực vào khoảng 10-15%³. Từ đó có thể thấy, các cửa hàng bán rau củ quả và thậm chí là các nhà hàng bán đồ ăn cũng góp một phần không nhỏ vào sự lãng phí trên.

Từ đó, chúng tôi quyết định phát triển một ứng dụng giúp giảm lượng thức ăn bị lãng phí cho đối tượng là những cửa hàng lương thực, thực phẩm. Mục tiêu là xây dựng một hệ thống giúp người bán hàng đăng bán các sản phẩm “cận date” trên cho khách hàng với mức giá hời hơn khá nhiều so với giá gốc. Qua đó giúp cắt cửa hàng giảm một phần tác nhân gây lãng phí lương thực và người tiêu dùng

¹<https://www.fao.org/nutrition/capacity-development/food-loss-and-waste/en/>, truy cập ngày 20/07/2023

²<https://en.hectar.co/articles/too-good-to-go-limpart-anti-gasps>, truy cập ngày 20/07/2023

³<https://thesaigontimes.vn/lang-phi-thuc-pham-bai-toan-kho-cua-phat-trien-ben-vung-tai-viet-nam/>, truy cập ngày 20/07/2023

cũng tiết kiệm được một khoản tiền trong khi chất lượng sản phẩm vẫn ở mức khá cho đến mức chấp nhận được.

1.2 Mục tiêu đề tài

Ứng dụng mua bán lương thực thực phẩm dư thừa của chúng tôi hướng đến việc kết nối giữa các cửa hàng và những người dùng có nhu cầu mua thực phẩm giá rẻ, thực phẩm “cận date”. Giúp các cửa hàng xử lý những thực phẩm có hạn sử dụng ngắn hạn, chất lượng giảm đi nhanh chóng và vẫn có thể thu về một khoảng chi phí đáng kể thay vì vứt bỏ chúng và lãng phí, và giúp người tiêu dùng muốn giảm khoảng chi phí cho thực phẩm tiết kiệm một khoản tiền không hề nhỏ. Ngoài ra, mục đích chính của ứng dụng là góp phần vào việc bảo vệ môi trường cũng được thực hiện thông qua việc giảm đi lượng rác thải ra ngoài thông qua việc giảm lượng thực phẩm, lương thực bị bỏ phí. Qua đó, nhóm hy vọng sẽ đạt được mục đích to lớn hơn là góp phần giảm việc sử dụng nguồn nguyên liệu tươi (nguyên liệu chưa qua chế biến), giảm tiêu tốn năng lượng, phát thải khí độc ra môi trường (thông qua đốt chất thải) và cuối cùng giúp giảm đáng kể việc ô nhiễm nước rỉ rác từ việc chôn lấp rác thải.

1.3 Phạm vi đề tài

Trong phạm vi luận văn này, chúng tôi chủ yếu thực hiện các module chính sau:

- Cung cấp cho các chủ cửa hàng một nền tảng cho phép đăng bán sản phẩm đúng theo tiêu chí của ứng dụng, kèm với các thông tin về khuyến mãi và cửa hàng để tiếp cận người dùng.
- Cung cấp cho người dùng một nền tảng mua bán thực phẩm, lương thực giá rẻ nhưng vẫn phần nào đảm bảo chất lượng thông qua thông tin đánh giá, review về sản phẩm.
- Cung cấp phương thức thanh toán online nhanh chóng và tiện lợi.
- Gợi ý cho người dùng các sản phẩm phù hợp với họ thông qua nhiều tiêu chí như đánh giá của người dùng khác về sản phẩm, số lượng sản phẩm đã bán ra, khoảng cách tới cửa hàng, các sản phẩm người dùng đã mua trước đó, ...
- Cho phép người dùng lọc các sản phẩm theo khoảng cách, đánh giá, loại thực phẩm, giá cả, ...
- Cho phép người dùng hiển thị các cửa hàng trên bản đồ một cách trực quan

1.4 Ý nghĩa đề tài

Với mục tiêu đề tài lần này, chúng tôi muốn phát triển một ứng dụng giúp giải quyết lượng lương thực, thực phẩm dư thừa, qua đó góp phần vào mục đích to lớn

hơn là bảo vệ môi trường, giảm chi phí xử lý chất thải và giúp chủ cửa hàng cũng như người dùng tiết kiệm một khoảng chi phí đáng kể thông qua việc tránh lãng phí lương thực, thực phẩm.

1.5 Cấu trúc của đồ án

Cấu trúc đồ án của chúng tôi bao gồm các chương sau:

- **Chương 2:** Mô tả giải thuật giải thuật gợi ý sản phẩm NBCF.
- **Chương 3:** Phân tích và thiết kế hệ thống:
 - Giới thiệu hệ thống: nêu ra các yêu cầu người dùng cần để thực hiện hệ thống, bao gồm các yêu cầu chức năng và phi chức năng.
 - Hệ thống tương tự: phân tích các hệ thống tương tự và so sánh với ứng dụng của nhóm.
 - Use-case: bao gồm usecase diagram của toàn bộ hệ thống và đặc tả usecase.
 - Activity: vẽ luồng thực thi những tính năng chính của hệ thống.
 - Kiến trúc phần mềm: mô tả kiến trúc của hệ thống, lí do vì sao lựa chọn kiến trúc này.
 - Class diagram: class diagram theo mô hình phân lớp của toàn bộ hệ thống.
 - Database diagram: phân tích, lựa chọn hệ cơ sở dữ liệu cho hệ thống và vẽ thiết kế database cho hệ thống.
- **Chương 4:** Hiện thực hệ thống:
 - Mô tả các công nghệ được sử dụng ở trong đồ án, bao gồm front-end, back-end và database.
 - Các màn hình mà nhóm đã hiện thực được.
 - Cách triển khai hệ thống.
- **Chương 5:** Kiểm thử và đánh giá hệ thống:
 - Kiểm thử hệ thống: đánh giá hệ thống bằng exception test.
- **Chương 6:** Tổng kết:
 - Tổng kết các kết quả đạt được, cũng như các hạn chế còn tồn tại của đề tài.
 - Hướng phát triển: định hướng phát triển thêm của hệ thống trong tương lai.

Chương 2

GIẢI THUẬT

Chương này trình bày cơ sở lý thuyết và hiện thực thuật toán NBCF của nhóm cùng với kết quả đã đạt được.

2.1 Thuật toán gợi ý sản phẩm NBCF

2.1.1 Ý tưởng

Hệ thống gợi ý (Recommender System) là một thành phần trong hệ thống thông tin. Mục đích của nó là hỗ trợ người dùng tìm kiếm được đúng thông tin cần thiết, dự đoán sở thích hay xếp hạng mà người dùng có thể dành cho một mục thông tin nào đó mà họ chưa xem xét tới trong quá khứ.

Ở đây nhóm sẽ đề cập đến Neighborhood-based Collaborative Filtering hay viết tắt là NBCF.

Ý tưởng cơ bản của NBCF là xác định mức độ quan tâm của một người dùng (user) tới một thông tin dựa trên các người dùng khác gần giống với người dùng này. Độ giống nhau giữa các người dùng có thể được xác định thông qua mức độ quan tâm của các người dùng này tới các thông tin khác mà hệ thống đã biết. Việc xác định mức độ quan tâm của mỗi người dùng tới một thông tin dựa trên mức độ quan tâm của các người dùng tương đồng tới thông tin đó còn được gọi là User-user collaborative filtering.

Việc cài đặt thuật toán sẽ gồm các bước sau:

- Similarity Functions: Xác định sự giống nhau giữa hai người dùng
- Cosine Similarity: Hàm để đo sự giống nhau được nhắc đến ở đây là **Cosine Similarity**

$$\text{Cos}\theta = \frac{\vec{a} \cdot \vec{b}}{||\vec{a}|| ||\vec{b}||} = \frac{\sum_1^n a_i b_i}{\sqrt{\sum_1^n a_i^2} \sqrt{\sum_1^n b_i^2}} \quad (2.1)$$

trong đó $\vec{a} \cdot \vec{b} = \sum_1^n a_i b_i = a_1 b_1 + a_2 b_2 + \dots + a_n b_n$ là kết quả phép nhân hai vector.

- Chuẩn hóa dữ liệu: tính giá trị trung bình ở bước trên cho mỗi người dùng. Giá trị cao tương ứng với các người dùng dễ tính và ngược lại. Khi đó, nếu

tiếp tục trừ từ mỗi rating đi giá trị này và thay các giá trị chưa biết bằng 0 ta sẽ được normalized utility matrix

- Xác định độ quan tâm: Công thức phổ biến được sử dụng để dự đoán rating của u cho i là

$$\hat{y}_{i,u} = \frac{\sum_{u_j \in N_{(u,i)}} \vec{y}_{i,u_j} \text{sim}(u, u_j)}{\sum_{u_j \in N_{(u,i)}} |\text{sim}(u, u_j)|} \quad (2.2)$$

Get Similarity Functions: (1)

```
function getSimilarity($matrix, $item, $otherProduct)
{
    $vectorUser = array();
    $vectorOtherUser = array();
    $match = 0;
    foreach ($matrix[$item] as $key => $value) {
        if (array_key_exists($key, $matrix[$otherProduct])) {
            $vectorUser[] = $value;
            $vectorOtherUser[] = $matrix[$otherProduct][$key];
            $match++;
        } else {
            $vectorUser[] = $value;
            $vectorOtherUser[] = 0;
        }
    }

    foreach ($matrix[$otherProduct] as $key => $value) {
        if (array_key_exists($key, $matrix[$item])) { } else {
            $vectorOtherUser[] = $value;
            $vectorUser[] = 0;
        }
    }

    $data = Cosine::similarity($vectorUser, $vectorOtherUser);
    if ($match == 0 || $temp < 0.5) {
        return -1;
    }

    return $data;
}
```

Get Recommendation Functions: (2)

```
function getRecommendation($matrix, $user)
{
    $total = array();
    $simsums = array();
    $ranks = array();
    foreach ($matrix as $otherUser => $value) {
        if ($otherUser != $user) {
            $sim = getSimilarity($matrix, $user, $otherUser);
```

```
// "Similarity between: " . $otherUser . " and " . $user . " is :  
    " . $sim . "<br/>";  
if ($sim == -1) continue;  
foreach ($matrix[$otherUser] as $key => $value) {  
    if (!array_key_exists($key, $matrix[$user])) {  
        if (!array_key_exists($key, $total)) {  
            $total[$key] = 0;  
        }  
        $total[$key] += $matrix[$otherUser][$key] * $sim;  
  
        if (!array_key_exists($key, $simsums)) {  
            $simsums[$key] = 0;  
        }  
        $simsums[$key] += $sim;  
    }  
}  
}  
  
foreach ($total as $key => $value) {  
    $ranks[$key] = $value / $simsums[$key];  
}  
array_multisort($ranks, SORT_DESC);  
return $ranks;  
}
```

Chương 3

PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

Ở chương này sẽ giới thiệu các chức năng của hệ thống, bao gồm các yêu cầu chức năng và phi chức năng và các diagram thể hiện thiết kế hệ thống như là Use-case diagram và Activity diagram. Kèm với đó là thiết kế của hệ thống thể hiện qua mô tả kiến trúc phần mềm, Class diagram và Database diagram.

3.1 Giới thiệu hệ thống

Hiện nay ở Việt Nam chưa có ứng dụng di động nào giúp người dùng tận dụng những sản phẩm “cận date”, dư thừa để giảm lượng rác thải và lãng phí tài nguyên trong khi những nước ở châu Âu đã có nhiều ứng dụng phục vụ mục đích này từ lâu, thậm chí xu hướng này đã bắt đầu ở châu Á mà điển hình là Thái Lan với ứng dụng **Yindii**¹ hay Singapore là **Makan Rescue**². Do đó nhóm muốn ứng dụng của mình sẽ là tiền đề cho việc nhân rộng việc chống lãng phí đồ ăn trên nền tảng công nghệ số để tiếp cận được nhiều người hơn, qua đó tạo một thói quen tiết kiệm lương thực, thực phẩm cho mọi người.

Với định hướng trên cũng như yêu cầu người dùng thông qua việc tham khảo các mô hình đã thành công ở nước khác cũng như các ứng dụng đặt hàng online trên thị trường, chúng tôi xác định các tính năng chính mà hệ thống chúng tôi mang đến cũng như các tính năng hỗ trợ cho các tính năng chính đó bao gồm **đặt hàng, xem các cửa hàng ở lân cận, gợi ý sản phẩm cho người dùng, tính năng người dùng và quản lý cửa hàng**.

Cụ thể hơn về nhóm tính năng đặt hàng, ứng dụng của chúng tôi sẽ có các tính năng đặt hàng, hủy đơn căn bản của một ứng dụng đặt đồ ăn thông thường. Kèm với đó là tính năng giỏ hàng và thanh toán online giúp người dùng thuận tiện trong việc thanh toán đơn hàng. Bên cạnh đó, người dùng cũng có thể đánh giá và xem

¹<https://play.google.com/store/apps/details?id=co.yindii&hl=vi&gl=US>, truy cập ngày 20/07/2023

²<https://play.google.com/store/apps/details?id=com.makanrescueapp&hl=vi&gl=US&pli=1>, truy cập ngày 20/07/2023

các đánh giá về một cửa hàng hoặc sản phẩm để lựa chọn sản phẩm hợp lý. Các cửa hàng nhận nhiều đánh giá thấp cũng sẽ được xem xét xử lý.

Về xem các cửa hàng ở lân cận, chúng tôi cung cấp cho người dùng một danh sách các cửa hàng ở xung quanh địa điểm được chọn để người dùng thuận tiện trong việc lựa chọn sản phẩm mình muốn mua. Địa điểm có thể lấy từ GPS hoặc được người dùng nhập vào.

Về thuật toán gợi ý sản phẩm, thuật toán được nhóm sử dụng là **thuật toán gợi ý sản phẩm NBCF**. Khi bấm vào mục **gợi ý** phía dưới thanh tìm kiếm, ứng dụng sẽ hiển thị các sản phẩm được gợi ý dựa trên thuật toán gợi ý sản phẩm NBCF. Chi tiết hơn về thuật toán này sẽ được nhóm đề cập ở phần **Giải thuật**.

Nhóm tính năng người dùng bao gồm các tính năng quản lý thông tin cá nhân người dùng như họ và tên, số điện thoại, địa chỉ email và mật khẩu, kèm với đó là thông tin thanh toán của khách hàng.

Nhóm tính năng quản lý cửa hàng dành cho chủ cửa hàng quản lý các đơn đặt hàng của khách hàng, nhắn tin trao đổi với khách hàng, quản lý sản phẩm của cửa hàng và xem lại lịch sử đơn hàng cũng như doanh thu của cửa hàng.

Các yêu cầu phi chức năng cũng là một yếu tố quan trọng nhằm tối ưu hóa trải nghiệm người dùng. Các yêu cầu phi chức năng dưới đây sẽ được chúng tôi đảm bảo cho hệ thống của mình:

- Độ trễ khi tương tác với ứng dụng là 0.1s
- Hệ thống chạy được ổn định với lượng truy cập tối đa 10.000 cùng lúc
- Ứng dụng tương thích với cả hai nền tảng Android và iOS
- Hệ thống dễ dàng mở rộng

3.2 Hệ thống tương tự

Trên thực tế, ý tưởng bán các sản phẩm gần hết hạn không phải quá mới lạ ở thị trường Việt Nam. Trong đó, đã có các sàn thương mại điện tử như Shopee, Lazada hay cả Tiki cũng có những người dùng đăng bán các sản phẩm như sữa, đồ đóng hộp nói chung, các loại lương thực tương tự khác khi chúng ở trong tình trạng gần hết hạn sử dụng nhưng vẫn đảm bảo cho sức khỏe người tiêu dùng.

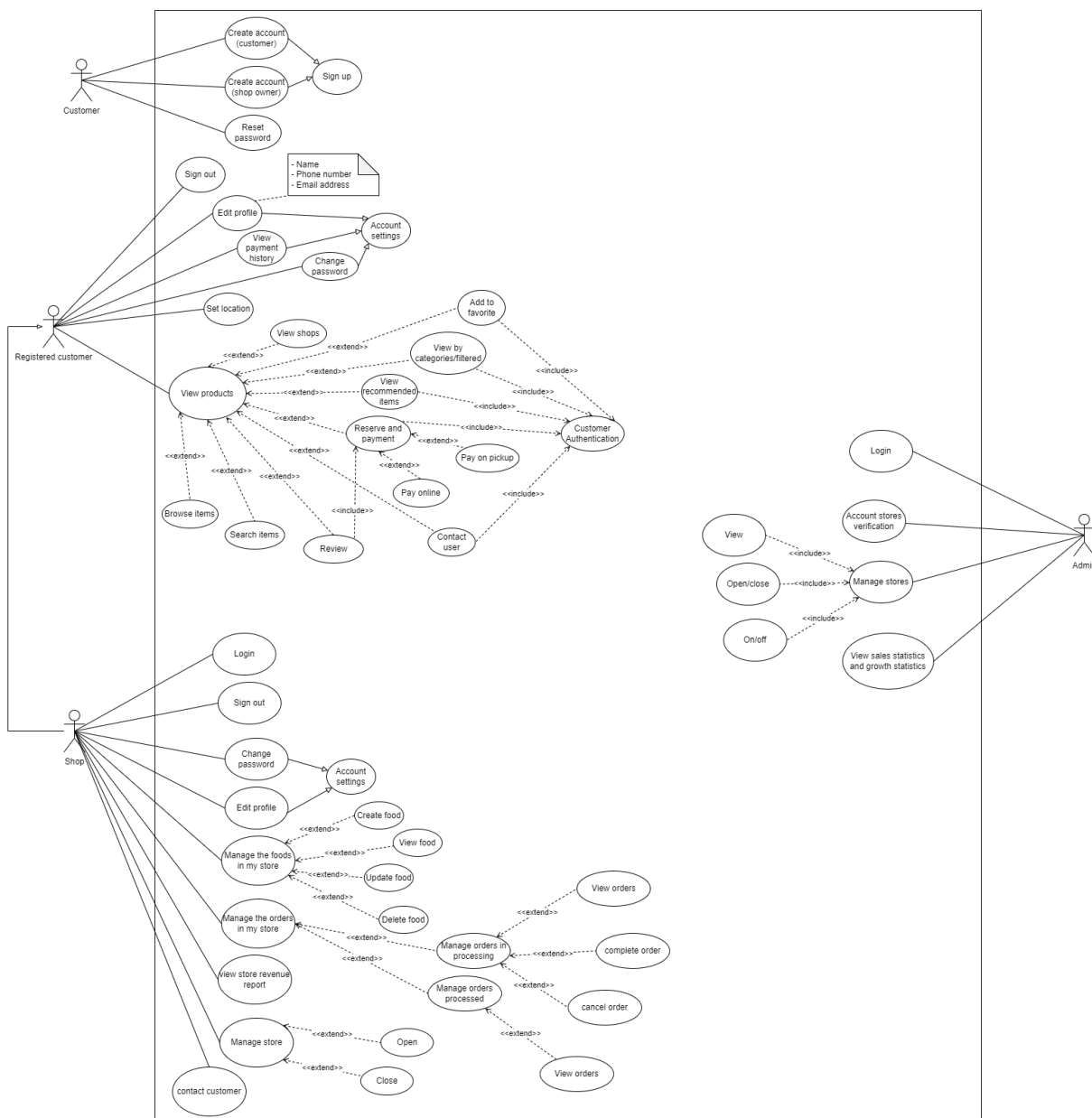
Tuy nhiên, nhược điểm của các sàn thương mại điện tử này là rất khó để đăng bán các sản phẩm có hạn sử dụng từ 01 tới một vài ngày hoặc tệ nhất là trong ngày do khó tiếp cận tới người dùng và hệ thống giao nhận sản phẩm phức tạp và tốn thời gian. Do đó, thị trường này vẫn chưa có một ứng dụng nào như Too Good To Go hay là Yindii với các tiêu chí như khả năng tiếp cận đúng tệp khách hàng, đặt hàng và nhận hàng diễn ra nhanh chóng và cơ chế gợi ý và hiển thị sản phẩm phù hợp. Do đó, nhóm tin rằng ứng dụng của mình sẽ khai thác tốt thị trường này.

3.3 Use-case

Mục này trình bày use-case diagram chung cho toàn hệ thống và các use-case description chi tiết cho từng chức năng.

3.3.1 Use-case diagram

Use-case diagram chung cho toàn bộ hệ thống, gồm các chủ thể là người dùng đã đăng ký, người dùng chưa đăng ký, chủ cửa hàng và admin.



Hình 3.1: Use-case diagram cho toàn bộ hệ thống

Use-case diagram chi tiết: <https://drive.google.com/file/d/19DZqlvnHqgkM1I-uXTHJwkF83bsdx5NH/view?usp=sharing>

3.3.2 Use-case description

- Cập nhật thông tin người dùng: Người dùng có thể thay đổi thông tin cá nhân, cập nhật email, đổi mã pin

Use-case name	Cập nhật thông tin người dùng
Use-case ID	01
Actor	Người dùng đã đăng ký
Description	Người dùng có thể thay đổi thông tin cá nhân, cập nhật email, đổi mã pin
Pre-conditions	Người dùng đã đăng nhập thành công
Post-conditions	Người dùng thay đổi thành công
Normal flow	<ol style="list-style-type: none"> 1. Người dùng bấm vào tab “Tài khoản” ở góc bên phải trên thanh navigation 2. Người dùng chọn thẻ “Chỉnh sửa tài khoản” 3. Hệ thống hiển thị màn hình thông tin người dùng 4. Người dùng bấm vào tên để thay đổi tên người dùng 5. Người dùng bấm “Cập nhật” để lưu lại những thay đổi 6. Hệ thống sẽ ghi nhận những thay đổi của người dùng
Alternative flows	<p>3.a Người dùng muốn thay đổi hoặc thêm email</p> <ol style="list-style-type: none"> 1. Người dùng bấm vào email để thay đổi hoặc thêm email 2. Người dùng bấm “Cập nhật” để lưu lại những thay đổi 3. Hệ thống gửi mail xác nhận đến email trên 4. Người dùng xác nhận email 5. Đi đến bước 5 trong Normal flow
Exceptions	3.a Người dùng bấm “ Hủy ” hoặc bấm ra ngoài khoảng trống

- Đăng ký: Người dùng đăng ký tài khoản trên hệ thống

Use-case name	Đăng ký
Use-case ID	02
Actor	Người dùng
Description	Người dùng đăng ký tài khoản trên hệ thống
Pre-conditions	Số điện thoại của người dùng chưa được đăng ký trên hệ thống
Post-conditions	Tài khoản của người dùng được tạo thành công
Normal flow	<ol style="list-style-type: none">1. Người dùng chưa đăng nhập truy cập vào hệ thống, hệ thống hiển thị màn hình đăng nhập với hai lựa chọn là “Đăng nhập” và “Đăng ký”2. Người dùng nhấn chọn nút “Đăng ký”, hệ thống hiển thị màn hình đăng ký bằng số điện thoại cho người dùng3. Người dùng nhập số điện thoại vào (9 hoặc 10 số). Người dùng chọn “Tiếp tục”4. Hệ thống gửi một mã xác nhận đến số điện thoại trên5. Người dùng nhập mã vào hoặc cấp quyền để hệ thống tự động nhập6. Hệ thống hiển thị màn hình đăng ký tên người dùng7. Người dùng nhập họ tên và email mình vào8. Hệ thống xác thực và kích hoạt tài khoản
Alternative flows	5.a Người dùng chọn “Gửi lại mã xác nhận”
Exceptions	3.a Số điện thoại người dùng đã được đăng ký: <ol style="list-style-type: none">1. Hệ thống tự động đăng nhập tài khoản với số điện thoại đã xác nhận ở trên

- Đăng nhập: Người dùng đăng nhập vào hệ thống

Use-case name	Đăng nhập
Use-case ID	03
Actor	Người dùng đã đăng ký
Description	Người dùng đăng nhập vào hệ thống
Pre-conditions	Người dùng đã đăng ký tài khoản trên hệ thống
Post-conditions	Người dùng đăng nhập vào hệ thống
Normal flow	<ol style="list-style-type: none">1. Người dùng chưa đăng nhập truy cập vào hệ thống, hệ thống hiển thị màn hình đăng nhập với hai lựa chọn là “Đăng nhập” và “Đăng ký”2. Người dùng nhấn chọn nút “Đăng nhập”3. Người dùng nhập số điện thoại và bấm vào nút “Tiếp tục”4. Người dùng nhập mật khẩu và bấm vào nút “Đăng nhập”5. Hệ thống xác thực và ghi nhận đăng nhập, sau đó chuyển đến màn hình sản phẩm
Exceptions	<p>3.a Người dùng nhập sai số điện thoại thì thông báo và nhập lại</p> <p>4.a Nếu người dùng nhập sai mật khẩu thì thông báo và nhập lại</p>

- Đăng xuất: Người dùng đăng xuất khỏi hệ thống

Use-case name	Đăng xuất
Use-case ID	04
Actor	Người dùng
Description	Người dùng đăng xuất khỏi hệ thống
Pre-conditions	Người dùng đã đăng nhập vào hệ thống
Post-conditions	Người dùng đăng xuất khỏi hệ thống
Normal flow	<ol style="list-style-type: none"> 1. Người dùng bấm vào tab “Tài khoản” ở góc bên phải trên thanh navigation 2. Người dùng nhấn chọn “Đăng xuất” 3. Hệ thống hiện một popup cảnh báo người dùng 4. Người dùng chọn “Xác nhận” và đăng xuất khỏi hệ thống
Exceptions	<p>3.c Người dùng chọn “Hủy”:</p> <ol style="list-style-type: none"> 1. Popup đóng lại và người dùng không đăng xuất

- Thay đổi mật khẩu: Người dùng đặt hoặc thay đổi mật khẩu

Use-case name	Thay đổi mật khẩu
Use-case ID	05
Actor	Người dùng đã đăng nhập
Description	Người dùng đặt hoặc thay đổi mật khẩu
Pre-conditions	Người dùng phải đăng nhập vào hệ thống
Post-conditions	Hệ thống ghi nhận mật khẩu mới

Normal flow	<ol style="list-style-type: none">1. Người dùng bấm vào tab “Tài khoản” ở góc bên phải trên thanh navigation2. Người dùng nhấn chọn thẻ “Chỉnh sửa tài khoản”3. Người dùng nhấn chọn thẻ “Đổi mật khẩu”4. Hệ thống hiện màn hình nhập mật khẩu cũ và người dùng nhập mật khẩu cũ5. Hệ thống hiện màn hình mật khẩu mới và người dùng nhập mật khẩu mới6. Hệ thống hiện màn hình xác nhận mật khẩu mới và người dùng nhập lại mật khẩu mới7. Hệ thống ghi nhận mật khẩu mới của người dùng và thông báo cho người dùng là đã thành công
Exceptions	<p>5.a Người dùng nhập sai mật khẩu cũ</p> <ol style="list-style-type: none">1. Hệ thống hiện thông báo người dùng nhập sai mật khẩu2. Quay lại bước 4 trong Normal flow <p>7.a Mật khẩu mới và nhập lại mật khẩu mới không khớp</p> <ol style="list-style-type: none">1. Hệ thống hiện thông báo người dùng nhập sai mật khẩu2. Quay lại bước 5 trong Normal flow

- Đặt hàng: Người dùng đặt hàng và chọn phương thức thanh toán

Use-case name	Đặt hàng
Use-case ID	06
Actor	Người dùng đã đăng ký
Description	Người dùng đặt hàng và chọn phương thức thanh toán

Pre-conditions	Người dùng phải đăng nhập vào hệ thống
Post-conditions	Hệ thống ghi nhận đơn đặt hàng của người dùng
Normal flow	<ol style="list-style-type: none"> 1. Người dùng đã đăng nhập truy cập vào hệ thống, hệ thống hiển thị các sản phẩm được đề xuất 2. Người dùng nhấn chọn sản phẩm mong muốn để xem thêm thông tin về cửa hàng, địa điểm, thời gian cho phép lấy hàng, giá cả, đánh giá, chi tiết sản phẩm 3. Người dùng nhấn nút “Thêm vào giỏ hàng” 4. Người dùng bấm vào “Giỏ hàng” 5. Người dùng điều chỉnh Thông tin đơn hàng trong giỏ hàng 6. Người dùng chọn hình thức thanh toán “Tại cửa hàng” hoặc “Online” 7. Người dùng bấm vào nút đặt hàng 8. Hệ thống hiện popup xác nhận 9. Người dùng bấm vào “Xác nhận” 10. Hệ thống xử lý yêu cầu và thông báo đặt hàng thành công
Alternative flows	<p>6.a Người dùng chọn phương thức thanh toán online:</p> <ol style="list-style-type: none"> 1. Người dùng nhấn vào phương thức thanh toán đang hiển thị 2. Hệ thống hiển thị danh sách các phương thức thanh toán đã được người dùng thêm vào 3. Người dùng chọn phương thức thanh toán mong muốn 4. Hệ thống hiển thị lại màn hình thanh toán với phương thức thanh toán đã chọn

Exceptions	9.a Người dùng không xác thực hoặc bấm quay lại 1. Hệ thống hiện thông báo Xác thực thất bại và quay về màn hình sản phẩm
-------------------	---

- Thay đổi mật khẩu: Người dùng đặt hoặc thay đổi mật khẩu

Use-case name	Thay đổi mật khẩu
Use-case ID	07
Actor	Người dùng đã đăng nhập
Description	Người dùng đặt hoặc thay đổi mật khẩu
Pre-conditions	Người dùng phải đăng nhập vào hệ thống
Post-conditions	Hệ thống ghi nhận mật khẩu mới
Normal flow	<ol style="list-style-type: none">1. Người dùng bấm vào tab “Tài khoản” ở góc bên phải trên thanh navigation2. Người dùng nhấn chọn thẻ “Cài đặt”3. Người dùng nhấn chọn thẻ “Đổi mật khẩu”4. Hệ thống hiện màn hình nhập mật khẩu cũ và người dùng nhập mật khẩu cũ5. Hệ thống hiện màn hình mật khẩu mới và người dùng nhập mật khẩu mới6. Hệ thống hiện màn hình xác nhận mật khẩu mới và người dùng nhập lại mật khẩu mới7. Hệ thống ghi nhận mật khẩu mới của người dùng và thông báo cho người dùng là đã thành công

Exceptions	<p>5.a Người dùng nhập sai mật khẩu cũ</p> <ol style="list-style-type: none">Hệ thống hiện thông báo người dùng nhập sai mật khẩuQuay lại bước 4 trong Normal flow <p>7.a Mật khẩu cũ và mật khẩu mới không khớp</p> <ol style="list-style-type: none">Hệ thống hiện thông báo người dùng nhập sai mật khẩuQuay lại bước 5 trong Normal flow
-------------------	--

- Người dùng nhấn tin cho cửa hàng

Use-case name	Người dùng nhấn tin cho cửa hàng
Use-case ID	08
Actor	Người dùng đã đăng nhập
Description	Người dùng nhấn tin thành công
Pre-conditions	Người dùng phải đăng nhập vào hệ thống và đã mua hàng ở cửa hàng muốn nhấn tin
Post-conditions	Hệ thống ghi nhận tin nhắn mới
Normal flow	<ol style="list-style-type: none">Người dùng bấm vào tab “Đơn hàng” trên thanh navigationNgười dùng nhấn chọn thẻ “Đang xử lý” hoặc “Lịch sử”Người dùng nhấn chọn Nhắn tin trên những thẻ cửa hàng đang hiện (nếu có)Hệ thống hiện màn hình nhấn tin với chủ cửa hàng vừa chọnNgười dùng nhập nội dung muốn nhắn và bấm gửiHệ thống ghi nhận nội dung tin nhắn và thời gian vào hệ thống

- Người dùng đánh giá đơn hàng

Use-case name	Người dùng đánh giá đơn hàng
Use-case ID	09
Actor	Người dùng đã đăng nhập
Description	Người dùng gửi đánh giá thành công
Pre-conditions	Người dùng phải đăng nhập vào hệ thống và đã mua sản phẩm muốn đánh giá
Post-conditions	Hệ thống ghi nhận đánh giá mới
Normal flow	<ol style="list-style-type: none">1. Sau khi người chủ cửa hàng xác nhận đơn hàng thành công, trong mục Đơn hàng nút Hủy đơn sẽ chuyển thành Đánh giá2. Người dùng nhấn chọn nút “Đánh giá”3. Hệ thống hiển thị màn hình đánh giá4. Người dùng nhập vào hình ảnh, số sao và nội dung đánh giá5. Người dùng bấm gửi6. Hệ thống ghi nhận nội dung đánh giá vào hệ thống
Exceptions	<p>5.a Người dùng bấm quay lại</p> <ol style="list-style-type: none">1. Hệ thống hiện màn hình đơn hàng lúc trước

- Đăng nhập dành cho cửa hàng: Nhân viên hoặc chủ cửa hàng có thể đăng nhập và sử dụng các tính năng trong app

Use-case name	Đăng nhập dành cho cửa hàng
Use-case ID	10
Actor	Nhân viên hoặc chủ cửa hàng
Description	Nhân viên hoặc chủ cửa hàng có thể đăng nhập và sử dụng các tính năng trong app
Pre-conditions	Cửa hàng đã đăng ký tài khoản và tài khoản đã được duyệt
Post-conditions	Đăng nhập thành công

Normal flow	<ol style="list-style-type: none"> 1. Người dùng vào trang đăng nhập dành cho cửa hàng 2. Người dùng nhập số điện thoại vào ô “Nhập số điện thoại” 3. Người dùng bấm vào nút “Tiếp tục” 4. Hệ thống kiểm tra và chuyển đến màn hình nhập mật khẩu 5. Người dùng nhập “Mật khẩu” và bấm vào nút “Đăng nhập” 6. Hệ thống xác thực và ghi nhận đăng nhập, sau đó chuyển đến trang chủ
Exceptions	<p>3.a Nếu “Số điện thoại” hoặc “Mật khẩu” sai</p> <ol style="list-style-type: none"> 1 Thông báo cho người dùng 2 Quay lại bước 2 trong Normal flow <p>b Nếu người dùng chưa nhập “Số điện thoại” hoặc mật khẩu:</p> <ol style="list-style-type: none"> 1 Thông báo cho người dùng 2 Quay lại bước nhập lại trường còn thiếu

- Đăng ký dành cho cửa hàng: Chủ cửa hàng có thể đăng ký tài khoản dành cho cửa hàng của mình

Use-case name	Đăng ký dành cho cửa hàng
Use-case ID	11
Actor	Chủ cửa hàng
Description	Chủ cửa hàng có thể đăng ký tài khoản dành cho cửa hàng của mình
Pre-conditions	Có số điện thoại
Post-conditions	Đăng ký thành công

<p>Normal flow</p>	<ol style="list-style-type: none"> 1. Người dùng vào trang đăng ký dành cho cửa hàng 2. Người dùng nhập số điện thoại của mình 3. Người dùng bấm vào nút “Tiếp tục” 4. Hệ thống gửi mã OTP đến số điện thoại người dùng đăng ký 5. Người dùng nhập mã OTP gửi đến tin nhắn trên điện thoại có số điện thoại người dùng đăng ký 6. Người dùng bấm vào nút “Tiếp tục” 7. Người dùng nhập email của mình 8. Người dùng nhập họ và tên của mình 9. Người dùng nhập số cmnd hoặc cccd của mình 10. Người dùng nhập tên cửa hàng của mình 11. Người dùng nhập địa chỉ cửa hàng của mình 12. Người dùng chọn địa chỉ của mình trên bản đồ 13. Người dùng nhập mật khẩu của mình 14. Người dùng nhập lại mật khẩu 15. Người dùng bấm vào nút “Tiếp tục” 16. Người dùng chọn ngân hàng và nhập số tài khoản ngân hàng của mình 17. Người dùng bấm vào nút “Đăng ký” 18. Hệ thống gửi thông báo “Bạn đã gửi đăng ký thành công, vui lòng chờ quản trị viên xác nhận tài khoản của bạn”
<p>Alternative flows</p>	<p>5.a Người dùng bấm trở về để trở về trang đăng ký</p> <p>5.b Nếu người dùng chưa thấy OTP hoặc muốn nhận mã OTP mới thì người dùng bấm vào nút gửi lại OTP</p>

Exceptions	<p>3.a Nếu người dùng nhập số điện thoại sai hoặc số điện thoại trùng với người đăng ký trước thì thông báo cho người dùng và quay lại nhập lại số điện thoại</p> <p>6.a. Nếu người dùng nhập mã OTP sai thì thông báo cho người dùng và nhập lại mã OTP</p> <p>15.a Nếu có trường nào trống thì thông báo cho người dùng và quay lại nhập trường đó</p> <p>15.b Nếu hai mật khẩu không giống nhau thì thông báo cho người dùng và quay lại nhập lại trường đó</p>
-------------------	--

- Xem các đơn hàng khách đã đặt: hủ cửa hàng hoặc nhân viên có thể xem các đơn hàng chưa xử lý

Use-case name	Xem các đơn hàng khách đã đặt
Use-case ID	12
Actor	Chủ cửa hàng hoặc nhân viên
Description	Chủ cửa hàng hoặc nhân viên có thể xem các đơn hàng chưa xử lý
Pre-conditions	Đã đăng nhập
Post-conditions	Xem các đơn hàng khách đã đặt và chưa xử lý
Normal flow	<ol style="list-style-type: none"> 1. Từ trang chủ người dùng bấm vào tab “Đơn hàng” 2. Hệ thống hiển thị danh sách đơn hàng chưa xử lý

- Xem chi tiết đơn hàng: Chủ cửa hàng hoặc nhân viên có thể xem chi tiết đơn hàng chưa xử lý

Use-case name	Xem chi tiết đơn hàng
Use-case ID	13
Actor	Chủ cửa hàng hoặc nhân viên
Description	Chủ cửa hàng hoặc nhân viên có thể xem chi tiết đơn hàng chưa xử lý
Pre-conditions	Đã đăng nhập
Post-conditions	Xem chi tiết đơn hàng

Normal flow	<ol style="list-style-type: none"> 1. Từ trang chủ người dùng bấm vào tab “Đơn hàng” 2. Hệ thống hiển thị danh sách đơn hàng chưa xử lý 3. Bấm vào đơn hàng muốn xem 4. Hệ thống hiển thị chi tiết đơn hàng
--------------------	--

- Hủy đơn hàng: Chủ cửa hàng hoặc nhân viên có thể hủy các đơn hàng chưa xử lý

Use-case name	Hủy đơn hàng
Use-case ID	14
Actor	Chủ cửa hàng hoặc nhân viên
Description	Chủ cửa hàng hoặc nhân viên có thể hủy các đơn hàng chưa xử lý
Pre-conditions	Đã đăng nhập
Post-conditions	Hủy đơn hàng thành công
Normal flow	<ol style="list-style-type: none"> 1. Từ trang chủ người dùng bấm vào tab “Đơn hàng” 2. Hệ thống hiển thị danh sách đơn hàng chưa xử lý 3. Bấm vào đơn hàng muốn hủy 4. Hệ thống hiển thị chi tiết đơn hàng 5. Bấm vào “Hủy đơn” 6. Hệ thống hiện popup nêu lý do hủy 7. Chọn lý do hủy 8. Bấm vào “Xác nhận” 9. Hệ thống thông báo hủy thành công
Exceptions	8.a Nếu người dùng chưa muốn hủy thì bấm vào nút trở về

- Hoàn thành đơn hàng: Chủ cửa hàng hoặc nhân viên có thể hoàn thành các đơn hàng chưa xử lý

Use-case name	Hoàn thành đơn hàng
Use-case ID	15
Actor	Chủ cửa hàng hoặc nhân viên
Description	Chủ cửa hàng hoặc nhân viên có thể hoàn thành các đơn hàng chưa xử lý
Pre-conditions	Đã đăng nhập
Post-conditions	Hoàn thành đơn hàng thành công
Normal flow	<ol style="list-style-type: none"> 1. Từ trang chủ người dùng bấm vào tab “Đơn hàng” 2. Hệ thống hiển thị danh sách đơn hàng chưa xử lý 3. Bấm vào đơn hàng muốn hoàn thành 4. Bấm vào nút “Hoàn thành” 5. Hệ thống hiện popup bạn có chắc chắn hoàn thành không 6. Bấm vào “Xác nhận” 7. Hệ thống thông báo đã hoàn thành đơn hàng
Exceptions	6.a Nếu người dùng chưa muốn hoàn thành thì bấm vào nút trở về

- Xem các món ăn của cửa hàng mình: Chủ cửa hàng hoặc nhân viên có thể xem món ăn của cửa hàng mình

Use-case name	Xem các món ăn của cửa hàng mình
Use-case ID	16
Actor	Chủ cửa hàng hoặc nhân viên
Description	Chủ cửa hàng hoặc nhân viên có thể xem món ăn của cửa hàng mình
Pre-conditions	Đã đăng nhập
Post-conditions	Xem các món ăn của cửa hàng mình

Normal flow	<ol style="list-style-type: none">1. Từ trang chủ người dùng bấm vào tab “Món ăn”2. Hệ thống hiển thị các món ăn của cửa hàng
Alternative flows	<ol style="list-style-type: none">2.a Có thể chuyển trang để xem món ăn2.b Có thể lọc món ăn theo tên, ngày thêm, trạng thái

- Thêm món ăn cho cửa hàng: Chủ cửa hàng hoặc nhân viên có thể thêm các món ăn cho cửa hàng mình

Use-case name	Thêm món ăn cho cửa hàng
Use-case ID	17
Actor	Chủ cửa hàng hoặc nhân viên
Description	Chủ cửa hàng hoặc nhân viên có thể thêm các món ăn cho cửa hàng mình
Pre-conditions	Đã đăng nhập
Post-conditions	Thêm món ăn thành công

Normal flow	<ol style="list-style-type: none"> 1. Từ trang chủ người dùng bấm vào tab “Món ăn” 2. Hệ thống hiển thị các món ăn của cửa hàng 3. Bấm vào nút “Thêm món ăn” 4. Hệ thống chuyển sang trang thêm món ăn 5. Thêm hình ảnh món 6. Thêm tên món 7. Thêm giá gốc món ăn 8. Thêm giá món ăn sau khuyến mãi 9. Thêm số lượng cần bán 10. Chọn trạng thái (Đang bán, Chưa bán) 11. Thêm thời gian bán 12. Thêm mô tả món ăn 13. Bấm vào nút “Thêm sản phẩm” 14. Hệ thống thông báo thêm món ăn thành công
Alternative flows	5.a Có thể thêm nhiều ảnh và xóa ảnh
Exceptions	<p>13.a Nếu không muốn thêm món bấm nút “Hủy”</p> <p>13.b Nếu có trường nào trống thì thông báo và nhập lại trường đó</p> <p>13.c Nếu giá sau khuyến mãi lớn hơn giá gốc thì thông báo và nhập lại giá khuyến mãi</p>

- Sửa món ăn của cửa hàng: Chủ cửa hàng hoặc nhân viên có thể sửa món ăn của cửa hàng mình

Use-case name	Sửa món ăn của cửa hàng
Use-case ID	18
Actor	Chủ cửa hàng hoặc nhân viên
Description	Chủ cửa hàng hoặc nhân viên có thể sửa món ăn của cửa hàng mình
Pre-conditions	Đã đăng nhập
Post-conditions	Sửa món ăn thành công
Normal flow	<ol style="list-style-type: none">1. Từ trang chủ người dùng bấm vào tab “Món ăn”2. Hệ thống hiển thị các món ăn của cửa hàng3. Bấm vào món ăn muốn sửa4. Hệ thống chuyển sang trang sửa món ăn5. Chỉnh sửa thông tin muốn sửa6. Bấm vào nút “Cập nhật”7. Hệ thống thông báo cập nhật thành công
Exceptions	<p>6.a Nếu không cập nhật món bấm nút “Hủy”</p> <p>6.b Nếu có trường nào trống thì thông báo và nhập lại trường đó</p> <p>6.c Nếu giá sau khuyến mãi lớn hơn giá gốc thì thông báo và nhập lại giá khuyến mãi</p>

- Xóa món ăn của cửa hàng: Chủ cửa hàng hoặc nhân viên có thể xóa các món ăn của cửa hàng

Use-case name	Xóa món ăn của cửa hàng
Use-case ID	19
Actor	Chủ cửa hàng hoặc nhân viên
Description	Chủ cửa hàng hoặc nhân viên có thể xóa các món ăn của cửa hàng
Pre-conditions	Đã đăng nhập
Post-conditions	Xóa món ăn thành công

Normal flow	<ol style="list-style-type: none"> 1. Từ trang chủ người dùng bấm vào tab “Món ăn” 2. Hệ thống hiển thị các món ăn của cửa hàng 3. Bấm vào món ăn muốn xóa 4. Hệ thống chuyển sang trang sửa món ăn 5. Bấm vào nút “Xóa” 6. Hệ thống hiển thị popup bạn có chắc muốn xóa món ăn không 7. Bấm vào nút “Xác nhận” 8. Hệ thống thông báo xóa thành công
Alternative flows	7.a Nếu không xóa món ăn bấm nút “Hủy” và quay lại bước 5
Exceptions	5.a Nếu không muốn xóa món ăn thì bấm nút trở về

- Xem thông tin cửa hàng: Chủ cửa hàng hoặc nhân viên có thể xem thông tin cửa hàng

Use-case name	Xem thông tin cửa hàng
Use-case ID	20
Actor	Chủ cửa hàng hoặc nhân viên
Description	Chủ cửa hàng hoặc nhân viên có thể xem thông tin cửa hàng
Pre-conditions	Đã đăng nhập
Post-conditions	Xem thông tin cửa hàng
Normal flow	<ol style="list-style-type: none"> 1. Từ trang chủ người dùng bấm vào tab “Cá nhân” 2. Bấm vào “Thông tin” 3. Hệ thống trả về trang thông tin cá nhân

- Sửa thông tin cửa hàng: Chủ cửa hàng hoặc nhân viên có thể sửa thông tin cửa hàng

Use-case name	Sửa thông tin cửa hàng
Use-case ID	21
Actor	Chủ cửa hàng hoặc nhân viên
Description	Chủ cửa hàng hoặc nhân viên có thể sửa thông tin cửa hàng
Pre-conditions	Đã đăng nhập
Post-conditions	Sửa thông tin cửa hàng thành công
Normal flow	<ol style="list-style-type: none"> 1. Từ trang chủ người dùng bấm vào tab “Cá nhân” 2. Bấm vào “Chỉnh sửa tài khoản” 3. Hệ thống trả về trang chỉnh sửa thông tin cá nhân 4. Sửa các trường muốn sửa 5. Bấm “Cập nhật” 6. Hệ thống thông báo cập nhật thành công
Exceptions	<p>5.a Nếu không muốn sửa bấm vào nút “Trở về”</p> <p>5.b Nếu trường nào trống thì thông báo và nhập lại trường đó</p>

- Cập nhật mật khẩu: Chủ cửa hàng hoặc nhân viên có thể cập nhật mật khẩu của cửa hàng

Use-case name	Cập nhật mật khẩu
Use-case ID	22
Actor	Chủ cửa hàng hoặc nhân viên
Description	Chủ cửa hàng hoặc nhân viên có thể cập nhật mật khẩu của cửa hàng
Pre-conditions	Đã đăng nhập
Post-conditions	Cập nhật mật khẩu thành công

Normal flow	<ol style="list-style-type: none">1. Từ trang chủ người dùng bấm vào tab “Cá nhân”2. Bấm vào “Chỉnh sửa tài khoản”3. Hệ thống trả về trang thông tin cá nhân4. Bấm vào “Cập nhật mật khẩu”5. Hệ thống trả về trang sửa mật khẩu6. Nhập mật khẩu cũ7. Nhập mật khẩu mới8. Nhập lại mật khẩu mới9. Bấm “Cập nhật”10. Hệ thống thông báo cập nhật thành công
Exceptions	<p>9.a Nếu không muốn cập nhật bấm vào nút “Hủy”</p> <p>9.b Nếu trường nào trống thì thông báo và nhập lại trường đó</p> <p>9.c Nếu mật khẩu cũ sai thì thông báo và nhập lại mật khẩu cũ</p> <p>9.d Nếu nhập lại mật khẩu mới không trùng thì thông báo và nhập lại</p> <p>9.e Nếu mật khẩu mới giống mật khẩu cũ thì thông báo và nhập lại</p>

- Đăng xuất cho chủ cửa hàng

Use-case name	Đăng xuất cho chủ cửa hàng
Use-case ID	23
Actor	Chủ cửa hàng hoặc nhân viên
Description	Đăng xuất khỏi tài khoản
Pre-conditions	Đã đăng nhập
Post-conditions	Đăng xuất khỏi tài khoản thành công
Normal flow	<ol style="list-style-type: none">1. Từ trang chủ người dùng bấm vào tab “Cá nhân”2. Bấm “Đăng xuất”3. Hệ thống trả về trang đăng nhập

- Xem lại các đơn hàng của cửa hàng: Chủ cửa hàng hoặc nhân viên có thể xem lại các đơn hàng của cửa hàng

Use-case name	Xem lại các đơn hàng của cửa hàng
Use-case ID	24
Actor	Chủ cửa hàng hoặc nhân viên
Description	Chủ cửa hàng hoặc nhân viên có thể xem lại các đơn hàng của cửa hàng
Pre-conditions	Đã đăng nhập
Post-conditions	Chủ cửa hàng hoặc nhân viên xem lại các đơn hàng của cửa hàng
Normal flow	<ol style="list-style-type: none"> 1. Từ trang chủ người dùng bấm vào tab “Trang chủ” 2. Bấm vào “Lịch sử đơn hàng” 3. Hệ thống trả về danh sách các đơn hàng đã xử lý của cửa hàng 4. Bấm vào đơn hàng mình muốn xem 5. Hệ thống trả về trang chi tiết đơn hàng

- Xem thống kê doanh thu của cửa hàng: Chủ cửa hàng hoặc nhân viên xem thống kê doanh thu của cửa hàng

Use-case name	Xem thống kê doanh thu của cửa hàng
Use-case ID	25
Actor	Chủ cửa hàng hoặc nhân viên
Description	Chủ cửa hàng hoặc nhân viên xem thống kê doanh thu của cửa hàng
Pre-conditions	Đã đăng nhập
Post-conditions	Chủ cửa hàng hoặc nhân viên xem lại các đơn hàng của cửa hàng

Normal flow	<ol style="list-style-type: none"> 1. Từ trang chủ người dùng bấm vào tab “Trang chủ” 2. Bấm vào “Báo cáo doanh thu” 3. Hệ thống trả về trang doanh thu của cửa hàng 4. Bấm vào tháng người dùng muốn xem 5. Hệ thống trả về doanh thu của cửa hàng theo tháng đó
--------------------	--

- **Bật/tắt cửa hàng:** Chủ cửa hàng hoặc nhân viên có thể Bật/tắt tình trạng hoạt động của cửa hàng

Use-case name	Bật/tắt cửa hàng
Use-case ID	26
Actor	Chủ cửa hàng hoặc nhân viên
Description	Chủ cửa hàng hoặc nhân viên có thể Bật/tắt cửa hàng
Pre-conditions	Đã đăng nhập
Post-conditions	Chủ cửa hàng hoặc nhân viên Bật/tắt cửa hàng thành công
Normal flow	<ol style="list-style-type: none"> 1. Từ trang chủ người dùng bấm vào nút “Bật/tắt” 2. Hệ thống thông báo Bật/tắt thành công

- **Đăng nhập cho quản trị viên:** Quản trị viên có thể đăng nhập và sử dụng các tính năng trong web

Use-case name	Đăng nhập
Use-case ID	27
Actor	Quản trị viên
Description	Quản trị viên có thể đăng nhập và sử dụng các tính năng trong web
Pre-conditions	Có tài khoản quản trị
Post-conditions	Đăng nhập thành công

Normal flow	<ol style="list-style-type: none">1. Người dùng vào trang đăng nhập2. Người dùng nhập Tên đăng nhập3. Người dùng nhập mật khẩu4. Người dùng bấm vào nút “Đăng nhập”5. Hệ thống chuyển đến trang chủ
Exceptions	<p>5.a Nếu “Tên đăng nhập” hoặc “Mật khẩu” sai thì thông báo cho người dùng và quay lại bước 2</p> <p>5.b Nếu người dùng chưa nhập “Tên đăng nhập” hoặc “Mật khẩu” thì thông báo cho người dùng và quay lại bước 2</p>

- Phê duyệt tài khoản các cửa hàng: Quản trị viên có thể phê duyệt tài khoản cửa hàng

Use-case name	Phê duyệt tài khoản các cửa hàng
Use-case ID	28
Actor	Quản trị viên
Description	Quản trị viên có thể phê duyệt tài khoản cửa hàng
Pre-conditions	Đã đăng nhập
Post-conditions	Quản trị viên phê duyệt tài khoản cửa hàng thành công
Normal flow	<ol style="list-style-type: none">1. Từ trang chủ người dùng bấm vào tab “Phê duyệt tài khoản”2. Hệ thống trả về trang phê duyệt3. Bấm vào nút “Đồng ý” cho cửa hàng đã phê duyệt4. Hệ thống hiển thị popup bạn có muốn phê duyệt không5. Người dùng bấm vào nút “Đồng ý”6. Hệ thống thông báo phê duyệt thành công và gửi mail cho cửa hàng thông báo tài khoản của bạn đã được phê duyệt

Exceptions	5.a Nếu người dùng không muốn cho cửa hàng đó tạo tài khoản thì bấm nút “Hủy”
-------------------	--

- Xem báo cáo doanh thu của ứng dụng: Quản trị viên có thể phê duyệt tài khoản cửa hàng

Use-case name	Xem báo cáo doanh thu của ứng dụng
Use-case ID	29
Actor	Quản trị viên
Description	Quản trị viên có thể phê duyệt tài khoản cửa hàng
Pre-conditions	Đã đăng nhập
Post-conditions	Quản trị viên phê duyệt tài khoản cửa hàng thành công
Normal flow	<ol style="list-style-type: none"> 1. Từ trang chủ người dùng bấm vào tab “Doanh thu” 2. Hệ thống trả về trang báo cáo 3. Người dùng xem báo cáo doanh thu

- Xem các cửa hàng đã đăng ký: Quản trị viên có thể xem các cửa hàng đã đăng ký

Use-case name	Xem các cửa hàng đã đăng ký
Use-case ID	30
Actor	Quản trị viên
Description	Quản trị viên có thể xem các cửa hàng đã đăng ký
Pre-conditions	Đã đăng nhập
Post-conditions	Quản trị viên xem các cửa hàng đã đăng ký
Normal flow	<ol style="list-style-type: none"> 1. Từ trang chủ người dùng bấm vào tab “Cửa hàng” 2. Hệ thống trả về trang cửa hàng 3. Người dùng xem các cửa hàng đã đăng ký

- **Bật/tắt cửa hàng đã đăng ký:** Quản trị viên có thể Bật/tắt trạng thái của cửa hàng đã đăng ký

Use-case name	Bật/tắt cửa hàng đã đăng ký
Use-case ID	31
Actor	Quản trị viên
Description	Quản trị viên có thể Bật/tắt cửa hàng đã đăng ký
Pre-conditions	Đã đăng nhập
Post-conditions	Quản trị viên Bật/tắt cửa hàng đã đăng ký
Normal flow	<ol style="list-style-type: none"> 1. Từ trang chủ người dùng bấm vào tab “Cửa hàng” 2. Hệ thống trả về trang cửa hàng 3. Người dùng bấm vào nút “Bật/tắt” cửa hàng đã đăng ký 4. Hệ thống thông báo đã Bật/tắt thành công

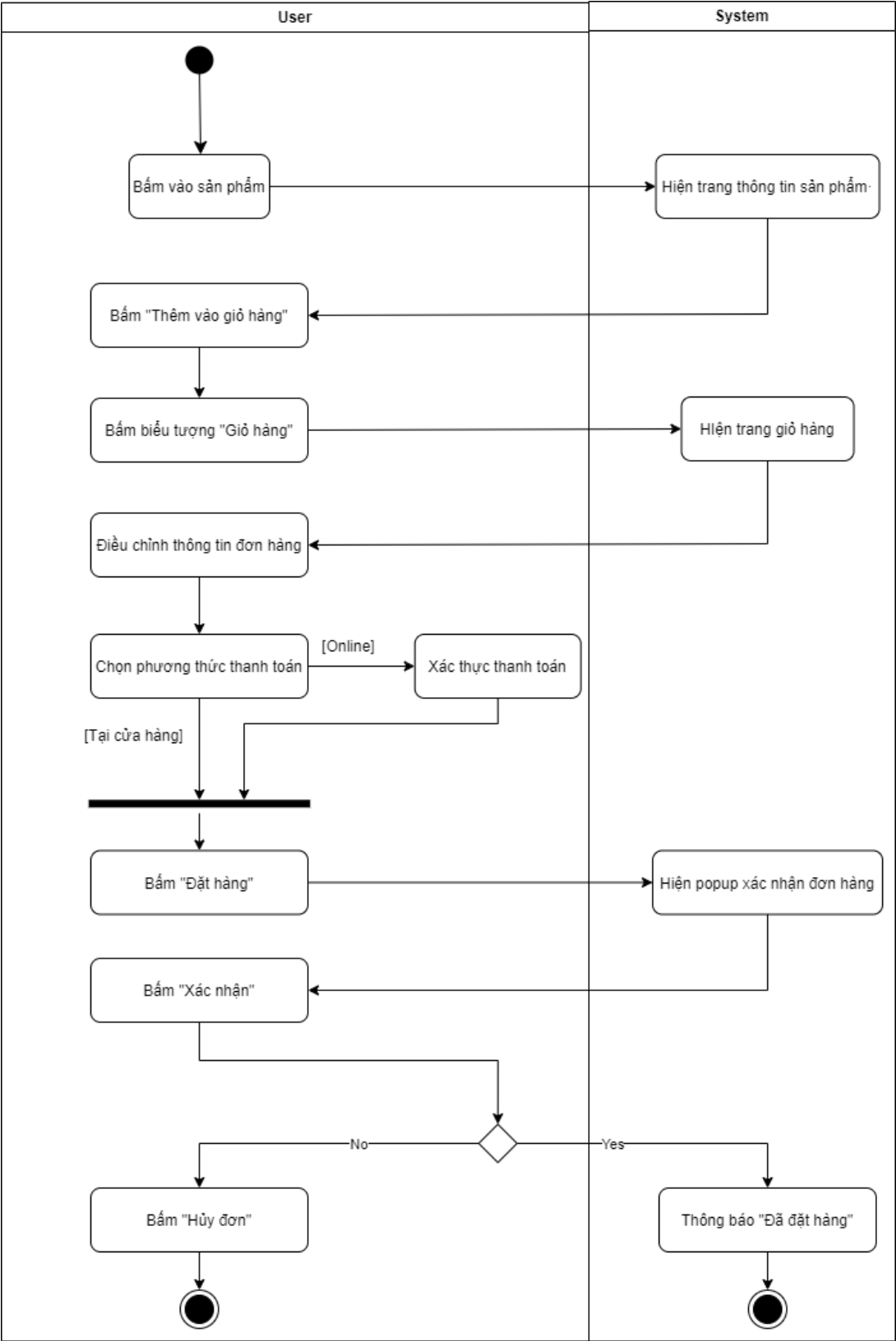
- **Cấm/mở cửa hàng đã đăng ký:** Quản trị viên có thể Cấm/mở cửa hàng đã đăng ký

Use-case name	Cấm/mở cửa hàng đã đăng ký
Use-case ID	32
Actor	Quản trị viên
Description	Quản trị viên có thể Cấm/mở cửa hàng đã đăng ký
Pre-conditions	Đã đăng nhập
Post-conditions	Quản trị viên Cấm/mở cửa hàng đã đăng ký

Normal flow	<ol style="list-style-type: none">1. Từ trang chủ người dùng bấm vào tab “Cửa hàng”2. Hệ thống trả về trang cửa hàng3. Người dùng bấm vào nút “Cấm/mở” của hàng đã đăng ký4. Hệ thống hiển thị popup bạn có muốn Cấm/mở cửa hàng không5. Người dùng bấm vào nút “Xác nhận”6. Hệ thống thông báo đã Cấm/mở thành công và gửi mail cho cửa hàng
Exceptions	<p>5.a Nếu người dùng không muốn Cấm/mở cửa hàng thì bấm vào nút “Hủy”</p>

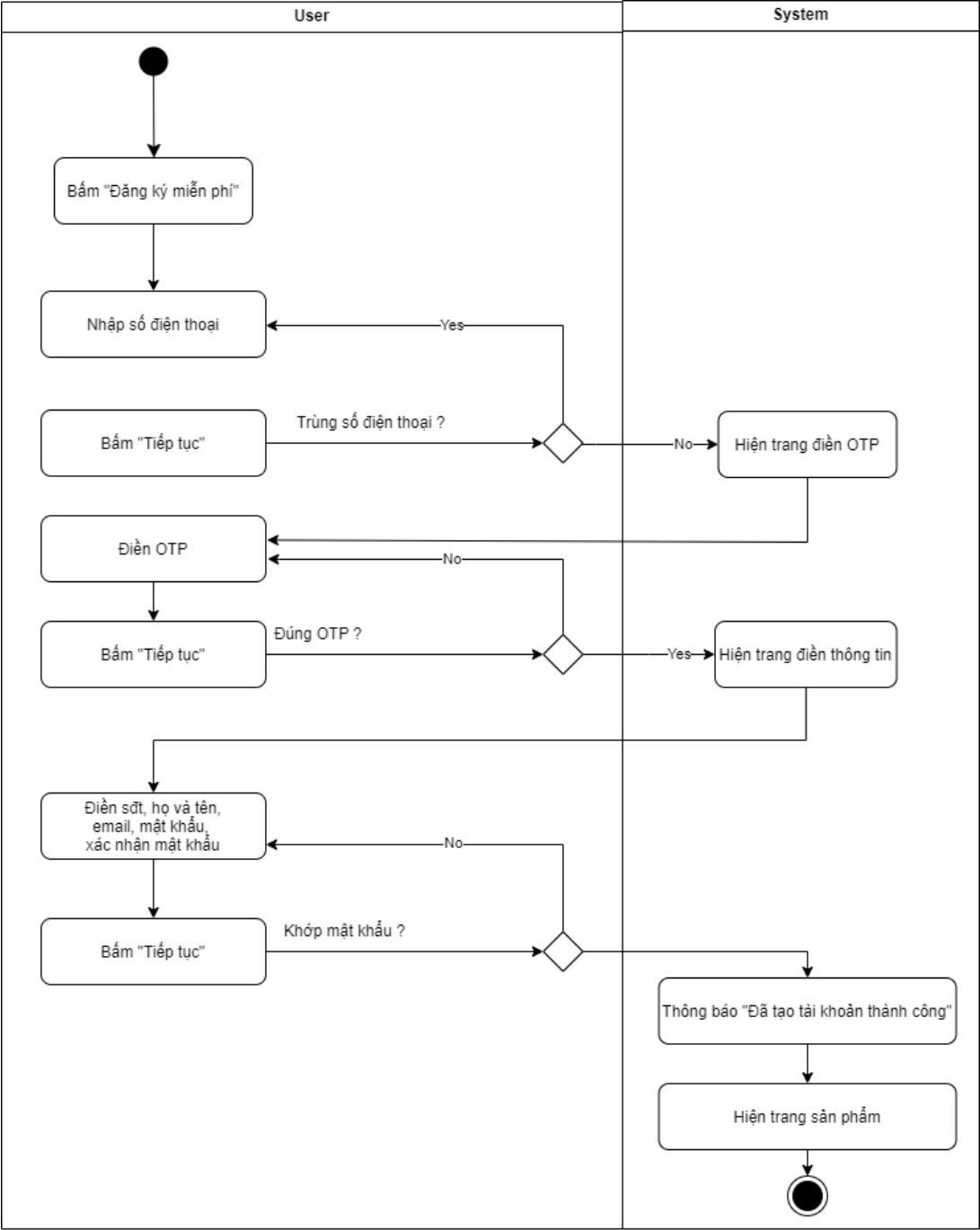
3.4 Activity diagram

Activity diagram cho tính năng Đặt hàng:



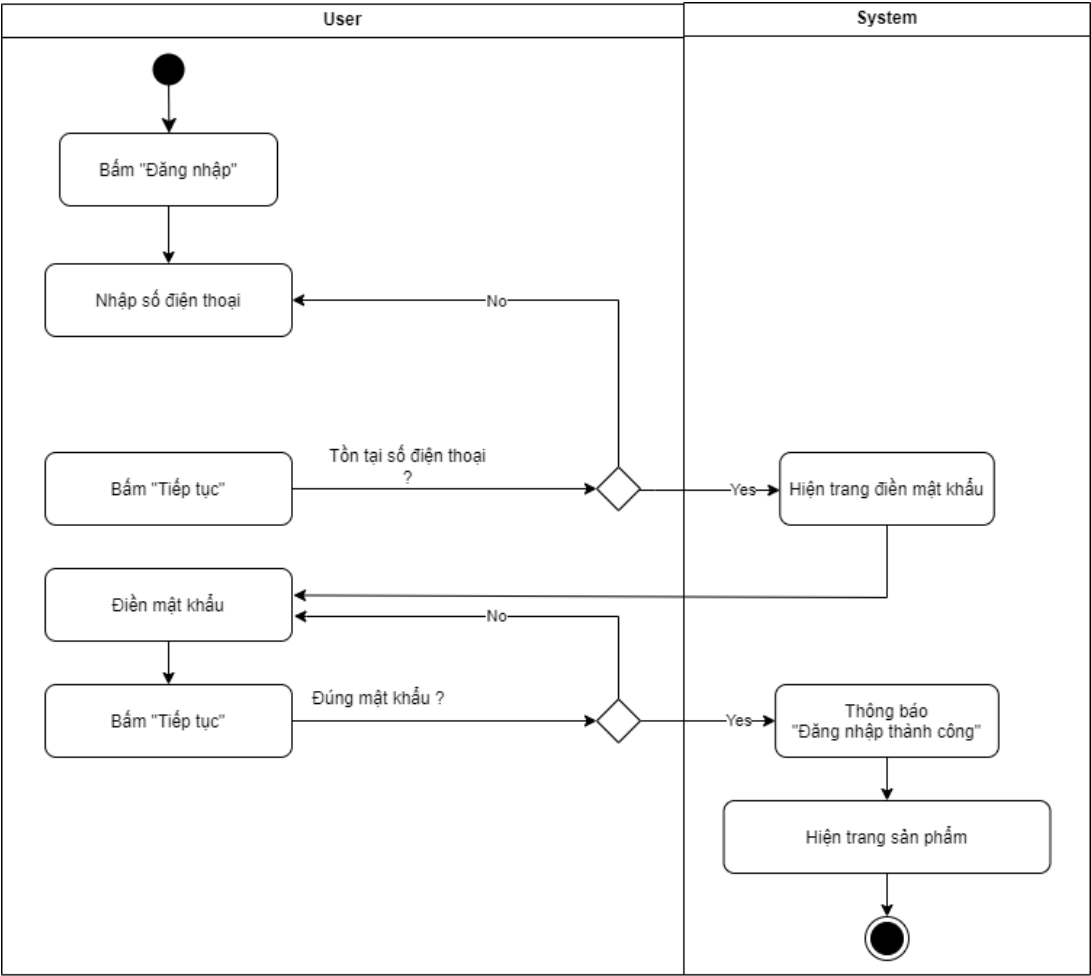
Hình 3.2: Activity cho tính năng Đặt hàng

Activity diagram cho tính năng Đăng ký người dùng:



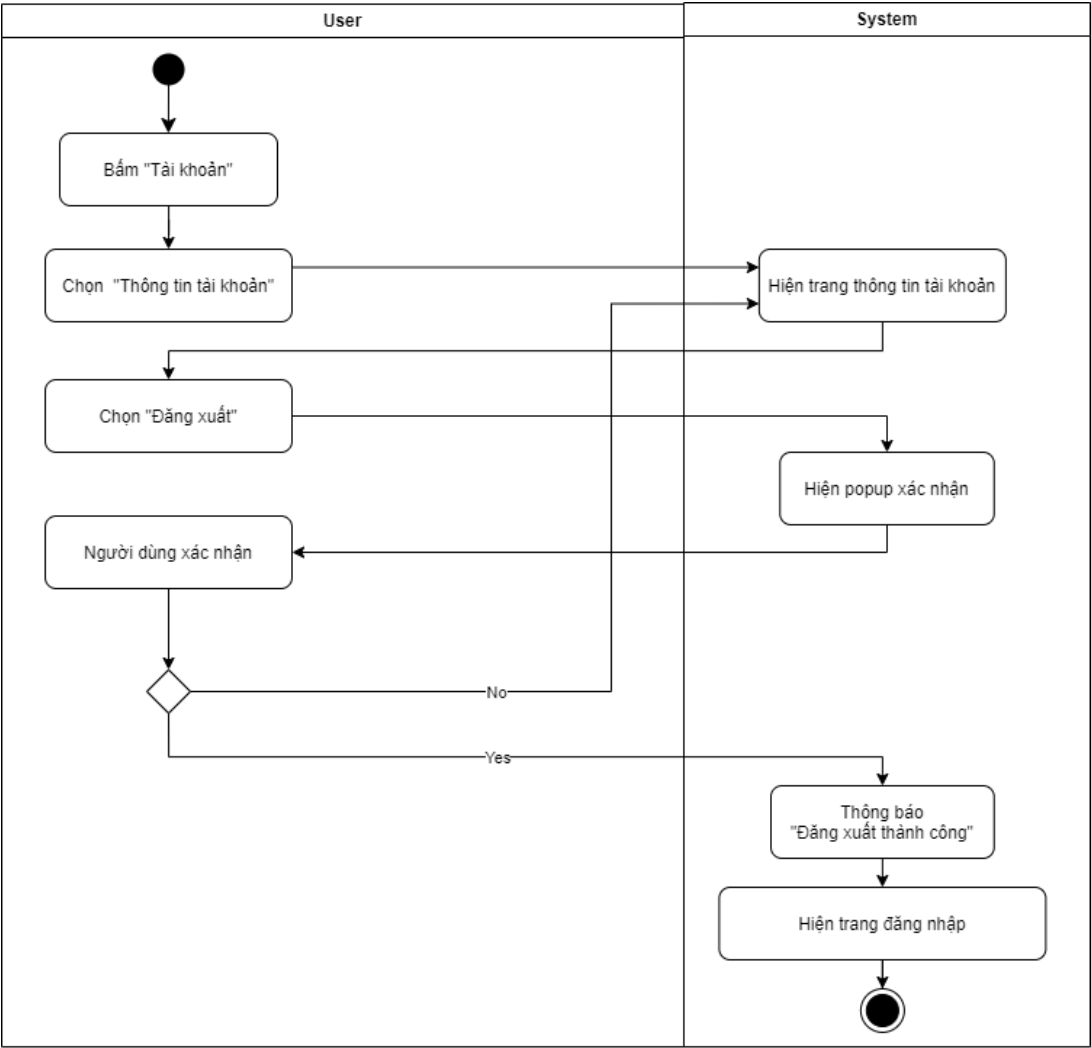
Hình 3.3: Activity cho tính năng Đăng ký người dùng

Activity diagram cho tính năng Đăng nhập người dùng:



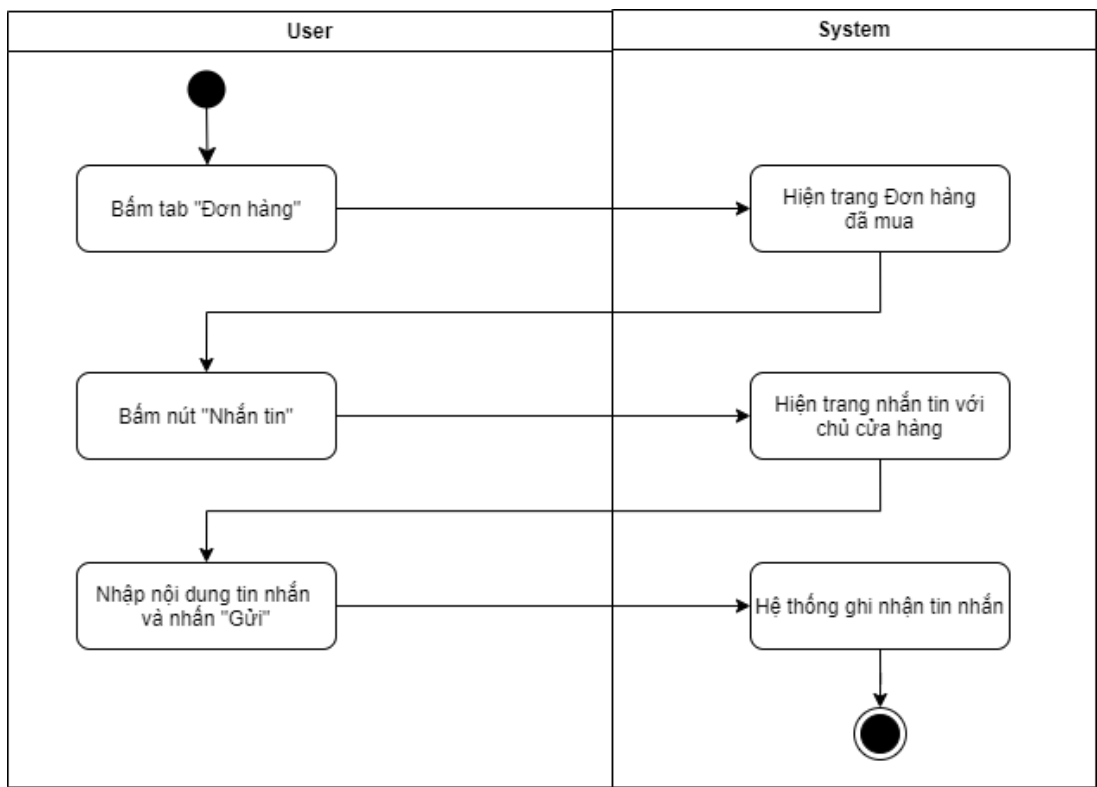
Hình 3.4: Activity cho tính năng Đăng nhập người dùng

Activity diagram cho tính năng Đăng xuất:



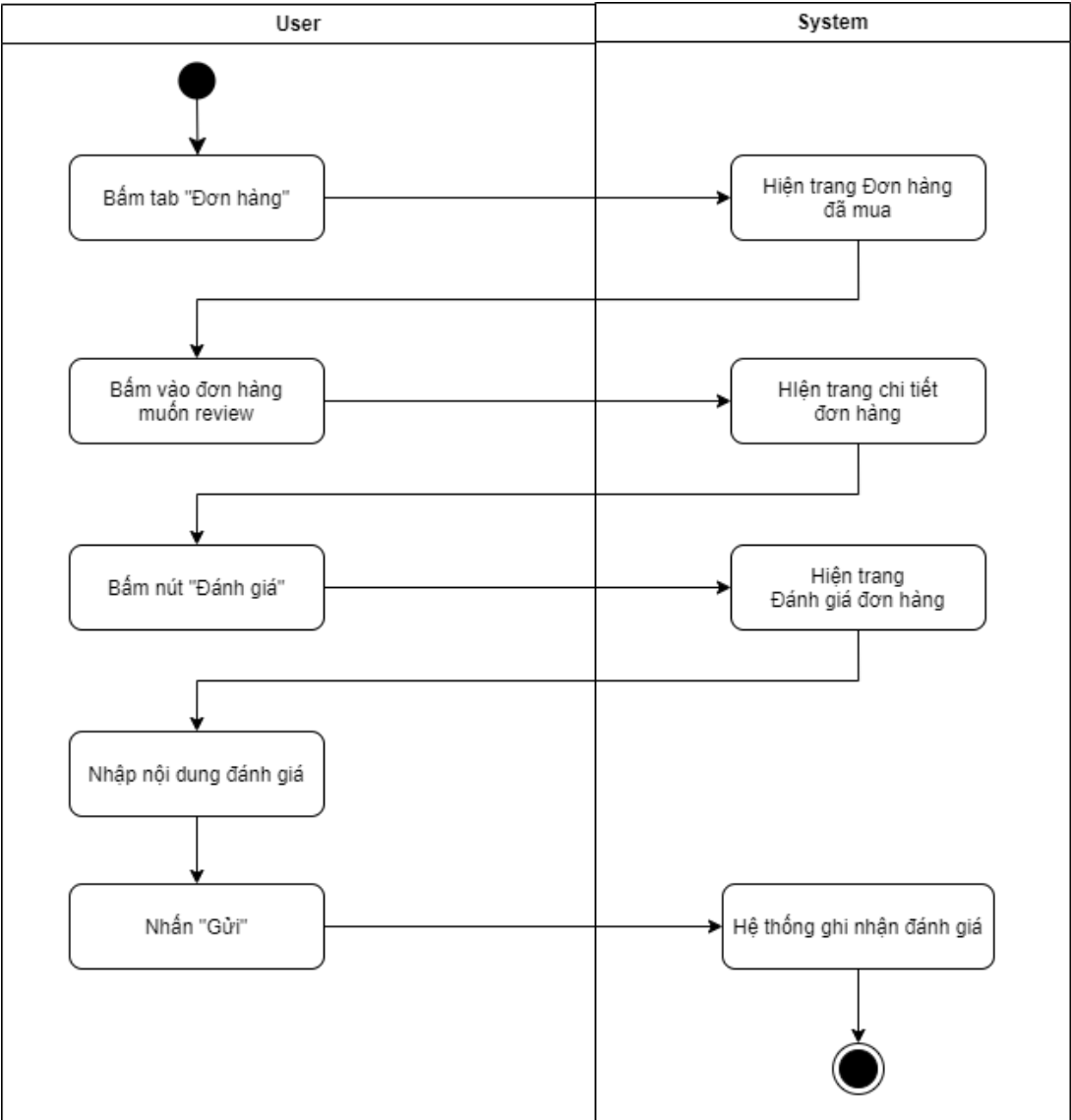
Hình 3.5: Activity cho tính năng Đăng xuất

Activity diagram cho tính năng Nhắn tin:



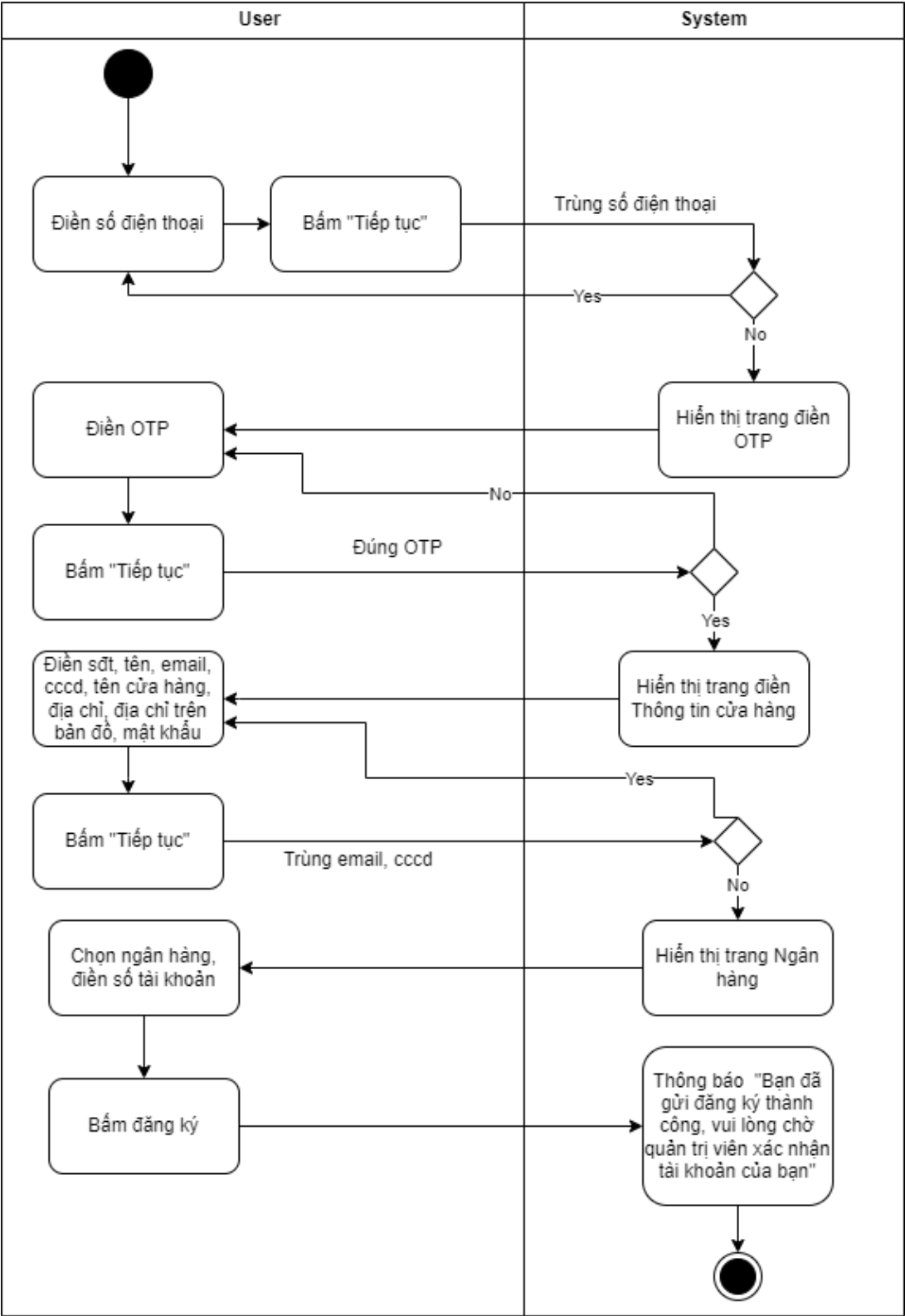
Hình 3.6: Activity cho tính năng Nhắn tin

Activity diagram cho tính năng Đánh giá đơn hàng:



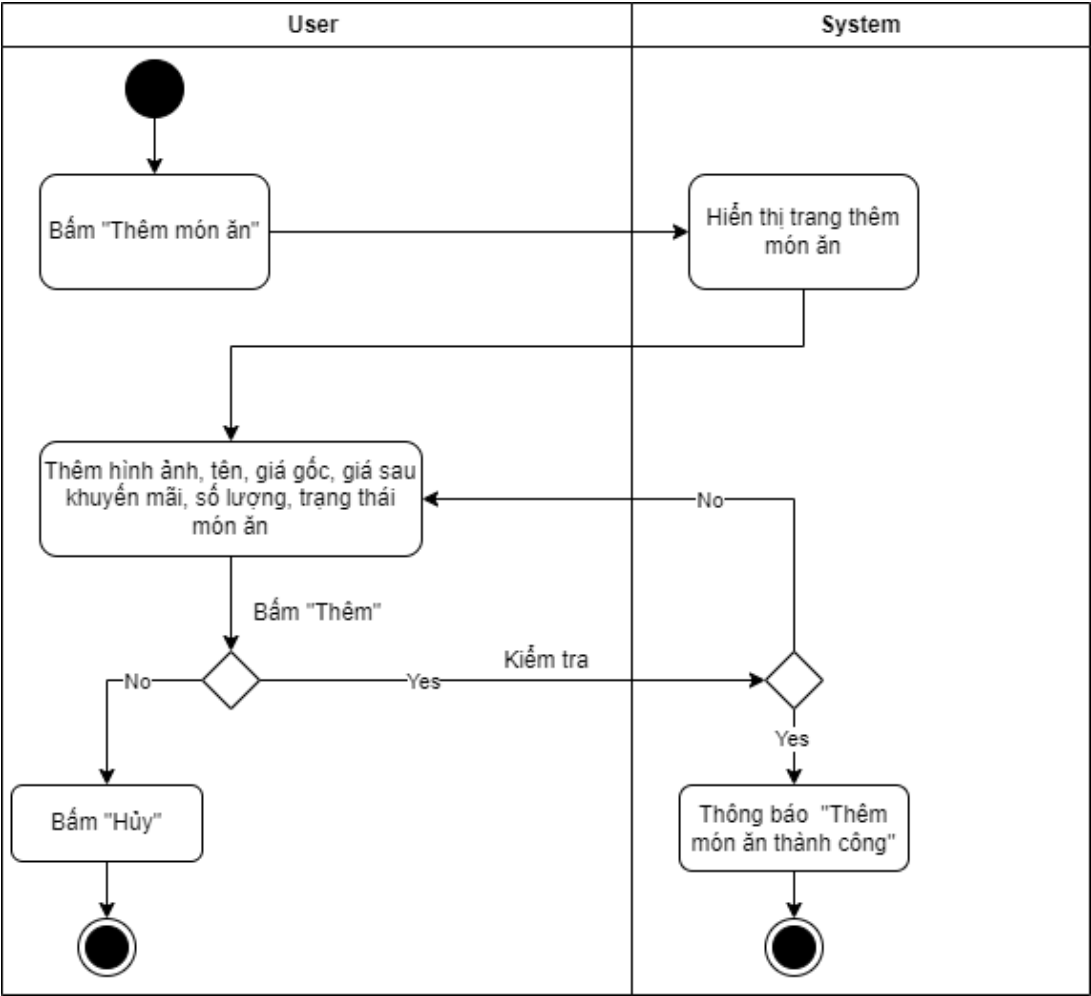
Hình 3.7: Activity cho tính năng Đánh giá đơn hàng

Activity cho tính năng Đăng ký chủ cửa hàng:



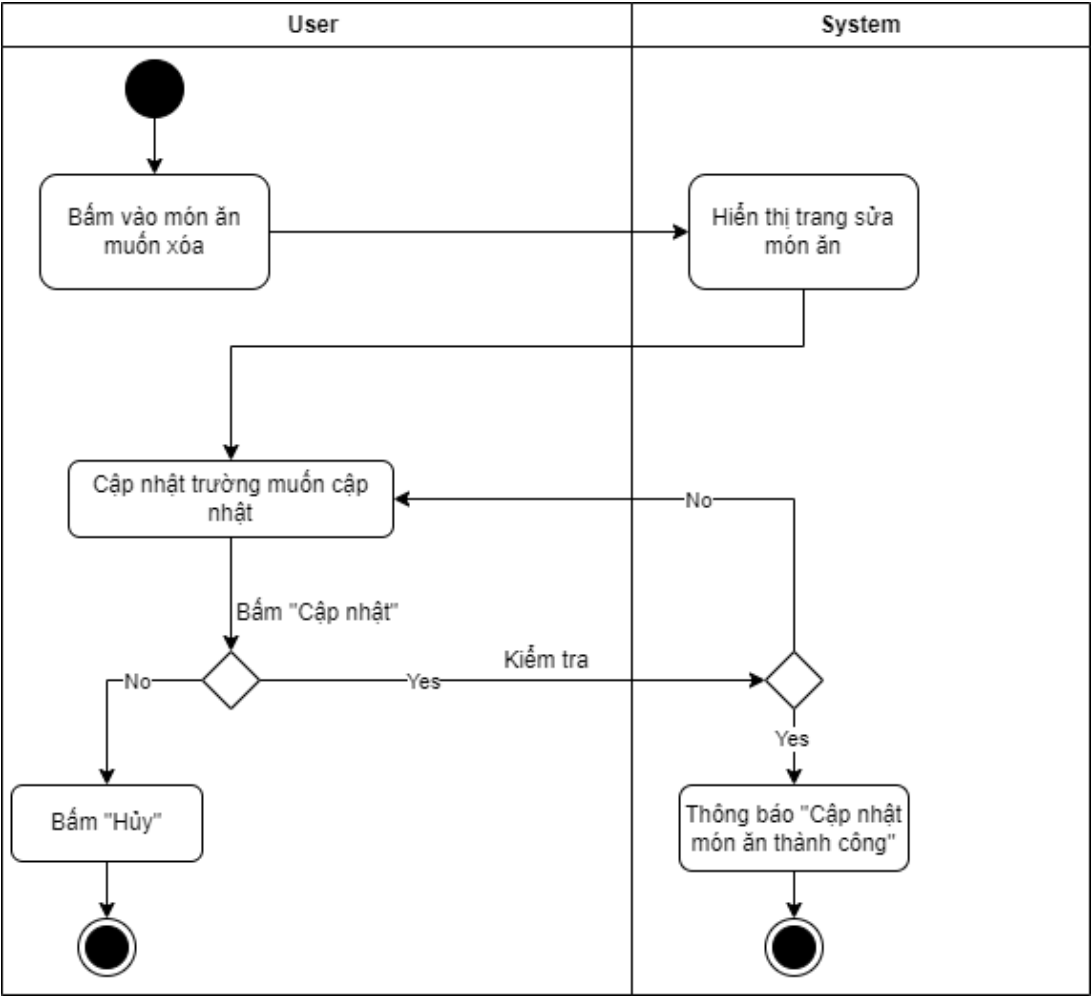
Hình 3.8: Activity cho tính năng Đăng ký chủ cửa hàng

Activity diagram cho tính năng Thêm sản phẩm:



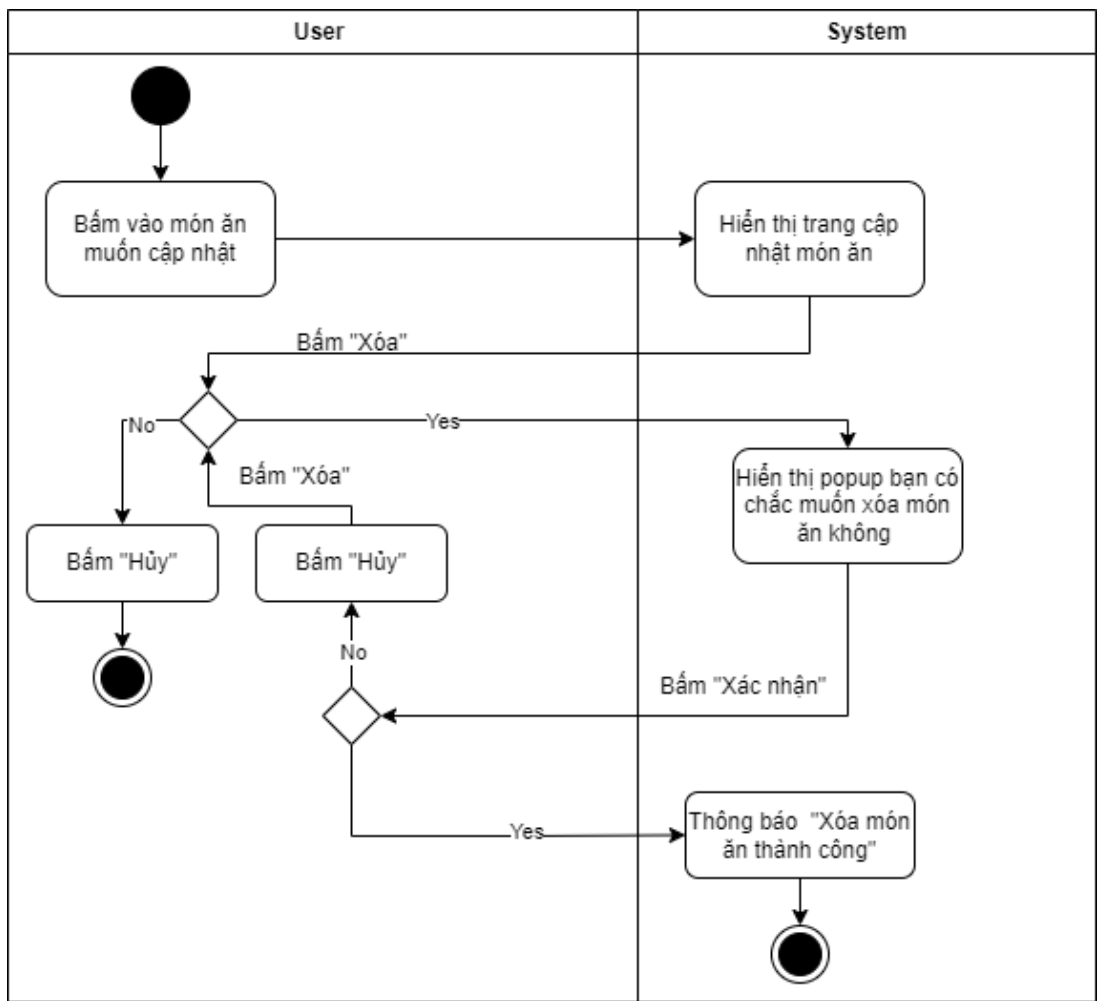
Hình 3.9: Activity cho tính năng Thêm sản phẩm

Activity diagram cho tính năng Cập nhật sản phẩm:



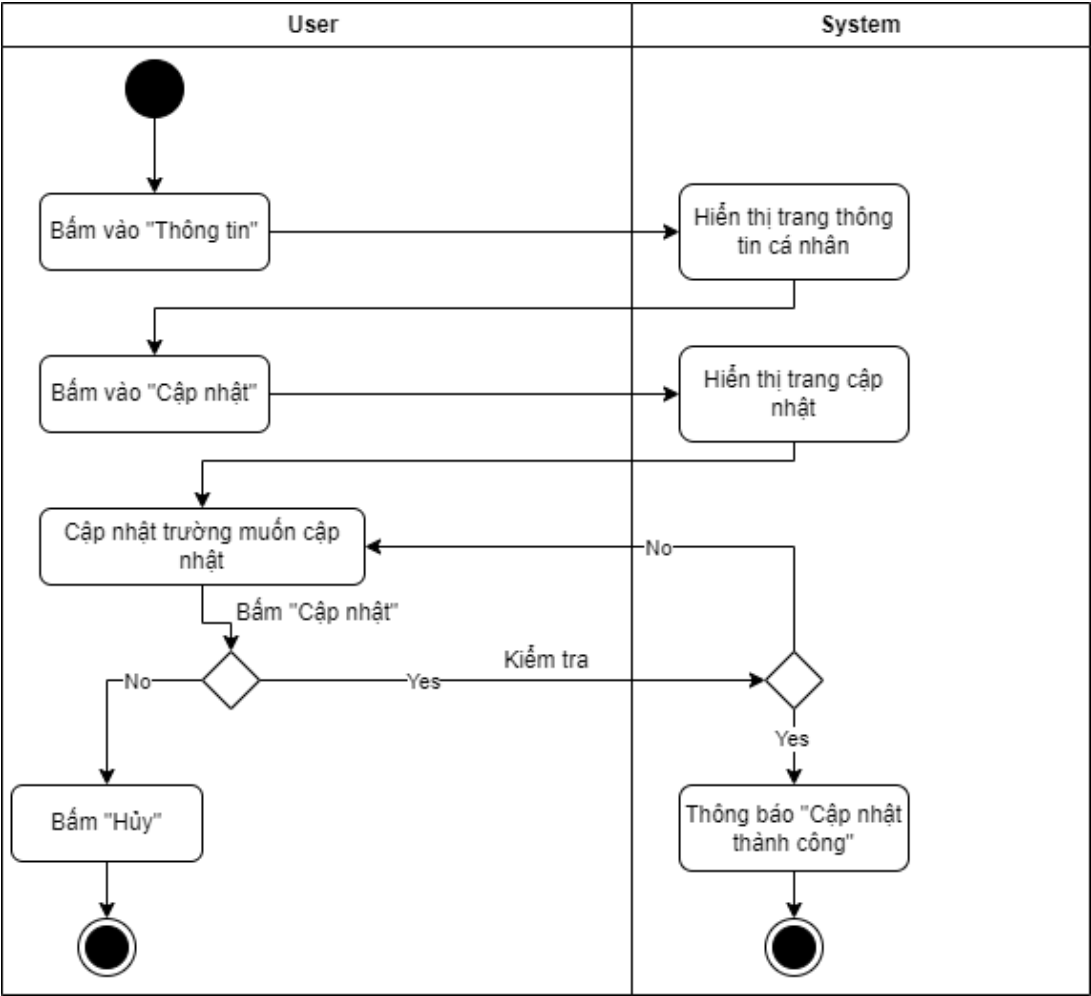
Hình 3.10: Activity cho tính năng Cập nhật sản phẩm

Activity diagram cho tính năng Xóa sản phẩm:



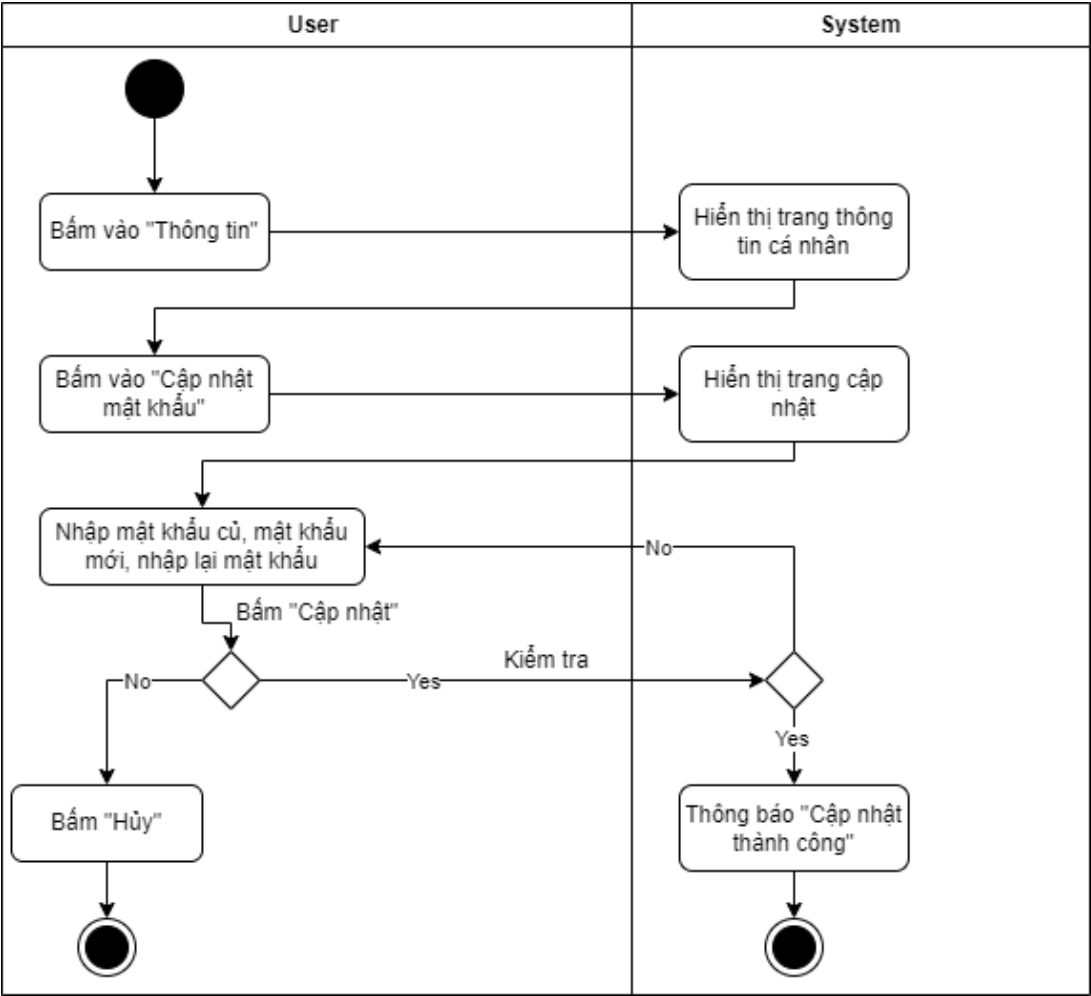
Hình 3.11: Activity cho tính năng Xóa sản phẩm

Activity diagram cho tính năng Cập nhật thông tin:



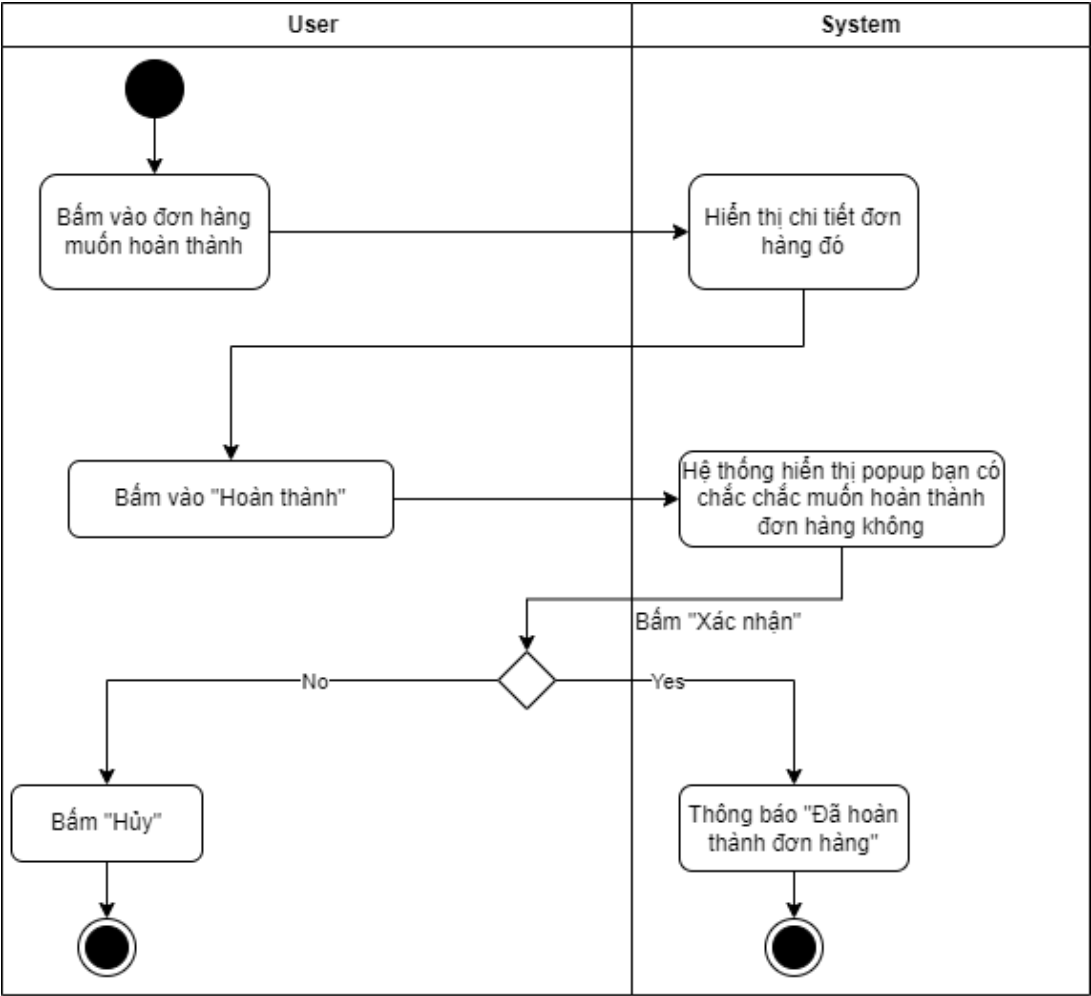
Hình 3.12: Activity cho tính năng Cập nhật thông tin

Activity diagram cho tính năng Thay đổi mật khẩu:



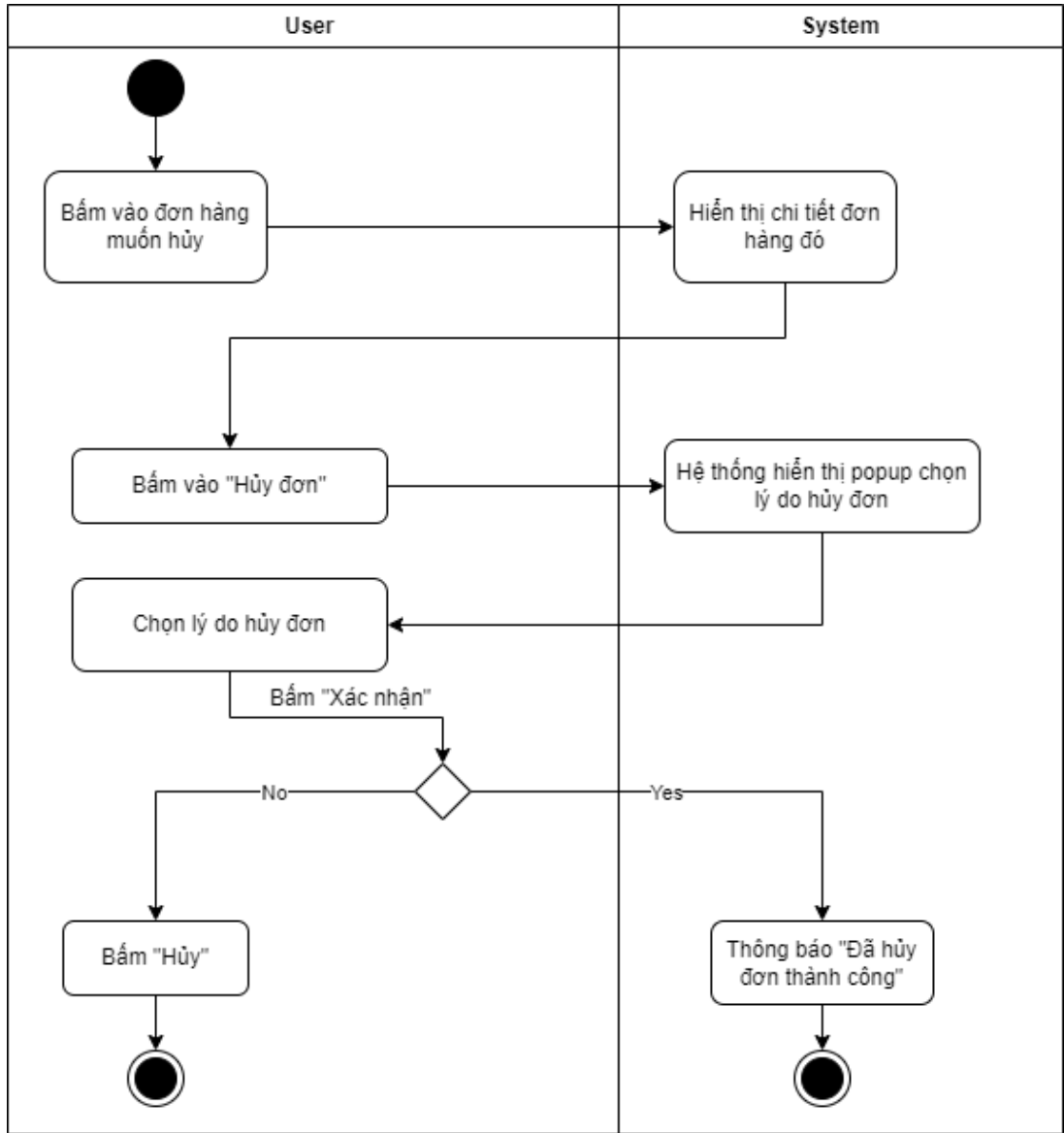
Hình 3.13: Activity cho tính năng Thay đổi mật khẩu

Activity diagram cho tính năng Hoàn thành đơn hàng:



Hình 3.14: Activity cho tính năng Hoàn thành đơn hàng

Activity diagram cho tính năng Hủy đơn hàng:



Hình 3.15: Activity cho tính năng Hủy đơn hàng

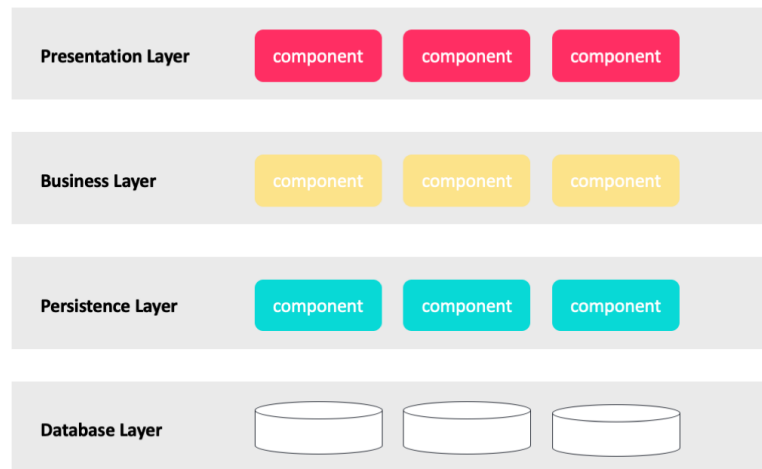
3.5 Kiến trúc phần mềm

Sau khi phân tích nghiệp vụ của hệ thống, chúng tôi quyết định chọn **Kiến trúc n-tier** để triển khai hệ thống của chúng tôi. Đây là một trong những kiến trúc phổ biến nhất nhờ chi phí thấp và sự đơn giản của nó.

Kiến trúc n-tier hay còn được gọi là kiến trúc đa lớp bởi vì nó nhóm các chức năng liên quan trong hệ thống thành các lớp logic riêng biệt theo chiều ngang và các lớp này xếp chồng lên nhau theo chiều dọc. Việc giao tiếp giữa các lớp cũng phải tuân theo trật tự, nghĩa là một request phải đi từ trên xuống, tuần tự qua các lớp và không bỏ qua lớp nào, đây còn gọi là layers of isolation. Và giao tiếp giữa các lớp này thường không đồng bộ để tăng khả năng mở rộng cho hệ thống.

Mô hình mà kiến trúc đa tầng cung cấp cũng giúp tăng tính linh hoạt cho hệ thống do nó tách biệt hệ thống thành nhiều lớp, từ đó tùy chỉnh thay đổi hoặc thêm lớp mới thay vì phải viết lại toàn bộ hệ thống.

Không có quy định cụ thể về số lớp của kiến trúc n-tier, **n** có thể thay đổi tùy vào quy mô của hệ thống. Tuy vậy, đa số các kiến trúc đa lớp thường bao gồm bốn lớp tiêu chuẩn: presentation, business, persistence và database.



Hình 3.16: Các lớp tiêu chuẩn trong kiến trúc n-tier

Như đã đề cập ở trên, kiến trúc n-tier là một kiến trúc khá linh hoạt trong việc mở rộng, thay đổi quy mô của hệ thống. Điều này phù hợp với hệ thống mà nhóm chúng tôi muốn xây dựng, vì đây là một ứng dụng mới, chưa có ứng dụng nào trên thị trường Việt Nam để tham khảo, do đó mà việc phân tích và thay đổi nghiệp vụ liên tục trong quá trình phát triển ứng dụng là khó tránh khỏi.

Vì những lí do nêu trên mà nhóm đã quyết định lựa chọn kiến trúc n-tiers cho hệ thống của mình.

3.6 Class diagram

Các class diagram thể hiện thiết kế của hệ thống, gồm 4 lớp

3.6.1 Overview

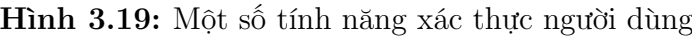
Tổng quan class diagram

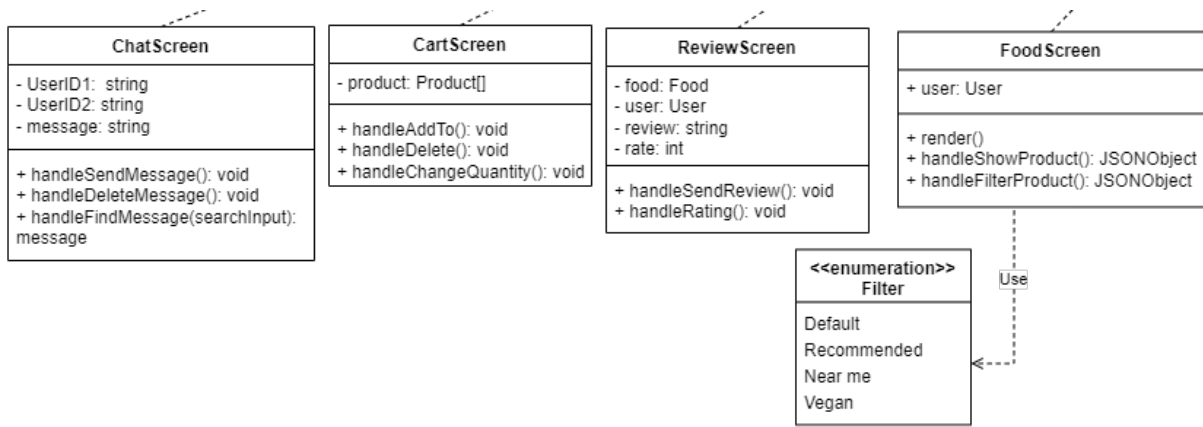


3.6.2 Presentation layer

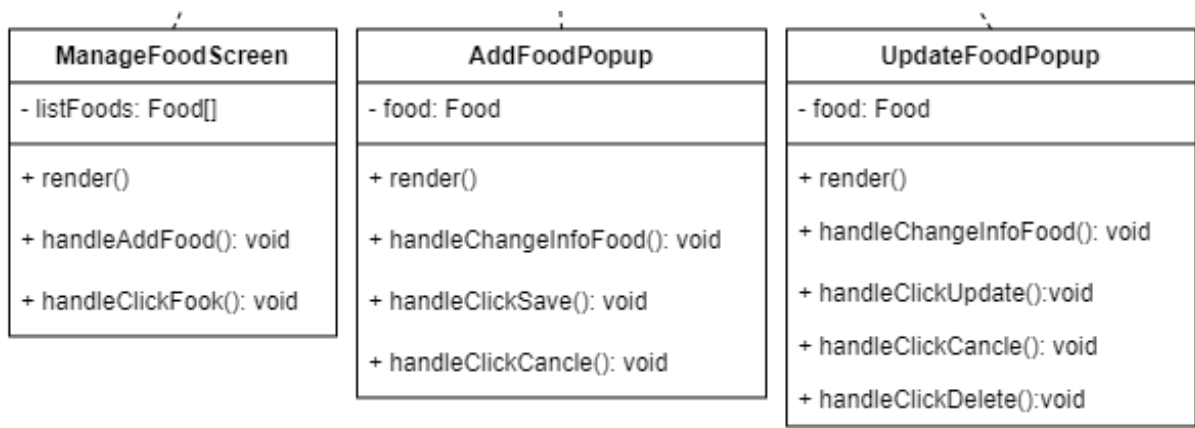


Một số class mà chúng tôi đã thiết kế:

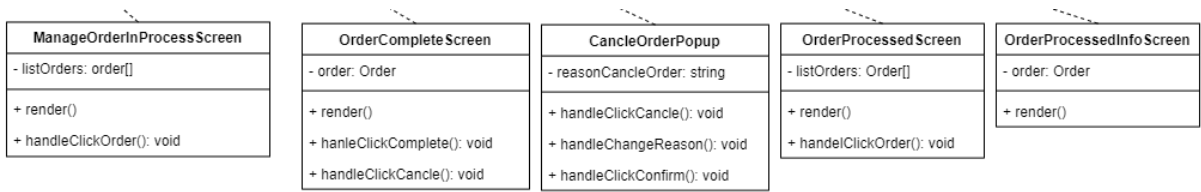




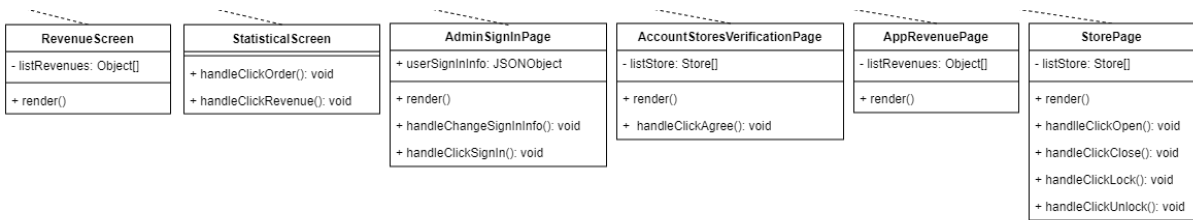
Hình 3.20: Một số tính năng của người dùng



Hình 3.21: Một số tính năng quản lý sản phẩm của chủ cửa hàng

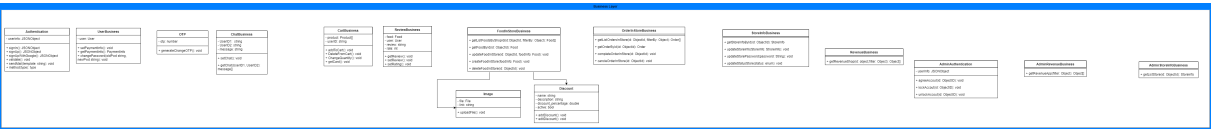


Hình 3.22: Một số tính năng quản lý đơn hàng của chủ cửa hàng



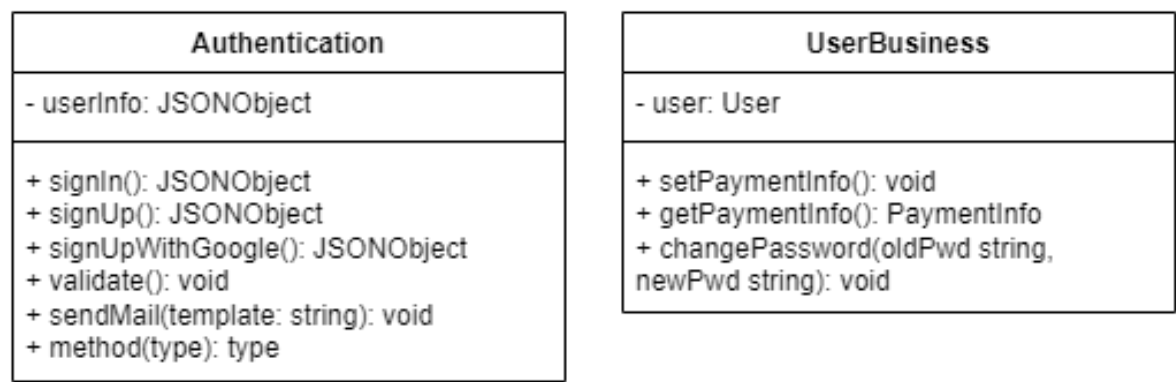
Hình 3.23: Các tính năng của admin

3.6.3 Business layer

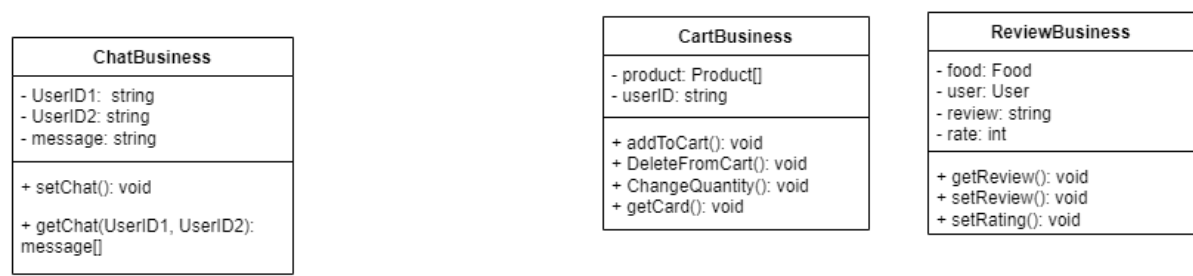


Hình 3.24: Business layer

Business layer là nơi đáp ứng các yêu cầu thao tác và xử lý dữ liệu từ Presentation layer trước khi truyền xuống Persistence layer để tương tác với cơ sở dữ liệu.



Hình 3.25: User business



Hình 3.26: User feature business

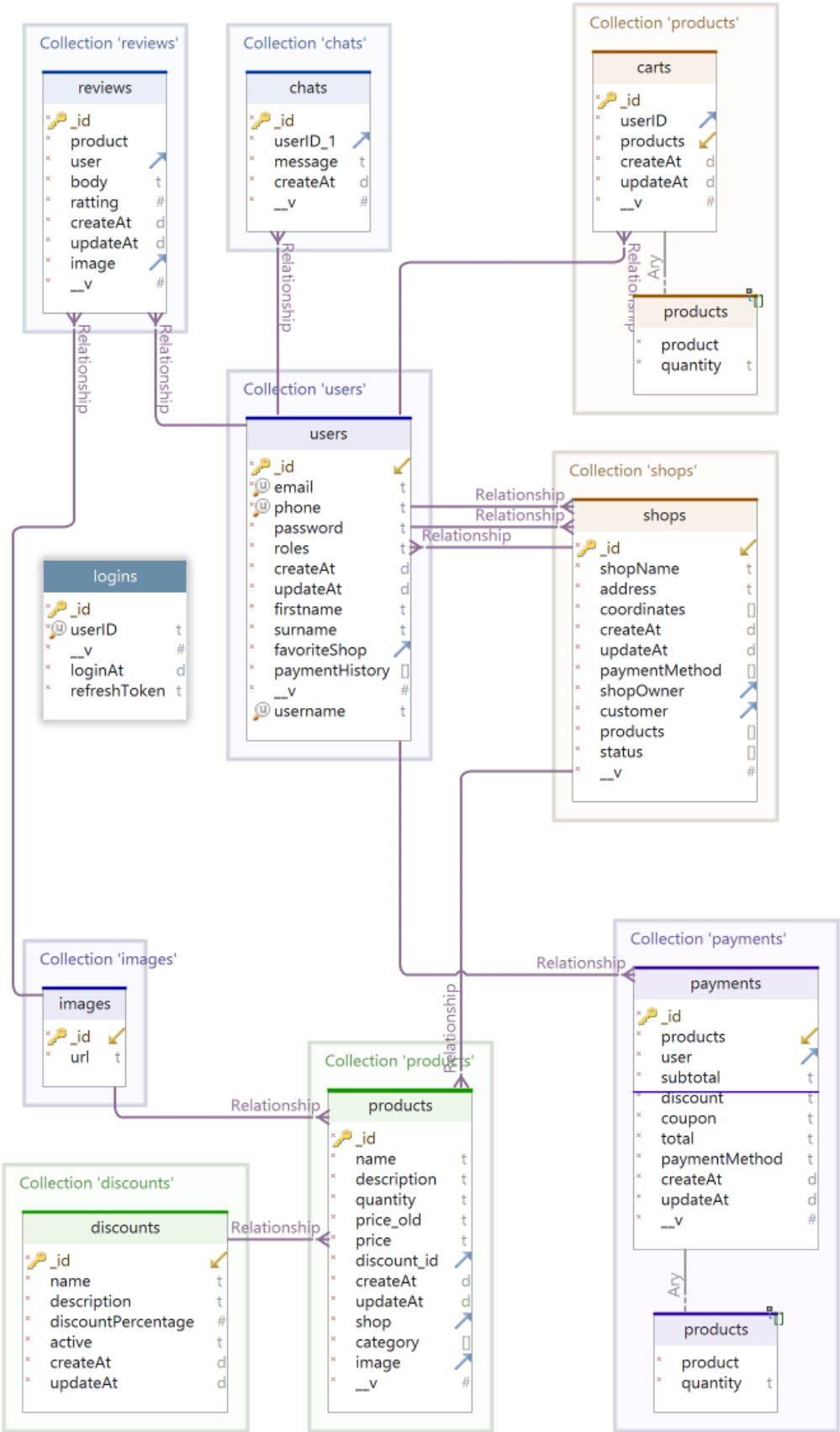
[illegible]

Các class ở layer này được chúng tôi thiết kế để giao tiếp với cơ sở dữ liệu ở bên dưới một cách hiệu quả.

Để thiết kế database cho mục đích lưu trữ dữ liệu, nhóm quyết định chọn cơ sở dữ liệu hướng tài liệu (document-oriented) để hiện thực, và hệ quản trị cơ sở dữ liệu được chọn là MongoDB. Những lợi ích của việc sử dụng hệ cơ sở dữ liệu hướng tài liệu như sau:

- Trang 62/105

- Indexes có thể được tạo để cải thiện hiệu suất của các tìm kiếm trong MongoDB. Bất kỳ trường nào trong document MongoDB đều có thể được index
- MongoDB sử dụng khái niệm sharding để chia tỷ lệ theo chiều ngang (vertical scaling) bằng cách chia nhỏ dữ liệu trên nhiều instances MongoDB. MongoDB có thể chạy trên nhiều máy chủ, do đó nó có khả năng mở rộng cực lớn và điều này cũng phù hợp với đặc điểm mà nhóm muốn khi tìm kiếm một database để lưu trữ dữ liệu



Hình 3.30: Schema design của database

Chương 4

HIỆN THỰC HỆ THỐNG

Chương này giới thiệu các công nghệ được sử dụng trong đề tài, các giao diện đã hiện thực và triển khai hệ thống.

4.1 Công nghệ sử dụng

Mục này giới thiệu các công nghệ được sử dụng trong đề tài.

4.1.1 Front-end

4.1.1.1 React Native

React Native là một framework mã nguồn mở được tạo ra bởi Facebook để phát triển ứng dụng di động. Nó sử dụng ngôn ngữ lập trình JavaScript để tạo ra ứng dụng di động cho cả hai hệ điều hành iOS và Android với cùng một cơ sở mã.

Lý do chọn React Native

- Khả năng tái sử dụng mã: Điểm mạnh lớn nhất của React Native là khả năng tái sử dụng mã. Nhà phát triển không cần phải viết mã riêng biệt cho từng nền tảng (Android và iOS). Khoảng 90% mã có thể được tái sử dụng giữa hai nền tảng, giúp tăng tốc quá trình phát triển và mang lại hiệu quả đáng kể. Việc tái sử dụng mã cũng giảm thời gian đưa sản phẩm ra thị trường và yêu cầu ít công sức bảo trì hơn.
- Live reload: Tính năng live reload của React Native cho phép nhìn thấy và làm việc với các thay đổi ngay lập tức. Bạn có thể sửa lỗi trong mã khi ứng dụng đang chạy, và tính năng live reload sẽ tự động làm mới ứng dụng để bạn có thể thấy những thay đổi trong mã.
- Tập trung vào giao diện người dùng: React Native sử dụng thư viện React JavaScript để xây dựng giao diện ứng dụng nhanh chóng và đáp ứng nhiều nhu cầu. Framework này cung cấp khả năng kết xuất tuyệt vời và sử dụng phương pháp dựa trên thành phần, giúp dễ dàng tạo ra giao diện người dùng đơn giản hoặc phức tạp.

- Plugin của bên thứ ba: Xây dựng một ứng dụng từ đầu có thể tốn kém, vì vậy React Native cung cấp một số lựa chọn plugin của bên thứ ba bao gồm cả các native modules và JavaScript-based để nhà phát triển sử dụng. Các plugin của bên thứ ba giúp cải thiện hiệu suất ứng dụng và tiết kiệm thời gian phát triển.

4.1.1.2 Redux

Redux là một thư viện quản lý trạng thái được sử dụng cùng với React để giải quyết vấn đề quản lý state trong ứng dụng. Với Redux, các state được tập trung và quản lý trong một object duy nhất gọi là store, và các component có thể lấy dữ liệu từ store và cập nhật state.

Redux có 3 thành phần chính là Store, Action và Reducer.

- Store: Store là nơi lưu trữ các state chung của ứng dụng. Nó là duy nhất và được quản lý bởi Redux. Các component có thể truy cập vào store để lấy dữ liệu, cập nhật state và lắng nghe sự thay đổi của các state.
- Action: Action là các object được gửi vào store để cập nhật state. Một action thường bao gồm hai phần chính là type và payload. Phần type xác định loại action và được sử dụng để Redux chọn reducer tương ứng để xử lý action đó. Phần payload chứa dữ liệu được sử dụng để cập nhật state.
- Reducer: Reducer là các hàm thực hiện việc cập nhật state trong store. Mỗi reducer nhận vào một state hiện tại và một action. Khi nhận được một action, reducer tính toán và trả về một state mới, sau đó Redux cập nhật state này vào store. Các reducer phải là pure function để đảm bảo tính đồng nhất và dễ hiểu. Pure function trong Redux trả về một state mới thay vì sửa đổi state hiện tại, điều này cho phép Redux hỗ trợ debug thông qua chức năng time travel (quay lại xem lại tác động của từng action lên store).

Sử dụng Redux giúp giải quyết các vấn đề quản lý state phức tạp trong ứng dụng, cho phép tập trung vào logic của phần mềm. Nó làm cho mã nguồn trở nên rõ ràng hơn, phát triển ứng dụng nhanh chóng và dễ bảo trì khi quy mô mã nguồn tăng lên.

4.1.2 Back-end

4.1.2.1 GraphQL

GraphQL là một ngôn ngữ thao tác và truy vấn dữ liệu nguồn mở cho API, được phát triển bởi Facebook và công bố vào năm 2015 (trước đó đã được phát triển nội bộ từ năm 2012). Nó cung cấp cho client một cách thức dễ dàng để yêu cầu chính xác những dữ liệu họ cần từ API, giúp việc phát triển API linh hoạt hơn theo thời gian.

Lý do chọn GraphQL

- GraphQL schema: GraphQL có khả năng tự động tạo ra một nguồn đáng tin cậy trong ứng dụng thông qua việc xác định schema. Schema này định nghĩa

các kiểu dữ liệu và quan hệ giữa chúng, giúp đảm bảo tính nhất quán trong việc truy vấn và trả về dữ liệu.

- Tiết kiệm số lượng request: Với GraphQL, khách hàng có thể yêu cầu chính xác thông tin mà họ cần một lần duy nhất, giảm thiểu số lượng request tới server. Điều này giúp tối ưu hóa hiệu suất mạng và giảm thời gian phản hồi.
- Kiểm soát và xử lý data type: GraphQL hỗ trợ mạnh mẽ việc kiểm soát và xử lý các kiểu dữ liệu. Điều này giúp hạn chế sự không nhất quán trong giao tiếp giữa server và client.
- Mở rộng API: GraphQL cho phép phát triển thêm API mà không ảnh hưởng đến các truy vấn hiện có. Điều này giúp tiện lợi trong việc mở rộng và cập nhật API mà không gây rối trong việc sử dụng.
- Tương thích và linh hoạt: GraphQL không yêu cầu một kiến trúc ứng dụng cụ thể và có thể hoạt động như một REST API. Nó cũng tương thích tốt với các công cụ API hiện có và có tài liệu học phong phú và chi tiết.

Ngoài ra, GraphQL cung cấp khả năng thực hiện các thao tác create, update và delete dữ liệu với cấu trúc sáng sủa và phân tầng. Điều này thuận lợi cho các lập trình viên phía client và tương thích tốt với các cơ sở dữ liệu phổ biến hiện nay. Thêm vào đó, GraphQL hỗ trợ websocket để client có thể lắng nghe dữ liệu từ server một cách dễ dàng, giúp việc phát triển ứng dụng realtime trở nên thuận tiện.

4.1.2.2 NodeJS

Node.js là một môi trường runtime mã nguồn mở và đa nền tảng, được sử dụng để chạy các ứng dụng ngoài trình duyệt của client. Nó được phát triển bởi Ryan Dahl vào năm 2009 và được coi là một giải pháp lý tưởng cho các ứng dụng đòi hỏi xử lý nhiều dữ liệu nhờ mô hình hướng sự kiện không đồng bộ.

Lý do chọn NodeJS

- Hướng sự kiện không đồng bộ (event-driven): Node.js sử dụng mô hình IO hướng sự kiện không đồng bộ, cho phép xử lý nhiều yêu cầu đồng thời mà không cần chờ đợi kết quả từng yêu cầu trước. Điều này giúp tăng hiệu suất và khả năng mở rộng của ứng dụng.
- Chia sẻ code phía client và server: Với Node.js, bạn có thể chia sẻ cùng một codebase giữa phía client và phía server, sử dụng JavaScript cho cả hai phía. Điều này giúp giảm thiểu sự trùng lặp và tăng khả năng tái sử dụng mã nguồn.
- NPM (Node Package Manager) và module Node.js: Node.js có NPM, một công cụ quản lý gói phổ biến và mạnh mẽ. Nó cung cấp hàng ngàn module được phát triển bởi cộng đồng, giúp bạn nhanh chóng tích hợp các chức năng và tính năng mới vào ứng dụng của mình.

- Cộng đồng hỗ trợ: Node.js có một cộng đồng phát triển tích cực và rộng lớn, đồng nghĩa với việc có sự hỗ trợ, tài liệu và nguồn thông tin phong phú từ cộng đồng này.
- Stream dữ liệu lớn: Node.js cho phép xử lý dữ liệu thông qua cơ chế stream, cho phép xử lý các file có kích thước lớn một cách hiệu quả và không gây tắc nghẽn.

4.2 Hiện thực giao diện

Mục này trình bày các màn hình của ứng dụng đã được nhóm hiện thực cùng với mô tả.

4.2.1 Xác thực người dùng

4.2.1.1 Đăng ký

Chúng tôi yêu cầu người dùng phải đăng ký mới có thể sử dụng ứng dụng của chúng tôi. Chỉ có một tùy chọn duy nhất là đăng ký bằng số điện thoại để đảm bảo số điện thoại người dùng có thể liên lạc được.

< Đăng ký

Họ và tên
Nhập họ và tên

Email
Nhập email

Mật khẩu
.....

Nhập lại mật khẩu
.....

Tiếp tục

Hình 4.1: Giao diện trang đăng ký người dùng

< Đăng ký

Họ và tên
Nhập họ và tên

Email
Nhập email

CCCD
Nhập CCCD

Tên cửa hàng
Nhập tên cửa hàng

Địa chỉ
Nhập địa chỉ

Chọn địa chỉ trên bản đồ

Mật khẩu
.....

Nhập lại mật khẩu
.....

Tiếp tục

Hình 4.2: Giao diện trang đăng ký chủ cửa hàng

< Đăng ký

Nhập otp mà chúng tôi đã gửi cho bạn
0905467834

.....

Tiếp tục

Hình 4.3: Giao diện trang đăng ký

< Đăng ký

Họ và tên
Nhập họ và tên

Email
Nhập email

Mật khẩu
.....

Nhập lại mật khẩu
.....

Tiếp tục

Hình 4.5: Giao diện trang đăng ký

< Đăng ký

Họ và tên
Nhập họ và tên

Email
Nhập email

CCCD
Nhập CCCD

Tên cửa hàng
Nhập tên cửa hàng

Địa chỉ
Nhập địa chỉ

Chọn địa chỉ trên bản đồ

Mật khẩu
.....

Nhập lại mật khẩu
.....

Tiếp tục

Hình 4.4: Giao diện trang đăng ký

< Đăng ký

Nhập số điện thoại của bạn:
Nhập số điện thoại

Tiếp tục

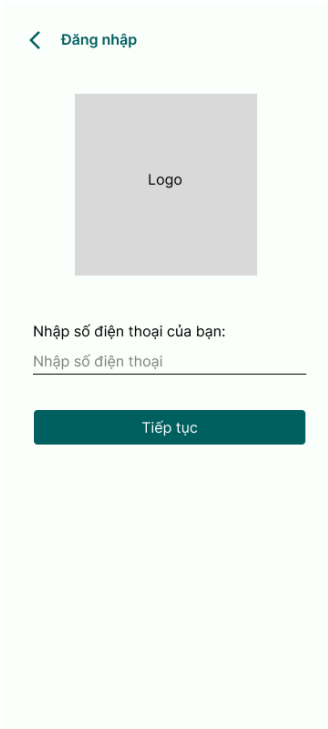
Hình 4.6: Giao diện trang đăng ký

4.2.1.2 Đăng nhập

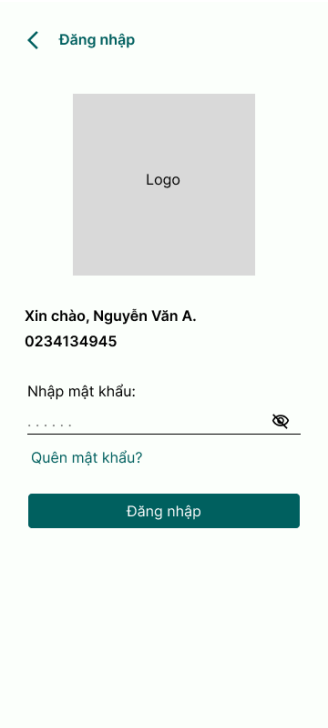
Sau khi đăng ký thì người dùng có thể đăng nhập vào ứng dụng với số điện thoại và mật khẩu ở trên



Hình 4.7: Giao diện trang đăng nhập

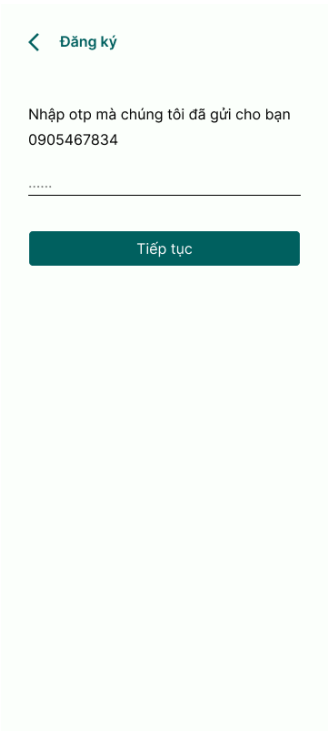


Hình 4.8: Giao diện trang đăng nhập

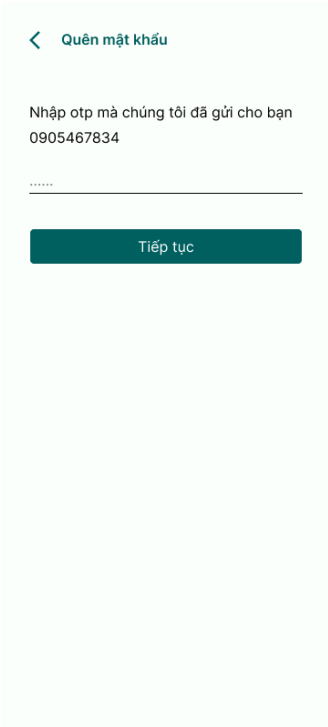


Hình 4.9: Giao diện trang đăng nhập

Người dùng quên mật khẩu có thể lấy lại mật khẩu bằng cách xác nhận OTP thông qua tin nhắn gửi về số điện thoại.



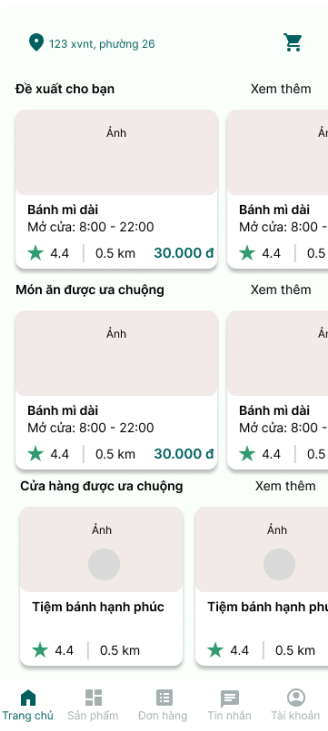
Hình 4.10: Giao diện trang nhập OTP



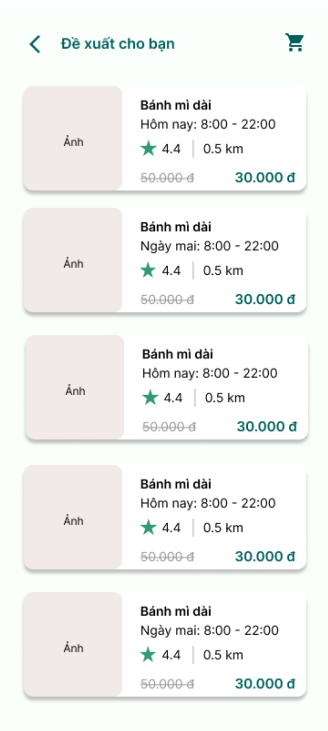
Hình 4.11: Giao diện trang quên mật khẩu

4.2.2 Tính năng người dùng

Sau khi đăng nhập thì đây là giao diện **Trang chủ** của người dùng. Người dùng có thể bấm vào các nút xem thêm để xem toàn bộ các sản phẩm tương ứng.



Hình 4.12: Giao diện trang chủ



Hình 4.13: Giao diện trang sản phẩm

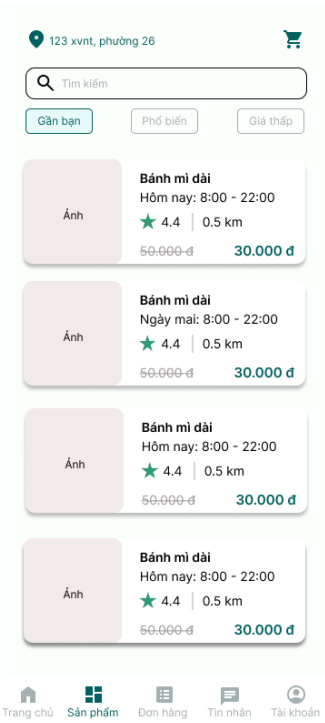
Hoặc người dùng cũng có thể chuyển sang trang **Sản phẩm** để tìm kiếm các sản phẩm mong muốn. Người dùng cũng có thể bấm vào sản phẩm để xem chi tiết về sản phẩm hoặc cửa hàng.



Hình 4.14: Giao diện trang sản phẩm

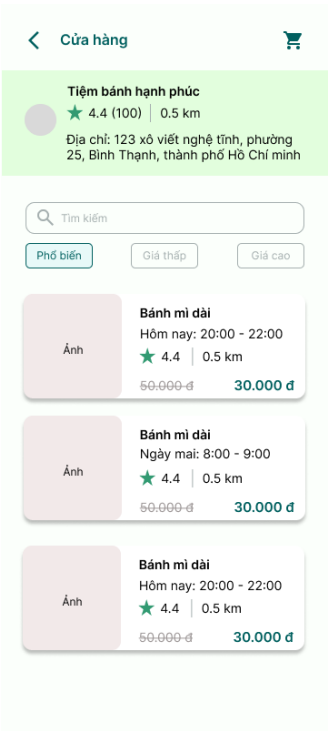


Hình 4.16: Giao diện trang sản phẩm - chi tiết

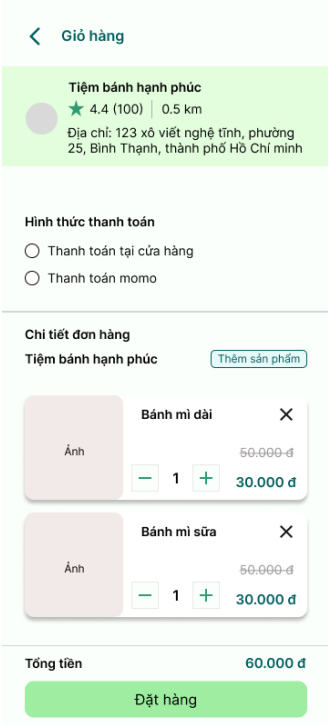


Hình 4.15: Giao diện trang sản phẩm

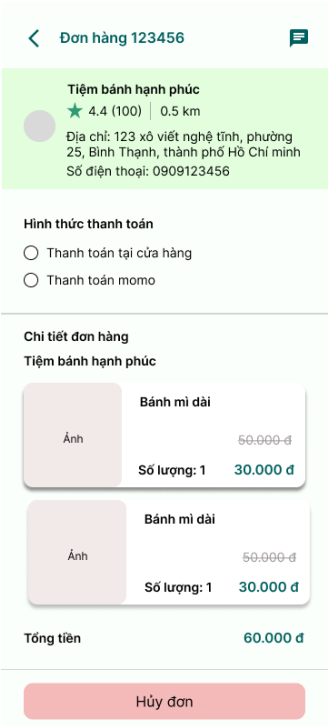
Sau khi tìm được sản phẩm mong muốn, người dùng bấm vào nút **Thêm vào giỏ hàng** để thêm sản phẩm vào giỏ hàng. Ở giao diện trang **Giỏ hàng**, sau khi điều chỉnh các thông tin đơn hàng và thanh toán thì người dùng có thể chọn **Đặt hàng**, hoặc người dùng cũng có thể chọn **Hủy đơn**. Sau khi nhận hàng thành công người dùng cũng có thể để lại đánh giá về chất lượng sản phẩm.



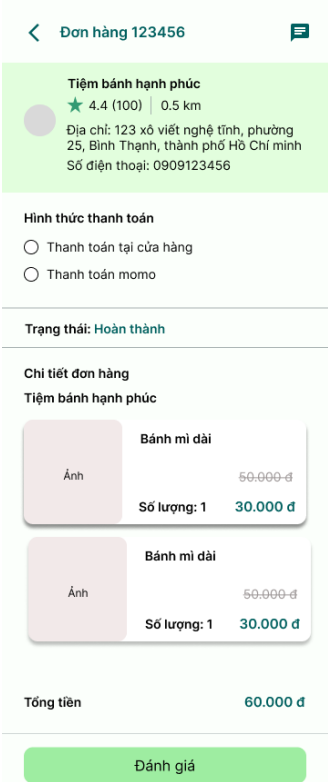
Hình 4.17: Giao diện trang sản phẩm - thông tin cửa hàng



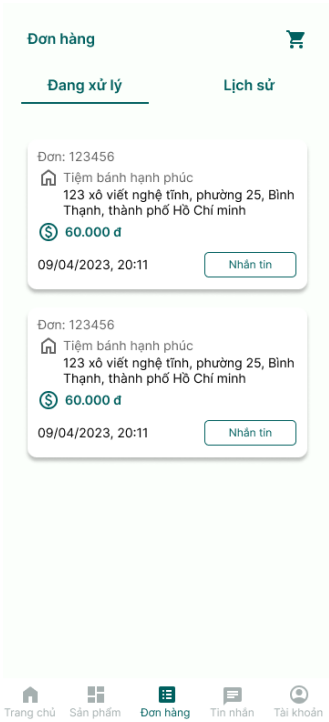
Hình 4.19: Giao diện trang giỏ hàng



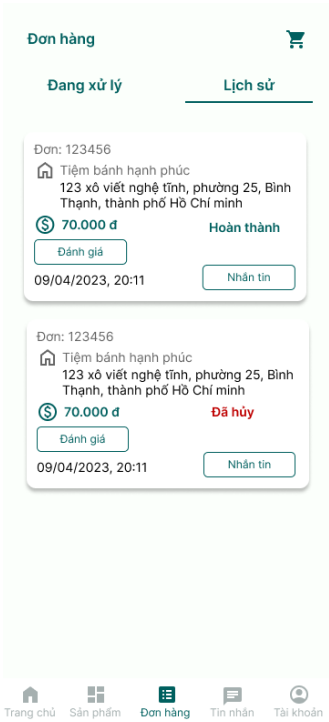
Hình 4.18: Giao diện trang đặt hàng



Hình 4.20: Giao diện trang đặt hàng

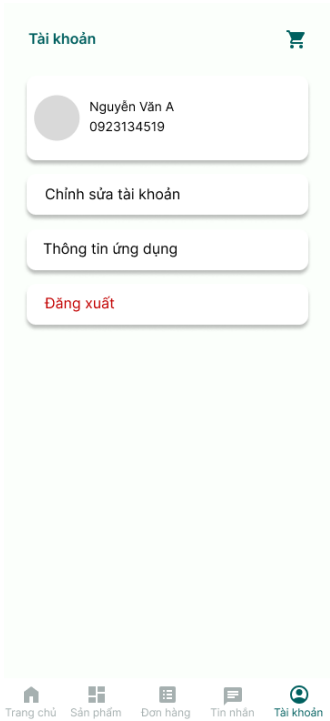


Hình 4.21: Giao diện trang giỏ hàng



Hình 4.22: Giao diện trang giỏ hàng

Ở giao diện trang **Thông tin người dùng**, người dùng có thể điều chỉnh các thông tin cá nhân như Họ và tên, email hay mật khẩu người dùng.



Hình 4.23: Giao diện trang thông tin người dùng



Hình 4.24: Giao diện trang chỉnh sửa thông tin người dùng

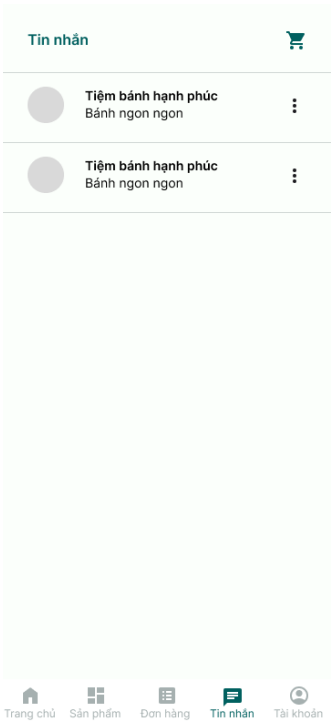


Hình 4.25: Giao diện trang đổi mật khẩu người dùng

Người dùng cũng có thể nhắn tin với chủ cửa hàng để trao đổi thêm các thông tin về sản phẩm cũng.



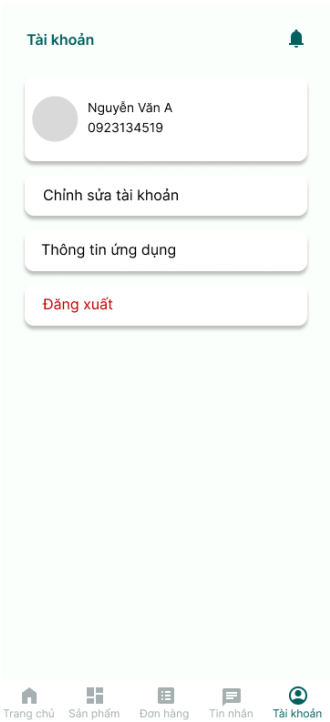
Hình 4.26: Giao diện trang tin nhắn



Hình 4.27: Giao diện trang tin nhắn

4.2.3 Tính năng chủ cửa hàng

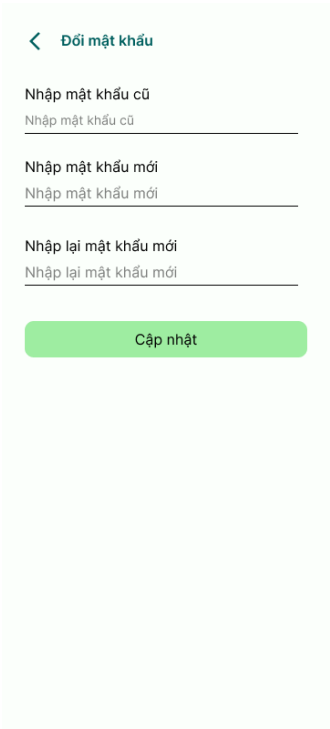
Đối với người dùng là chủ cửa hàng, người dùng cũng có các tính năng chỉnh sửa thông tin người dùng cơ bản. Bên cạnh đó có thêm các tính năng dành riêng cho chủ cửa hàng như xem **Lịch sử đơn hàng**, **Thống kê doanh thu**, **thêm sản phẩm** cùng với thông tin mô tả hoặc **xóa sản phẩm** khỏi cửa hàng. Chủ cửa hàng cũng có thể quản lý đơn hàng của mình với tùy chọn Hủy đơn hoặc Hoàn thành ở giao diện trang **Quản lý đơn hàng**.



Hình 4.28: Giao diện trang thông tin chủ cửa hàng



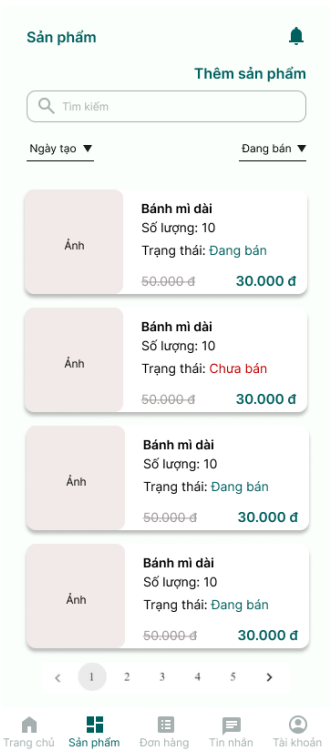
Hình 4.29: Giao diện trang chỉnh sửa thông tin chủ cửa hàng



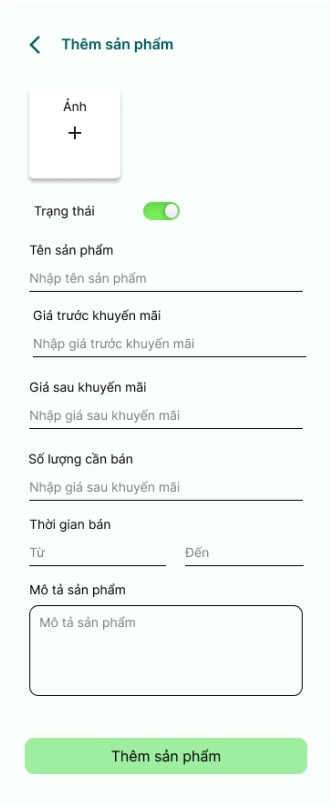
Hình 4.30: Giao diện trang đổi mật khẩu chủ cửa hàng



Hình 4.32: Giao diện trang chủ chủ cửa hàng



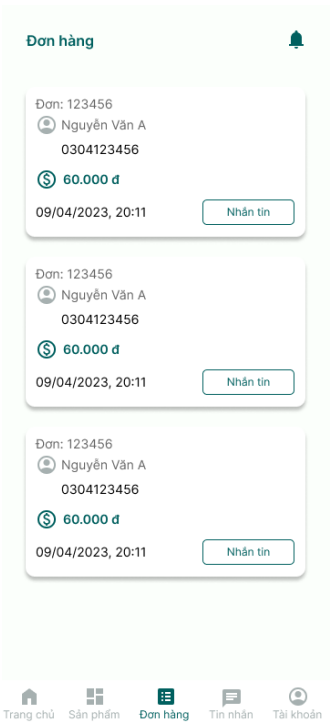
Hình 4.31: Giao diện trang sản phẩm chủ cửa hàng



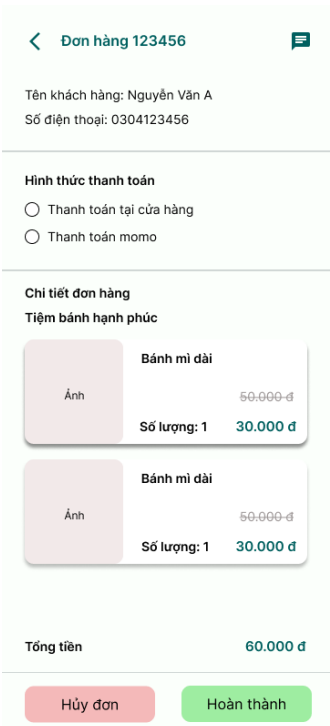
Hình 4.33: Giao diện trang thêm sản phẩm



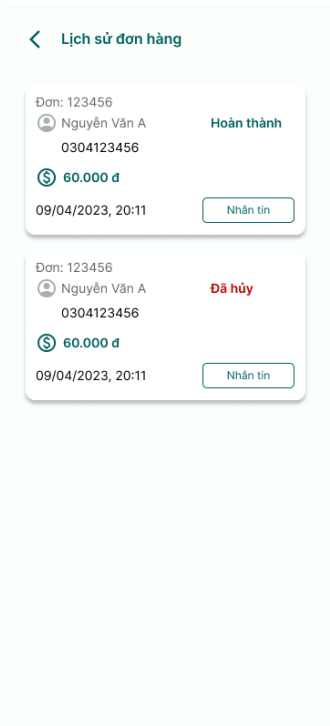
Hình 4.34: Giao diện trang thêm sản phẩm



Hình 4.36: Giao diện trang quản lý đơn hàng



Hình 4.35: Giao diện trang quản lý đơn hàng



Hình 4.37: Giao diện trang quản lý đơn hàng

<

Đơn hàng 123456

Tên khách hàng: Nguyễn Văn A

Số điện thoại: 0304123456

Hình thức thanh toán

☐ Thanh toán tại cửa hàng

☐ Thanh toán momo

Trạng thái: Hoàn thành

Chi tiết đơn hàng

Tiệm bánh hạnh phúc

Ảnh

Bánh mì dài

50.000 đ

Số lượng: 1

30.000 đ

Ảnh

Bánh mì dài

50.000 đ

Số lượng: 1

30.000 đ

Tổng tiền

60.000 đ

Hình 4.38: Giao diện trang quản lý đơn hàng

<

Doanh thu

01/05/2023 - 0705/2023

▼

Số đơn hàng đã bán: 20

Số sản phẩm đã bán: 100

Doanh thu: 2.000.000

Hình 4.39: Giao diện trang thống kê doanh thu

4.2.4 Tính năng admin

Admin có thể quản lý phê duyệt tài khoản chủ cửa hàng hoặc thay đổi tình trạng hoạt động của các cửa hàng. Bên cạnh đó admin cũng có thể xem doanh thu tổng của các cửa hàng.

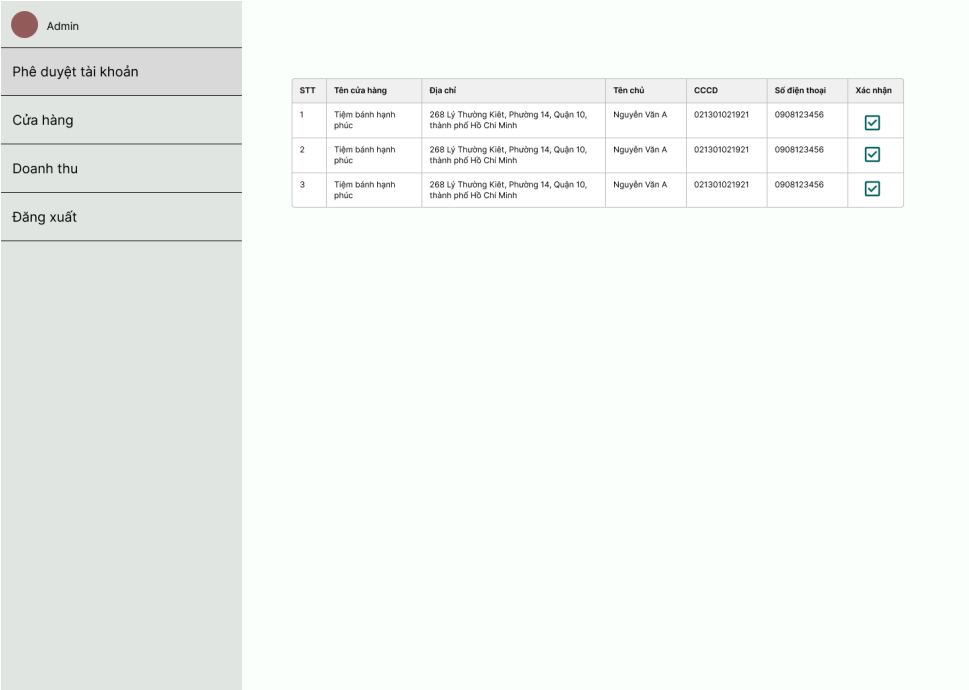
ADMIN

Tên đăng nhập

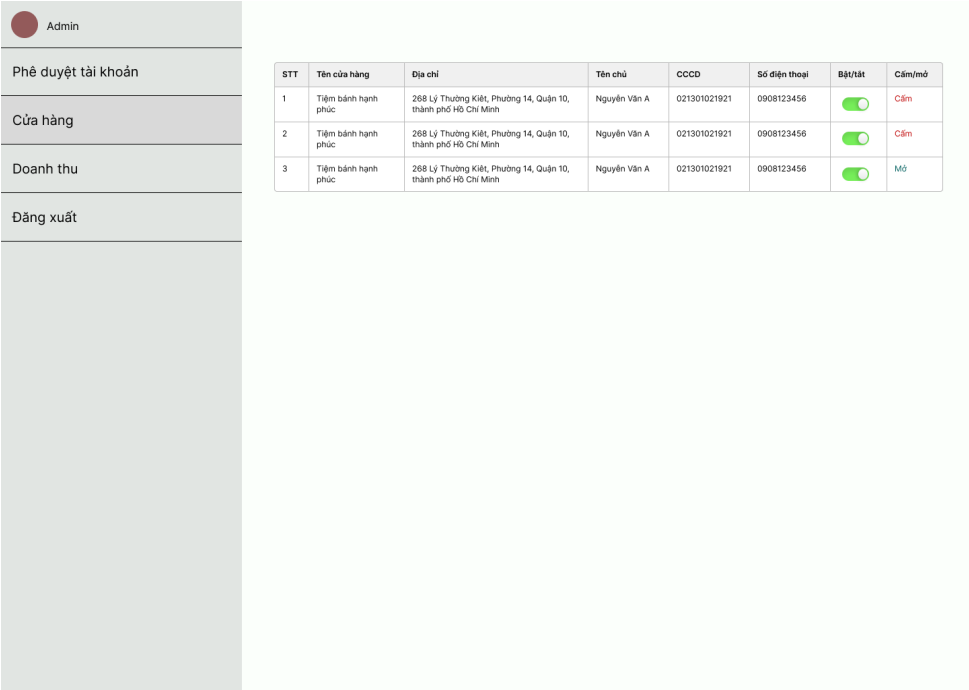
Mật khẩu

ĐĂNG NHẬP

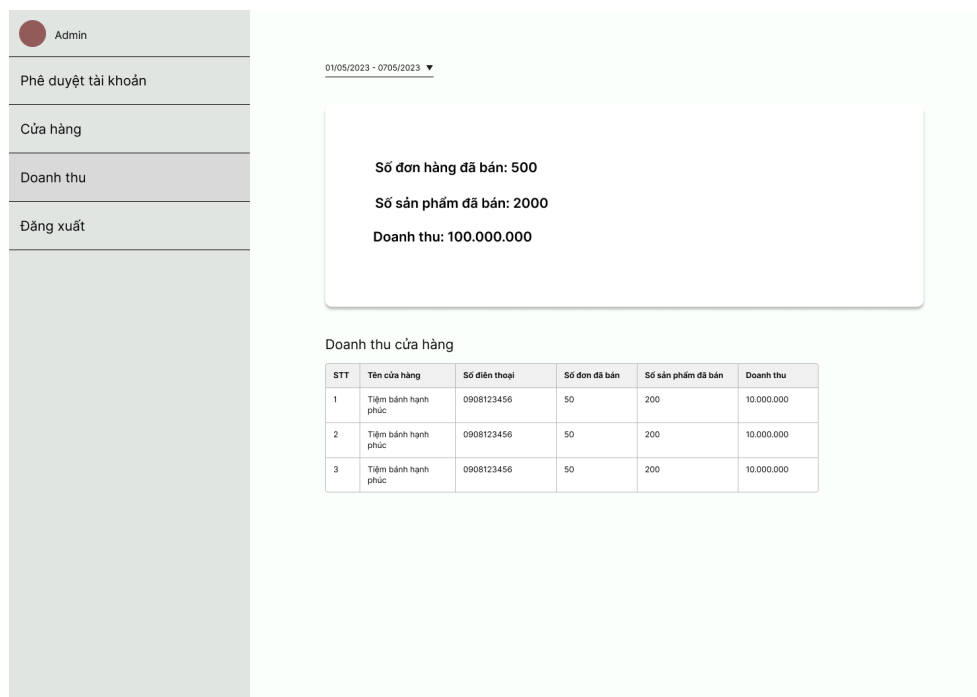
Hình 4.40: Giao diện trang đăng nhập admin



Hình 4.41: Giao diện trang phê duyệt tài khoản



Hình 4.42: Giao diện trang quản lý cửa hàng



Hình 4.43: Giao diện trang quản lý doanh thu

4.3 Triển khai hệ thống

Có nhiều cách để triển khai hệ thống phần mềm online, chúng tôi đã tìm hiểu được một số cách như sau:

- Tự host server riêng: phương pháp này đòi hỏi lập trình viên phải thiết lập mọi thứ thủ công, bao gồm việc cài đặt server, database server và deploy ứng dụng lên server vừa cài đặt. Phương pháp này cho phép lập trình viên quyền tự chủ hoàn toàn về dữ liệu cũng như là sự bảo mật, đi kèm với đó là khả năng tùy biến theo ý muốn cực tốt. Tuy nhiên phương pháp này cũng đòi hỏi lập trình viên phải có kiến thức về quản trị hệ thống và bảo mật dữ liệu. Và do phải đầu tư một máy chủ đủ mạnh để phục vụ cho việc hosting cùng bảo trì và nâng cấp hệ thống, phương pháp này cũng đòi hỏi khoản phí đầu tư ban đầu khá lớn và phí duy trì không hề nhỏ.
- Platform-as-a-Service (PaaS): đây là một mô hình cung cấp dịch vụ điện toán đám mây, trong đó nhà cung cấp cloud chịu trách nhiệm quản lý và cung cấp các tài nguyên phần mềm và phần cứng cho khách hàng. PaaS cho phép người dùng xây dựng, triển khai và quản lý ứng dụng của họ trên nền tảng điện toán đám mây được quản lý. Phương pháp này khắc phục phần nào những nhược điểm của việc tự host một server và kèm với đó là sự linh động nhưng cũng đánh mất tính tùy biến và tính đa dạng.
- Infrastructure-as-a-Service (IaaS): đây là một mô hình cung cấp dịch vụ điện toán đám mây, trong đó người dùng có thể thuê tài nguyên máy chủ ảo và các thành phần hạ tầng khác từ nhà cung cấp dịch vụ. IaaS cho phép tổ chức và cá nhân xây dựng và quản lý các ứng dụng, lưu trữ dữ liệu và triển khai các công nghệ thông tin mà không cần đầu tư vào việc xây dựng và duy trì

hạ tầng vật lý. Với phương pháp này người dùng có thể dễ dàng bắt đầu dự án một cách nhanh chóng, tiết kiệm đáng kể thời gian và tiền bạc đầu tư cho việc hosting, cùng với đó là sự linh hoạt về khả năng mở rộng và đa dạng dịch vụ tùy theo nhu cầu, công nghệ của dự án, thậm chí là vị trí địa lý. Tuy không cần đầu tư quá nhiều về chi phí ban đầu nhưng mô hình này vẫn đòi hỏi một khoản chi phí đáng kể về đường truyền kèm với đó là lập trình viên phải có kiến thức kỹ thuật cao để xử lý các vấn đề liên quan đến việc triển khai, bảo trì và giám sát hạ tầng.

- Docker containers: đây là phương pháp triển khai ứng dụng lên các docker container. Chúng là các run-time environment mà ta có thể chạy các ứng dụng một cách độc lập. Cụ thể, docker cho phép lập trình viên đóng gói ứng dụng và các phụ thuộc của nó trong một container, và deploy các container này trên bất kỳ máy chủ nào. Phương pháp này đảm bảo tính di động và tương thích với mọi môi trường và ngôn ngữ.

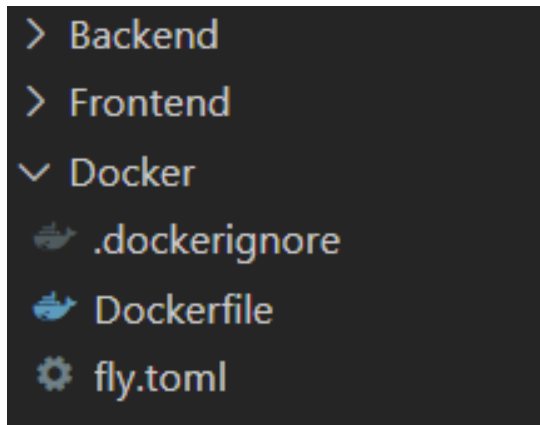
Sau khi cân nhắc thì chúng tôi đã lựa chọn cách tiếp cận docker containers vì:

- Tính di động: docker cho phép lập trình viên đóng gói ứng dụng và các dependencies của nó trong container, và deploy container này lên bất kỳ máy chủ nào. Việc này tăng tính di động và tương thích với mọi công nghệ được sử dụng trong ứng dụng.
- Tính tương thích: như đã đề cập ở trên, docker containers có thể triển khai trên bất kỳ hệ điều hành hoặc máy chủ nào, giúp cho việc phát triển và deploy ứng dụng dễ dàng hơn.
- Quản lý tài nguyên hiệu quả: docker giúp quản lý tài nguyên một cách hiệu quả hơn, cho phép chia sẻ tài nguyên giữa các container khác nhau và sử dụng tài nguyên một cách tối ưu.
- Tính bảo mật cao: docker containers cung cấp một môi trường tách biệt giữa các container, giúp giảm thiểu rủi ro bảo mật và đảm bảo tính riêng tư của dữ liệu.

Tuy nhiên, cách tiếp cận trên cũng có một số nhược điểm như:

- Đòi hỏi kiến thức cao về docker: Triển khai docker containers đòi hỏi lập trình viên phải có kiến thức cao về docker và cách quản lý container.
- Tốn thời gian đóng gói: Đóng gói ứng dụng và các dependencies vào một container đòi hỏi thời gian để tạo ra một container hoàn chỉnh.

Sau khi đã quyết định chọn docker containers, chúng tôi triển khai hệ thống như sau:



Hình 4.44: Cấu trúc thư mục mã nguồn

Trong đó:

- Backend: chứa toàn bộ source code của backend.
- Frontend: chứa toàn bộ source code của frontend.
- Docker: chứa Dockerfile Docker để build cho backend.
- fly.toml: là file docker compose để deploy backend lên server của **fly.io**

Dockerfile của chúng tôi như sau:

Dockerfile: (3)

```
# syntax = docker/dockerfile:1

# Adjust NODE_VERSION as desired
ARG NODE_VERSION=18.16.1
FROM node:${NODE_VERSION}-slim as base

LABEL fly_launch_runtime="Node.js"

# Node.js app lives here
WORKDIR /app

# Set production environment
ENV NODE_ENV=development
ARG YARN_VERSION=1.22.4
RUN npm install -g yarn@$YARN_VERSION --force

# Throw-away build stage to reduce size of final image
FROM base as build

# Install packages needed to build node modules
RUN apt-get update -qq && \
    apt-get install -y python-is-python3 pkg-config build-essential
```

```
# Install node modules
COPY --link package.json yarn.lock ./
RUN yarn install

# Copy application code
COPY . .
COPY src/ src/

# Final stage for app image
FROM base

# Copy built application
COPY --from=build /app /app

# Start the server by default, this can be overwritten at runtime
EXPOSE 3000
CMD [ "yarn", "run", "start" ]
```

Sau khi đã hoàn thành các bước cấu hình trên thì khi mà chúng tôi muốn triển khai hệ thống cần chạy lệnh sau:

flyctl deploy

Sau khi câu lệnh trên thực thi thì backend sẽ được triển khai lên hệ thống của **fly.io** và sẽ được truy cập qua <https://daily-groceries.fly.dev> để gọi tới api của backend.

Chương 5

KIỂM THỬ VÀ ĐÁNH GIÁ HỆ THỐNG

Kiểm thử là phần quan trọng và rất cần thiết trong quá trình phát triển sản phẩm nhằm đảm bảo chức năng hoạt động chính xác cũng như phát hiện kịp thời và khắc phục những lỗi sai có thể gây ảnh hưởng nghiêm trọng đến hệ thống và người dùng.

5.1 Tổng quát

Đối với đề tài này, nhóm đã thực hiện **exception test** với tổng cộng 36 test case. Trong đó 36 test case được đánh giá là đạt, đạt tỉ lệ 100%.

5.1.1 Người dùng

5.1.1.1 Đăng nhập

ID	Mô tả	Số testcase	Kết quả
DNSN01	Đăng nhập thành công	1	Đạt
DNSN02	Đăng nhập thất bại	3	Đạt

5.1.1.2 Đăng ký

ID	Mô tả	Số testcase	Kết quả
DKSN01	Đăng ký thành công	1	Đạt
DKSN02	Đăng ký thất bại	2	Đạt

5.1.1.3 Cập nhật thông tin

ID	Mô tả	Số testcase	Kết quả
CNTTSN01	Cập nhật thông tin cá nhân	2	Đạt
CNTTSN02	Cập nhật mật khẩu	3	Đạt

5.1.1.4 Sản phẩm

ID	Mô tả	Số testcase	Kết quả
SPSN01	Xem danh sách sản phẩm	1	Đạt
SPSN02	Xem chi tiết sản phẩm	1	Đạt

5.1.1.5 Giỏ hàng

ID	Mô tả	Số testcase	Kết quả
GHSN01	Thêm vào giỏ hàng	3	Đạt
GHSN02	Xóa sản phẩm khỏi giỏ hàng	1	Đạt
GHSN03	Cập nhật số lượng sản phẩm trong giỏ hàng	2	Đạt
GHSN04	Đặt hàng	1	Đạt

5.1.1.6 Đơn hàng

ID	Mô tả	Số testcase	Kết quả
DHSN01	Xem danh sách các đơn hàng đang xử lý	1	Đạt
DHSN02	Xem chi tiết đơn hàng đang xử lý	1	Đạt
DHSN03	Hủy đơn hàng	1	Đạt
DHSN04	Xem danh sách lịch sử đơn hàng	1	Đạt
DHSN05	Xem chi tiết lịch đơn hàng	1	Đạt

5.1.1.7 Đánh giá đơn hàng

ID	Mô tả	Số testcase	Kết quả
DGDHSN01	Đánh giá đơn hàng	2	Đạt

5.1.1.8 Đăng xuất

ID	Mô tả	Số testcase	Kết quả
DXSN01	Đăng xuất thành công	1	Đạt

5.1.2 Cửa hàng

5.1.2.1 Đăng nhập

ID	Mô tả	Số testcase	Kết quả
CHDNSN01	Đăng nhập thành công	1	Đạt
CHDNSN02	Đăng nhập thất bại	2	Đạt

5.1.2.2 Đăng ký

ID	Mô tả	Số testcase	Kết quả
CHDKSN01	Đăng ký thành công	1	Đạt
CHDKSN02	Đăng ký thất bại	2	Đạt

5.1.2.3 Cập nhật thông tin

ID	Mô tả	Số testcase	Kết quả
CHCNTTSN01	Cập nhật thông tin cá nhân	2	Đạt
CHCNTTSN02	Cập nhật mật khẩu	3	Đạt

5.1.2.4 Sản phẩm

ID	Mô tả	Số testcase	Kết quả
CHSPSN01	Xem danh sách sản phẩm	1	Đạt
CHSPSN02	Thêm Sản phẩm	2	Đạt
CHSPSN03	Sửa thông tin sản phẩm	2	Đạt
CHSPSN04	Xóa sản phẩm	1	Đạt

5.1.2.5 Đơn hàng

ID	Mô tả	Số testcase	Kết quả
CHDHSN01	Xem danh sách các đơn hàng đang xử lý	1	Đạt
CHDHSN02	Xem chi tiết đơn hàng đang xử lý	1	Đạt
CHDHSN03	Hoàn thành đơn hàng	1	Đạt
CHDHSN04	Hủy đơn hàng	1	Đạt
CHDHSN05	Xem danh sách lịch sử đơn hàng	1	Đạt
CHDHSN06	Xem chi tiết lịch đơn hàng	1	Đạt

5.1.2.6 Đăng xuất

ID	Mô tả	Số testcase	Kết quả
CHDXSN01	Đăng xuất thành công	1	Đạt

5.2 Chi tiết test case

Mục này trình bày chi tiết các test case đã được nêu ở trên.

5.2.1 Người dùng

5.2.1.1 DNSN01 – Đăng nhập thành công

Bước kiểm thử:

- Bước 1: Vào trang đăng nhập
- Bước 2: Nhập số điện thoại và mật khẩu
- Bước 3: Bấm vào nút đăng nhập và xem thông báo lỗi

Test case:

ID	Số điện thoại	Mật khẩu	Mong đợi	Kết quả
DNTCTC01	0387234156	123456	Đăng nhập thành công	Đạt

5.2.1.2 DNSN02 – Đăng nhập thất bại

Bước kiểm thử:

- Bước 1: Vào trang đăng nhập
- Bước 2: Nhập số điện thoại và mật khẩu
- Bước 3: Bấm vào nút đăng nhập và xem thông báo lỗi

Test case:

ID	Số điện thoại	Mật khẩu	Mong đợi	Kết quả
DNTBTC01	0387234156	12345	Thông báo “Mật khẩu không đúng định dạng”	Đạt
DNTBTC02	0387234156	123465	Thông báo “Tên đăng nhập hoặc mật khẩu không chính xác”	Đạt

5.2.1.3 DKS01 – Đăng ký thành công

Bước kiểm thử:

- Bước 1: Vào trang đăng ký
- Bước 2: Nhập số điện thoại và bấm tiếp tục
- Bước 3: Nhập OTP và bấm tiếp tục
- Bước 4: Nhập ảnh, email, họ và tên, mật khẩu, nhập lại mật khẩu và bấm vào nút đăng ký

Test case:

ID	Số điện thoại	OTP	Họ và tên	Email
DKTCTC01	0334759347	111111	Nguyễn văn A	test@gmail.com

Mật khẩu	Nhập lại mật khẩu	Mong đợi	Kết quả
123456	123456	Đăng ký thành công	Đạt

5.2.1.4 DKS02 – Đăng ký thất bại

Bước kiểm thử:

- Bước 1: Vào trang đăng ký
- Bước 2: Nhập số điện thoại và bấm tiếp tục
- Bước 3: Nhập OTP và bấm tiếp tục
- Bước 4: Nhập ảnh, email, họ và tên, mật khẩu, nhập lại mật khẩu và bấm vào nút đăng ký

Test case:

ID	Số điện thoại	OTP	Họ và tên	Email
DKTBTC01	0331272234	111111	Nguyễn văn A	test
DKTBTC02	0331272754	111111	Nguyễn văn A	test@gmail.com
DKTBTC03	0334759347			

Mật khẩu	Nhập lại mật khẩu	Mong đợi	Kết quả
123456	123654	Thông báo “Email không đúng định dạng”	Đạt
123456	123654	Thông báo “Mật khẩu không khớp”	Đạt
		Thông báo “Số điện thoại đã tồn tại”	Đạt

5.2.1.5 CNTTSN01 – Cập nhật thông tin cá nhân

Bước kiểm thử:

- Bước 1: Vào trang chỉnh sửa tài khoản
- Bước 2: Chỉnh sửa họ và tên, email
- Bước 3: Bấm vào nút “Cập nhật”

Test case:

ID	Họ và tên	Email	Mong đợi	Kết quả
CNTTCNTC01	Nguyễn văn A	test	Thông báo “Email không đúng định dạng”	Đạt
CNTTCNTC02	Nguyễn văn B	test@gmail.com	Cập nhật thông tin thành công	Đạt

5.2.1.6 CNTTSN02 – Cập nhật mật khẩu

Bước kiểm thử:

- Bước 1: Vào trang đổi mật khẩu
- Bước 2: Nhập mật khẩu cũ, mật khẩu mới, nhập lại mật khẩu mới
- Bước 3: Bấm vào nút cập nhật

Test case:

ID	Mật khẩu cũ	Mật khẩu mới	Nhập lại mật khẩu mới	Mong đợi	Kết quả
CNMKTC01	123455	123455	123455	Thông báo “Mật khẩu cũ không đúng”	Đạt
CNMKTC02	123456	123455	123456	Thông báo “Mật khẩu không khớp”	Đạt
CNMKTC03	123456	123455	123455	Cập nhật thành công	Đạt

5.2.1.7 SPSN01 – Xem danh sách sản phẩm

Bước kiểm thử:

- Bước 1: Vào trang sản phẩm
- Bước 2: Xem danh sách

Test case:

ID	Mong đợi	Kết quả
DSSPTC01	Hiện danh sách sản phẩm	Đạt

5.2.1.8 SPSN02 – Xem chi tiết sản phẩm

Bước kiểm thử:

- Bước 1: Vào trang sản phẩm
- Bước 2: Bấm vào sản phẩm

Test case:

ID	Mong đợi	Kết quả
CTSPTC01	Hiện chi tiết sản phẩm	Đạt

5.2.1.9 GDSN01 – Thêm vào giỏ hàng

Bước kiểm thử:

- Bước 1: Vào trang sản phẩm
- Bước 2: Chọn số lượng

- Bước 3: Bấm “Thêm vào giỏ hàng”

Test case:

ID	Sản phẩm	Số lượng	Mong đợi	Kết quả
TVGHTC01	Bánh mì	1	Thêm vào giỏ hàng thành công	Đạt
TVGHTC02	Bánh mì	100	Thông báo “Số lượng sản phẩm không đủ”	Đạt
TVGHTC03	Bánh bao	1	Thông báo “Bạn không thể thêm sản phẩm từ cửa hàng khác”	Đạt

5.2.1.10 GHSN02 – Xóa sản phẩm khỏi giỏ hàng

Bước kiểm thử:

- Bước 1: Vào trang giỏ hàng
- Bước 2: Bấm vào nút xóa sản phẩm

Test case:

ID	Mong đợi	Kết quả
XSPKGHTC01	Xóa sản phẩm khỏi giỏ hàng thành công	Đạt

5.2.1.11 GHSN03 – Cập nhật số lượng sản phẩm trong giỏ hàng

Bước kiểm thử:

- Bước 1: Vào trang giỏ hàng
- Bước 2: Thay đổi số lượng sản phẩm

Test case:

ID	Số lượng	Mong đợi	Kết quả
CNSLSPTC01	3	Cập nhật số lượng sản phẩm thành công	Đạt
CNSLSPTC02	101	Thông báo “Số lượng sản phẩm không đủ”	Đạt

5.2.1.12 GHSN04 – Đặt hàng

Bước kiểm thử:

- Bước 1: Vào trang giỏ hàng
- Bước 2: Bấm vào nút “Đặt hàng” và xác nhận đặt hàng

Test case:

ID	Mong đợi	Kết quả
DHTC01	Đặt hàng thành công	Đạt

5.2.1.13 DHSN01 – Xem danh sách các đơn hàng đang xử lý

Bước kiểm thử:

- Bước 1: Vào trang đơn hàng
- Bước 2: Bấm vào trang đơn hàng đang xử lý

Test case:

ID	Mong đợi	Kết quả
DHDSLTC01	Hiện danh sách đơn hàng đang xử lý	Đạt

5.2.1.14 DHSN02 - Xem chi tiết đơn hàng đang xử lý

Bước kiểm thử:

- Bước 1: Vào trang đơn hàng đang xử lý
- Bước 2: Bấm vào đơn hàng

Test case:

ID	Mong đợi	Kết quả
CTDHDSLTC01	Hiện chi tiết đơn hàng đang xử lý	Đạt

5.2.1.15 DHSN03 – Hủy đơn hàng

Bước kiểm thử:

- Bước 1: Vào trang đơn hàng đang xử lý
- Bước 2: Bấm vào đơn hàng
- Bước 3: Bấm vào nút “Hủy đơn hàng” và xác nhận hủy đơn hàng

Test case:

ID	Mong đợi	Kết quả
HDHTC01	Hủy đơn hàng thành công	Đạt

5.2.1.16 DHSN01 – Xem danh sách lịch sử đơn hàng

Bước kiểm thử:

- Bước 1: Bấm vào trang đơn hàng
- Bước 2: Bấm vào lịch sử đơn hàng

Test case:

ID	Mong đợi	Kết quả
LSDHTC01	Hiện danh sách lịch sử đơn hàng	Đạt

5.2.1.17 DHSN02 - Xem chi tiết lịch sử đơn hàng

Bước kiểm thử:

- Bước 1: Vào trang lịch sử đơn hàng
- Bước 2: Bấm vào đơn hàng

Test case:

ID	Mong đợi	Kết quả
CTLSDHTC01	Hiện chi tiết lịch sử đơn hàng	Đạt

5.2.1.18 DGDHSN01 – Đánh giá đơn hàng

Bước kiểm thử:

- Bước 1: Vào trang lịch sử đơn hàng
- Bước 2: Bấm vào đơn hàng
- Bước 3: Bấm vào nút Đánh giá
- Bước 4: Chọn ảnh, số sao và nội dung đánh giá
- Bước 5: Bấm vào nút Gửi

Test case:

ID	Số sao	Nội dung	Mong đợi	Kết quả
CHCNSPTC01	5	Bánh ngon	Đánh giá thành công	Đạt
CHCNSPTC02	0	Bánh ngon	Thông báo “Bạn chưa chọn số sao”	Đạt

5.2.1.19 CHDXSN01 – Đăng xuất thành công

Bước kiểm thử:

- Bước 1: Vào trang tài khoản
- Bước 2: Bấm vào nút “Đăng xuất”

Test case:

ID	Mong đợi	Kết quả
DXTC01	Đăng xuất thành công	Đạt

5.2.2 Cửa hàng

5.2.2.1 CHDNSN01 – Đăng nhập thành công

Bước kiểm thử:

- Bước 1: Vào trang đăng nhập
- Bước 2: Nhập số điện thoại và mật khẩu

- Bước 3: Bấm vào nút Đăng nhập

Test case:

ID	Số điện thoại	Mật khẩu	Mong đợi	Kết quả
CHDNTCTC01	0387234156	123456	Đăng nhập thành công	Đạt

5.2.2.2 CHDNSN02 – Đăng nhập thất bại

Bước kiểm thử:

- Bước 1: Vào trang đăng nhập
- Bước 2: Nhập số điện thoại và mật khẩu
- Bước 3: Bấm vào nút Đăng nhập và xem thông báo lỗi

Test case:

ID	Số điện thoại	Mật khẩu	Mong đợi	Kết quả
CHDNTBTC01	0387234156	12345	Thông báo “Mật khẩu không đúng định dạng”	Đạt
CHDNTBTC02	0387234156	123465	Thông báo “Tên đăng nhập hoặc mật khẩu không chính xác”	Đạt

5.2.2.3 CHDKSN01 – Đăng ký thành công

Bước kiểm thử:

- Bước 1: Vào trang đăng ký
- Bước 2: Nhập số điện thoại và bấm tiếp tục
- Bước 3: Nhập OTP và bấm Tiếp tục
- Bước 4: Nhập ảnh, email, họ và tên, CCCD, tên cửa hàng, vị trí, mật khẩu, nhập lại mật khẩu và bấm vào nút Đăng ký

Test case:

ID	Số điện thoại	OTP	Họ và tên	Email
CHDKTCTC01	0334759347	111111	Nguyễn văn A	test1@gmail.com

CCCD	Tên cửa hàng	Vị trí	Mật khẩu	Nhập lại mật khẩu	Mong đợi	Kết quả
024465363560	tesst	15,33	123456	123456	Đăng ký thành công	Đạt

5.2.2.4 CHDKSN02 – Đăng ký thất bại

Bước kiểm thử:

- Bước 1: Vào trang đăng ký
- Bước 2: Nhập số điện thoại và bấm tiếp tục
- Bước 3: Nhập OTP và bấm Tiếp tục
- Bước 4: Nhập ảnh, email, họ và tên, CCCD, tên cửa hàng, vị trí, mật khẩu, nhập lại mật khẩu và bấm vào nút Đăng ký
- Bước 5: Xem thông báo

Test case:

ID	Số điện thoại	OTP	Họ và tên	Email
CHDKTCTC01	0334759356	111111	Nguyễn văn A	test1gmail.com
CHDKTCTC02	0334759356	111111	Nguyễn văn A	test1@gmail.com
CHDKTCTC03	0334759347			

CCCD	Tên cửa hàng	Vị trí	Mật khẩu	Nhập lại mật khẩu	Mong đợi	Kết quả
12433546536356	tesst	15,33	123456	123456	Thông báo “Email không đúng định dạng”	Đạt
12433546536356	tesst	15,33	123456	123455	Thông báo “Mật khẩu không khớp”	Đạt
					Thông báo “Số điện thoại đã tồn tại”	Đạt

5.2.2.5 CHCNTTSN01 – Cập nhật thông tin cá nhân

Bước kiểm thử:

- Bước 1: Vào trang chỉnh sửa tài khoản
- Bước 2: Chỉnh sửa họ và tên, email, tên cửa hàng
- Bước 3: Bấm vào nút “Cập nhật”

Test case:

ID	Họ và tên	Email	Tên cửa hàng	Mong đợi	Kết quả
CHCNTTCNTC01	Nguyễn văn A	test	Test	Thông báo “Email không đúng định dạng”	Đạt
CHCNTTCNTC02	Nguyễn văn B	test@gmail.com	test	Cập nhật thông tin thành công	Đạt

5.2.2.6 CHCNTTSN02 – Cập nhật mật khẩu

Bước kiểm thử:

- Bước 1: Vào trang đổi mật khẩu
- Bước 2: Nhập mật khẩu cũ, mật khẩu mới, nhập lại mật khẩu mới
- Bước 3: Bấm vào nút “Cập nhật”

Test case:

ID	Mật khẩu cũ	Mật khẩu mới	Nhập lại mật khẩu mới	Mong đợi	Kết quả
CHCNMKTC01	123455	123455	123455	Thông báo “Mật khẩu cũ không đúng”	Đạt
CHCNMKTC02	123456	123455	123456	Thông báo “Mật khẩu không khớp”	Đạt
CHCNMKTC03	123456	123455	123455	Cập nhật thành công	Đạt

5.2.2.7 CHSPSN01 – Xem danh sách sản phẩm

Bước kiểm thử:

- Bước 1: Vào trang sản phẩm
- Bước 2: Xem danh sách

Test case:

ID	Mong đợi	Kết quả
CHDSSPTC01	Hiện danh sách sản phẩm	Đạt

5.2.2.8 CHSPSN02 – Thêm sản phẩm

Bước kiểm thử:

- Bước 1: Vào trang sản phẩm
- Bước 2: Bấm vào Thêm sản phẩm
- Bước 3: Thêm ảnh, tên, giá trước khuyến mãi, giá sau khuyến mãi, số lượng cần bán, thời gian bán, mô tả sản phẩm
- Bước 4: Bấm vào nút Thêm sản phẩm

Test case:

ID	Tên sản phẩm	Giá trước khuyến mãi	Giá sau khuyến mãi
CHTSPTC01	Bánh mì	50000	30000
CHTSPTC02	Bánh mì	30000	40000

Số lượng	Thời gian	Mô tả	Mong đợi	Kết quả
100	17:00 - 19:00	Bánh ngon	Thêm thành công	Đạt
100	17:00 - 19:00	Bánh ngon	Thông báo “Giá trước khuyến mãi phải lớn hơn giá sau khuyến mãi”	Đạt

5.2.2.9 CHSPSN03 – Sửa thông tin sản phẩm

Bước kiểm thử:

- Bước 1: Vào trang sản phẩm
- Bước 2: Bấm vào sản phẩm
- Bước 3: Chỉnh sửa ảnh, tên, giá trước khuyến mãi, giá sau khuyến mãi, số lượng cần bán, thời gian bán, mô tả sản phẩm
- Bước 4: Bấm vào nút Cập nhật

Test case:

ID	Tên sản phẩm	Giá trước khuyến mãi	Giá sau khuyến mãi
CHCNSPTC01	Bánh mì	60000	40000
CHCNSPTC02	Bánh mì	30000	40000

Số lượng	Thời gian	Mô tả	Mong đợi	Kết quả
20	17:00 - 19:00	Bánh ngon	Thêm thành công	Đạt
100	17:00 - 19:00	Bánh ngon	Thông báo “Giá trước khuyến mãi phải lớn hơn giá sau khuyến mãi”	Đạt

5.2.2.10 CHSPSN04 – Xóa sản phẩm

Bước kiểm thử:

- Bước 1: Vào trang sản phẩm
- Bước 2: Bấm vào sản phẩm
- Bước 3: Bấm vào nút Xóa và xác nhận

Test case:

ID	Mong đợi	Kết quả
CHXSPTC01	Xóa sản phẩm thành công	Đạt

5.2.2.11 CHDHSN01 – Xem danh sách các đơn hàng đang xử lý

Bước kiểm thử:

- Bước 1: Bấm vào trang đơn hàng

Test case:

ID	Mong đợi	Kết quả
CHDHDSLTC01	Hiện danh sách đơn hàng đang xử lý	Đạt

5.2.2.12 CHDHSN02 - Xem chi tiết đơn hàng đang xử lý

Bước kiểm thử:

- Bước 1: Vào trang đơn hàng đang xử lý
- Bước 2: Bấm vào đơn hàng

Test case:

ID	Mong đợi	Kết quả
CHCTDHDSLTC01	Hiện chi tiết đơn hàng đang xử lý	Đạt

5.2.2.13 CHDHSN03 – Hoàn thành đơn hàng

Bước kiểm thử:

- Bước 1: Vào trang đơn hàng đang xử lý

- Bước 2: Bấm vào đơn hàng
- Bước 3: Bấm vào nút “Hoàn thành” và xác nhận hoàn thành đơn hàng

Test case:

ID	Mong đợi	Kết quả
CHHTDHTC01	Hủy đơn hàng thành công	Đạt

5.2.2.14 CHDHSN04 – Hủy đơn hàng

Bước kiểm thử:

- Bước 1: Vào trang đơn hàng đang xử lý
- Bước 2: Bấm vào đơn hàng
- Bước 3: Bấm vào nút “Hủy đơn hàng” và xác nhận hủy đơn hàng

Test case:

ID	Mong đợi	Kết quả
HDHTC01	Hủy đơn hàng thành công	Đạt

5.2.2.15 CHDHSN05 – Xem danh sách lịch sử đơn hàng

Bước kiểm thử:

- Bước 1: Từ trang chủ bấm vào Lịch sử đơn hàng
- Bước 2: Xem danh sách

Test case:

ID	Mong đợi	Kết quả
CHLSDHTC01	Hiện danh sách lịch sử đơn hàng	Đạt

5.2.2.16 DHSN06 - Xem chi tiết lịch sử đơn hàng

Bước kiểm thử:

- Bước 1: Vào trang lịch sử đơn hàng
- Bước 2: Bấm vào đơn hàng

Test case:

ID	Mong đợi	Kết quả
CHCTLSDHTC01	Hiện chi tiết lịch sử đơn hàng	Đạt

5.2.2.17 CHCHDXSN01 – Đăng xuất thành công

Bước kiểm thử:

- Bước 1: Vào trang tài khoản
- Bước 2: Bấm vào nút “Đăng xuất”

Test case:

ID	Mong đợi	Kết quả
DXTC01	Đăng xuất thành công	Đạt

Chương 6

TỔNG KẾT

6.1 Tổng kết kết quả đạt được và những hạn chế của đề tài

6.1.1 Kết quả đạt được

Đối với đề tài này, chúng tôi đã đạt được mục đích ban đầu đặt ra là xây dựng một ứng dụng mua bán thực phẩm dư thừa giúp các chủ cửa hàng giải quyết các thực phẩm có hạn sử dụng ngắn thay vì bỏ đi và giúp người tiêu dùng tiết kiệm một khoản tiền chi cho lương thực thực phẩm. Bên cạnh đó, nhóm còn đạt được những điều sau:

- Phân tích và tìm hiểu nghiệp vụ của một hệ thống mua bán hàng online.
- Tìm hiểu các hệ thống khác trong lĩnh vực để đưa ra phân tích, so sánh nhằm hoàn thiện các tính năng của hệ thống.
- Tìm hiểu, phân tích đánh giá và áp dụng các công nghệ nổi bật hiện nay về ứng dụng mobile như React Native, Expo, Redux, đối với server như Apollo, Redis và đối với database là MongoDB.

6.1.2 Các điểm hạn chế

Vì hạn chế về thời gian và về nhân lực nên hệ thống của chúng tôi vẫn tồn đọng một số vấn đề:

- Chưa hiện thực được API thanh toán online.
- Nghiệp vụ liên quan tới mã giảm giá chưa được phân tích kỹ.
- Nền tảng dựa trên phản hồi của người dùng nhưng hệ thống đánh giá sản phẩm chưa được hoàn thiện, hiện chỉ ở mức sơ sài.

6.2 Hướng phát triển

- Tiếp tục hoàn thiện các nghiệp vụ liên quan.

- Hoàn thiện các chức năng còn thiếu.
- Hoàn thiện quy trình CI/CD.

Tài liệu tham khảo

[1] Facebook Inc., *Reactjs*, [Online; URL: <https://github.com/gorilla/mux>, truy cập vào ngày 10-05-2023]

[2] Dan Abramov and the Redux documentation authors, *Redux*, [Online; URL: <https://redux.js.org/>, truy cập vào ngày 10-05-2023]

[3] Phương Nam, *Chống lãng phí thực phẩm thời công nghệ số*, [Online; URL: <https://foodbankvietnam.com/chong-lang-phi-thuc-pham-thoi-cong-nghe-so/>, truy cập vào ngày 04-03-2023]

[4] Khalil Stemmler, *What is GraphQL? GraphQL introduction*, [Online; URL: <https://www.apollographql.com/blog/graphql/basics/what-is-graphql-introduction/>, truy cập vào ngày 05-03-2023]

[5] Khalil Stemmler, *What is GraphQL? GraphQL introduction*, [Online; URL: <https://www.apollographql.com/blog/graphql/basics/what-is-graphql-introduction/>, truy cập vào ngày 05-05-2023]

[6] Microsoft, *N-tier architecture style*, [Online; URL: <https://learn.microsoft.com/en-us/azure/architecture/guide/architecture-styles/n-tier>, truy cập vào ngày 15-04-2023]

[7] glints, *React Native Là Gì? Vì Sao Bạn Nên Học React Native?*, [Online; URL: <https://glints.com/vn/blog/react-native-la-gi/#.ZGr5R3ZBxPY>, truy cập vào ngày 05-03-2023]

[8] Ngô Tiến Phát, *GraphQL là gì? Bắt đầu với GraphQL*, [Online; URL: <https://viblo.asia/p/graphql-la-gi-bat-dau-voi-graphql-gDVK2nX0KLj>, truy cập vào ngày 26-04-2023]

[9] Duong Trung Hieu, *Cài đặt thuật toán gợi ý sản phẩm NBCF*, [Online; URL: <https://viblo.asia/p/cai-dat-thuat-toan-goi-y-san-pham-nbcf-ORNZqx8LK0n>, truy cập vào ngày 15-03-2023]