

BỘ THÔNG TIN VÀ TRUYỀN THÔNG
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



ĐỒ ÁN
TỐT NGHIỆP ĐẠI HỌC

Đề tài:

**“XÂY DỰNG HỆ THỐNG THƯƠNG MẠI
VĂN PHÒNG PHẨM”**

Người hướng dẫn : TS. NGUYỄN THỊ TUYẾT HẢI

Sinh viên thực hiện : NGUYỄN NGỌC PHƯƠNG TRINH

Mã số sinh viên : N18DCCN231

Lớp : D18CQCP02-N

Hệ : Đại học chính quy

TP. HỒ CHÍ MINH, NĂM 2022

BỘ THÔNG TIN VÀ TRUYỀN THÔNG
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



ĐỒ ÁN
TỐT NGHIỆP ĐẠI HỌC

Đề tài:

**“XÂY DỰNG HỆ THỐNG THƯƠNG MẠI
VĂN PHÒNG PHẨM”**

Người hướng dẫn : TS. NGUYỄN THỊ TUYẾT HẢI
Sinh viên thực hiện : NGUYỄN NGỌC PHƯƠNG TRINH
Mã số sinh viên : N18DCCN231
Lớp : D18CQCP02-N
Khoa : 2018 -2023
Hệ : Đại học chính quy

TP. HỒ CHÍ MINH, NĂM 2022

HỌC VIỆN
CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG
CƠ SỞ TẠI TP. HỒ CHÍ MINH
Khoa: Công nghệ thông tin 2

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập – Tự do – Hạnh phúc

TP. Hồ Chí Minh, ngày 23 tháng 09 năm 2022

ĐỀ TÀI ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC

Căn cứ Quyết định số: 500/QĐ-HVCS ngày 23 tháng 09 năm 2022 của Phó Giám đốc Học viện – Phụ trách Cơ sở tại TP. Hồ Chí Minh về việc “phê duyệt danh sách giáo viên hướng dẫn và giao đề tài đồ án tốt nghiệp cho sinh viên Đại học chính quy Khóa 2018-2022 Ngành Công nghệ thông tin, An toàn thông tin và Công nghệ đa phương tiện”;

Khoa Công nghệ thông tin 2 giao nhiệm vụ thực hiện Đề án tốt nghiệp cho sinh viên:

1. Họ và tên sv : Nguyễn Ngọc Phương Trinh Mã SV : N18DCCN231
Lớp : D18CQCP02-N Khóa : 2018-2023
Ngành đào tạo : Công nghệ thông tin Hệ đào tạo : Đại học Chính quy
2. Tên đề tài tốt nghiệp: Xây dựng hệ thống thương mại văn phòng phẩm
3. Nội dung chính của Đề án:

Lý thuyết:

- Tìm hiểu nghiệp vụ bán hàng
- Tìm hiểu Spring Framework (Spring Boot, JPA, ...) và Restful API để triển khai Web service
- Tìm hiểu Java, Retrofit,... để xây dựng giao diện Android dành cho khách hàng và truy xuất tới API
- Tìm hiểu Java Swing, Retrofit,.. để xây dựng giao diện Win Application dành cho quản trị viên và truy xuất tới API

Thực hành:

- Phân tích và thiết kế cơ sở dữ liệu
- Xây dựng ứng dụng di động dành cho cả admin và khách hàng:

1. Admin gồm các chức năng:
 - + Quản lý khách hàng
 - + Quản lý nhập kho
 - + Quản lý hóa đơn
 - + Quản lý danh mục sản phẩm
 - + Thống kê doanh thu

2. Khách hàng gồm các chức năng:
 - + Đăng ký, đăng nhập
 - + Chỉnh sửa thông tin cá nhân
 - + Quản lý giỏ hàng
 - + Mua hàng, theo dõi đơn hàng
 - + Nếu đăng nhập với role admin thì có thể thực hiện các chức năng admin trên di động

4. Cơ sở dữ liệu ban đầu:

5. Giáo viên hướng dẫn: Nguyễn Thị Tuyết Hải

6. Ngày giao đề tài: 23/09/2022

7. Ngày nộp quyền: 09/12/2022

TRƯỞNG KHOA CNTT2

Nơi nhận:

- Sinh viên có tên tại khoản I;
- Lưu: VP Khoa.



TS. Nguyễn Hồng Sơn

LỜI CẢM ƠN

Ba tháng làm đồ án tốt nghiệp ngắn ngủi là cơ hội cho em tổng hợp và hệ thống hóa lại những kiến thức đã học, đồng thời kết hợp với thực tế để nâng cao kiến thức chuyên môn. Tuy chỉ có ba tháng, nhưng qua quá trình làm đồ án, em đã được mở rộng tầm nhìn và tiếp thu rất nhiều kiến thức thực tế. Từ đó em nhận thấy, việc cọ sát thực tế là vô cùng quan trọng – nó giúp sinh viên xây dựng nền tảng lý thuyết được học ở trường vững chắc hơn. Trong quá trình làm đồ án, từ chỗ còn bỡ ngỡ cho đến thiếu kinh nghiệm, em đã gặp phải rất nhiều khó khăn nhưng với sự giúp đỡ tận tình của quý thầy cô Khoa Công nghệ thông tin 2 và sự nhiệt tình của cô Nguyễn Thị Tuyết Hải đã giúp em có được những kinh nghiệm quý báu để hoàn thành tốt kì đồ án tốt nghiệp này cũng như viết lên bài báo cáo cuối kỳ. Em xin chân thành cảm ơn.

Lời cảm ơn đầu tiên em xin gửi đến Ban giám hiệu, quý thầy cô khoa Công nghệ thông tin 2 đã tận tâm giảng dạy và truyền đạt những kiến thức, kinh nghiệm quý báu cho em. Đặc biệt, em xin cảm ơn cô Nguyễn Thị Tuyết Hải, người đã tận tình hướng dẫn em hoàn thành bài báo cáo này.

Em cũng xin cảm ơn gia đình và bạn bè, đã ở bên cạnh động viên, góp ý, truyền thêm động lực để em có thể chỉnh sửa và hoàn thiện đồ án tốt nghiệp.

Vì thời gian và kiến thức còn hạn hẹp nên bài báo cáo không thể tránh khỏi những thiếu sót, rất mong sự góp ý của quý thầy cô và các bạn, để em rút kinh nghiệm và hoàn thành tốt hơn. Em xin chân thành cảm ơn!

Thành phố Hồ Chí Minh, tháng 12 năm 2022
Sinh viên thực hiện

Nguyễn Ngọc Phương Trinh

MỤC LỤC

LỜI CẢM ƠN	I
MỤC LỤC	II
DANH MỤC CÁC KÝ HIỆU VÀ CHỮ VIẾT TẮT.....	V
DANH MỤC CÁC HÌNH	VI
DANH MỤC CÁC BẢNG.....	VIII
CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI.....	1
1.1. Sự cần thiết của đề tài.....	1
1.2. Mục tiêu của đề tài	1
1.3. Kết quả phải đạt.....	1
1.4. Cơ sở lý luận và nền tảng xây dựng ứng dụng	2
1.4.1. Ngôn ngữ lập trình Java	2
1.4.2. Giới thiệu về thư viện Retrofit	4
1.4.3. XML trong Android	5
1.4.4. RESTful API	5
1.4.5. Spring Boot.....	6
1.4.6. Giới thiệu về hệ quản trị cơ sở dữ liệu MySQL và công vụ MySQL Workbench.....	7
1.4.7. Android Studio	8
CHƯƠNG 2. KHẢO SÁT HIỆN TRẠNG	9
2.1. Khảo sát hiện trạng	9
2.2. Giải pháp đề xuất.....	10
CHƯƠNG 3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG	13
3.1. Sơ đồ phân rã chức năng BFD.....	13
3.2. Sơ đồ ngũ cảnh	14
3.3. Lược đồ Usecase.....	15
3.3.1. Usecase Tổng quát.....	15
3.3.2. Usecase Đăng nhập.....	16
3.3.3. Usecase Sửa thông tin cá nhân	17
3.3.4. Usecase Quản lý sản phẩm	19
3.3.5. Usecase Quản lý nhãn hiệu.....	20

3.3.6. Usecase Quản lý loại sản phẩm	21
3.3.7. Usecase Quản lý đơn hàng	22
3.3.8. Usecase Quản lý người dùng.....	23
3.3.9. Usecase Quản lý giỏ hàng	24
3.3.10. Usecase Quản lý đánh giá.....	25
3.3.11. Usecase Theo dõi đơn hàng.....	26
3.4. Sơ đồ tuần tự.....	27
3.4.1. Sơ đồ tuần tự đăng nhập	27
3.4.2. Sơ đồ tuần tự sửa thông tin cá nhân	27
3.4.3. Sơ đồ tuần tự sửa, thêm sản phẩm.....	28
3.4.4. Sơ đồ tuần tự xóa sản phẩm.....	28
3.4.5. Sơ đồ tuần tự theo dõi đơn hàng.....	29
3.4.6. Sơ đồ tuần tự sửa, thêm sản phẩm vào giỏ hàng	29
3.4.7. Sơ đồ tuần tự xóa sản phẩm khỏi giỏ hàng	30
3.4.8. Sơ đồ tuần tự thay đổi mật khẩu.....	30
3.5. Thiết kế dữ liệu.....	31
3.5.1. Mô hình thực thể quan hệ ER.....	31
3.5.2. Lược đồ quan hệ	31
3.5.3. Hiện thực hóa.....	32
3.5.4. Mô tả các bảng dữ liệu	33
CHƯƠNG 4. THIẾT KẾ PHẦN MỀM.....	34
4.1. Front-end (Forms)	34
4.1.1. Windows Application.....	34
4.1.2. Mobile Application.....	48
4.2. Back-End (APIs)	94
4.2.1. Authentication API.....	94
4.2.2. User API	95
4.2.3. Admin API.....	110
CHƯƠNG 5. KẾT LUẬN VÀ KIẾN NGHỊ	115
5.1. Kết quả đạt được.....	115
5.2. Hạn chế của đề tài.....	115
5.3. Kiến nghị	115

DANH MỤC CÁC KÝ HIỆU VÀ CHỮ VIẾT TẮT

HTML: HyperText Markup Language

CSS: Cascading Style Sheets

SQL: Structured Query Language

PHP: Hypertext Preprocessor

ERD: Entity Relation Diagram

DFD: Data Flow Diagram

BFD: Business Function Diagram

ORM: Object Relational Mapping

DANH MỤC CÁC HÌNH

Hình 1.1. Các tính năng của ngôn ngữ Java	3
Hình 1.2. Retrofit trong android	4
Hình 1.3. RESTful API	6
Hình 1.4. Spring Boot.....	6
Hình 1.5. Hệ quản trị cơ sở dữ liệu MySQL	7
Hình 1.6. MySQL Workbench	8
Hình 1.7. Android Studio	8
Hình 2.1. Sơ đồ cộng tác cho khách hàng	10
Hình 2.2. Sơ đồ cộng tác cho nhân viên cửa hàng	11
Hình 2.3. Sơ đồ cộng tác cho shipper.....	11
Hình 3.1. Sơ đồ phân rã chức năng BFD.....	13
Hình 3.2. Sơ đồ ngữ cảnh	14
Hình 3.3. Usecase Tổng quát.....	15
Hình 3.4. Usecase Đăng nhập.....	16
Hình 3.5. Usecase Sửa thông tin cá nhân	17
Hình 3.6. Usecase Quản lý sản phẩm	19
Hình 3.7. Usecase Quản lý nhãn hiệu.....	20
Hình 3.8. Usecase Quản lý loại sản phẩm	21
Hình 3.9. Usecase Quản lý đơn hàng	22
Hình 3.10. Usecase Quản lý người dùng.....	23
Hình 3.11. Usecase Quản lý giỏ hàng	24
Hình 3.12. Usecase Quản lý đánh giá.....	25
Hình 3.13. Usecase Theo dõi đơn hàng.....	26
Hình 4.1. Login Form	34
Hình 4.2. Order Form	35
Hình 4.3. Product Form	36
Hình 4.4. Price History Form	37
Hình 4.5. Feedback Form	38
Hình 4.6. User Form	39
Hình 4.7. Category Form.....	40
Hình 4.8. Brand Form.....	41
Hình 4.9. Import List	42
Hình 4.10. Create Import Form	43
Hình 4.11. Import Product Form	44
Hình 4.12. Profile Form.....	45
Hình 4.13. Address Form	46
Hình 4.14. Change Password Form	47
Hình 4.15. Giao diện trang chủ	48
Hình 4.16. Giao diện danh mục sản phẩm.....	50
Hình 4.17. Giao diện sản phẩm	52

Hình 4.18. Giao diện chi tiết sản phẩm	54
Hình 4.19. Giao diện giỏ hàng (1)	56
Hình 4.20. Giao diện giỏ hàng (2)	57
Hình 4.21. Giao diện giỏ hàng (3)	59
Hình 4.22. Giao diện giỏ hàng (4)	61
Hình 4.23. Giao diện thanh toán (1)	63
Hình 4.24. Giao diện thanh toán (2)	65
Hình 4.25. Giao diện tài khoản.....	66
Hình 4.26. Giao diện chỉnh sửa thông tin cá nhân	68
Hình 4.27. Giao diện đổi mật khẩu.....	70
Hình 4.28. Giao diện hóa đơn (1)	71
Hình 4.29. Giao diện hóa đơn (2)	73
Hình 4.30. Giao diện hóa đơn (3)	75
Hình 4.31. Giao diện hóa đơn (4)	77
Hình 4.32. Giao diện hóa đơn (5)	79
Hình 4.33. Giao diện chi tiết hóa đơn (1)	81
Hình 4.34. Giao diện chi tiết hóa đơn (2)	82
Hình 4.35. Giao diện chi tiết hóa đơn (3)	83
Hình 4.36. Giao diện chi tiết hóa đơn (4)	84
Hình 4.37. Giao diện chi tiết hóa đơn (5)	85
Hình 4.38. Giao diện đánh giá đơn hàng.....	86
Hình 4.39. Giao diện số địa chỉ	88
Hình 4.40. Giao diện chỉnh sửa địa chỉ	90
Hình 4.41. Giao diện đăng nhập	91
Hình 4.42. Giao diện đăng ký.....	93

DANH MỤC CÁC BẢNG

Bảng 3.1. Usecase Đăng nhập	17
Bảng 3.2. Usecase sửa thông tin cá nhân	18
Bảng 3.3. Usecase Thay đổi mật khẩu	19
Bảng 3.4. Usecase Quản lý sản phẩm.....	20
Bảng 3.5. Usecase Quản lý nhãn hiệu	21
Bảng 3.6. Usecase Quản lý loại sản phẩm.....	22
Bảng 3.7. Usecase Quản lý đơn hàng.....	23
Bảng 3.8. Usecase Quản lý người dùng	24
Bảng 3.9. Usecase Quản lý giỏ hàng	25
Bảng 3.10. Usecase Quản lý đánh giá	25
Bảng 3.11. Usecase Theo dõi đơn hàng	26

LỜI MỞ ĐẦU

Trong thời buổi hiện nay, khi mà công nghệ phát triển, thương mại điện tử ngày càng phát triển trên thế giới cũng như Việt Nam đang dần khẳng định được vị thế của mình trong nền kinh tế thị trường. Cùng với đó là sự ra đời của các ngôn ngữ lập trình cho phép thiết kế và xây dựng các ứng dụng thương mại điện tử dưới nhiều hình thức khác nhau. Và một trong những ứng dụng của thương mại điện tử phổ biến ở nước ta là dịch vụ bán hàng qua mạng internet. Dịch vụ này cho phép người dùng tìm kiếm chọn lựa sản phẩm để mua, thực hiện giao dịch mà không cần phải trực tiếp đến cửa hàng, mà chỉ cần sử dụng một thiết bị máy tính có kết nối internet. Tuy nhiên, hiện nay dịch vụ này vẫn còn khá mới với nhiều người tiêu dùng nước ta, đặc biệt là những người không có kiến thức về tin học nói chung và thương mại điện tử nói riêng. Nên hiện giờ các đối tượng chính của thương mại điện tử vẫn chủ yếu là tầng lớp tri thức và học sinh, sinh viên. Là những người sinh viên, cũng có chút kiến thức về tin học cũng như tiếp xúc với thương mại điện tử, tuy nhiên đôi lúc cũng gặp một số khó khăn trong việc mua bán trên các hệ thống bán hàng qua mạng hiện tại. Như hình thức thanh toán, nhận hàng, chất lượng sản phẩm thực tế không được như quảng cáo....

Hiểu được các vấn đề đó cũng như mong muốn đưa thương mại điện tử đến với nhiều người hơn, em đã quyết định thực hiện đề tài “Xây dựng ứng dụng Android bán đồ dùng văn phòng phẩm”. Với mục đích xây dựng một hệ thống mua hàng qua điện thoại uy tín, đơn giản, thân thiện, cũng như đem lại cho khách hàng những lựa chọn tối ưu để ngay cả những người không có kiến thức về tin học cũng có thể tham gia mua hàng qua mạng chỉ với một số thao tác đơn giản. Các sản phẩm bán trên shop là đồ văn phòng phẩm hướng đến các học sinh sinh viên và người làm văn phòng.

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

1.1. Sơ lược về đề tài

Văn phòng phẩm là các sản phẩm, vật phẩm và vật dụng dành cho văn phòng làm việc tại các công ty, trường học, bệnh viện, cơ quan, ban ngành đều cần dùng đến văn phòng phẩm. VD như bìa hồ sơ, giấy photo, các dụng cụ văn phòng như dao rọc giấy, kéo, băng keo, giấy note, bút viết các loại, kim bấm, bấm lỗ.

Trên thực tế hiện nay văn phòng phẩm có nghĩa rộng hơn nó bao gồm dụng cụ học và một số dụng cụ liên quan đến ngành mộc.

Theo phong cách truyền thống, chúng ta thường thấy các nhà sách thường tập trung trưng bày sản phẩm trên kệ trong đó hơn 70% là sách, 30% còn lại dành cho các sản phẩm văn phòng phẩm và đồ dùng học tập...

Xu hướng công nghệ phát triển khá mạnh trong những năm gần đây, hỗ trợ rất lớn cho tất cả các ngành hàng, không chỉ riêng về ngành hàng văn phòng phẩm, đã tạo ra xu thế mới thay đổi không ít trong cách nhìn nhận của người tiêu dùng. Sự xuất hiện của những chiếc điện thoại thông minh, máy tính bảng, laptop, máy tính bảng đọc sách...đã tạo nên một làn sóng mới, dẫn đến việc người dùng nhất là đọc giả sẽ tìm đọc những cuốn sách hay, những bài báo mới, những kiến thức về cuộc sống và xã hội được người tiêu dùng cập nhật thường xuyên, liên tục, không nhảm chán và điều quan trọng nữa là rất tiện lợi về thời gian và chi phí.

Chính vì vậy, ngành hàng kinh doanh văn phòng phẩm truyền thống ngày càng phát triển chậm dần, khó phát triển vì không đủ điều kiện đáp ứng được các yêu tố nhanh, gọn, tiện lợi, không mất quá nhiều thời gian đi lại, lựa chọn, thực tế có thể thấy trong những năm qua.

Để thay đổi cho phù hợp với xu thế hiện nay, bắt buộc ngành hàng văn phòng phẩm truyền thống phải có những nghiên cứu, có những bước đi mới phù hợp và thu hút, níu kéo lại khách hàng của mình bằng cách đổi mới tu duy nhìn nhận, đổi mới chương trình, đổi mới cách trình bày, đổi mới sự tư vấn và cung cách phục vụ khách hàng...và thậm chí là mở thêm kênh online để phục vụ khách hàng chuyên dùng mạng internet.

1.2. Mục tiêu của đề tài

- Khách hàng: sản phẩm được phân chia theo cây danh mục rất chi tiết giúp khách hàng dễ dàng tìm kiếm các mặt hàng trong shop, quản lý tài khoản cá nhân cùng các sản phẩm đã quan tâm, yêu thích. Ngoài ra, người dùng cũng có thể theo dõi đơn hàng từ lúc đặt mua đến lúc nhận hàng.
- Admin: thêm mới, cập nhật các sản phẩm của shop, phân loại sản phẩm để dễ dàng quản lý, xét duyệt đơn hàng và theo dõi các phản hồi từ khách hàng

1.3. Kết quả phải đạt

Xây dựng hệ thống bán hàng gồm:

- 1) Win App dành cho Admin gồm các chức năng:

- + Quản lý khách hàng
 - + Quản lý nhập kho
 - + Quản lý hóa đơn
 - + Quản lý danh mục sản phẩm
 - + Quản lý khuyến mãi
 - + Phân tích và lập báo cáo thống kê
- 2) Mobile App dành cho khách hàng:
- + Đăng ký, đăng nhập
 - + Quên mật khẩu
 - + Chính sửa thông tin cá nhân
 - + Quản lý giỏ hàng
 - + Mua hàng, theo dõi đơn hàng, đánh giá,...

1.4. Cơ sở lý luận và nền tảng xây dựng ứng dụng

1.4.1. Ngôn ngữ lập trình Java

Java là một trong những ngôn ngữ lập trình hướng đối tượng. Nó được sử dụng trong phát triển phần mềm, trang web, game hay ứng dụng trên các thiết bị di động.

Java được khởi đầu bởi James Gosling và bạn đồng nghiệp ở Sun MicroSystem năm 1991. Ban đầu Java được tạo ra nhằm mục đích viết phần mềm cho các sản phẩm gia dụng, và có tên là Oak.

Java được phát hành năm 1994, đến năm 2010 được Oracle mua lại từ Sun MicroSystem. Java được tạo ra với tiêu chí “Viết (code) một lần, thực thi khắp nơi” (Write Once, Run Anywhere – WORA). Chương trình phần mềm viết bằng Java có thể chạy trên mọi nền tảng (platform) khác nhau thông qua một môi trường thực thi với điều kiện có môi trường thực thi thích hợp hỗ trợ nền tảng đó.

1.4.1.1. Các tính năng của ngôn ngữ lập trình Java

Hướng đối tượng: Trong Java, tất cả đều là một Object. Java có thể mở rộng và bảo trì dễ dàng bởi nó được xây dựng dựa trên mô hình Object.

Nền tảng độc lập: Khi được biên dịch, Java không được biên dịch thành ngôn ngữ máy trên nền tảng cụ thể mà thay vào mã byte – một nền tảng độc lập. Mã byte này được thông dịch từ máy ảo (JVM) trên một nền tảng nào đó mà nó đang chạy.

Đơn giản: Java được thiết kế đơn giản, dễ học. Chỉ hiểu khái niệm cơ bản về OOP Java, để trở thành master về java rất dễ.

Bảo mật: Tính năng an toàn của Java cho phép phát triển các hệ thống không virus, không giả mạo, việc xác thực dựa trên mã hoá công khai.

Kiến trúc – trung lập: Trình biên dịch của Java tạo ra các định dạng tệp đối tượng kiến trúc trung lập, khiến mã biên dịch được thực thi trên nhiều bộ vi xử lý thông qua hệ điều hành Java.

Portable: Một loại kiến trúc trung lập, phụ thuộc vào việc thực hiện, là những đặc điểm chính khi nói về Portable của Java. Các trình biên dịch tại Java được viết bằng ANSI C với ranh giới Portable gọn gàng, gọi là Subset POSIX, bạn có thể mang Byte Code của Java lên bất cứ một nền tảng nào.

Mạnh mẽ: Java luôn nỗ lực loại trừ các tình huống dễ bị lỗi thông qua việc kiểm tra lỗi tại thời điểm biên dịch và kiểm tra lỗi tại runtime.

Đa luồng: Giúp tạo ra các chương trình thực hiện cùng lúc nhiều tác vụ, cho phép các nhà phát triển xây dựng ứng dụng tương tác chạy trơn tru hơn.

Thông dịch: Các mã byte Java được dịch trực tiếp tới các máy tính gốc và không được lưu trữ ở bất cứ đâu.

Hiệu năng cao: Khi sử dụng trình biên dịch Just-In-Time, Java cho phép thực hiện hiệu năng cao.

Phân tán: Java được sử dụng thiết kế cho môi trường phân tán của Internet.

Năng động: Java năng động hơn C hoặc C++ do được thiết kế với mục đích thích ứng môi trường đang phát triển. Các chương trình Java sở hữu lượng lớn thông tin tại runtime có thể được sử dụng để xác minh hay giải quyết các truy cập vào các đối tượng tại runtime.



Hình 1.1. Các tính năng của ngôn ngữ Java

1.4.1.2. Ưu điểm của Java

- Java là ngôn ngữ độc lập với nền tảng vì chúng ta có thể chạy mã Java ở bất kỳ máy nào mà không cần phần mềm đặc biệt.

- Là ngôn ngữ lập trình cấp cao dễ học và dễ hiểu.
- Là ngôn ngữ hướng đối tượng làm tăng khả năng phát triển mã dễ dàng và tăng hiệu quả.
- Là ngôn ngữ an toàn vì Java không sử dụng con trỏ.
- Quản lý bộ nhớ hiệu quả.
- Hỗ trợ đa luồng, người dùng có thể thực hiện cùng lúc nhiều chương trình.
- Có nhiều tính năng: tự động thu gom rác, không sử dụng con trỏ, xử lý ngoại lệ...

1.4.1.3. Nhược điểm của Java

- Là ngôn ngữ cấp cao nên phải xử lý các mức biên dịch và trừu tượng của một máy ảo.
- Hiệu suất kém.
- Có ít trình xây dựng GUI (Giao diện người dùng đồ họa) như Swing, SWT, JSF và JavaFX.
- Nếu viết những đoạn mã dài phức tạp dễ ảnh hưởng đến khả năng đọc của mã.

1.4.2. Giới thiệu về thư viện Retrofit

Retrofit được định nghĩa là một type-safe HTTP client cho Java, Android và Kotlin do công ty Square phát triển. Retrofit hỗ trợ cho các nhà lập trình chuyển đổi API thành Java Interface để dễ dàng kết nối đến một dịch vụ REST trên web.

Retrofit là một trong những thư viện Rest-Client được sử dụng phổ biến hiện nay trong việc hỗ trợ giao tiếp giữa client và server trở lên thuận tiện hơn. Thư viện này rất mạnh mẽ cho phép bạn dễ dàng xử lý dữ liệu JSON hoặc XML và sau đó sẽ phân tích cú pháp thành Plain Old Java Objects (POJOs). Tại đây, hầu hết các yêu cầu như GET, POST, PUT, PATCH, và DELETE đều có thể được thực hiện.



Hình 1.2. Retrofit trong android

Tương tự như các phần mềm mã nguồn mở khác, Retrofit được xây dựng dựa trên các thư viện và công cụ mạnh mẽ. Retrofit làm cho việc sử dụng OkHttp trở nên đơn giản

để xử lý các yêu cầu trên mạng. Bên cạnh đó, kể từ Retrofit2 sẽ không tích hợp với bất kỳ một bộ chuyển đổi JSON nào để phân tích JSON thành các đối tượng Java. Để xử lý điều đó, Retrofit sẽ đi kèm với các thư viện chuyển đổi JSON sau đây:

- **Gson:** com.squareup.retrofit:converter-gson
- **Jackson:** com.squareup.retrofit:converter-jackson
- **Moshi:** com.squareup.retrofit:converter-moshi

1.4.3. XML trong Android

XML là viết tắt của eXtensible Markup Language (Ngôn ngữ đánh dấu mở rộng), là một cách mô tả dữ liệu sử dụng tài liệu dựa trên văn bản. XML có thể mở rộng và rất linh hoạt nên được sử dụng cho nhiều mục đích khác nhau, bao gồm định nghĩa bố cục giao diện người dùng cho các ứng dụng Android.

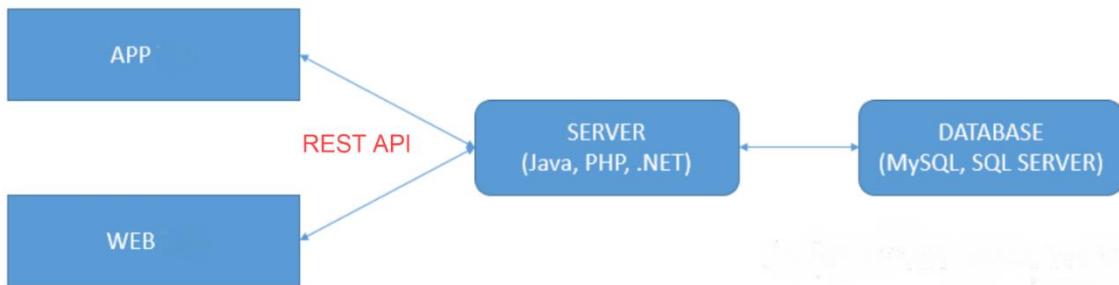
1.4.4. RESTful API

API (Application Programming Interface) là một tập các quy tắc và cơ chế mà theo đó, một ứng dụng hay một thành phần sẽ tương tác với một ứng dụng hay thành phần khác. API có thể trả về dữ liệu mà bạn cần cho ứng dụng của mình ở những kiểu dữ liệu phổ biến như JSON hay XML.

REST (REpresentational State Transfer) là một dạng chuyển đổi cấu trúc dữ liệu, một kiểu kiến trúc để viết API. Nó sử dụng phương thức HTTP đơn giản để tạo cho giao tiếp giữa các máy. Vì vậy, thay vì sử dụng một URL cho việc xử lý một số thông tin người dùng, REST gửi một yêu cầu HTTP như GET, POST, DELETE, vv đến một URL để xử lý dữ liệu.

RESTful API là một tiêu chuẩn dùng trong việc thiết kế các API cho các ứng dụng web để quản lý các resource. RESTful là một trong những kiểu thiết kế API được sử dụng phổ biến ngày nay để cho các ứng dụng (web, mobile...) khác nhau giao tiếp với nhau.

Chức năng quan trọng nhất của **REST** là quy định cách sử dụng các HTTP method (như GET, POST, PUT, DELETE...) và cách định dạng các URL cho ứng dụng web để quản lý các resource. RESTful không quy định logic code ứng dụng và không giới hạn bởi ngôn ngữ lập trình ứng dụng, bất kỳ ngôn ngữ hoặc framework nào cũng có thể sử dụng để thiết kế một **RESTful API**.

*Hình 1.3. RESTful API*

1.4.5. Spring Boot

Spring Boot là một dự án khá nổi bật trong hệ sinh thái Spring Framework. Nếu như trước đây, công đoạn khởi tạo một dự án Spring khá vất vả từ việc khai báo các dependency trong file pom.xml cho đến cấu hình bằng XML hoặc annotation phức tạp, tạo server cấu hình spring mvc, thì giờ đây với Spring Boot, chúng ta có thể tạo dự án Spring một cách nhanh chóng và cấu hình cũng đơn giản hơn. Dưới đây là một số tính năng nổi bật của Spring Boot:

- Tạo các ứng dụng Spring độc lập
- Nhúng trực tiếp Tomcat, Jetty hoặc Undertow (không cần phải deploy ra file WAR)
- Các starter dependency giúp việc cấu hình Maven đơn giản hơn
- Tự động cấu hình Spring khi cần thiết
- Không sinh code cấu hình và không yêu cầu phải cấu hình bằng XML

*Hình 1.4. Spring Boot*

1.4.6. Giới thiệu về hệ quản trị cơ sở dữ liệu MySQL và công vụ MySQL Workbench

1.4.6.1. Hệ quản trị cơ sở dữ liệu MySQL

Hệ quản trị cơ sở dữ liệu MySQL được hiểu như là chương trình dùng để quản lý hệ thống cơ sở dữ liệu, trong đó, cơ sở dữ liệu là một hệ thống lưu trữ thông tin được sắp xếp rõ ràng, phân lớp ngăn nắp. Nó giúp bạn có thể truy cập dữ liệu một cách thuận lợi và nhanh chóng nhất. Vì hỗ trợ đa số các ngôn ngữ lập trình nên MySQL chính là hệ quản trị cơ sở dữ liệu tự do nguồn mở phổ biến nhất trên thế giới. Hiện MySQL đang được các nhà phát triển rất ưa chuộng trong quá trình phát triển ứng dụng.

MySQL là cơ sở dữ liệu tốc độ cao, ổn định hoạt động trên nhiều hệ điều hành, cung cấp một hệ thống lớn các hàm tiện ích rất mạnh. Đặc biệt, hệ quản trị cơ sở dữ liệu MySQL hoàn toàn miễn phí nên người dùng có thể thoải mái tải về từ trang chủ. Nó có rất nhiều những phiên bản cho các hệ điều hành khác nhau. MySQL được sử dụng cho việc hỗ trợ PHP, Perl và nhiều ngôn ngữ khác. Là nơi lưu trữ những thông tin trên các trang web viết bằng framework PHP hay Perl...



Hình 1.5. Hệ quản trị cơ sở dữ liệu MySQL

1.4.6.2. Công vụ MySQL Workbench

MySQL Workbench chính là một chương trình giúp cho người lập trình có thể giao tiếp với hệ cơ sở dữ liệu MySQL thay vì phải sử dụng các lệnh Command-line phức tạp và mất thời gian. MySQL Workbench được thiết kế đơn giản, dễ sử dụng và có thể thích ứng với nhiều hệ điều hành như là Microsoft Windows, Mac OS, Linux hay Ubuntu.



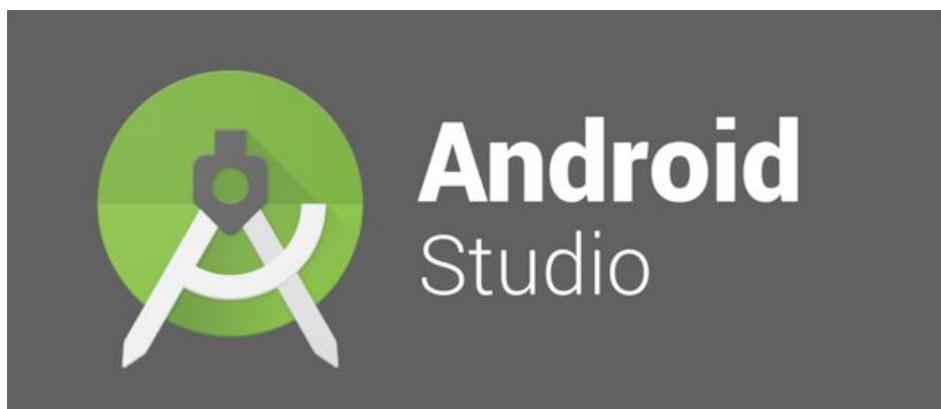
Hình 1.6. MySQL Workbench

1.4.7. Android Studio

Android Studio là IDE chính thức được sử dụng trong phát triển ứng dụng Android dựa trên IntelliJ IDEA.

Chức năng chính của Android Studio là cung cấp các giao diện giúp người dùng có thể tạo các ứng dụng và xử lý các công cụ file phức tạp sau hậu trường. Ngôn ngữ lập trình được sử dụng trong Android Studio là Java và nó sẽ được cài đặt sẵn trên thiết bị của bạn.

Khi sử dụng Android Studio thì chúng ta chỉ cần viết, chỉnh sửa và lưu trữ chúng trên các dự án của mình và các file nằm trong dự án đó. Đồng thời, Android Studio còn cung cấp quyền truy cập vào Android SDK.



Hình 1.7. Android Studio

CHƯƠNG 2. KHẢO SÁT HIỆN TRẠNG

2.1. Khảo sát hiện trạng

Thông tin về cửa hàng như sau:

- Trong cửa hàng có nhiều loại sản phẩm gồm: sách, giấy, bút, đồ dùng học tập, công cụ văn phòng, kẹp giấy....
- Mỗi khách hàng muốn đặt hàng online cần đăng ký tài khoản gồm các thông tin: họ tên, email, mật khẩu,...
- Sau khi đăng ký tài khoản thì thông tin khách hàng gồm: họ tên, email, username, điện thoại,...
- Hàng ngày có khách hàng vào ứng dụng để tìm kiếm sản phẩm và đặt hàng.
- Nhân viên cửa hàng cần kiểm tra hàng tồn kho để duyệt đơn hàng, sau đó giao hàng cho shipper để giao cho khách.
- Shipper đăng nhập vào app kiểm tra các đơn hàng thuộc trách nhiệm của mình, giao hàng cho khách
- Hàng tháng, nhân viên sẽ thống kê doanh thu và báo cáo lợi nhuận, đồng thời nhập thêm hàng vào kho.

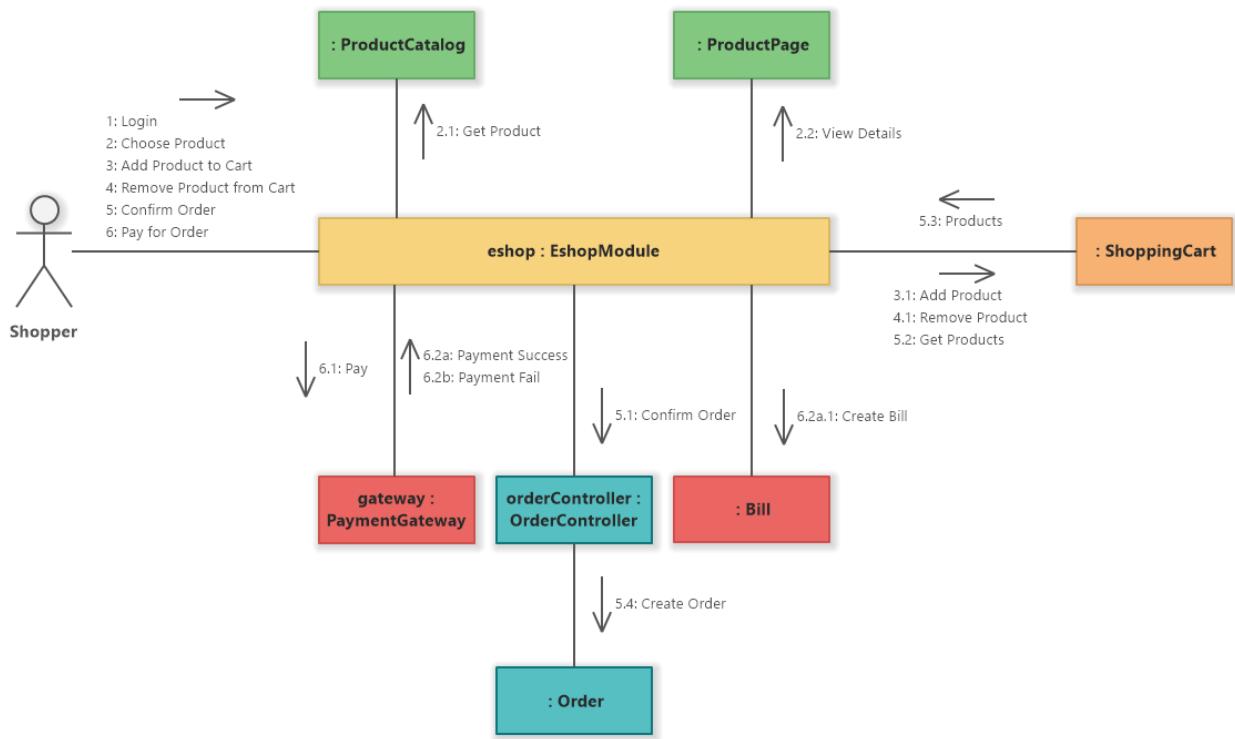
Hệ thống người dùng cho phép:

- Tra cứu sản phẩm có trong cửa hàng mà không cần đăng nhập
- Đăng nhập và thêm sản phẩm vào giỏ hàng
- Đặt hàng và theo dõi đơn hàng
- Nhận hàng và đánh giá đơn hàng
- Trả hàng nếu không hài lòng với sản phẩm

Hệ thống quản lý cho phép:

- Quản trị:
 - Quản lý nhân viên:
 - + Tiếp nhận nhân viên mới, nhập thông tin vào dữ liệu quản lý.
 - + Cập nhật thông tin nhân viên.
 - + Xóa thông tin nhân viên.
 - Lập báo cáo thống kê doanh thu
 - Quản lý sản phẩm, thể loại, nhãn hàng
 - Duyệt đơn hàng
- Nhân viên:
 - Duyệt đơn hàng.
 - Quản lý sản phẩm, thể loại, nhãn hàng
- Shipper:
 - Xem đơn hàng thuộc trách nhiệm của mình
 - Giao hàng và xác nhận đã giao.

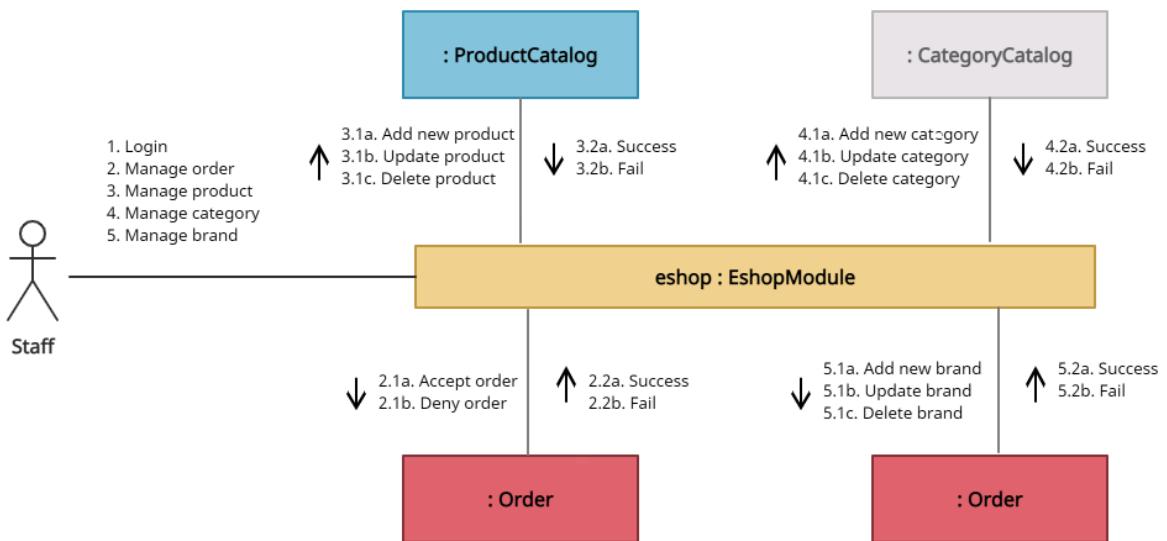
2.2. Giải pháp đề xuất



Hình 2.1. Sơ đồ công tác cho khách hàng

Khách hàng:

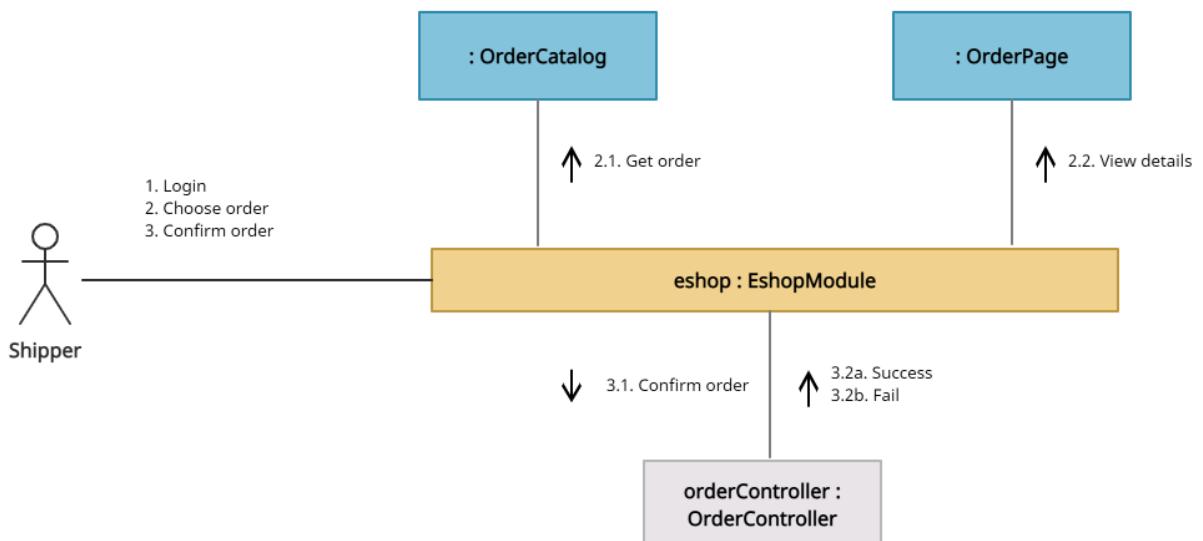
- 1) Đăng nhập vào ứng dụng
- 2) Xem danh sách sản phẩm và chi tiết từng sản phẩm
- 3) Thêm sản phẩm vào giỏ hàng
- 4) Xoá sản phẩm khỏi giỏ hàng
- 5) Từ trang giỏ hàng, khách chọn các sản phẩm muốn mua và ấn xác nhận → sản phẩm từ giỏ hàng được chuyển vào đơn hàng
- 6) Di chuyển đến trang thanh toán, khách chọn hình thức thanh toán và nhấn thanh toán → hệ thống báo thành công/thất bại



Hình 2.2. Sơ đồ công tác cho nhân viên cửa hàng

Nhân viên:

- 1) Đăng nhập vào ứng dụng
- 2) Từ trang quản lý đơn hàng, nhân viên có thể chấp nhận/từ chối các đơn hàng → hệ thống báo thành công/thất bại
- 3) Từ trang quản lý sản phẩm, nhân viên có thể thêm/xoá/sửa các sản phẩm → hệ thống báo thành công/thất bại
- 4) Từ trang quản lý thể loại, nhân viên có thể thêm/xoá/sửa các thể loại → hệ thống báo thành công/thất bại
- 5) Từ trang quản lý nhãn hiệu, nhân viên có thể thêm/xoá/sửa các nhãn hiệu → hệ thống báo thành công/thất bại



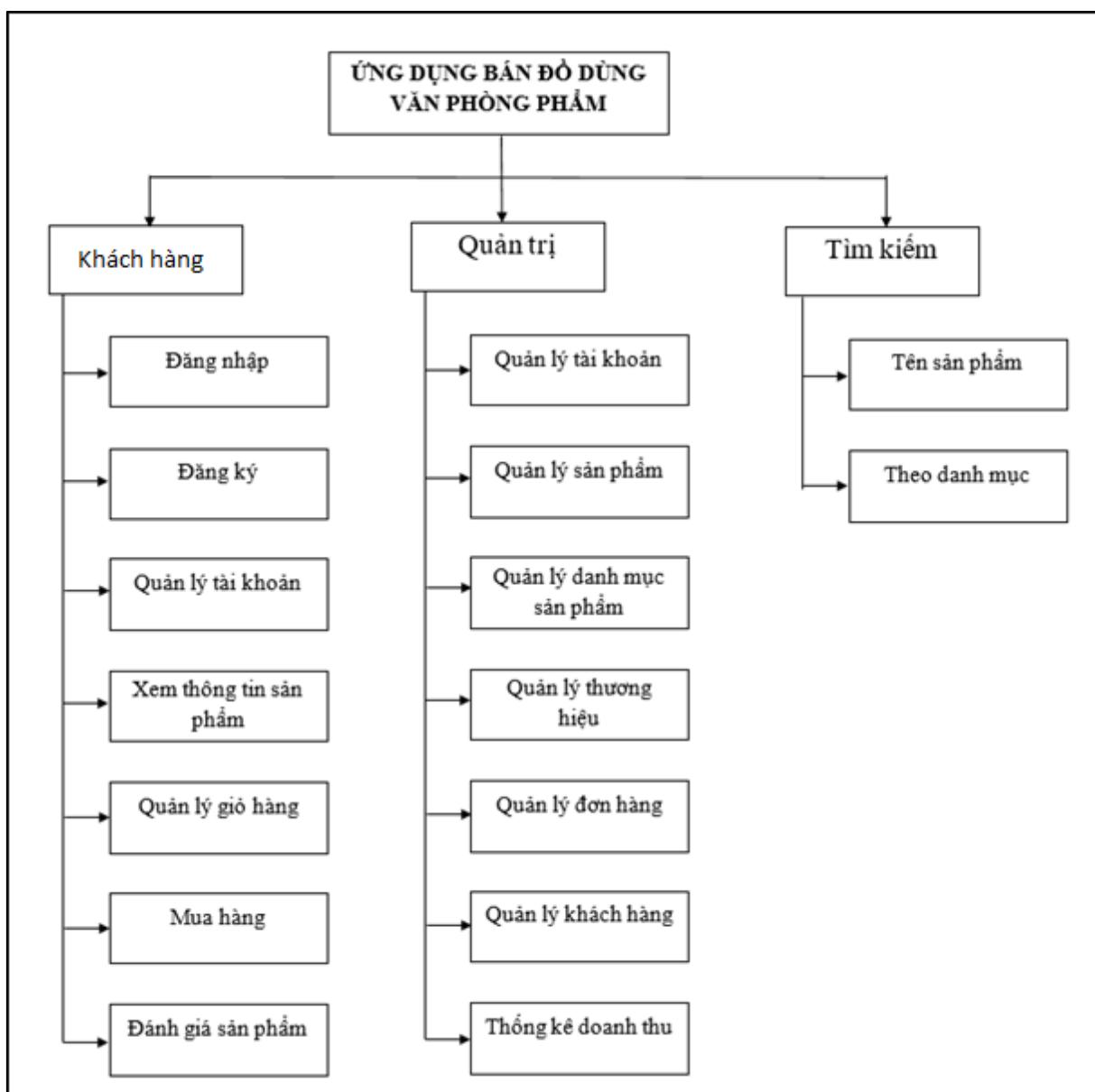
Hình 2.3. Sơ đồ công tác cho shipper

Shipper:

- 1) Đăng nhập vào ứng dụng
- 2) Xem danh sách đơn hàng thuộc trách nhiệm của mình và chi tiết đơn hàng
- 3) Xác nhận đã giao hàng → hệ thống báo thành công/thất bại

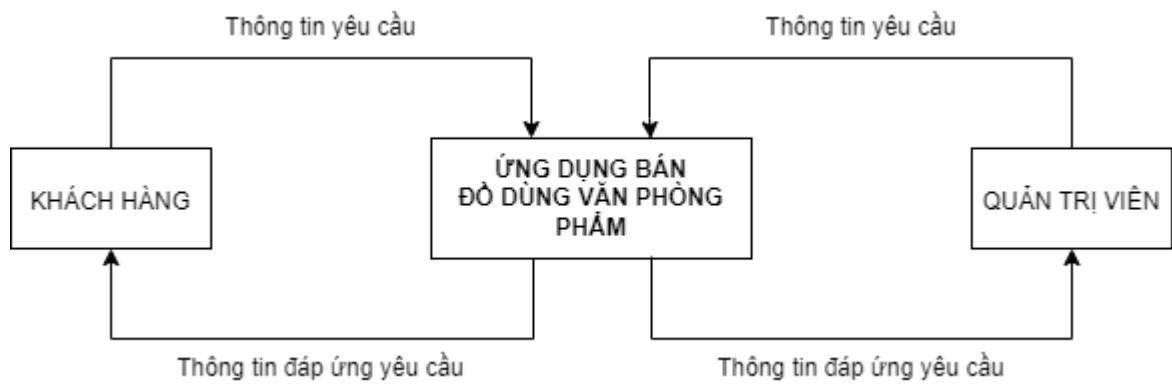
CHƯƠNG 3. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

3.1. Sơ đồ phân rã chức năng BFD



Hình 3.1. Sơ đồ phân rã chức năng BFD

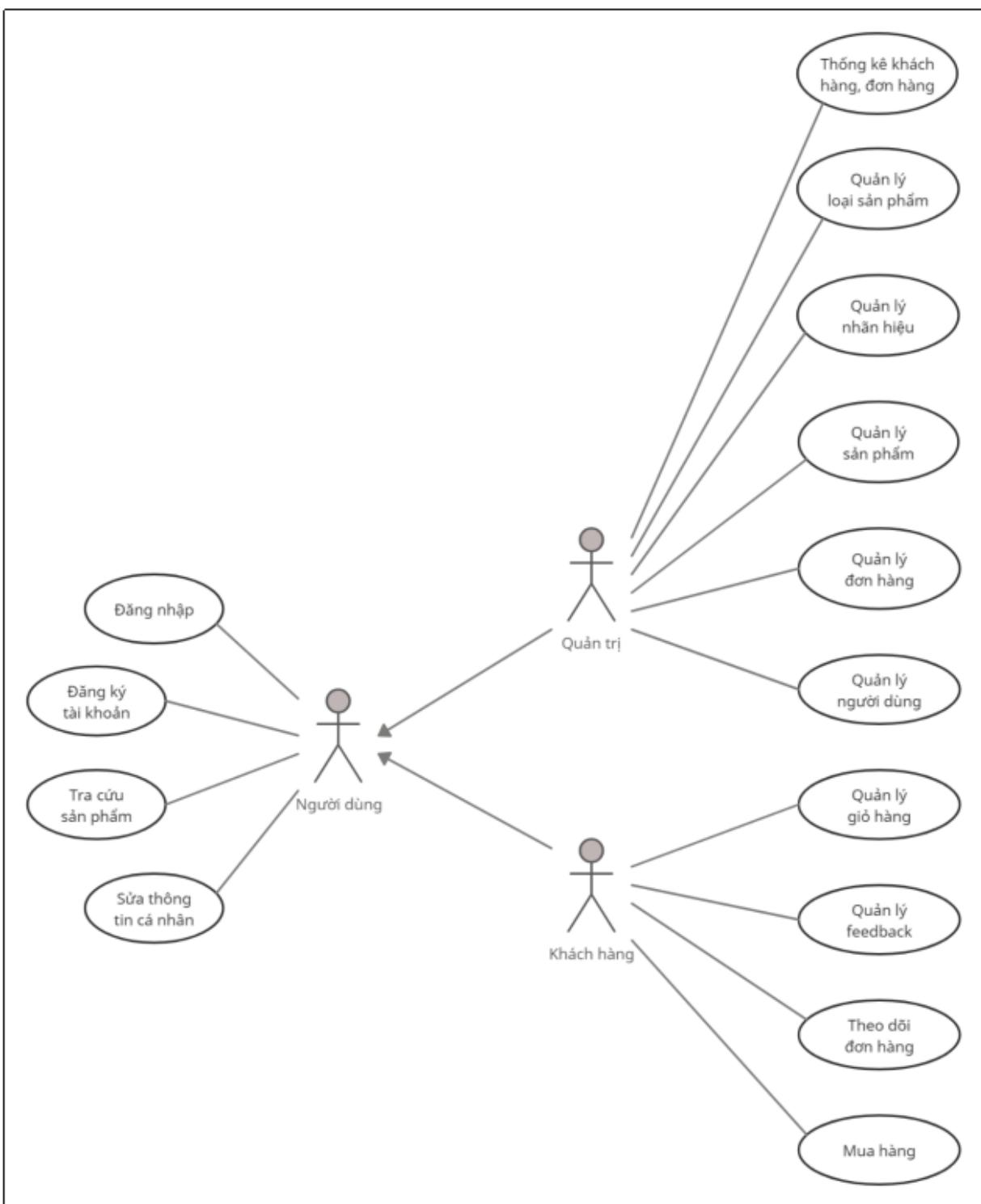
3.2. Sơ đồ ngữ cảnh



Hình 3.2. Sơ đồ ngữ cảnh

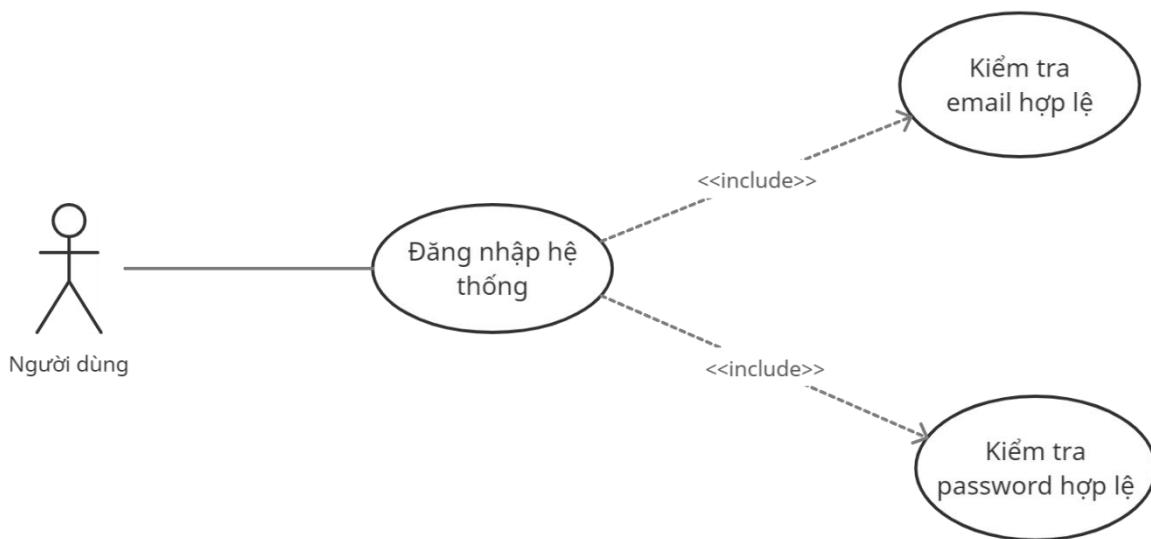
3.3. Lược đồ Usecase

3.3.1. Usecase Tổng quát



Hình 3.3. Usecase Tổng quát

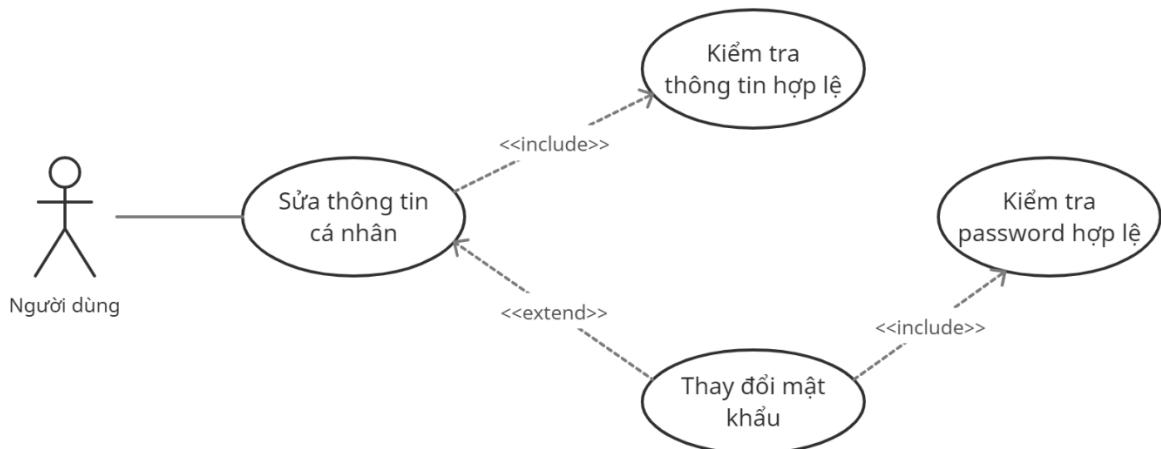
3.3.2. Usecase Đăng nhập



Hình 3.4. Usecase Đăng nhập

Tên Use-Case	Đăng nhập
Actor	Quản trị, khách hàng
Mô tả	Tác nhân sử dụng Usecase này để thực hiện đăng nhập vào hệ thống
Điều kiện kích hoạt	Chọn chức năng Đăng nhập ở hệ thống
Tiền điều kiện	Không đòi hỏi yêu cầu gì trước đó.
Hậu điều kiện	<p>Nếu thành công: Hệ thống sẽ hiển thị giao diện chính. Người dùng có thể thực hiện các chức năng theo đúng quyền hạn của mình</p> <p>Nếu thất bại: Hệ thống sẽ đưa ra thông báo “Thông tin đăng nhập không hợp lệ” và yêu cầu đăng nhập lại</p>
Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Hệ thống hiển thị giao diện đăng nhập 2. Người dùng nhập tài khoản bao gồm Tên đăng nhập và mật khẩu. 3. Hệ thống kiểm tra và xác nhận thông tin đăng nhập. 4. Hiển thị giao diện chính của phần mềm 5. Kết thúc Use case.
Luồng sự kiện phụ	<p>Dòng sự kiện phụ 1</p> <ol style="list-style-type: none"> 1. Người dùng hủy yêu cầu đăng nhập. 2. Hệ thống đóng lại.

	<p>3. Kết thúc Usecase.</p> <p>Dòng sự kiện phụ 2</p> <ol style="list-style-type: none"> Người dùng nhập thông tin sai. Hệ thống hiển thị thông báo lỗi. Kết thúc Use case.
Các yêu cầu đặc biệt	Không có

*Bảng 3.1. Usecase Đăng nhập***3.3.3. Usecase Sửa thông tin cá nhân***Hình 3.5. Usecase Sửa thông tin cá nhân*

Tên Use-Case	Sửa thông tin cá nhân
Actor	Quản trị, khách hàng
Mô tả	Tác nhân sử dụng Usecase này để thực hiện chức năng sửa các thông tin liên quan đến người dùng như: họ tên, chức danh, giới tính, email, điện thoại.
Điều kiện kích hoạt	Tác nhân chọn chức năng Sửa thông tin cá nhân
Tiền điều kiện	Tác nhân phải đăng nhập vào hệ thống
Hậu điều kiện	<p>Nếu thành công: Thông tin người dùng được cập nhật thành công vào hệ thống.</p> <p>Nếu thất bại: Hệ thống trả về thông báo lỗi, thông tin không được cập nhật thành công</p>
Luồng sự kiện chính	<ol style="list-style-type: none"> Tác nhân chọn chức năng Sửa thông tin người dùng Tác nhân chọn thông tin cần sửa Tác nhân sửa thông tin Tác nhân chọn lưu thông tin

	5. Hệ thống kiểm tra sự hợp lệ của thông tin đưa vào. 6. Hệ thống lưu thông tin vào cơ sở dữ liệu 7. Usecase kết thúc
Luồng sự kiện phụ	Dòng sự kiện phụ 1 1. Tác nhân hủy bỏ việc sửa người dùng 2. Hệ thống bỏ qua và trở về giao diện chính 3. Kết thúc Usecase. Dòng sự kiện phụ 2 1. Thông tin tác nhân nhập vào không hợp lệ 2. Hệ thống hiển thị thông báo lỗi 3. Kết thúc Usecase 2. Hệ thống hiển thị lỗi 3. Kết thúc Usecase
Các yêu cầu đặc biệt	Không có

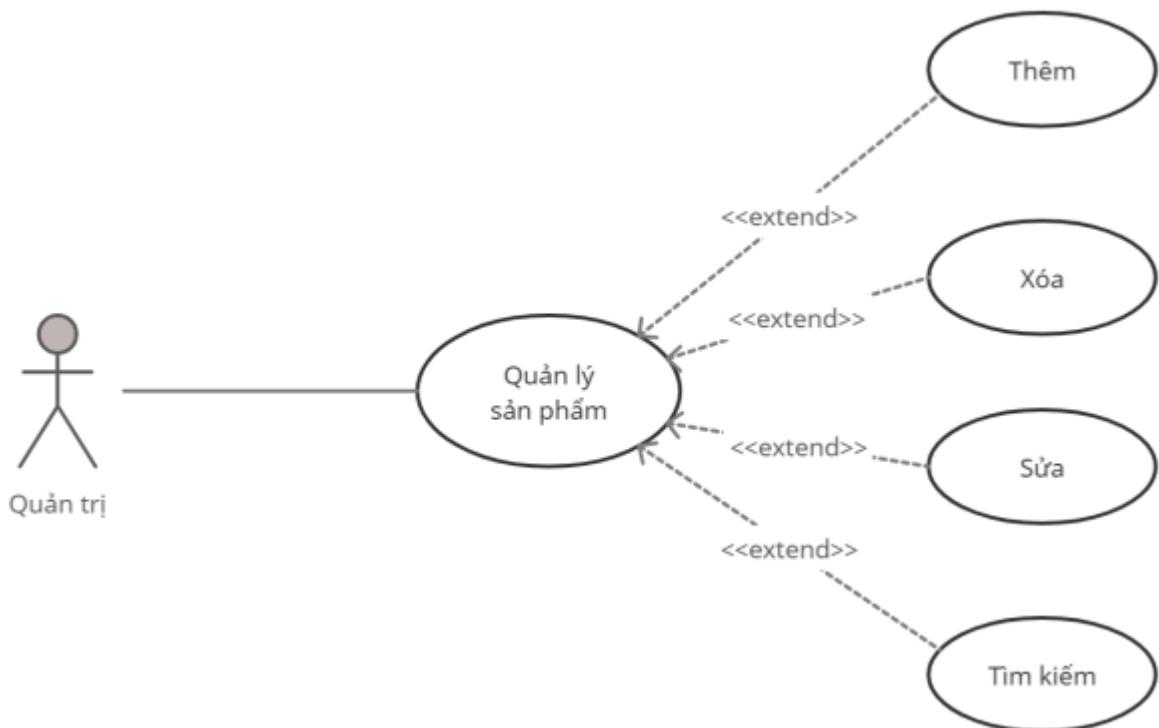
Bảng 3.2. Usecase sửa thông tin cá nhân

Tên Use-Case	Thay đổi mật khẩu
Actor	Quản trị, khách hàng
Mô tả	Tác nhân chọn usecase này để thay đổi mật khẩu.
Điều kiện kích hoạt	Chọn đổi mật khẩu ở hệ thống
Tiền điều kiện	Tác nhân phải đăng nhập vào hệ thống
Hậu điều kiện	Nếu thành công: Mật khẩu của người dùng được thay đổi thành công, hệ thống hiển thị form “Đăng nhập” cho phép người dùng đăng nhập lại. Nếu thất bại: hệ thống thông báo lỗi và mật khẩu của người dùng không được thay đổi.
Luồng sự kiện chính	1. Người dùng chọn chức năng thay đổi mật khẩu 2. Hệ thống hiển thị form cho phép người dùng thay đổi mật khẩu. 3. Người dùng nhập lại mật khẩu cũ và nhập mật khẩu mới vào textbox. 4. Hệ thống kiểm tra sự hợp lệ của thông tin đưa vào và lưu thông tin vào hệ thống. 5. Kết thúc Usecase

Luồng sự kiện phụ	<p>Dòng thứ 1</p> <ol style="list-style-type: none"> Người dùng hủy bỏ việc thay đổi mật khẩu. Hệ thống không thực hiện chức năng thay đổi mật khẩu. Kết thúc Usecase <p>Dòng thứ 2</p> <ol style="list-style-type: none"> Thông tin người dùng nhập vào không hợp lệ Hệ thống không thực hiện chức năng thay đổi mật khẩu. Kết thúc Use case
Các yêu cầu đặc biệt	Không có

Bảng 3.3. Usecase Thay đổi mật khẩu

3.3.4. Usecase Quản lý sản phẩm

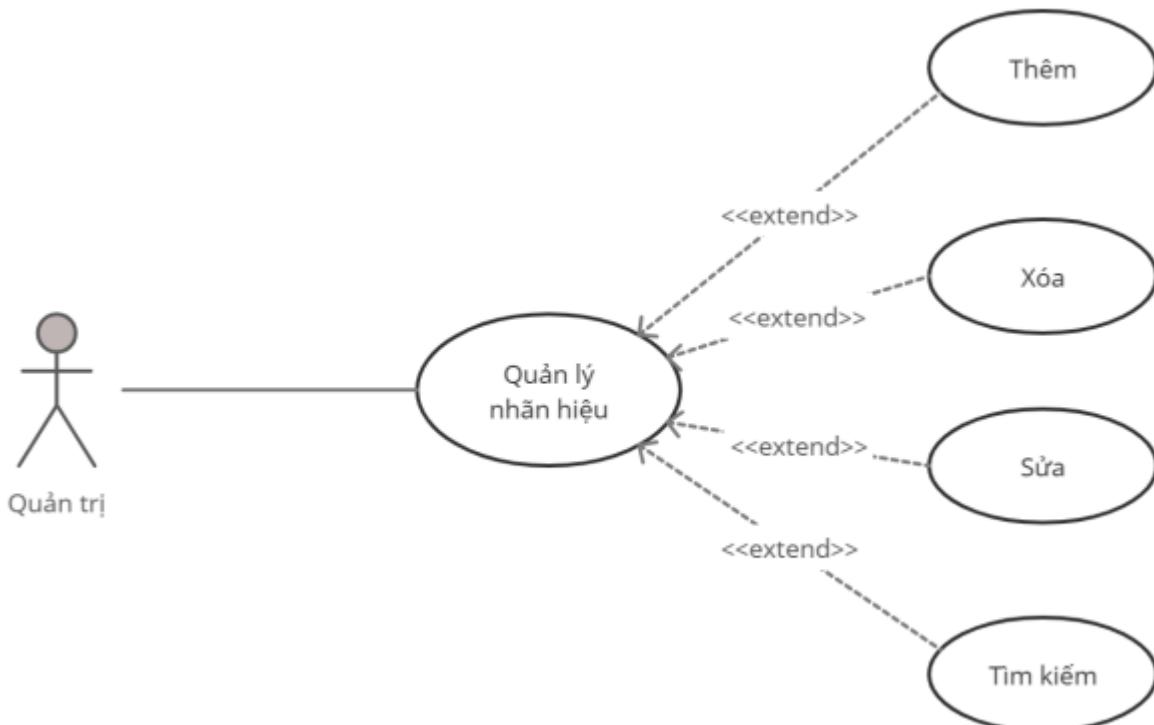
*Hình 3.6. Usecase Quản lý sản phẩm*

Use-Case	Nội dung
Mô tả	Cho phép quản trị viên tìm kiếm, thêm, xóa, cập nhật thông tin sản phẩm
Actor	Quản trị viên

Điều kiện kích hoạt	Quản trị viên chọn chức năng quản lý sản phẩm
Điều kiện trước	Quản trị viên đăng nhập vào hệ thống
Điều kiện sau	Hệ thống hiển thị màn hình quản lý sản phẩm và cung cấp chức năng cần thiết.
Luồng sự kiện chính	<ol style="list-style-type: none"> 1. Quản trị viên chọn một trong các chức năng: thêm, xóa, thay đổi thông tin sản phẩm 2. Hệ thống kiểm tra dữ liệu khi xóa sản phẩm 3. Hệ thống cập nhật CSDL. 4. Kết thúc Use-Case
Luồng sự kiện phụ	<ul style="list-style-type: none"> ❖ Dữ liệu xóa không thỏa các ràng buộc dữ liệu <ol style="list-style-type: none"> 1. Hệ thống thông báo lỗi 2. Quay lại bước 1 trong luồng sự kiện chính.

Bảng 3.4. Usecase Quản lý sản phẩm

3.3.5. Usecase Quản lý nhãn hiệu



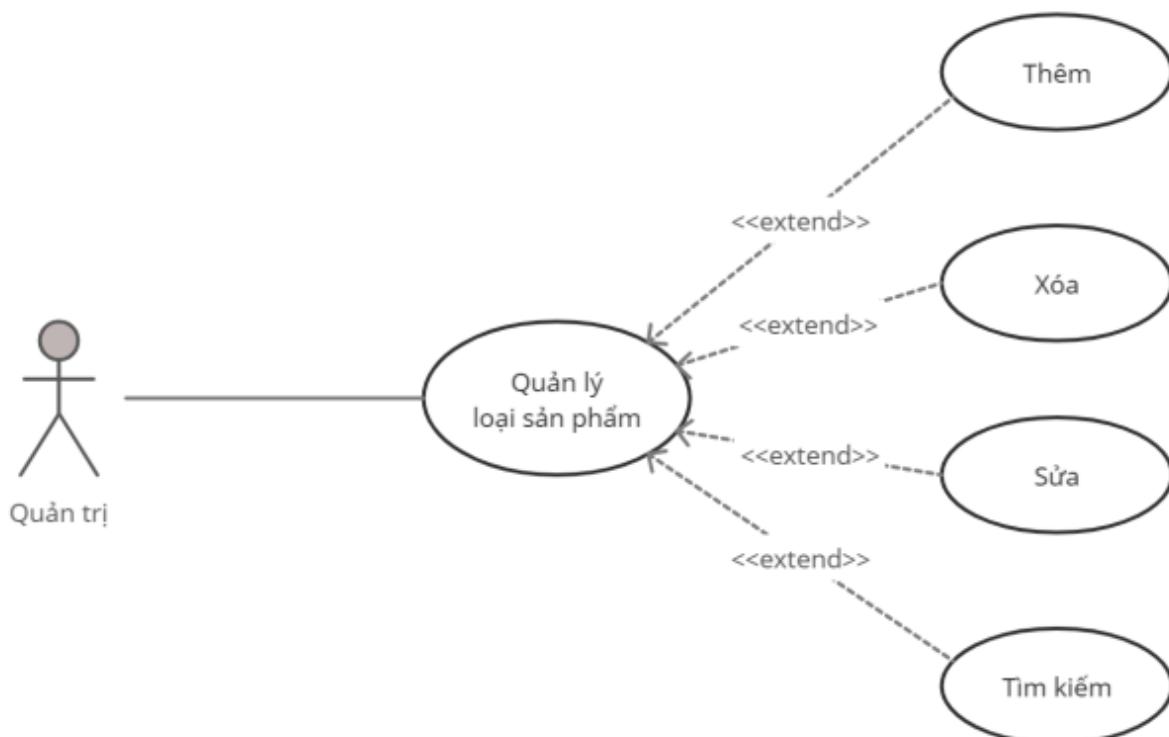
Hình 3.7. Usecase Quản lý nhãn hiệu

Use-Case	Nội dung
Mô tả	Cho phép quản trị viên tìm kiếm, thêm, xóa, cập nhật thông tin nhãn hiệu
Actor	Quản trị viên

Điều kiện kích hoạt	Quản trị viên chọn chức năng quản lý nhãn hiệu
Điều kiện trước	Quản trị viên đăng nhập vào hệ thống
Điều kiện sau	Hệ thống hiển thị màn hình quản lý nhãn hiệu và cung cấp chức năng cần thiết.
Luồng sự kiện chính	<ol style="list-style-type: none"> Quản trị viên chọn một trong các chức năng: thêm, xóa, thay đổi thông tin nhãn hiệu Hệ thống kiểm tra dữ liệu khi xóa nhãn hiệu Hệ thống cập nhật CSDL. Kết thúc Use-Case
Luồng sự kiện phụ	<ul style="list-style-type: none"> ❖ Dữ liệu xóa không thỏa các ràng buộc dữ liệu <ol style="list-style-type: none"> Hệ thống thông báo lỗi Quay lại bước 1 trong luồng sự kiện chính.

Bảng 3.5. Usecase Quản lý nhãn hiệu

3.3.6. Usecase Quản lý loại sản phẩm



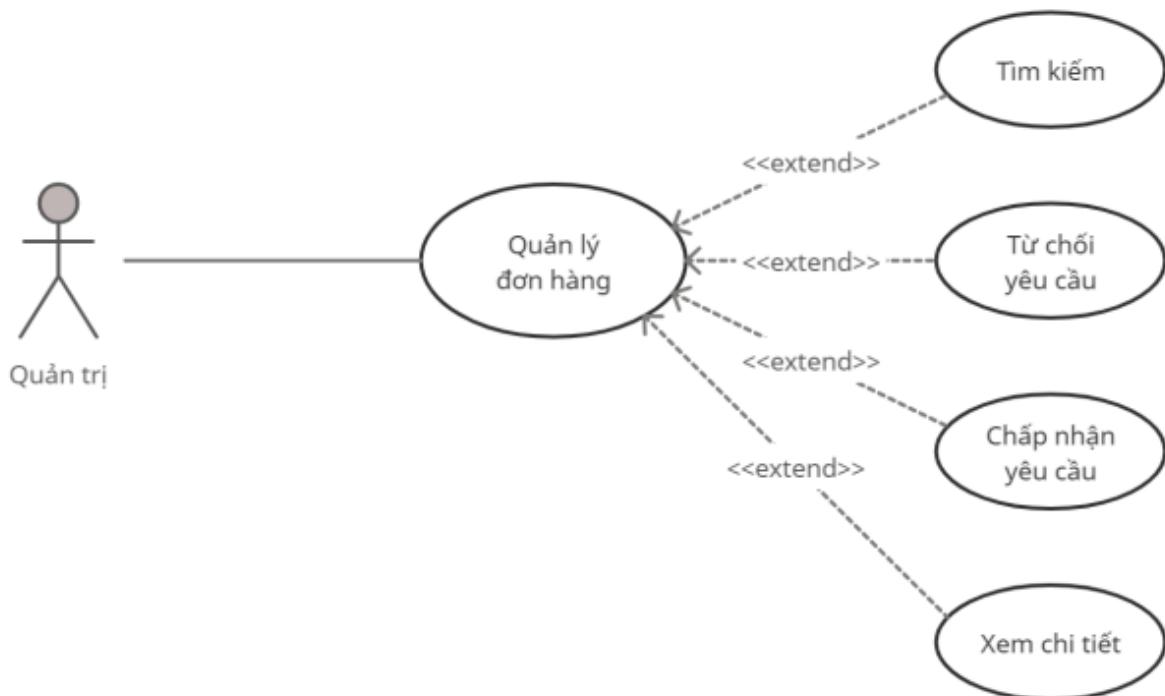
Hình 3.8. Usecase Quản lý loại sản phẩm

Use-Case	Nội dung
Mô tả	Cho phép quản trị viên tìm kiếm, thêm, xóa, cập nhật thông tin loại sản phẩm
Actor	Quản trị viên

Điều kiện kích hoạt	Quản trị viên chọn chức năng quản lý loại sản phẩm
Điều kiện trước	Quản trị viên đăng nhập vào hệ thống
Điều kiện sau	Hệ thống hiển thị màn hình quản lý loại sản phẩm và cung cấp chức năng cần thiết.
Luồng sự kiện chính	<ol style="list-style-type: none"> Quản trị viên chọn một trong các chức năng: thêm, xóa, thay đổi thông tin loại sản phẩm Hệ thống kiểm tra dữ liệu khi xóa loại sản phẩm Hệ thống cập nhật CSDL. Kết thúc Use-Case
Luồng sự kiện phụ	<p>❖ Dữ liệu xóa không thỏa các ràng buộc dữ liệu</p> <ol style="list-style-type: none"> Hệ thống thông báo lỗi Quay lại bước 1 trong luồng sự kiện chính.

Bảng 3.6. Usecase Quản lý loại sản phẩm

3.3.7. Usecase Quản lý đơn hàng



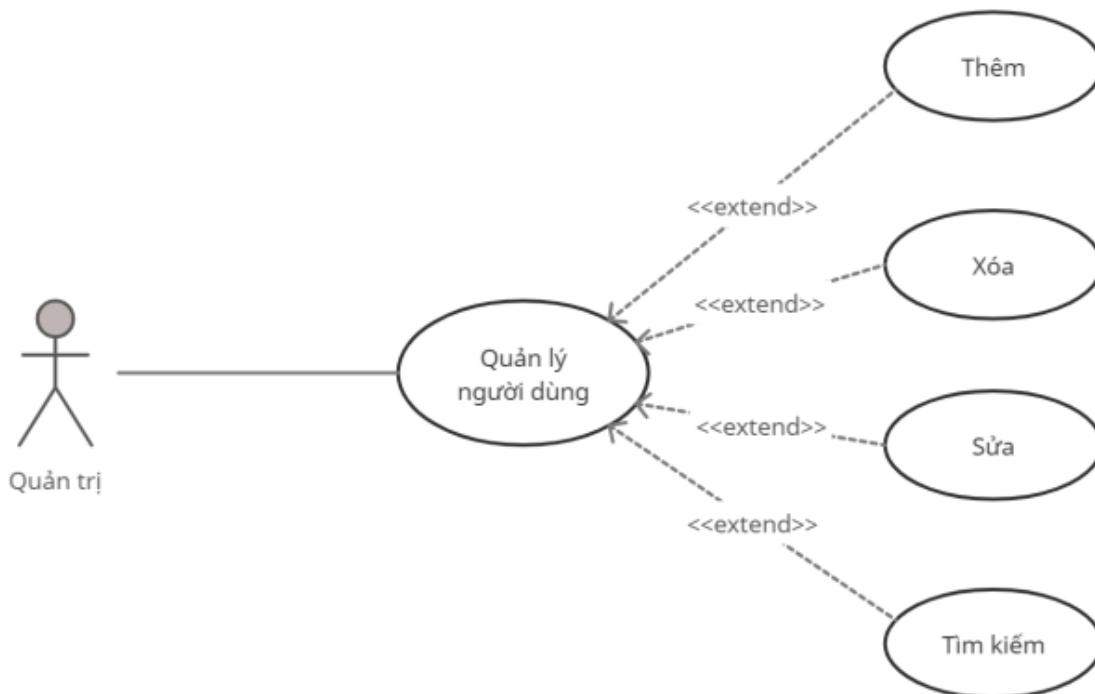
Hình 3.9. Usecase Quản lý đơn hàng

Use-Case	Nội dung
Mô tả	Cho phép quản trị viên tìm kiếm, từ chối, chấp nhận đơn đặt hàng
Actor	Quản trị viên

Điều kiện kích hoạt	Quản trị viên chọn chức năng quản lý đơn hàng
Điều kiện trước	Quản trị viên đăng nhập vào hệ thống
Điều kiện sau	Hệ thống hiển thị màn hình quản lý đơn hàng và cung cấp chức năng cần thiết.
Luồng sự kiện chính	<ol style="list-style-type: none"> Quản trị viên chọn một trong các trạng thái: Chờ xử lý, Yêu cầu hủy, Đang giao, Đã giao, Đã hủy Quản trị viên chọn một đơn hàng và chọn tiếp chức năng xem chi tiết, chấp nhận hoặc từ chối Hệ thống cập nhật CSDL. Kết thúc Use-Case
Luồng sự kiện phụ	Không có

Bảng 3.7. Usecase Quản lý đơn hàng

3.3.8. Usecase Quản lý người dùng



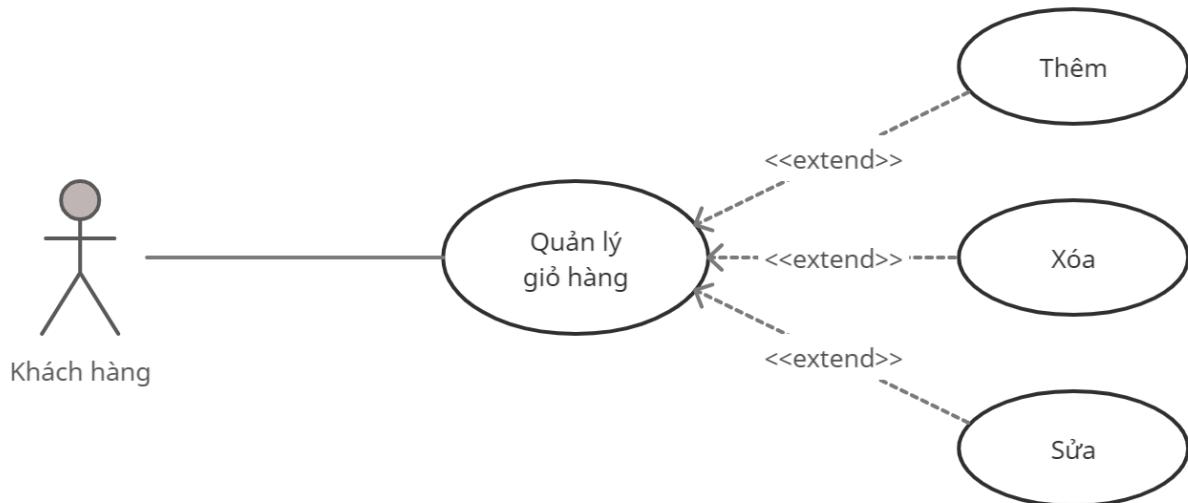
Hình 3.10. Usecase Quản lý người dùng

Use-Case	Nội dung
Mô tả	Cho phép quản trị viên tìm kiếm, thêm, xóa, cập nhật thông tin khách hàng
Actor	Quản trị viên
Điều kiện kích hoạt	Quản trị viên chọn chức năng quản lý khách hàng

Điều kiện trước	Quản trị viên đăng nhập vào hệ thống
Điều kiện sau	Hệ thống hiển thị màn hình quản lý khách hàng và cung cấp chức năng cần thiết.
Luồng sự kiện chính	<ol style="list-style-type: none"> Quản trị viên chọn một trong các chức năng: thêm, xóa, thay đổi thông tin khách hàng Hệ thống kiểm tra dữ liệu khi xóa loại sản phẩm Hệ thống cập nhật CSDL. Kết thúc Use-Case
Luồng sự kiện phụ	<p>❖ Dữ liệu xóa không thỏa các ràng buộc dữ liệu</p> <ol style="list-style-type: none"> Hệ thống thông báo lỗi Quay lại bước 1 trong luồng sự kiện chính.

Bảng 3.8. Usecase Quản lý người dùng

3.3.9. Usecase Quản lý giỏ hàng



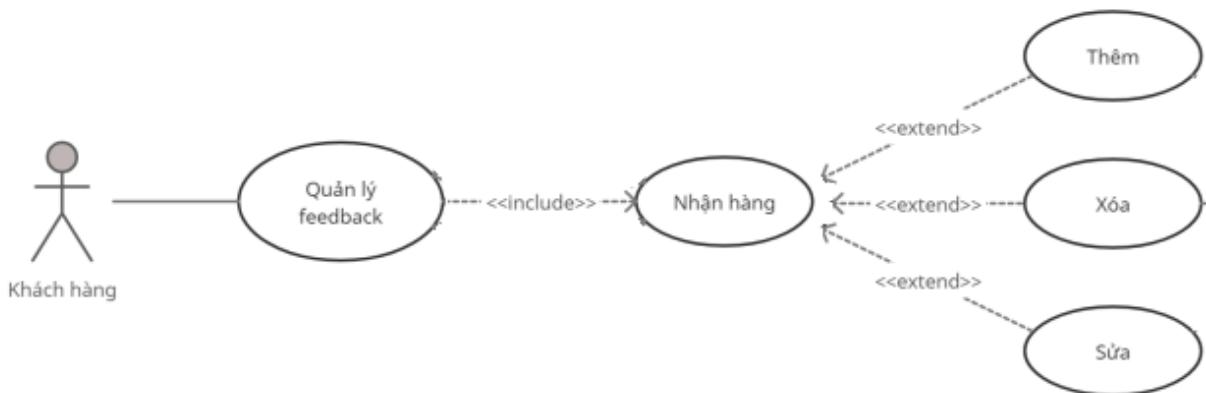
Hình 3.11. Usecase Quản lý giỏ hàng

Use-Case	Nội dung
Mô tả	Cho phép khách hàng tìm kiếm, thêm, xóa, cập nhật thông tin giỏ hàng
Actor	Khách hàng
Điều kiện kích hoạt	Khách hàng chọn chức năng quản lý giỏ hàng
Điều kiện trước	Khách hàng đăng nhập vào hệ thống
Điều kiện sau	Hệ thống hiển thị màn hình quản lý giỏ hàng và cung cấp chức năng cần thiết.

Luồng sự kiện chính	<ol style="list-style-type: none"> Quản trị viên chọn một trong các chức năng: thêm, xóa, thay đổi thông tin giỏ hàng Hệ thống cập nhật CSDL. Kết thúc Use-Case
Luồng sự kiện phụ	Không có

Bảng 3.9. Usecase Quản lý giỏ hàng

3.3.10. Usecase Quản lý đánh giá

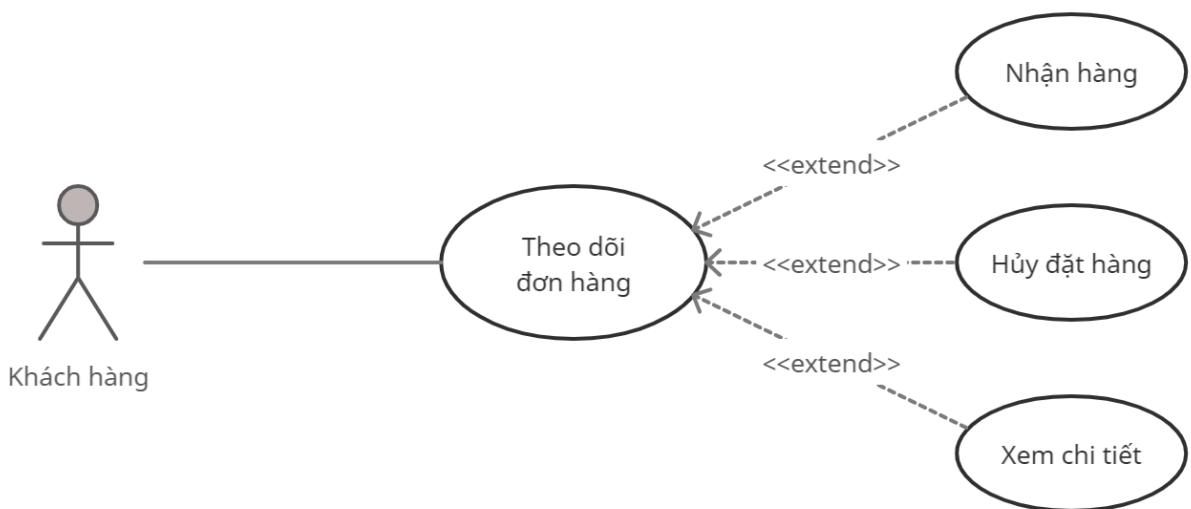


Hình 3.12. Usecase Quản lý đánh giá

Use-Case	Nội dung
Mô tả	Cho phép khách hàng thêm, xóa, cập nhật thông tin đánh giá sản phẩm
Actor	Khách hàng
Điều kiện kích hoạt	Khách hàng chọn chức năng đánh giá
Điều kiện trước	Khách hàng nhận hàng
Điều kiện sau	Hệ thống hiển thị màn hình nhập nội dung đánh giá và các chức năng cần thiết
Luồng sự kiện chính	<ol style="list-style-type: none"> Khách hàng nhập nội dung đánh giá Khách hàng chọn một trong các chức năng: thêm, xóa, thay đổi thông tin đánh giá Hệ thống cập nhật CSDL. Kết thúc Use-Case
Luồng sự kiện phụ	Không có

Bảng 3.10. Usecase Quản lý đánh giá

3.3.11. Usecase Theo dõi đơn hàng



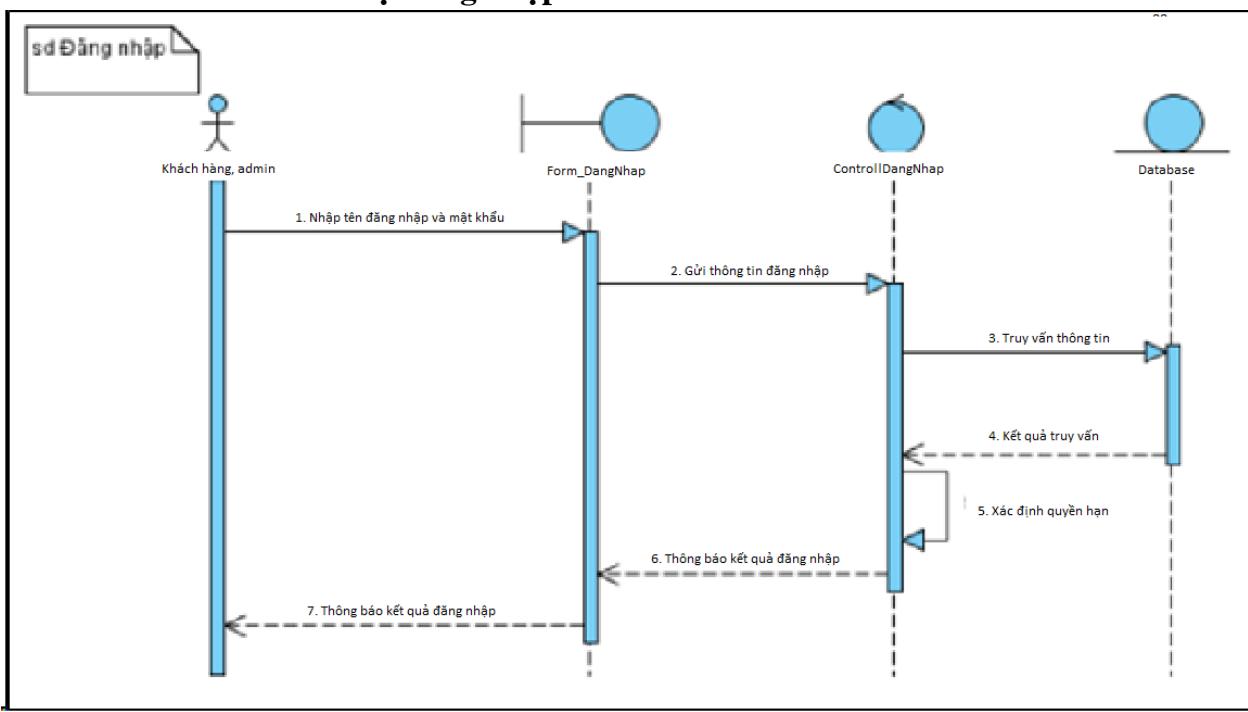
Hình 3.13. Usecase Theo dõi đơn hàng

Use-Case	Nội dung
Mô tả	Cho phép khách hàng xem chi tiết, nhận/hủy đơn hàng
Actor	Khách hàng
Điều kiện kích hoạt	Khách hàng chọn chức năng theo dõi đơn hàng
Điều kiện trước	Khách hàng đã mua hàng
Điều kiện sau	Hệ thống hiển thị màn hình thông tin đơn hàng và cung cấp chức năng cần thiết.
Luồng sự kiện chính	<ol style="list-style-type: none"> Khách hàng chọn một đơn hàng Khách hàng chọn một trong các chức năng: xem chi tiết, nhận/hủy đơn hàng Hệ thống cập nhật CSDL. Kết thúc Use-Case
Luồng sự kiện phụ	Không có

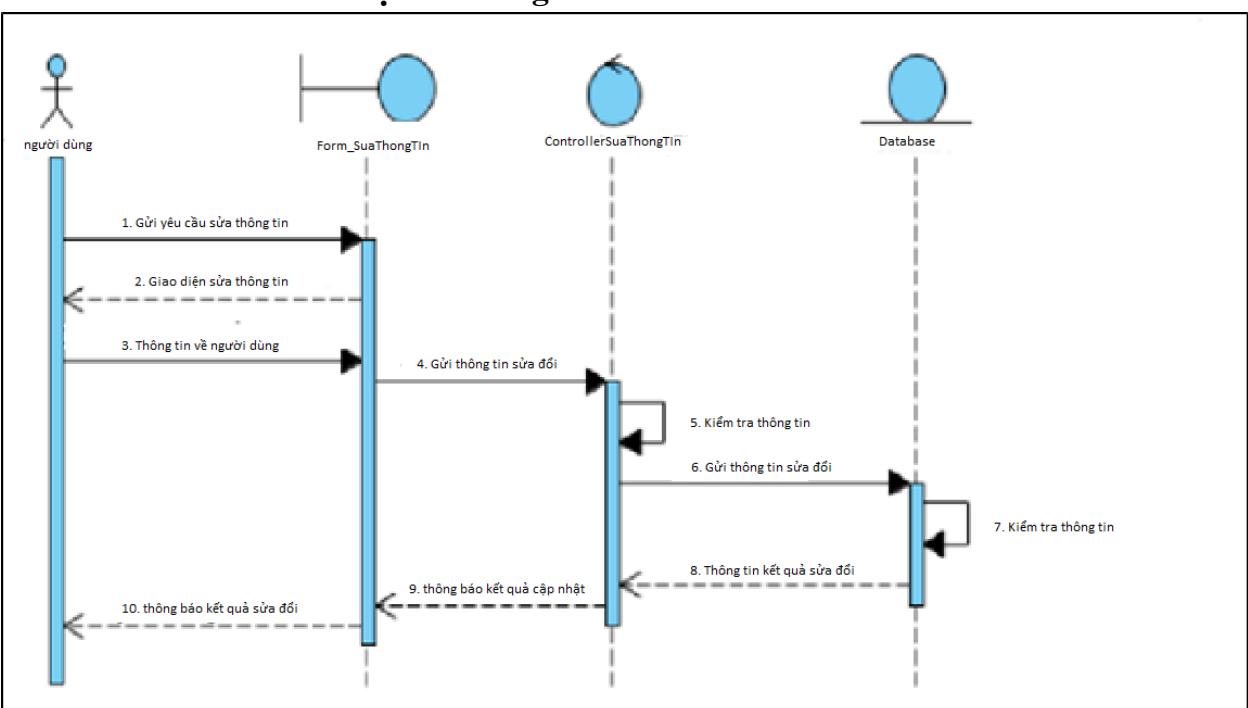
Bảng 3.11. Usecase Theo dõi đơn hàng

3.4. Sơ đồ tuần tự

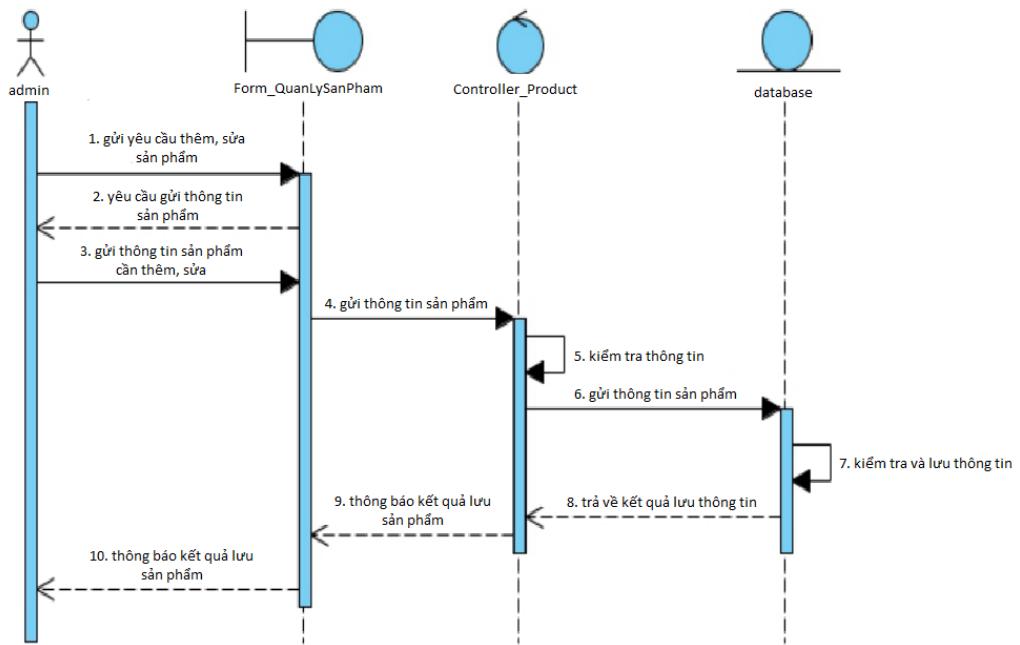
3.4.1. Sơ đồ tuần tự đăng nhập



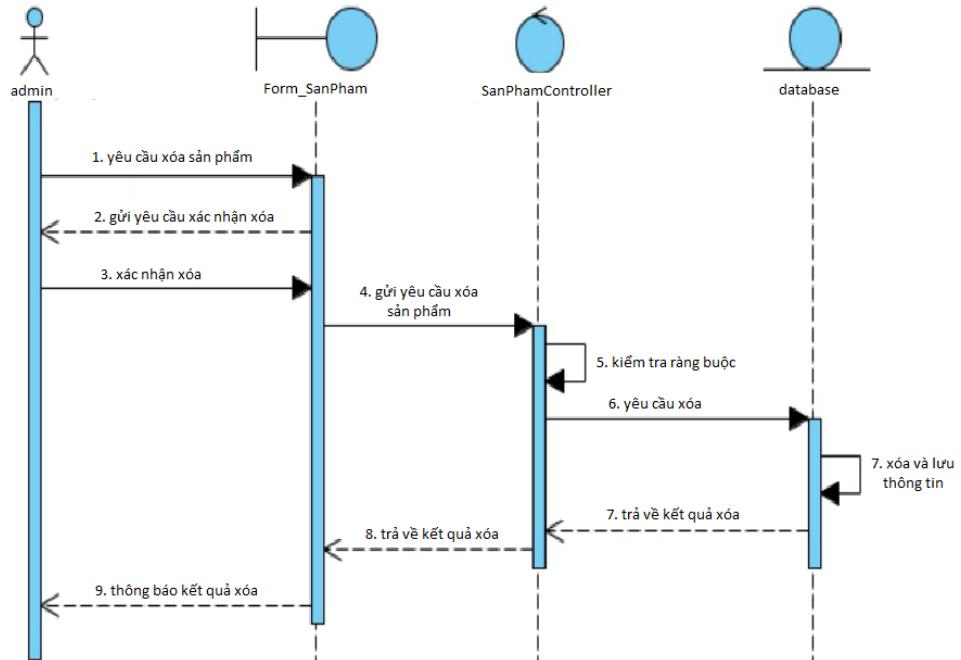
3.4.2. Sơ đồ tuần tự sửa thông tin cá nhân



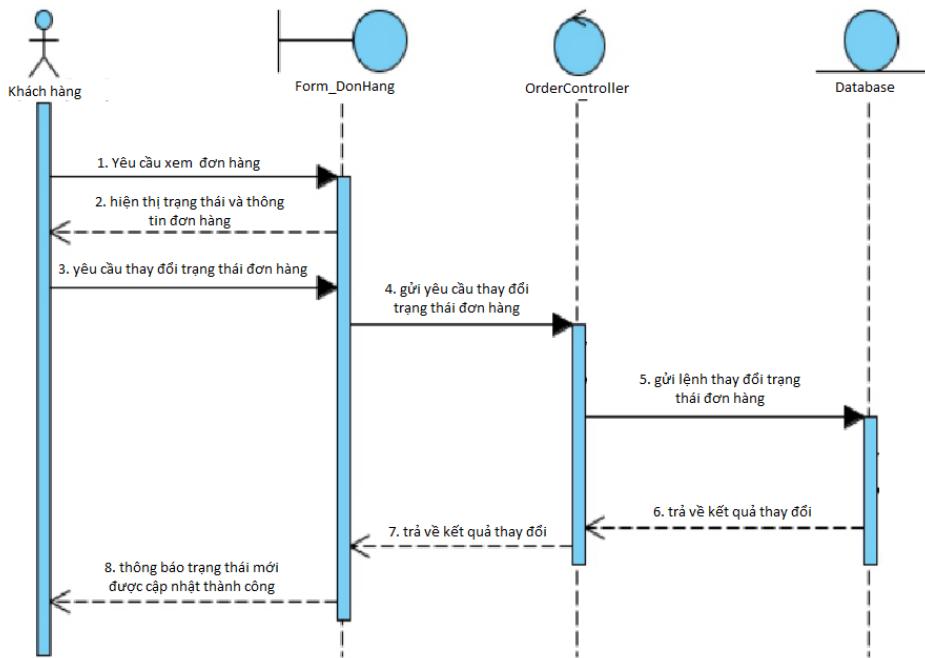
3.4.3. Sơ đồ tuần tự sửa, thêm sản phẩm



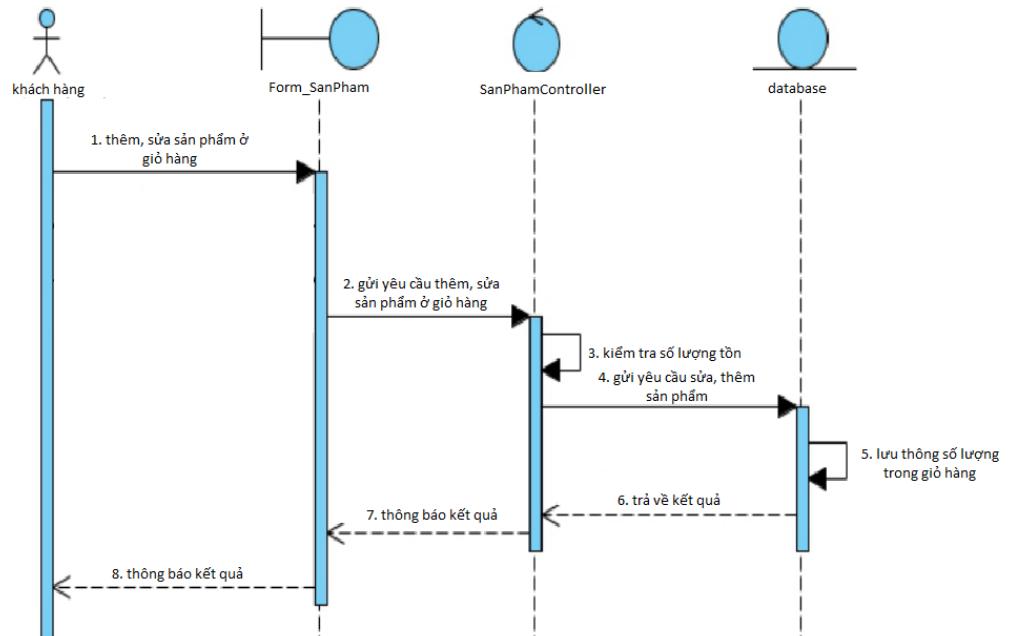
3.4.4. Sơ đồ tuần tự xóa sản phẩm



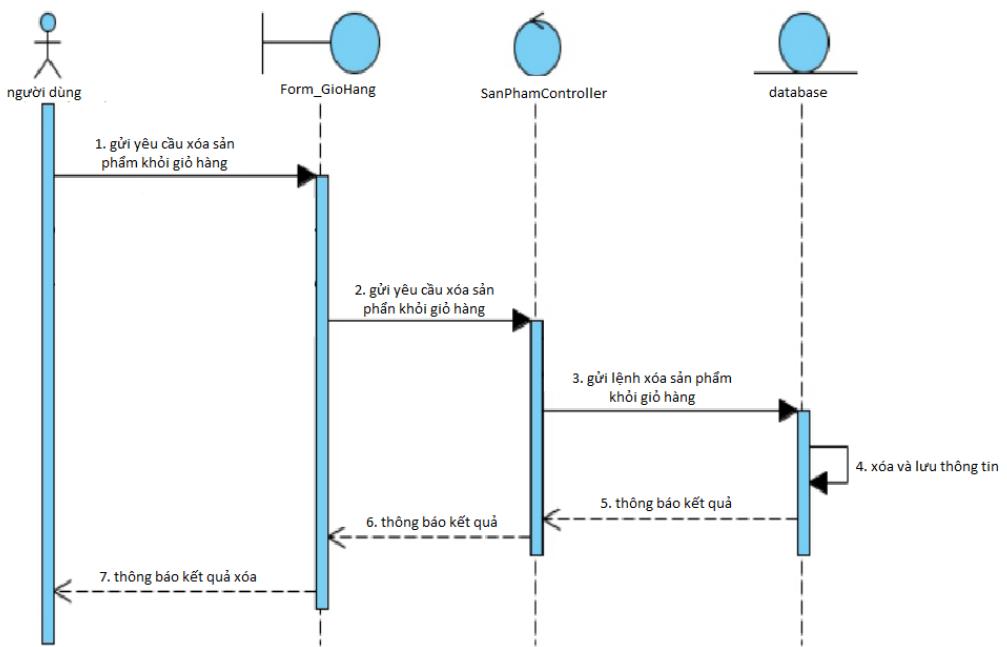
3.4.5. Sơ đồ tuần tự theo dõi đơn hàng



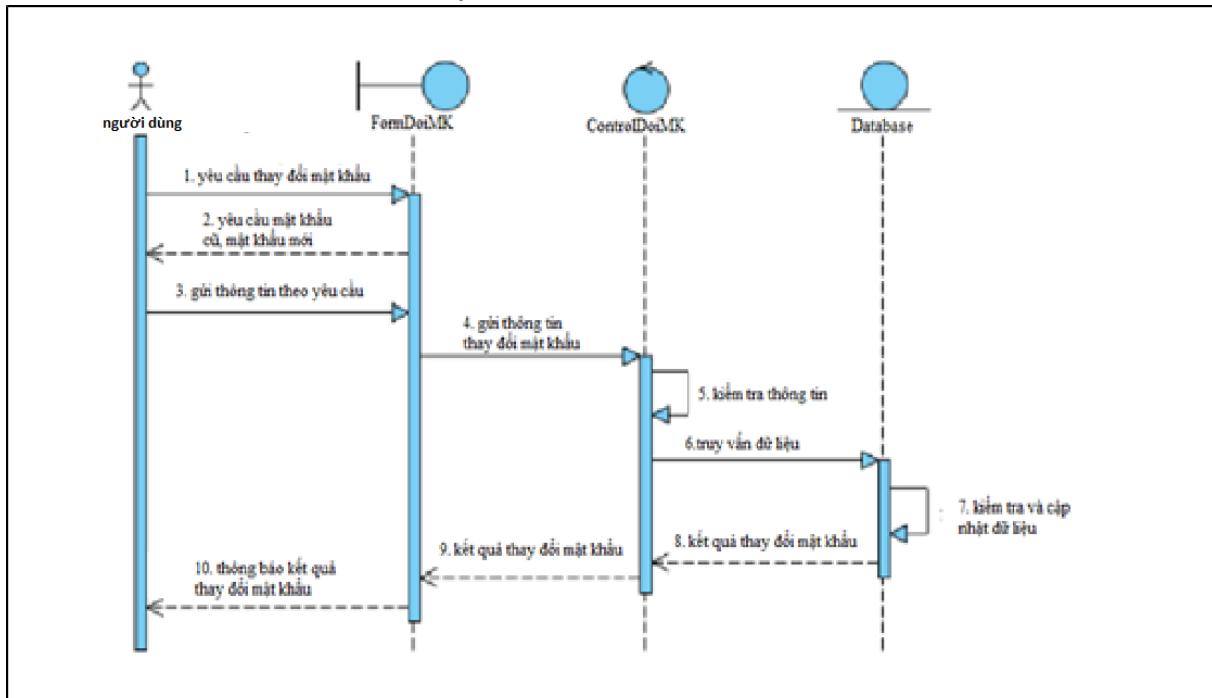
3.4.6. Sơ đồ tuần tự sửa, thêm sản phẩm vào giỏ hàng



3.4.7. Sơ đồ tuần tự xóa sản phẩm khỏi giỏ hàng

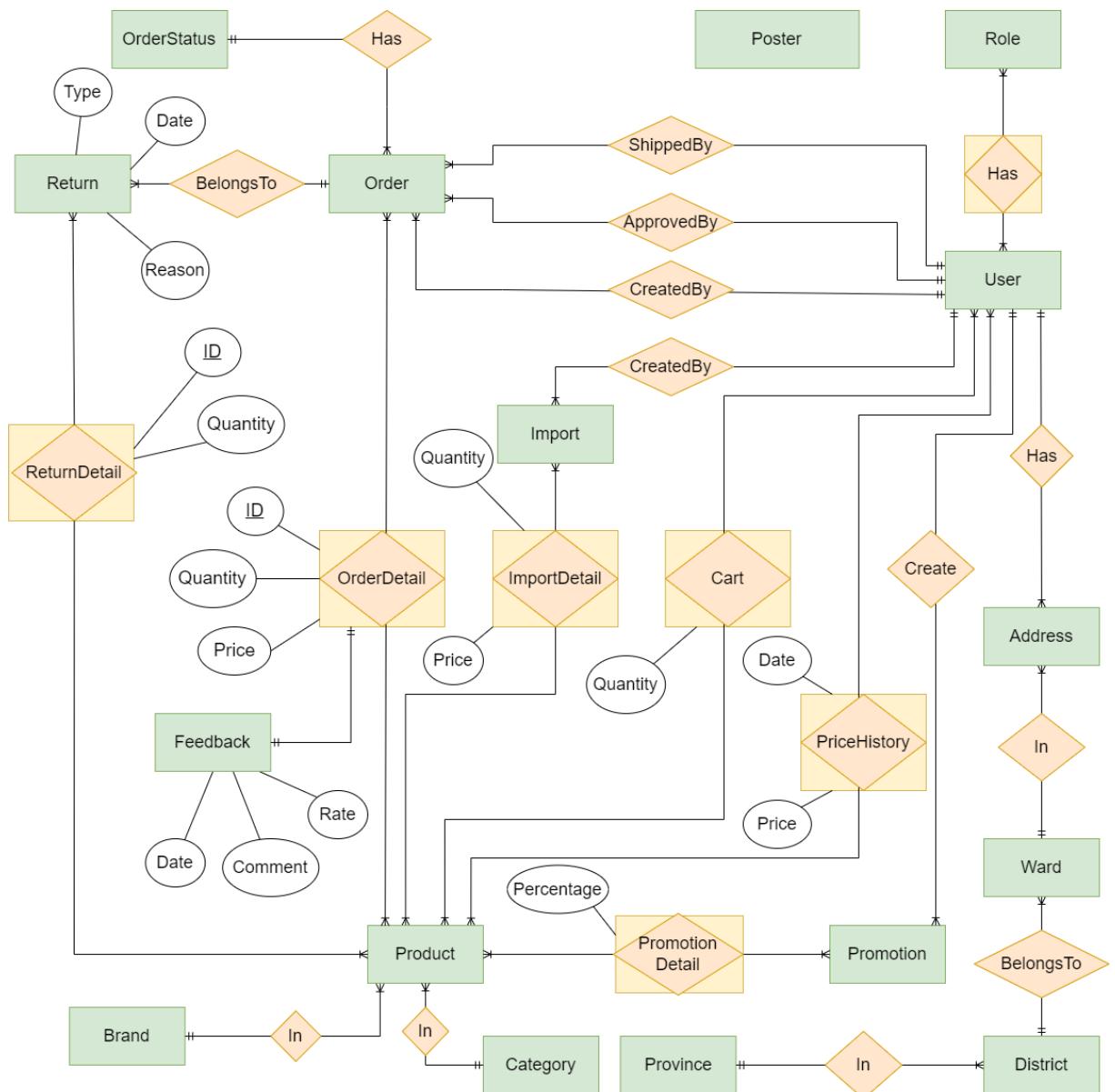


3.4.8. Sơ đồ tuần tự thay đổi mật khẩu



3.5. Thiết kế dữ liệu

3.5.1. Mô hình thực thể quan hệ ER



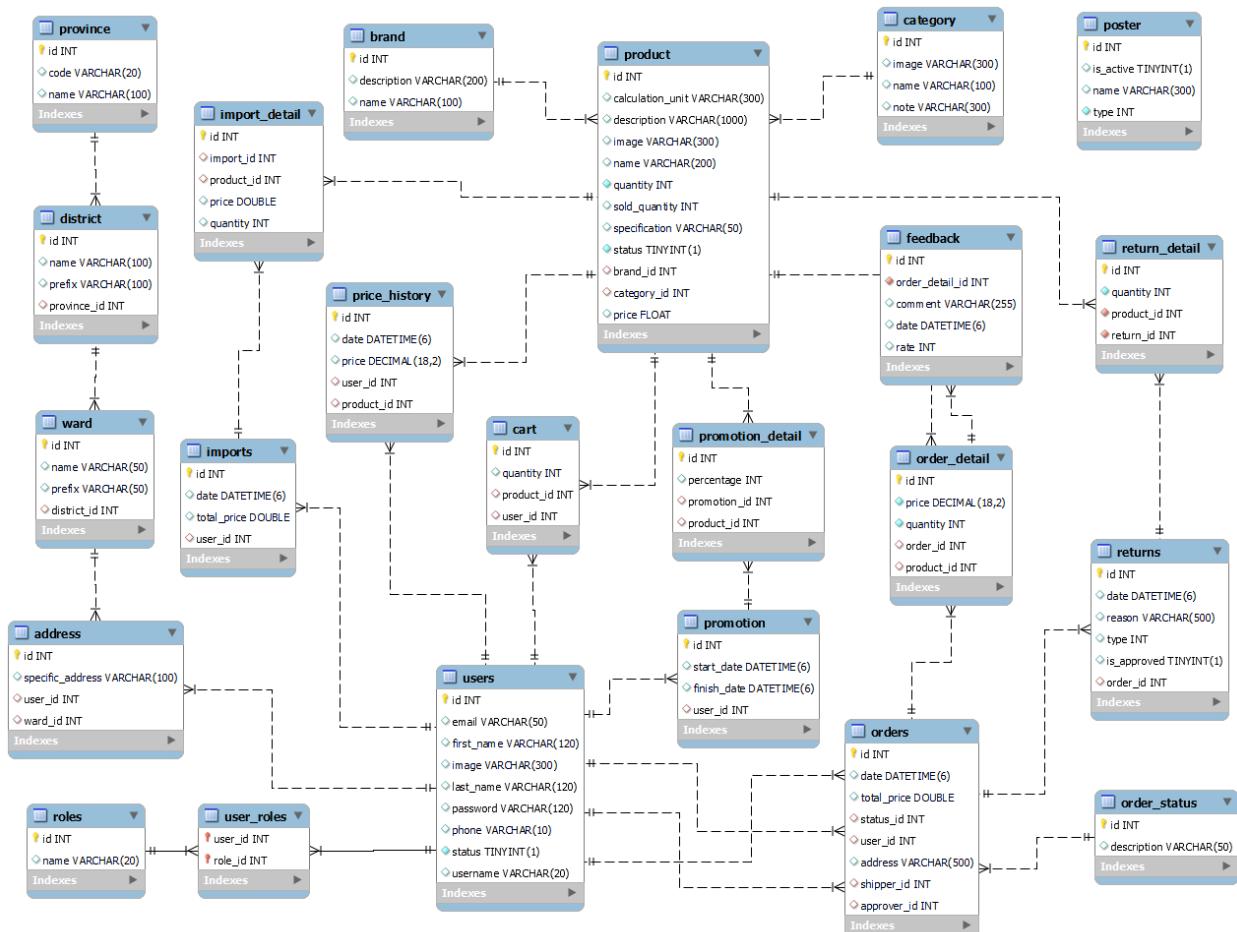
Hình 3.14. Mô hình thực thể quan hệ ER

3.5.2. Lược đồ quan hệ

- **Province** (ProvinceID, Name, Code)
 - **District** (DistrictID, Name, Prefix, *ProvinceID*)
 - **Ward** (WardID, Name, Prefix, *DistrictID*)
 - **Poster** (PosterID, Is_active, Name, Type)
 - **User** (UserID, Username, Email, FirstName, LastName, Phone, Password, Status)
 - **Role** (RoleID, Name)
 - **User_Role** (UserID, RoleID)
 - **Address** (AddressID, SpecificAddress, *WardID*, *UserID*)
 - **Brand** (BrandID, Name, Description)

- **Category** (CategoryID, Image, Name, Note)
- **Product** (ProductID, Name, CalculationUnit, Description, Discount, Image, Price, Quantity, SoldQuantity, Specification, Status, *BrandID*, *CategoryID*)
- **Cart** (CartID, Quantity, *ProductID*, *UserID*)
- **OrderStatus** (StatusID, Description)
- **Order** (OrderID, Date, TotalPrice, Address, *StatusID*, *UserID*, *ShipperID*, *ApproverID*)
- **OrderDetail** (OrderDetailID, Price, Quantity, *OrderID*, *ProductID*)
- **Import** (ImportID, Date, TotalPrice, *UserID*)
- **ImportDetail** (ImportDetailID, Price, Quantity, *ImportID*, *ProductID*)
- **Feedback** (FeedbackID, Comment, Rate, Date, *OrderDetailID*)
- **PriceHistory** (PriceHistoryID, Date, Price, *ProductID*, *UserID*)
- **Promotion** (PromotionID, StartDate, FinishDate, *UserID*)
- **PromotionDetail** (PromotionDetailID, Percentage, *PromotionID*, *ProductID*)
- **Return** (ReturnID, Date, Reason, Type, *OrderID*)
- **ReturnDetail** (ReturnDetailID, Quantity, *ReturnID*, *ProductID*)

3.5.3. Hiện thực hóa



Hình 3.15. Sơ đồ cơ sở dữ liệu trên MySQL

3.5.4. Mô tả các bảng dữ liệu

STT	Tên bảng	Diễn giải
1	User	Thông tin về người dùng
2	Roles	Thông tin về các vai trò
3	User_Roles	Thông tin về vai trò của từng user
4	Province	Thông tin về các tỉnh thành
5	District	Thông tin về quận, huyện
6	Ward	Thông tin về xã, phường
7	Address	Thông tin về địa chỉ của người dùng
8	Category	Thông tin về các danh mục sản phẩm
9	Brand	Thông tin về các thương hiệu
10	Poster	Thông tin về các poster
11	Cart	Thông tin về sản phẩm trong giỏ hàng
12	Order	Thông tin về đơn hàng của khách hàng
13	OrderDetail	Thông tin chi tiết của đơn hàng
14	Import	Thông tin về phiếu nhập hàng
15	ImportDetail	Thông tin chi tiết phiếu nhập
16	Product	Thông tin về các sản phẩm của cửa hàng
17	Feedback	Thông tin về đánh giá sản phẩm của khách hàng
18	OrderStatus	Thông tin về các trạng thái của đơn hàng
19	Return	Thông tin về phiếu trả hàng của khách
20	ReturnDetail	Thông tin chi tiết phiếu trả
21	Promotion	Thông tin đợt khuyến mãi
22	Promotion	Thông tin chi tiết trong đợt khuyến mãi
23	PriceHistory	Thông tin lịch sử giá của một sản phẩm

CHƯƠNG 4. THIẾT KẾ PHẦN MỀM**4.1. Front-end (Forms)****4.1.1. Windows Application****4.1.1.1. Login Form**

Username:

admin

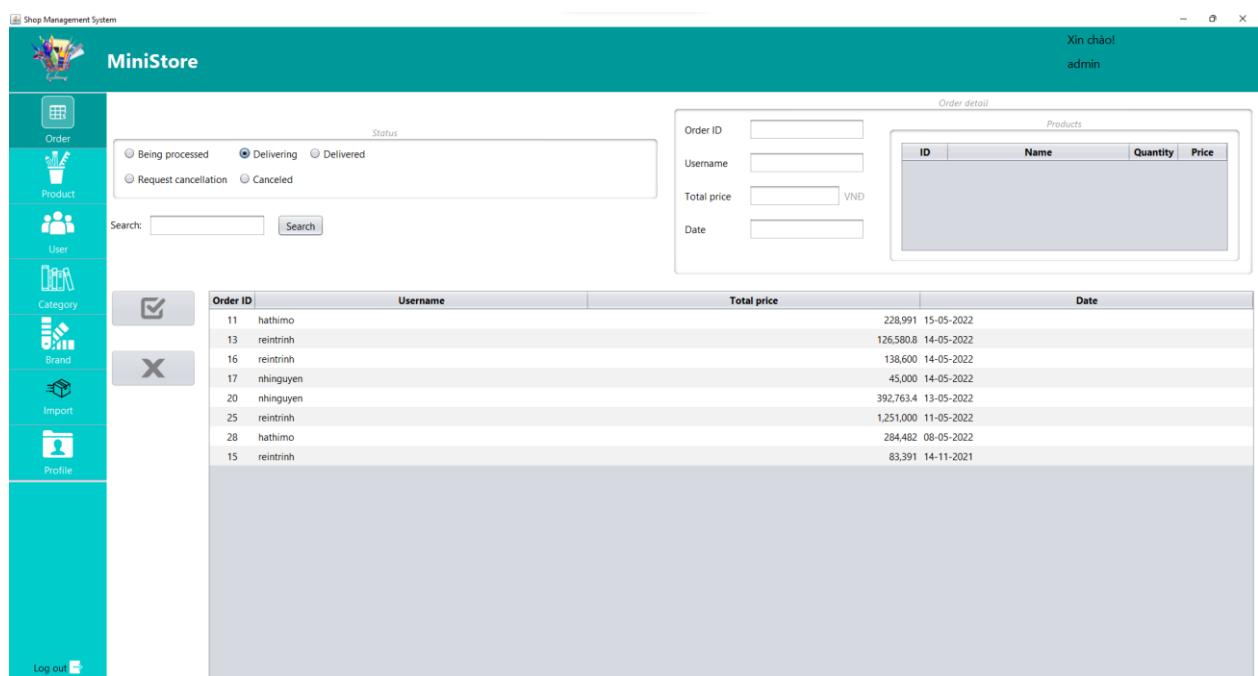
Password:

**LOGIN***Hình 4.1. Login Form*

- Nhiệm vụ: dành cho người dùng thực hiện chức năng login vào app

STT	Chức năng	API	Xử lý response
1	Người dùng nhập username, password và click nút LOGIN	Sign in	Kiểm tra role của user, nếu role là admin thì sẽ chuyển sang màn hình chính, nếu không phải admin thì sẽ thông báo lỗi

4.1.1.2. Order Form



Hình 4.2. Order Form

- Nhiệm vụ: dành cho admin thực hiện chức năng xem, chấp nhận/huỷ đơn hàng của khách

STT	Chức năng	API	Xử lý response
1	Admin click chọn các radiobutton để chuyển qua các đơn hàng với trạng thái khác nhau	Lấy thông tin các đơn hàng theo trạng thái	Response là danh sách đơn hàng với trạng thái tương ứng, form sẽ refresh dữ liệu trong table
2	Admin click vào 1 đơn hàng trong bảng đơn hàng thì sẽ xem được chi tiết đơn hàng trong khung Order Detail	Lấy thông tin 1 đơn hàng	Form refresh dữ liệu order detail với orderId tương ứng trong khung order detail
3	Đối với các đơn hàng có trạng thái Being proccesed và Request Cancellation thì hai nút Accept và Deny sẽ được enable để click chọn	Thay đổi trạng thái đơn hàng	Refesh lại danh sách đơn hàng

4.1.1.3. Product Form

The screenshot displays the 'Product Form' within the 'MiniStore' application. The left sidebar contains navigation icons for Order, Product, User, Category, Brand, Import, and Profile. The top right corner shows a greeting ('Xin chào! admin') and a 'Logout' button. The main content area is titled 'Product information' and contains fields for Product ID (36), Name (Túi đựng bút vải dinh huy hiệu Maimalbaby), Unit (túi), Specification, Price (86,000), Discount (0%), and Quantity (50). The 'Sold quantity' field is set to 12. To the right, there's a detailed description of the product: 'Đúng đắn một chút rủi minh ngầm TÚI BÚT!!! Ủi chui túi bút xanh nung nịnh như này hỏng chỉ là ngầm thử đâu mà còn phả hot le lun mê! Đì học hay đì làm đem túi bút là tiện nhất rú. Đúng đắn dung học tập hay đồ mỹ phẩm be bé xinh xinh vừa nhu in lun à!'. There are dropdowns for 'Category' (Dụng cụ học tập) and 'Brand' (Điểm 10). An 'Image' section with a 'Choose file' button is shown. Below the form is a table listing various products with their details. A search bar and several action buttons ('Feedback', 'Export', '+ Add', 'Modify', 'Remove') are located at the bottom.

ID	Name	Category	Brand	Unit	Price	Remain Quantity
49	Set 20 chiếc bút bi mực xanh	Bút viết	Thiên Long	hộp	150,000	5
48	New Product	Máy tính	Tomy	chiếc	80,000	5
47	Laptop Dell	Thiết bị điện tử	Flexoffice	chiếc	10,000	5
46	Laptop	Thiết bị điện tử	Tomy	chiếc	10,000	5
45	test new	Bút viết	Flexoffice	cây	8,300	29
40	test new bút	Bút viết	Flexoffice	cây	8,300	29
36	Túi đựng bút vải dinh huy hiệu Mai... Dụng cụ học tập	Điểm 10	túi	86,000	36	
35	Hộp bút vải có khía mặt mă 20k...	Dụng cụ học tập	Điểm 10	hộp	265,980	29
34	Chổi vệ sinh bàn phím	Dụng cụ văn phòng	SAHAKA	cây	96,720	39
33	Đè Tản Nhiệt 4 Quạt	Thiết bị văn phòng	Logitech	miếng	381,600	19
32	Đè Tản Nhiệt 4 Quạt	Thiết bị văn phòng	Logitech	miếng	381,600	19
31	Chuột máy tính không dây Logitec...	Thiết bị văn phòng	Logitech	cái	352,800	48
30	Chuột máy tính không dây Logitec...	Thiết bị văn phòng	Logitech	cái	352,800	48
29	Gôm Pentel Nhô 03 - Loại 1	Dụng cụ văn phòng	Thiên Long	cục	6,400	28
28	Gôm Pentel Nhô 03 - Loại 1	Dụng cụ văn phòng	Thiên Long	cục	6,400	28
27	Kéo lông 35cm - K20	Băng leo - Dao kéo - Hộ dán	Điểm 10	cây	34,200	70
26	Máy tính VINACAL 570 ES PLUS II -	Máy tính	Vinacal	chiếc	863,000	69

Hình 4.3. Product Form

- Nhiệm vụ: dành cho admin thực hiện chức năng xem, xoá, sửa sản phẩm

STT	Chức năng	API	Xử lý response
1	Admin click chọn các sản phẩm trong bảng để xem chi tiết sản phẩm	Lấy thông tin sản phẩm	Response là thông tin sản phẩm, lấy thông tin để hiển thị lên trong khung Product information
2	Admin click Add để thêm mới sản phẩm		Form clear dữ liệu trong khung Product information
3	Admin click Edit để sửa sản phẩm có sẵn		Form enable các field để admin chỉnh sửa
4	Sau khi nhập dữ liệu thêm mới/chỉnh sửa sản phẩm, admin click OK	Thêm sản phẩm/Cập nhật sản phẩm	Form hiện thông báo kết quả, success hoặc failed
5	Admin nhập dữ liệu vào field tìm kiếm để lọc sản phẩm		Table lọc sản phẩm tương ứng

4.1.1.4. Price History Form

The screenshot shows a web-based application window titled "Price History". At the top, there are four input fields: "Product Name" (New Product), "Min Price" (40000.0), "Max Price" (80000.0), and "Average Price" (57777.777). Below these fields is a table with four columns: Date, Username, Email, and Price. The table contains ten rows of data, all belonging to the user "admin" and showing price values of 80000.0, 50000.0, 60000.0, 60000.0, 60000.0, 60000.0, 60000.0, 60000.0, 50000.0, and 40000.0 respectively.

Date	Username	Email	Price
09-11-2022 04:30	admin	admin@gmail.com	80000.0
09-11-2022 04:30	admin	admin@gmail.com	50000.0
09-11-2022 04:16	admin	admin@gmail.com	60000.0
09-11-2022 04:13	admin	admin@gmail.com	60000.0
09-11-2022 04:09	admin	admin@gmail.com	60000.0
09-11-2022 04:08	admin	admin@gmail.com	60000.0
09-11-2022 03:41	admin	admin@gmail.com	60000.0
09-11-2022 03:40	admin	admin@gmail.com	50000.0
15-05-2022 00:00	admin	admin@gmail.com	40000.0

Hình 4.4. Price History Form

- Nhiệm vụ: dành cho admin xem lịch sử cập nhật giá sản phẩm

STT	Chức năng	API	Xử lý response
1	Admin click chọn lần cập nhật trong table để xem chi tiết	Lấy thông tin lịch sử giá	Response là thông tin cập nhật giá, lấy thông tin để hiện lên chi tiết

4.1.1.5. Feedback Form

The screenshot shows a feedback form for a product. The product details are:

- Product Name: Gôm Pentel Nhỏ 03 - Loại 1
- Date: 10-11-2022 22:36
- Username: reintrinh
- Rate: 4
- Comment: Đóng gói cẩn thận, shop giao hàng nhanh, nhưng shipper thái độ kém quá, trừ 1 sao vì shipper nhé

Below the form is a table of previous feedbacks:

Date	Username	Comment	Rate
10-11-20...	anhyeuem	Sản phẩm đúng với mô tả. Shop tư vấn và hỗ trợ nhiệt tình! Vote shop 5 sao nhé!	5
10-11-20...	reintrinh	Đóng gói cẩn thận, shop giao hàng nhanh, nhưng shipper thái độ kém quá, trừ 1 sao vì sh...	4

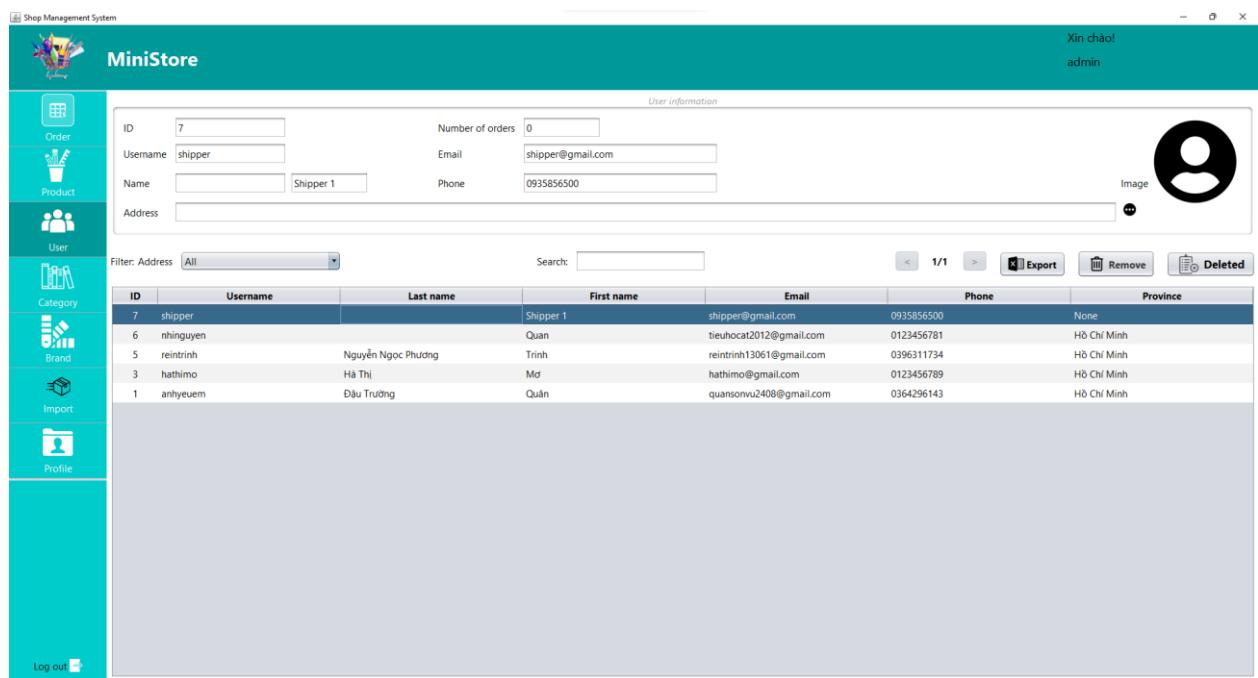
Average Rate: 4.5

Hình 4.5. Feedback Form

- Nhiệm vụ: dành cho admin xem đánh giá của sản phẩm

STT	Chức năng	API	Xử lý response
1	Admin click chọn feedback trong table để xem chi tiết	Lấy thông tin feedback	Response là thông tin feedback, lấy thông tin để hiện lên chi tiết

4.1.1.6. User Form



Hình 4.6. User Form

- Nhiệm vụ: dành cho admin thực hiện chức năng xem, xoá, sửa user

STT	Chức năng	API	Xử lý response
1	Admin click chọn các user trong bảng để xem chi tiết sản phẩm	Lấy thông tin user	Response là thông tin user, lấy thông tin để hiển thị lên trong khung User information
2	Admin click Remove để xoá user có sẵn	Cập nhật trạng thái user	Form refresh tab
3	Admin nhập dữ liệu vào field tìm kiếm để lọc user		Table lọc user tương ứng

4.1.1.7. Category Form

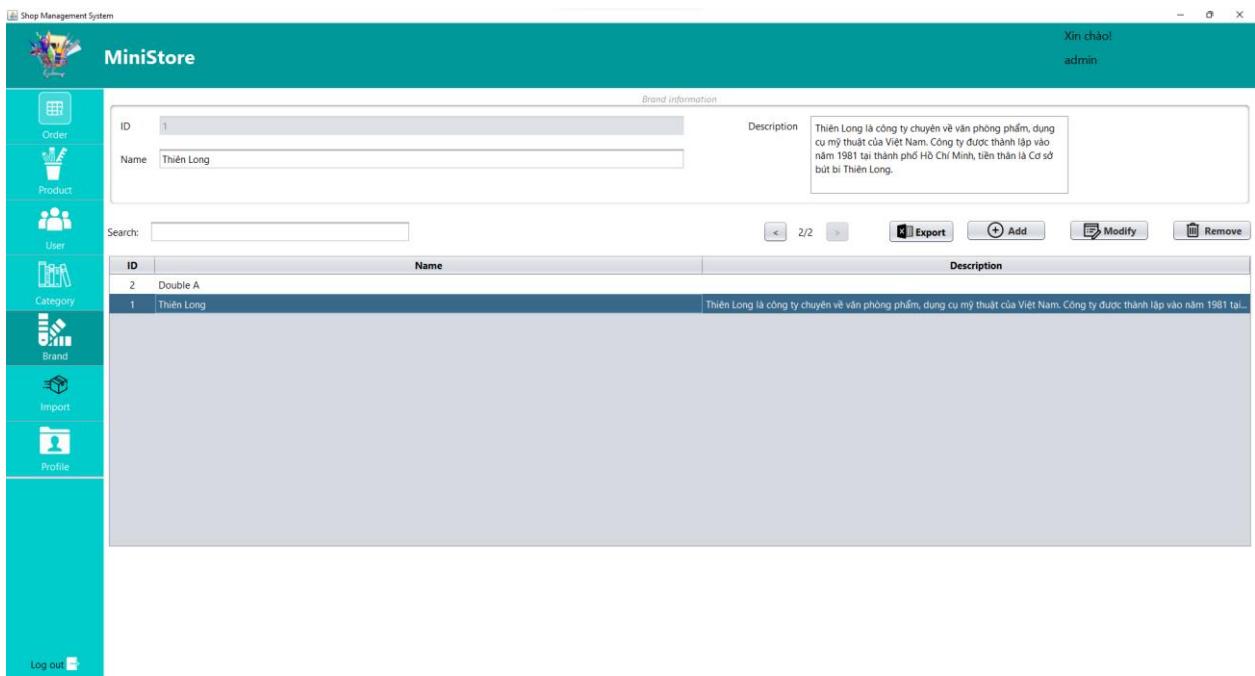
ID	Name	Note	Image
24	Gỗm tẩy chất lượng tốt	Gỗm tẩy chất lượng tốt, giá cả vừa phải	defaul.png
23	Máy tính cao cấp	Máy tính nước ngoài chất lượng bền bỉ, nhiều tính năng	defaul.png
22	Bút cao cấp	Bút viết chất lượng cao, giá cả mắc	211118231826tai xuong.jpg
21	Ăn vật văn phòng		anVat.jpg
20	Dụng cụ đóng dấu		dungCuDongDau.jpg
19	Dụng cụ vệ sinh văn phòng		DungCuVSVP.jpg
18	Kế - Rổ		keRo.jpg
17	Mực in - Bơm mực		mucIn.jpg
16	Mỹ thuật		myThuat.jpg
15	Thiết bị điện tử		thietBiDienTu.jpg
14	Thiết bị văn phòng		thietBiVP.jpg
13	Phụ kiện điện thoại		PKdienThoai.jpg
12	Giấy in ảnh - in màu chất lượng cao		giaInAnh.jpg
11	Pin các loại		pin.jpg
10	Bảo hộ lao động		baoHold.jpg
9	Bảng văn phòng		bangVP.jpg
8	Máy tính		mayTinh.png

Hình 4.7. Category Form

- Nhiệm vụ: dành cho admin thực hiện chức năng xem, xoá, sửa user

STT	Chức năng	API	Xử lý response
1	Admin click chọn thẻ loại trong bảng để xem chi tiết thẻ loại	Lấy thông tin thẻ loại	Response là thông tin sản phẩm, lấy thông tin để hiện lên trong khung Category information
2	Admin click Add để thêm mới thẻ loại		Form clear dữ liệu trong khung Category information
3	Admin click Edit để sửa thẻ loại có sẵn		Form enable các field để admin chỉnh sửa
4	Sau khi nhập dữ liệu thêm mới/chỉnh sửa thẻ loại, admin click OK	Thêm sản phẩm/Cập nhật thẻ loại	Form hiện thông báo kết quả, success hoặc failed
5	Admin nhập dữ liệu vào field tìm kiếm để lọc thẻ loại		Table lọc thẻ loại tương ứng

4.1.1.8. Brand Form



Hình 4.8. Brand Form

- Nhiệm vụ: dành cho admin thực hiện chức năng xem, xoá, sửa nhãn hiệu

STT	Chức năng	API	Xử lý response
1	Admin click chọn các nhãn hiệu trong bảng để xem chi tiết	Lấy thông tin nhãn hiệu	Response là thông tin nhãn hiệu, lấy thông tin để hiện lên trong khung Brand information
2	Admin click Add để thêm mới nhãn hiệu		Form clear dữ liệu trong khung Brand information
3	Admin click Edit để sửa nhãn hiệu có sẵn		Form enable các field để admin chỉnh sửa
4	Sau khi nhập dữ liệu thêm mới/chỉnh sửa, admin click OK	Thêm sản phẩm/Cập nhật nhãn hiệu	Form hiện thông báo kết quả, success hoặc failed

5	Admin nhập dữ liệu vào field tìm kiếm để lọc		Table lọc brand tương ứng
---	--	--	---------------------------

4.1.1.9. Import List

ID	Username	Total price	Date
6	admin	9,300,000	11-11-2022 01:28
5	admin	9,300,000	11-11-2022 01:25
4	admin	9,300,000	11-11-2022 01:23
3	admin	114,000	09-11-2022 01:23
2	admin	166,000	09-11-2022 01:11
1	admin	9,300,000	08-11-2022 11:33

Hình 4.9. Import List

- Nhiệm vụ: dành cho admin thực hiện chức năng xem, tạo mới phiếu nhập hàng

STT	Chức năng	API	Xử lý response
1	Admin click chọn các phiếu nhập trong bảng để xem chi tiết phiếu nhập	Lấy thông tin chi tiết phiếu nhập	Response là thông tin chi tiết phiếu nhập, lấy thông tin để hiển thị
2	Admin click Add để thêm mới phiếu nhập		Create Import Form hiển thị
5	Admin chọn khoảng thời gian và nhấn Apply để lọc phiếu nhập		Table lọc phiếu nhập tương ứng

4.1.1.10. Create Import Form

The screenshot shows a software application window titled "New Import".

- Staff information:** Contains fields for User ID (9), Name (admin), and Email (admin@gmail.com).
- Import detail:** A table with columns ID, Name, Quantity, and Price. It currently has one row: ID 1, Name Bút gel 1 viên bi, Quantity 26, Price 8,300.
- Product List:** A table showing a list of products with columns ID, Name, Category, Brand, Unit, Price, and Remain Quantity. The list includes items like Bút gel 1 viên bi, Bút gel 2 viên bi, etc.
- Buttons:** Choose, Edit, Delete, and a Create button at the bottom.

Hình 4.10. Create Import Form

- Nhiệm vụ: dành cho admin thực hiện chức năng thêm chi tiết phiếu nhập để tạo mới phiếu nhập

STT	Chức năng	API	Xử lý response
1	Admin click chọn các sản phẩm trong bảng Product list → click Choose		Form kiểm tra sản phẩm này đã thêm vào danh sách chi tiết phiếu nhập chưa, nếu chưa thì hiển thị Import Product Form, nếu rồi thì báo lỗi
2	Admin click Edit để chỉnh sửa chi tiết phiếu nhập		Hiển thị Import Product Form

3	Admin click Delete để xoá chi tiết phiếu nhập		Form refresh list chi tiết phiếu nhập
4	Admin click Create để tạo phiếu nhập	Thêm phiếu nhập	Form hiện thông báo kết quả, success hoặc failed

x

Import Product

ID

Name

Amount Unit

Price đ

 Save

 Cancel

Hình 4.11. Import Product Form

- Nhiệm vụ: dành cho admin thực hiện chức năng thêm sản phẩm vào danh sách chi tiết phiếu nhập

STT	Chức năng	API	Xử lý response
1	Admin click Save để thêm sản phẩm		Thêm sản phẩm vào list chi tiết phiếu nhập

4.1.1.11. Profile Form

Hình 4.12. Profile Form

- Nhiệm vụ: dành cho admin thực hiện chức năng chỉnh sửa thông tin cá nhân

STT	Chức năng	API	Xử lý response
1	Admin sửa các thông tin cá nhân → click Save	Cập nhập thông tin user	Form hiển thị thông báo api trả về, success hoặc failed
2	Admin click Change password để chỉnh sửa mật khẩu		Hiển thị Change password Form
3	Admin click Change address để chỉnh sửa địa chỉ		Hiển thị Change address Form

4.1.1.12. Address Form

ID	Specific address	Commune	District	Province
6	1/15c, Tô Ký	Tân Chánh Hiệp	Quận 12	Hồ Chí Minh

Hình 4.13. Address Form

- Nhiệm vụ: dành cho admin thực hiện chức năng xem, xoá, sửa địa chỉ

STT	Chức năng	API	Xử lý response
1	Admin click chọn các địa chỉ trong bảng để xem chi tiết	Lấy thông tin địa chỉ	Response là thông tin địa chỉ, lấy thông tin để hiện lên trong khung Address information
2	Admin click Add để thêm mới địa chỉ		Form clear dữ liệu trong khung Address information
3	Admin click Edit để sửa địa chỉ có sẵn		Form enable các field để admin chỉnh sửa
4	Sau khi nhập dữ liệu thêm mới/chỉnh sửa, admin click OK	Thêm sản phẩm/Cập nhật địa chỉ	Form hiện thông báo kết quả, success hoặc failed

4.1.1.13. Change Password Form

The screenshot shows a window titled "Change your password". Inside, there are three text input fields labeled "Old password", "New password", and "Verify new password", each accompanied by a small icon of an eye with a slash, likely for password visibility. At the bottom are two buttons: "Update password" and "Cancel".

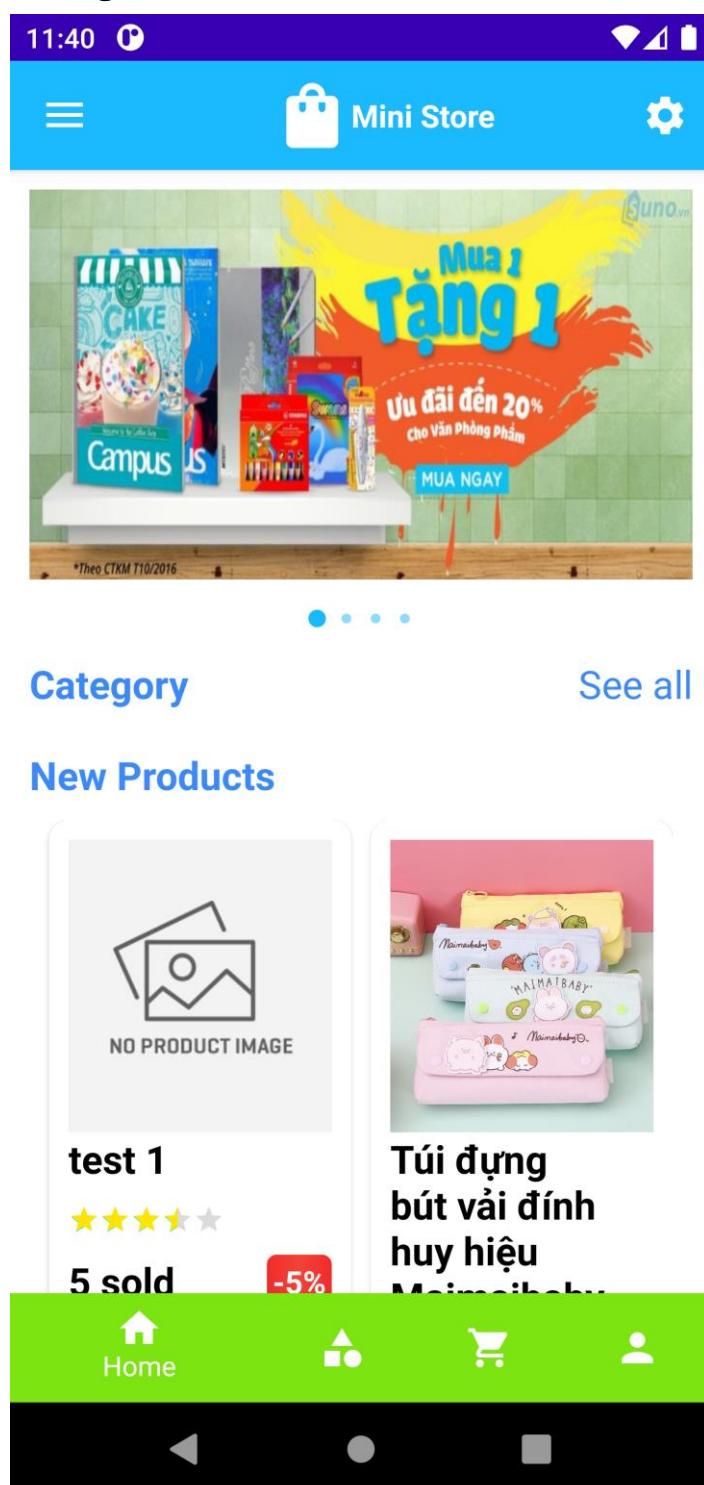
Hình 4.14. Change Password Form

- Nhiệm vụ: dành cho admin thực hiện chức năng đổi password

STT	Chức năng	API	Xử lý response
1	Admin nhập thông tin và click Update password	Cập nhật thông tin user	Form hiển thị thông báo trả về từ api, success hoặc failed

4.1.2. Mobile Application

4.1.2.1. Trang chủ (Home)



Hình 4.15. Giao diện trang chủ

- Nhiệm vụ: dành cho khách hàng xem các sản phẩm của shop

STT	Chức năng	API	Xử lý response
-----	-----------	-----	----------------

1	Hiển thị sản phẩm mới, sản phẩm bán chạy, tất cả sản phẩm, thanh menu	Lấy danh sách sản phẩm mới Lấy danh sách sản phẩm bán chạy Lấy danh sách tất cả sản phẩm	Form hiển thị thông tin các sản phẩm lên màn hình
2	Khách hàng nhập thông tin tìm kiếm vào ô tìm kiếm để tìm sản phẩm	Tìm kiếm sản phẩm	Hiển thị form với các sản phẩm chứa từ khoá
3	Khách hàng nhấn icon giỏ hàng để chuyển đến màn hình giỏ hàng		Hiển thị form giỏ hàng
4	Khách hàng nhấn icon thêm sản phẩm vào giỏ	Thêm sản phẩm vào giỏ	Form cập nhật số sản phẩm trong giỏ hàng đang hiển thị

4.1.2.2. Danh mục sản phẩm (Category)



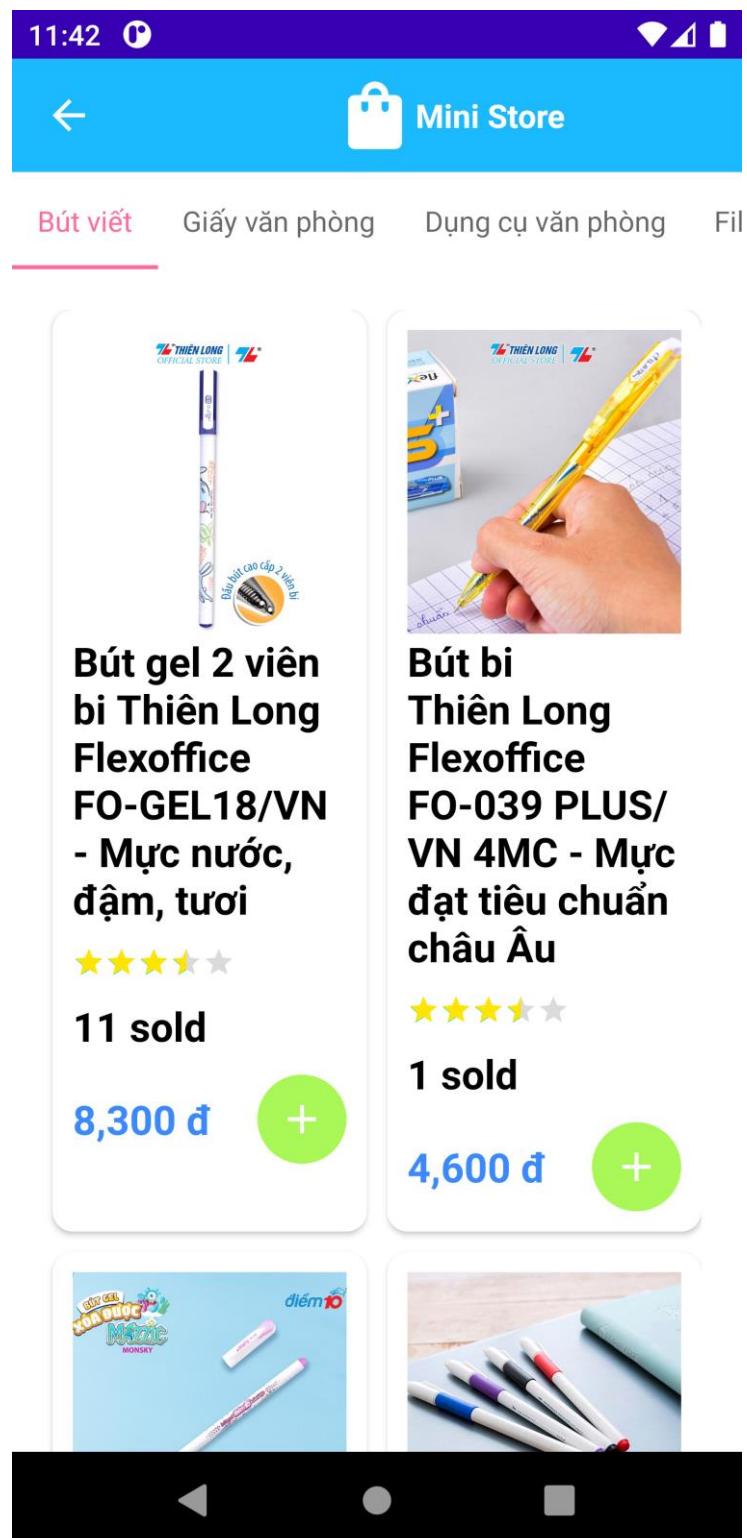
Hình 4.16. Giao diện danh mục sản phẩm

- Nhiệm vụ: dành cho khách hàng xem danh sách các thể loại của shop

STT	Chức năng	API	Xử lý response
-----	-----------	-----	----------------

1	Hiển thị danh sách thể loại	Lấy danh sách thể loại	Form hiển thị thông tin các thể loại lên màn hình
2	Người dùng click vào thể loại để xem các sản phẩm thuộc thể loại đó		Hiển thị Product Form

4.1.2.3. Sản phẩm (Product)



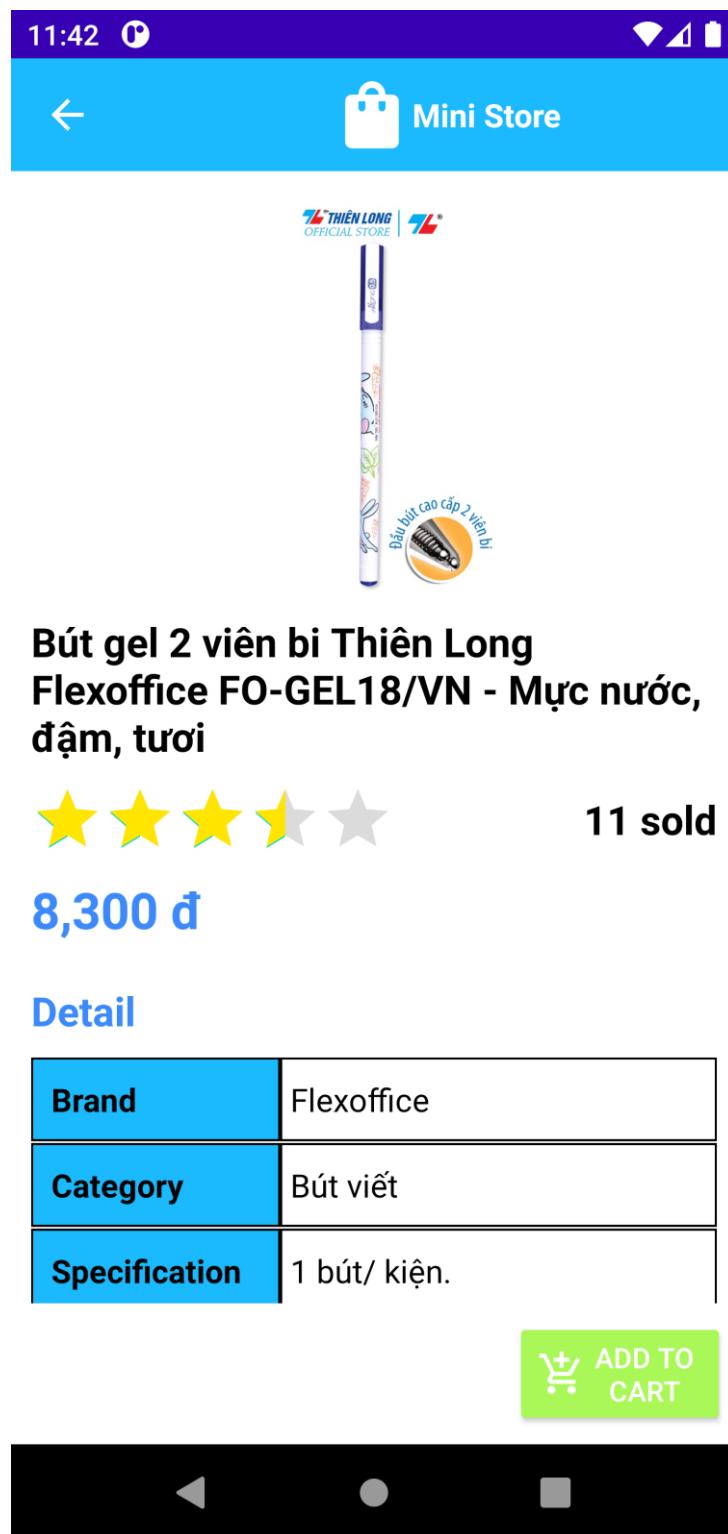
Hình 4.17. Giao diện sản phẩm

- Nhiệm vụ: dành cho khách hàng xem danh sách các sản phẩm theo thể loại

STT	Chức năng	API	Xử lý response
-----	-----------	-----	----------------

1	Hiển thị danh sách sản phẩm theo tab thẻ loại	Lấy danh sách sản phẩm theo thẻ loại	Form hiển thị thông tin các sản phẩm lên màn hình
2	Người dùng click vào nút thêm sản phẩm vào giỏ	Thêm sản phẩm vào giỏ	Form cập nhật số lượng giỏ hàng trên màn hình

4.1.2.4. Chi tiết sản phẩm (Detail)



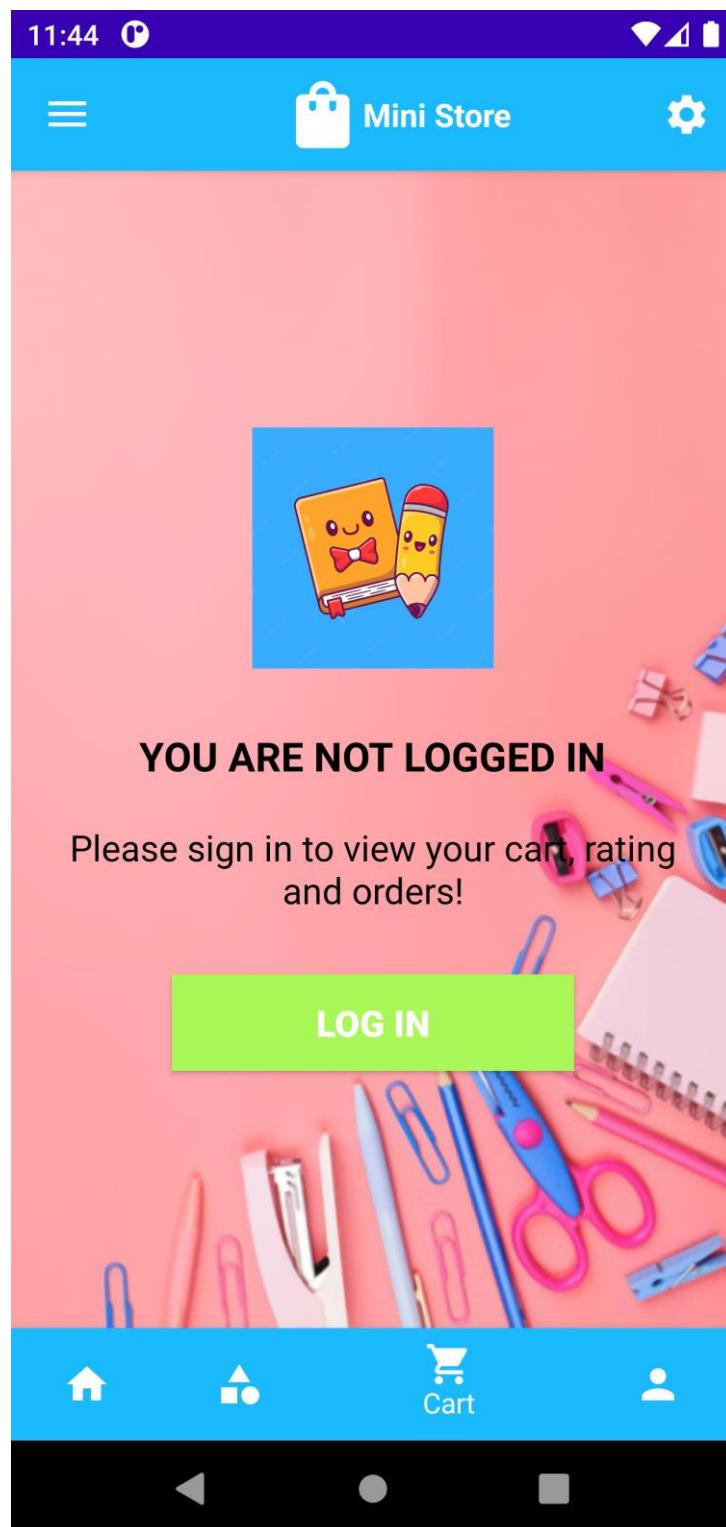
Hình 4.18. Giao diện chi tiết sản phẩm

- Nhiệm vụ: dành cho khách hàng xem chi tiết sản phẩm

STT	Chức năng	API	Xử lý response

1	Hiển thị chi tiết sản phẩm	Lấy thông tin chi tiết của một sản phẩm	Form hiển thị thông tin chi tiết của sản phẩm lên màn hình
2	Người dùng click vào Add to cart để thêm sản phẩm vào giỏ	Thêm sản phẩm vào giỏ	Form cập nhật số lượng giỏ hàng

4.1.2.5. Giỏ hàng (Cart)

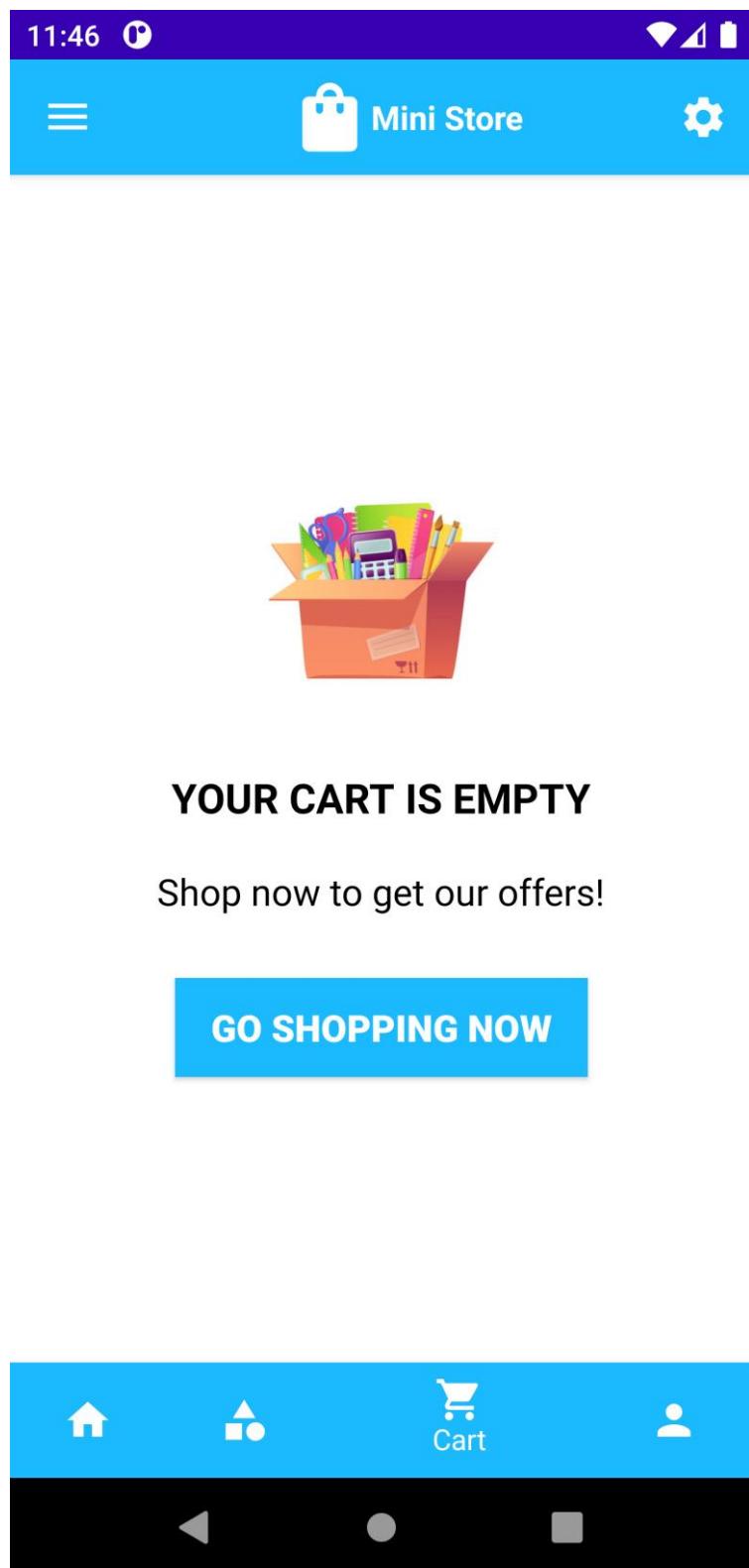


Hình 4.19. Giao diện giỏ hàng (1)

- Nhiệm vụ: màn hình giỏ hàng khi khách hàng chưa login

STT	Chức năng	API	Xử lý response

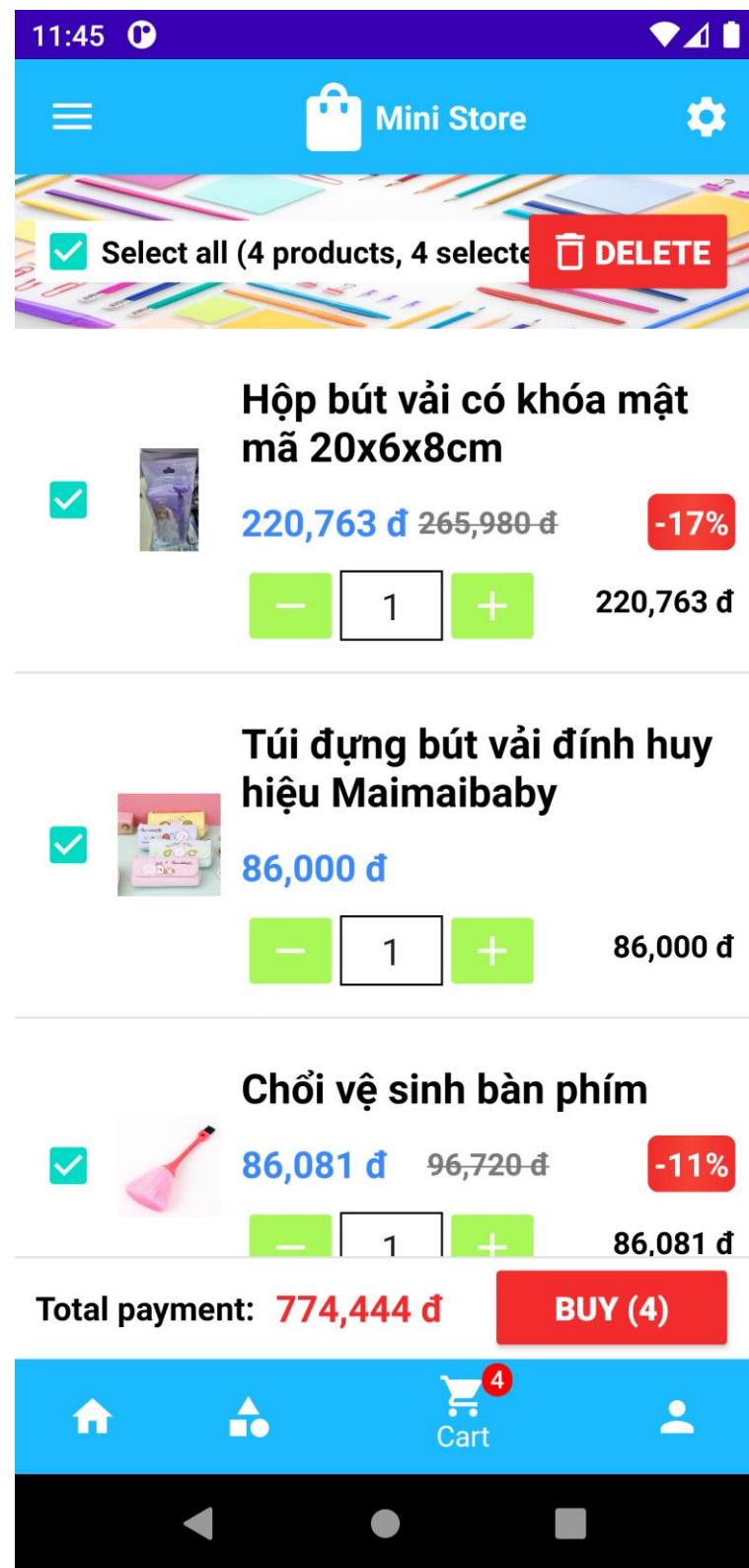
1	Người dùng click vào login		Chuyển đến màn hình login
---	----------------------------	--	---------------------------



Hình 4.20. Giao diện giỏ hàng (2)

- Nhiệm vụ: màn hình giỏ hàng khi khách hàng đã login nhưng giỏ hàng trống

STT	Chức năng	API	Xử lý response
1	Người dùng click vào Go shopping now		Chuyển đến màn hình chính

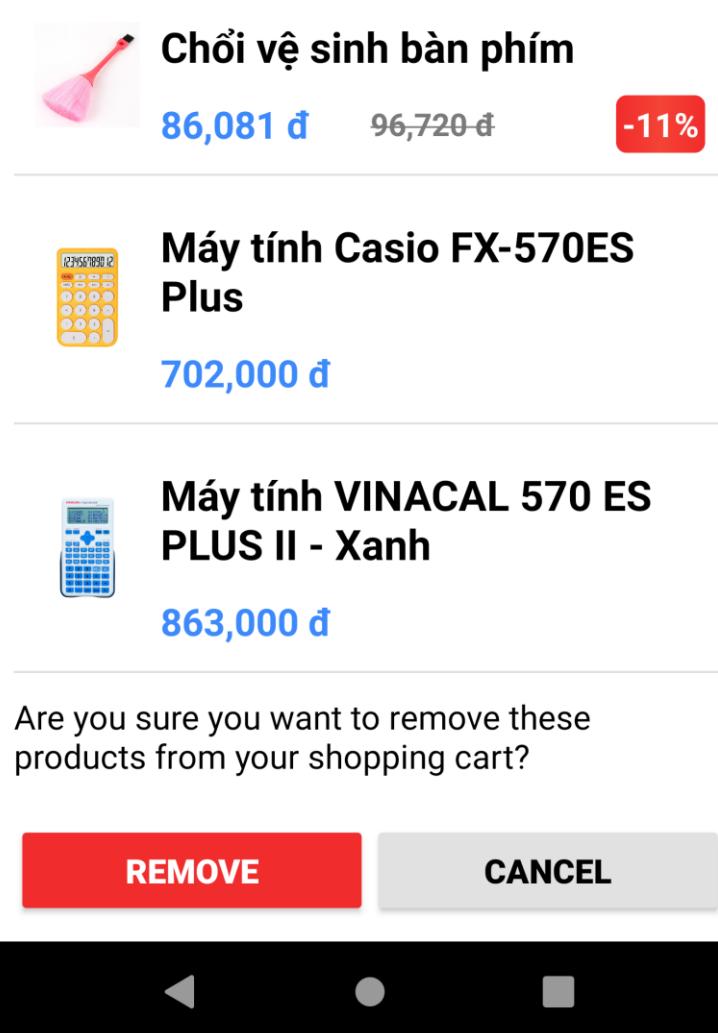


Hình 4.21. Giao diện giỏ hàng (3)

- Nhiệm vụ: màn hình giỏ hàng khi khách hàng đã login và có sản phẩm trong giỏ

STT	Chức năng	API	Xử lý response
-----	-----------	-----	----------------

1	Người dùng click vào checkbox để chọn sản phẩm muốn mua		Form cập nhật tổng giá tiền trên màn hình
2	Người dùng click +/- sản phẩm	Chỉnh sửa giỏ hàng	Form gọi API cập nhật số lượng sản phẩm trong giỏ và hiển thị thông báo trả về
3	Người dùng click Delete	Chỉnh sửa giỏ hàng	Form hiển thị popup xác nhận xoá sản phẩm đã chọn
4	Người dùng click Buy		Chuyển đến màn hình Xác nhận đơn hàng



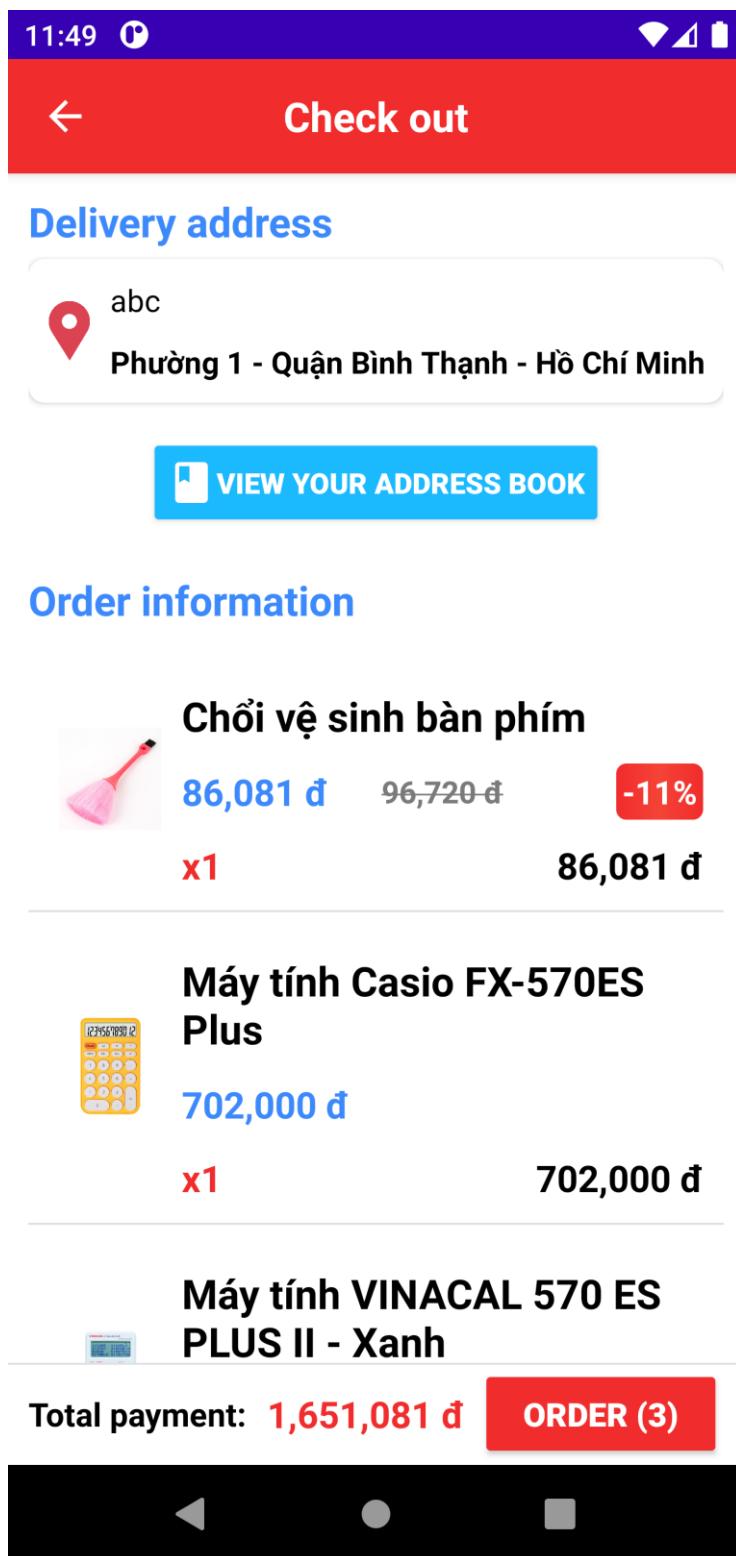
Hình 4.22. Giao diện giỏ hàng (4)

- Nhiệm vụ: xác nhận hoặc huỷ xoá những sản phẩm đã chọn

STT	Chức năng	API	Xử lý response
-----	-----------	-----	----------------

1	Người dùng click vào Remove	Chỉnh sửa giỏ hàng	Refesh lại giỏ hàng
2	Người dùng click Cancel		Đóng popup

4.1.2.6. Thanh toán (Check out)

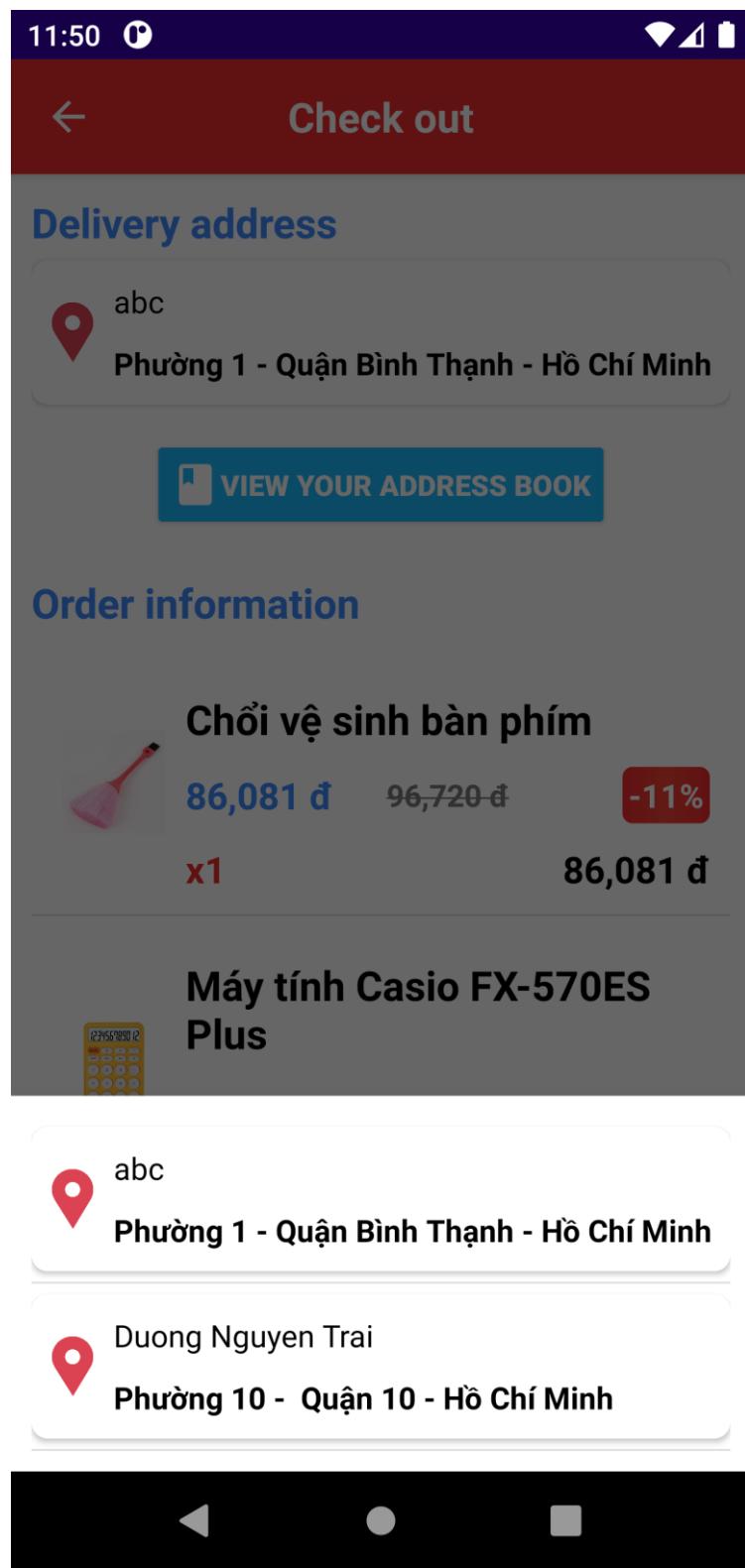


Hình 4.23. Giao diện thanh toán (1)

- Nhiệm vụ: màn hình check out đơn hàng

STT	Chức năng	API	Xử lý response
-----	-----------	-----	----------------

1	Người dùng click vào ORDER	Thêm mới đơn hàng	Hiển thị thông báo api trả về và chuyển đến màn hình chọn phương thức thanh toán
2	Người dùng click chọn địa chỉ		Hiển thị popup chọn địa chỉ giao hàng
3	Người dùng click chọn View your address book		Chuyển đến màn hình quản lý địa chỉ



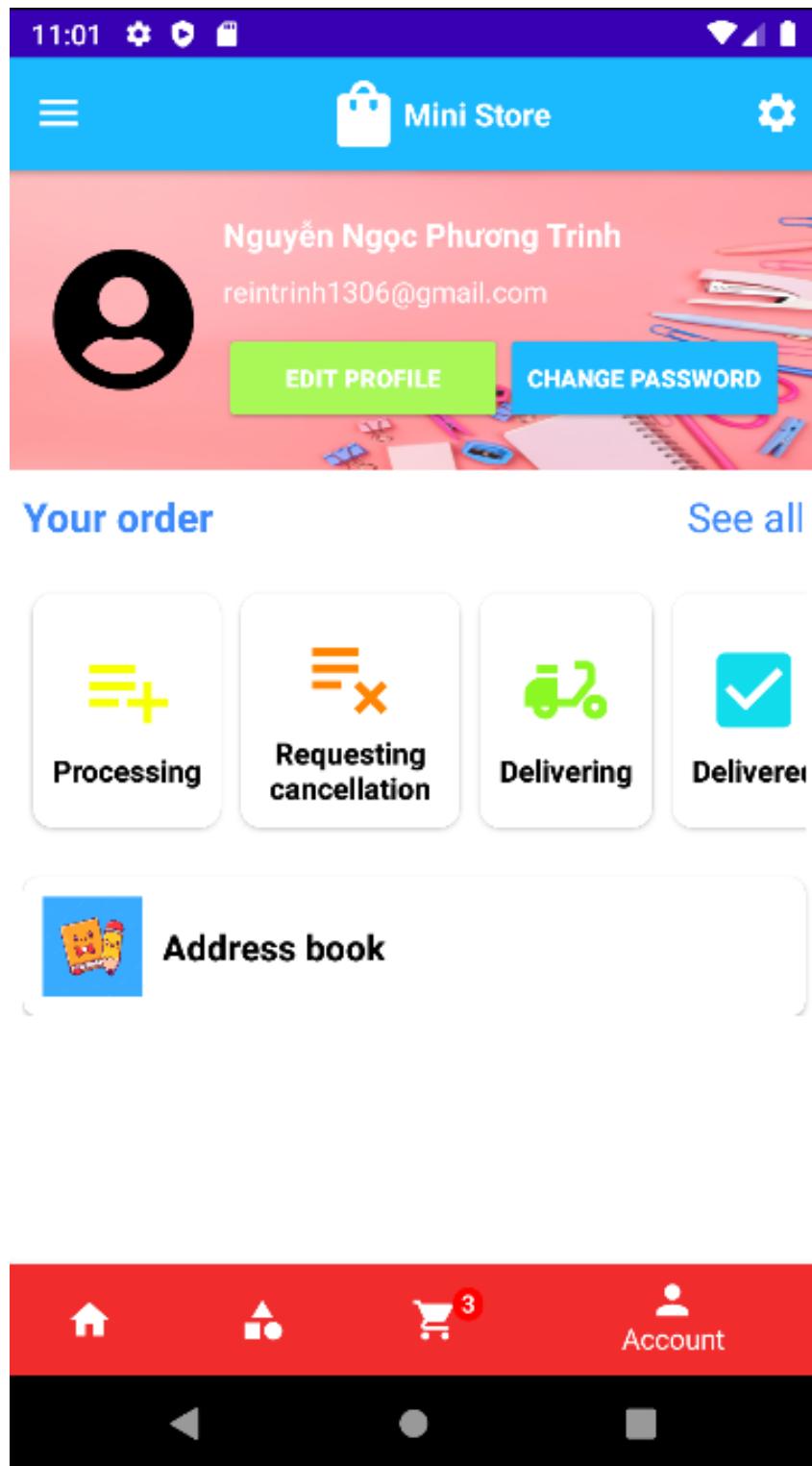
Hình 4.24. Giao diện thanh toán (2)

- Nhiệm vụ: màn hình chọn địa chỉ giao hàng

STT	Chức năng	API	Xử lý response

1	Người dùng click chọn địa chỉ		Form đóng popup và cập nhật địa chỉ đã chọn trên màn hình
---	-------------------------------	--	---

4.1.2.7. Tài khoản (Account)

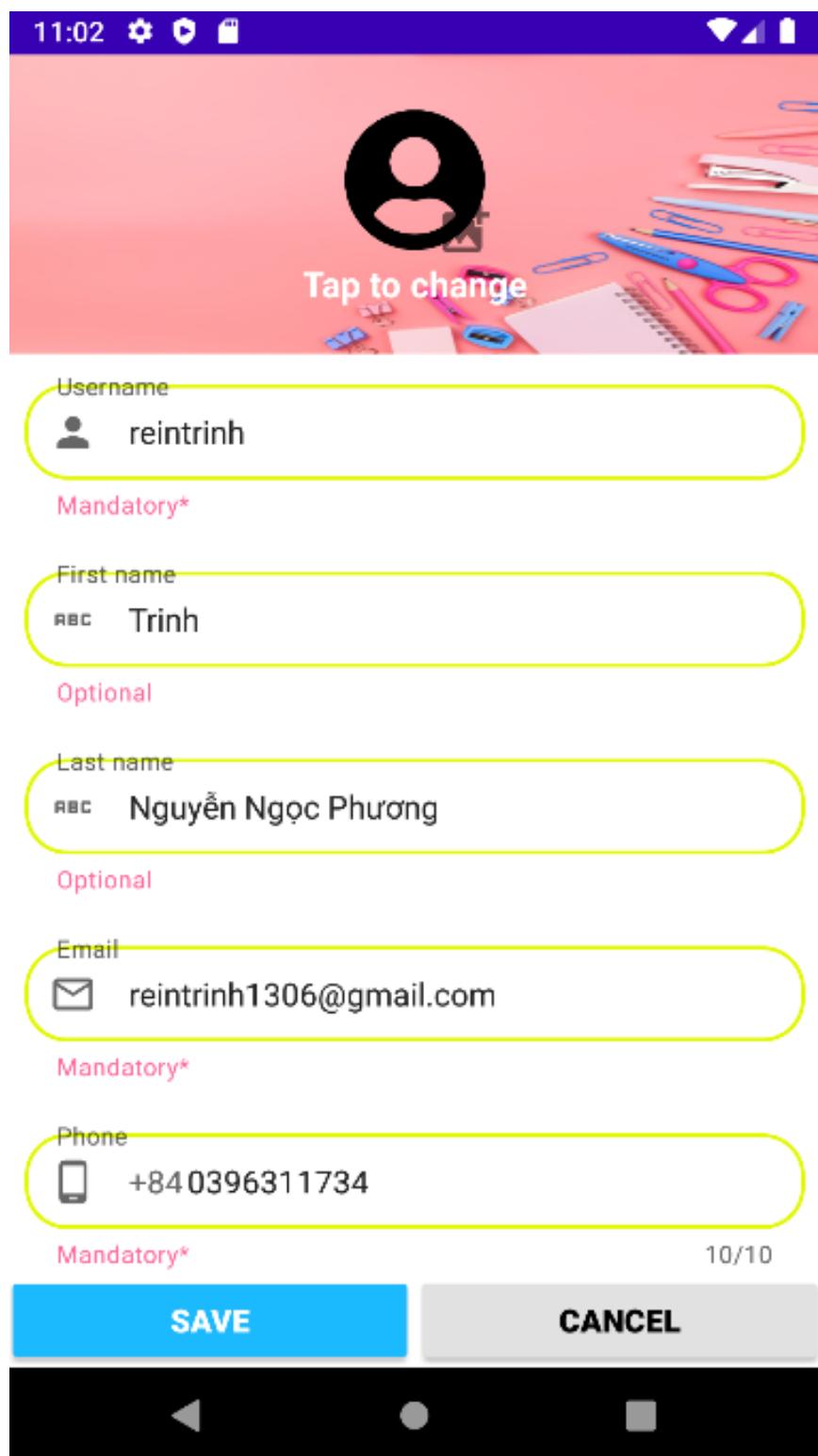


Hình 4.25. Giao diện tài khoản

- Nhiệm vụ: màn hình gồm các chức năng chỉnh sửa thông tin cá nhân, đổi mật khẩu, xem đơn hàng, xem số địa chỉ

STT	Chức năng	API	Xử lý response
1	Người dùng click vào Edit profile		Chuyển đến màn hình Edit profile
2	Người dùng click vào Change password		Chuyển đến màn hình Change password
3	Người dùng click vào Address book		Chuyển đến màn hình Address book
4	Người dùng click vào các tab đơn hàng		Chuyển đến màn hình Phân loại đơn hàng theo trạng thái

4.1.2.8. Chính sửa hồ sơ (Edit profile)



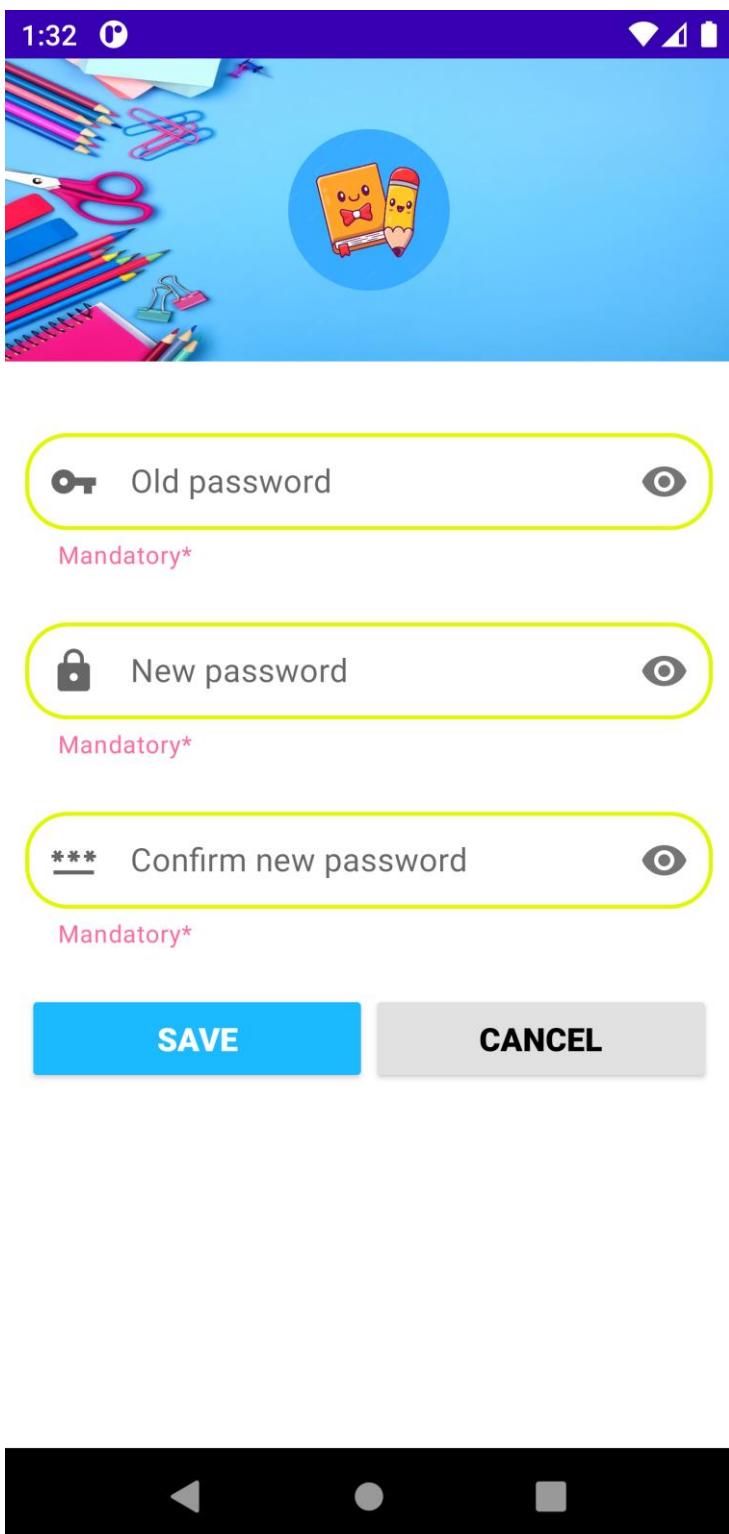
Hình 4.26. Giao diện chỉnh sửa thông tin cá nhân

- Nhiệm vụ: màn hình chỉnh sửa thông tin cá nhân

STT	Chức năng	API	Xử lý response
-----	-----------	-----	----------------

1	Người dùng click vào Save	Chỉnh sửa thông tin user	Báo kết quả
2	Người dùng click vào Cancel		Quay trở về form trước

4.1.2.9. Đổi mật khẩu (Change password)



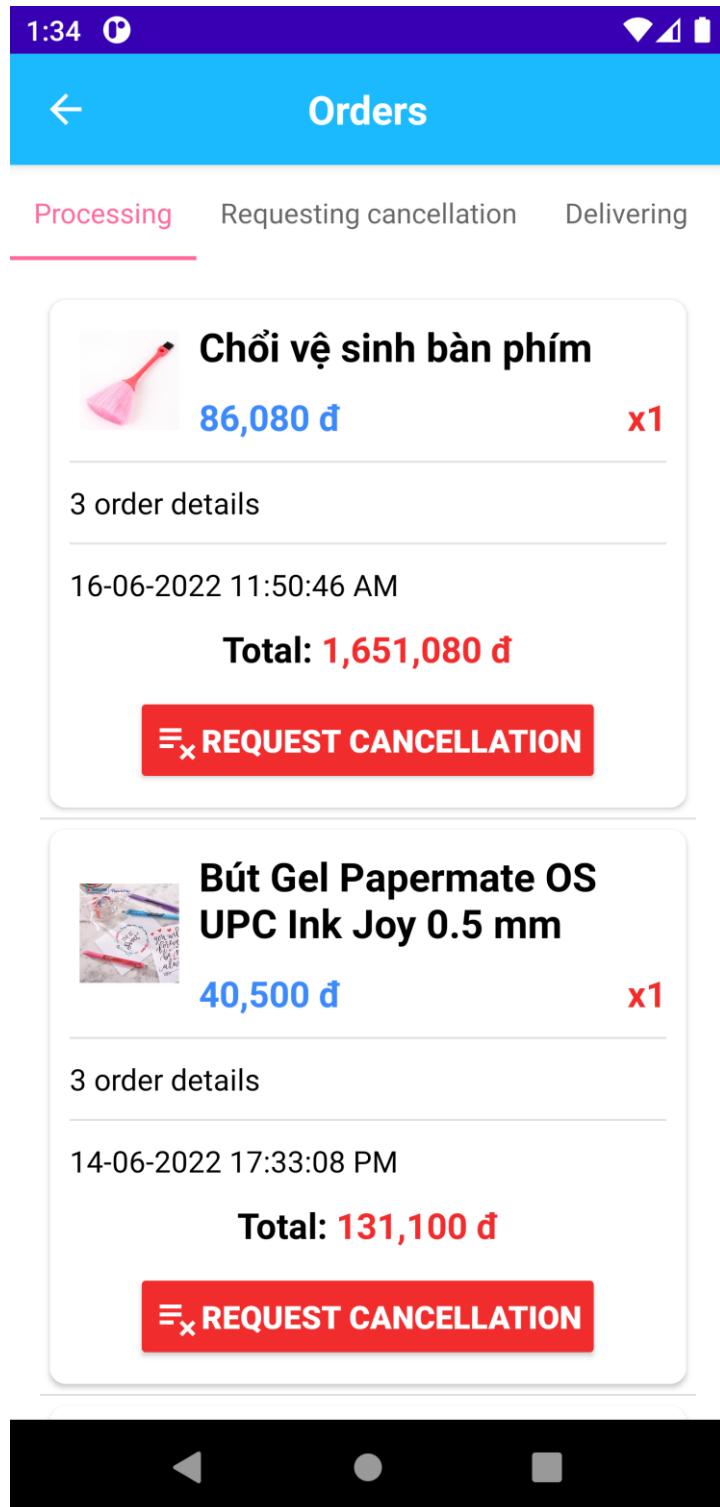
Hình 4.27. Giao diện đổi mật khẩu

- Nhiệm vụ: màn hình chỉnh sửa password

STT	Chức năng	API	Xử lý response
-----	-----------	-----	----------------

1	Người dùng click vào Save	Chỉnh sửa thông tin user	Báo kết quả
2	Người dùng click vào Cancel		Quay trở về form trước

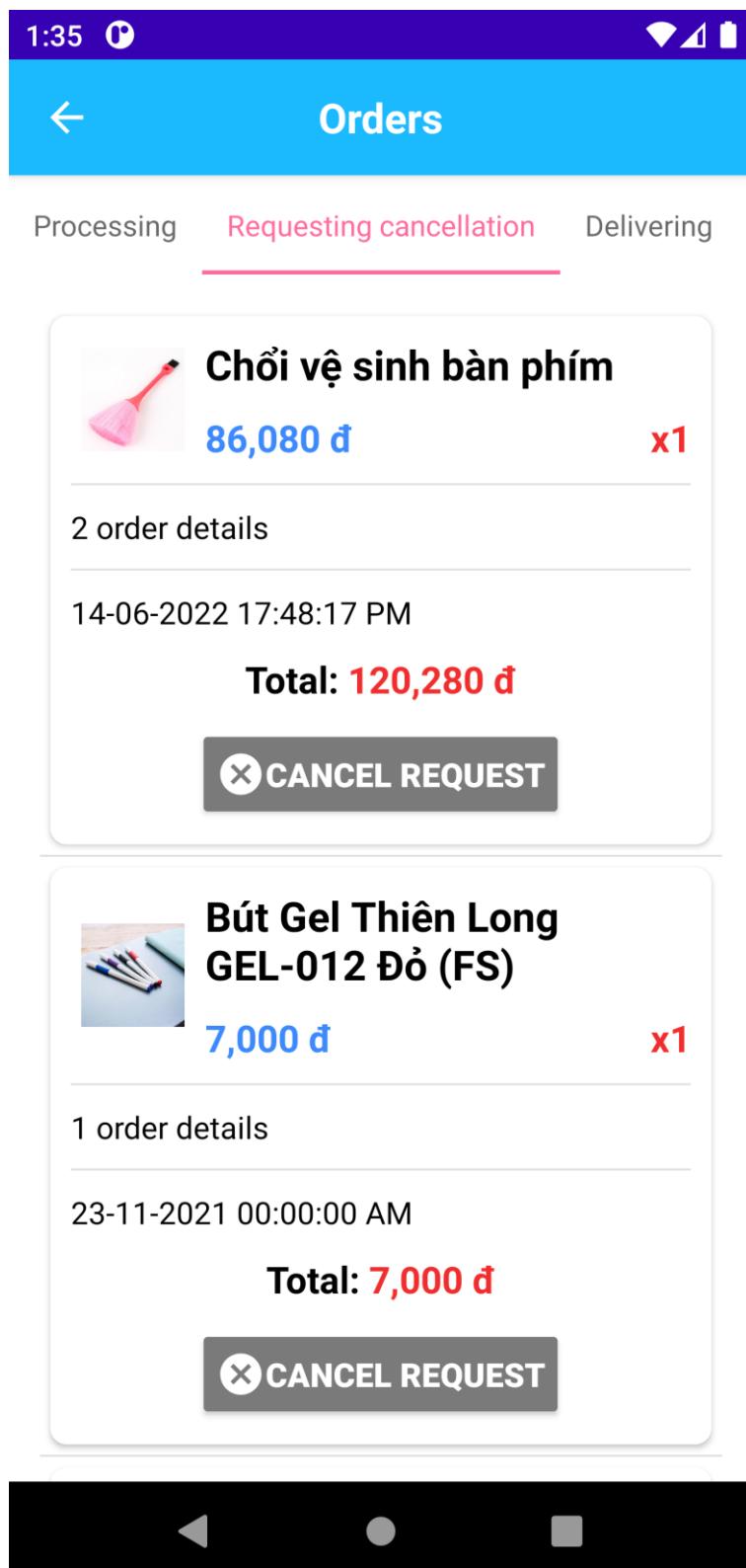
4.1.2.10. Hóa đơn (Order)



Hình 4.28. Giao diện hóa đơn (1)

- Nhiệm vụ: màn hình các đơn hàng đang Chờ xử lý

STT	Chức năng	API	Xử lý response
1	Người dùng click vào đơn hàng		Chuyển đến màn hình chi tiết đơn hàng
2	Người dùng click vào Cancel	Chỉnh sửa trạng thái đơn hàng	Refesh danh sách

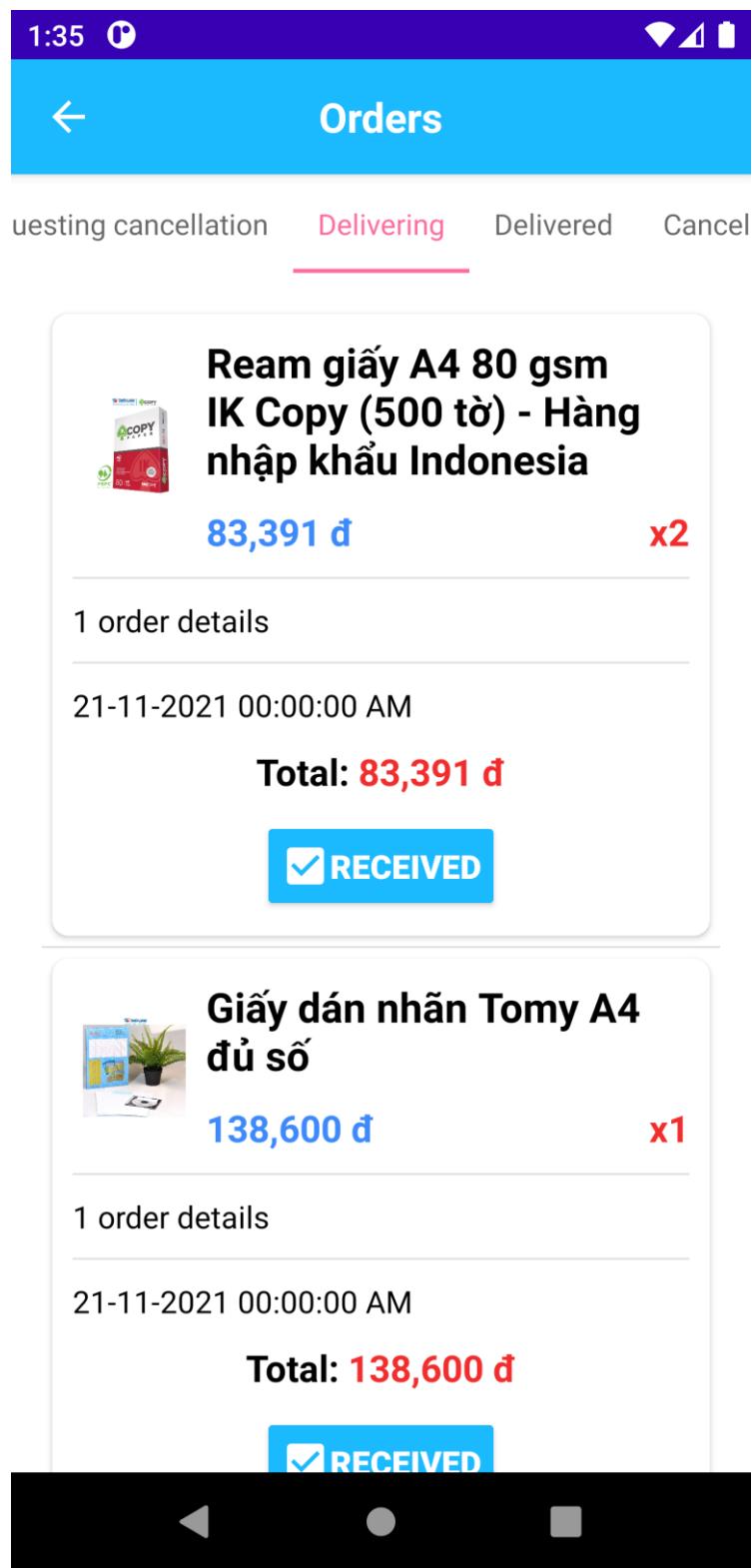


Hình 4.29. Giao diện hóa đơn (2)

- Nhiệm vụ: màn hình các đơn hàng đang Yêu cầu huỷ

STT	Chức năng	API	Xử lý response

1	Người dùng click vào đơn hàng		Chuyển đến màn hình chi tiết đơn hàng
2	Người dùng click vào Cancel	Chỉnh sửa trạng thái đơn hàng	Refesh danh sách

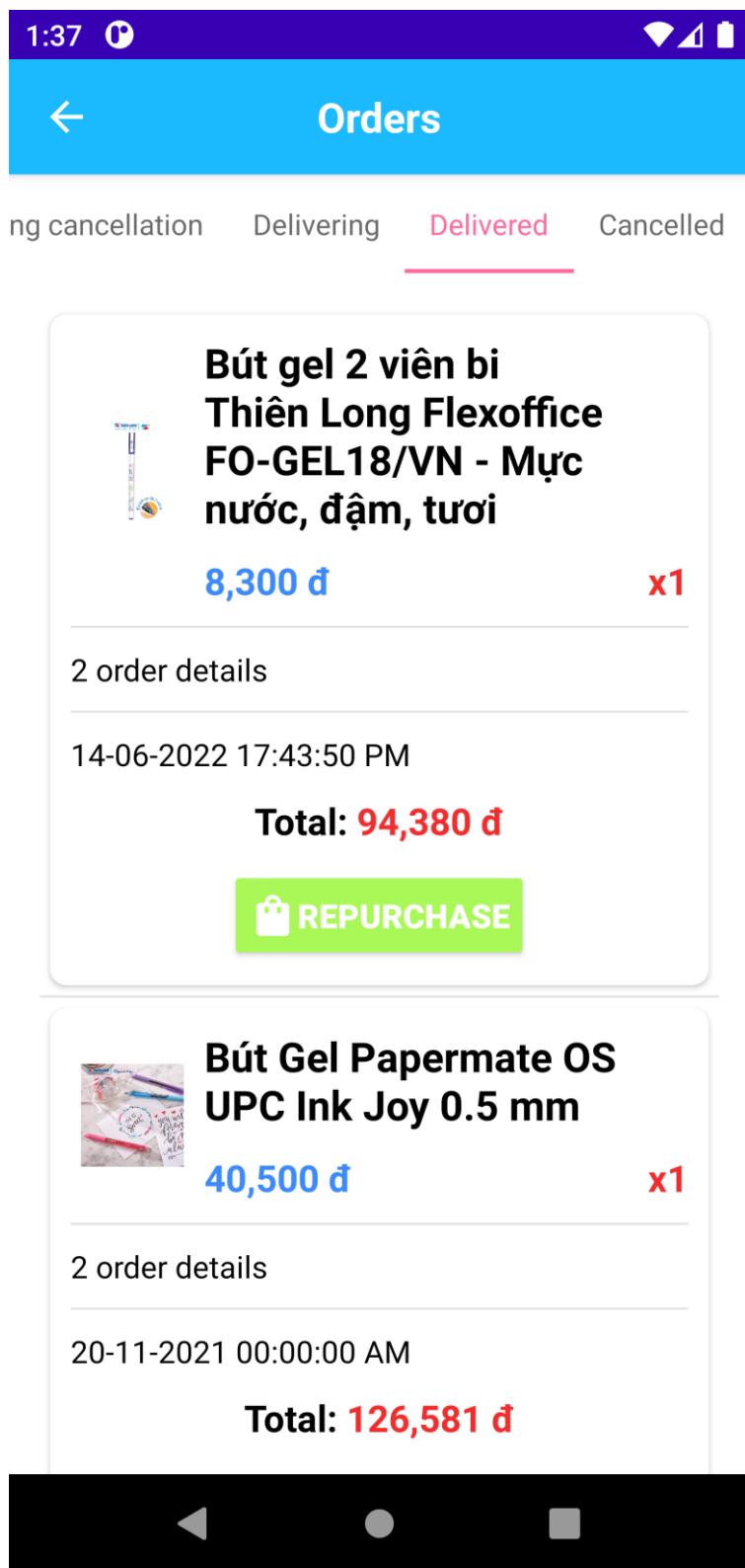


Hình 4.30. Giao diện hóa đơn (3)

- Nhiệm vụ: màn hình các đơn hàng đang Đang giao

STT	Chức năng	API	Xử lý response
-----	-----------	-----	----------------

1	Người dùng click vào đơn hàng		Chuyển đến màn hình chi tiết đơn hàng
2	Người dùng click vào Received	Chỉnh sửa trạng thái đơn hàng	Refesh danh sách

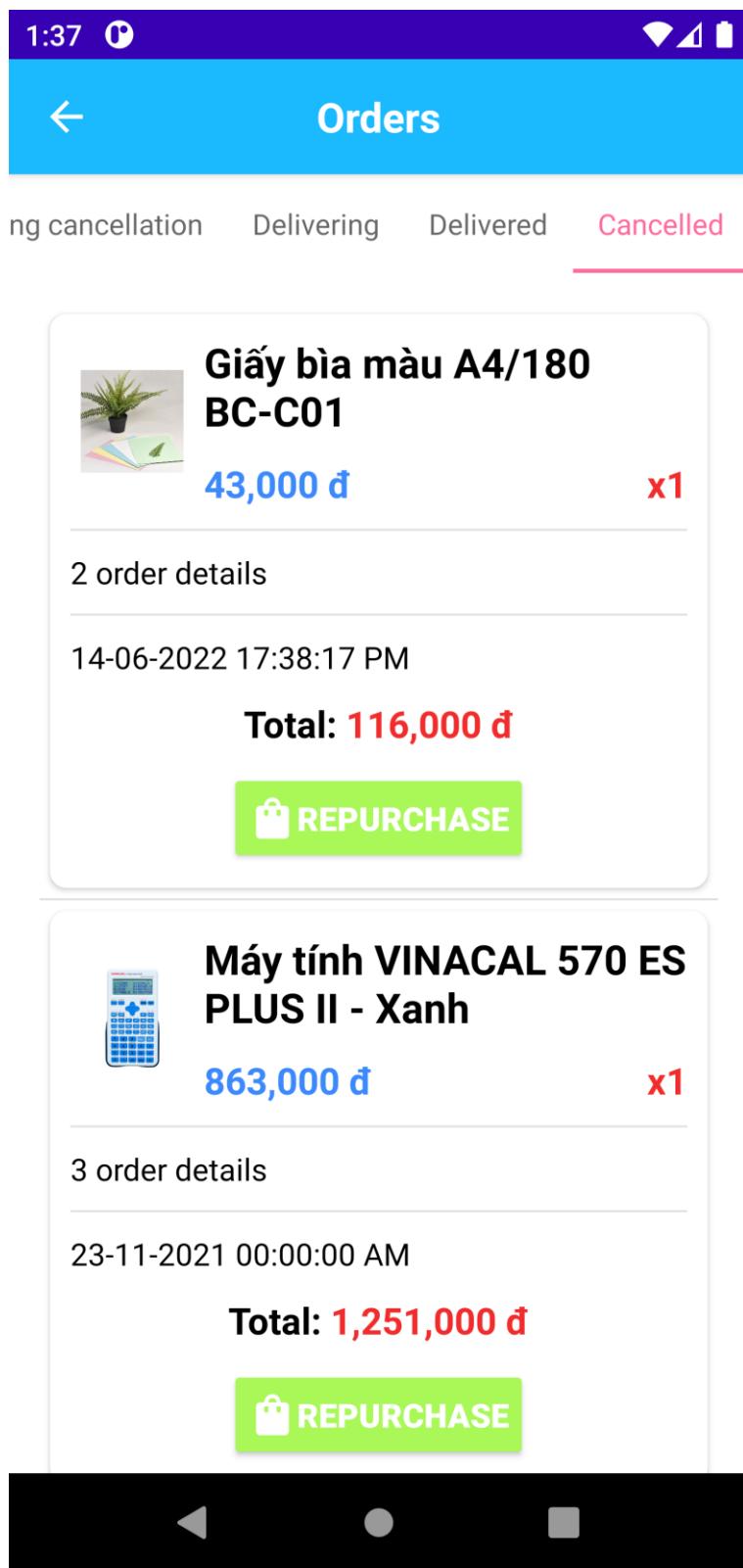


Hình 4.31. Giao diện hóa đơn (4)

- Nhiệm vụ: màn hình các đơn hàng đang Đã giao

STT	Chức năng	API	Xử lý response

1	Người dùng click vào đơn hàng		Chuyển đến màn hình chi tiết đơn hàng
2	Người dùng click vào Repurchase	Thêm sản phẩm vào giỏ hàng	Refesh danh sách

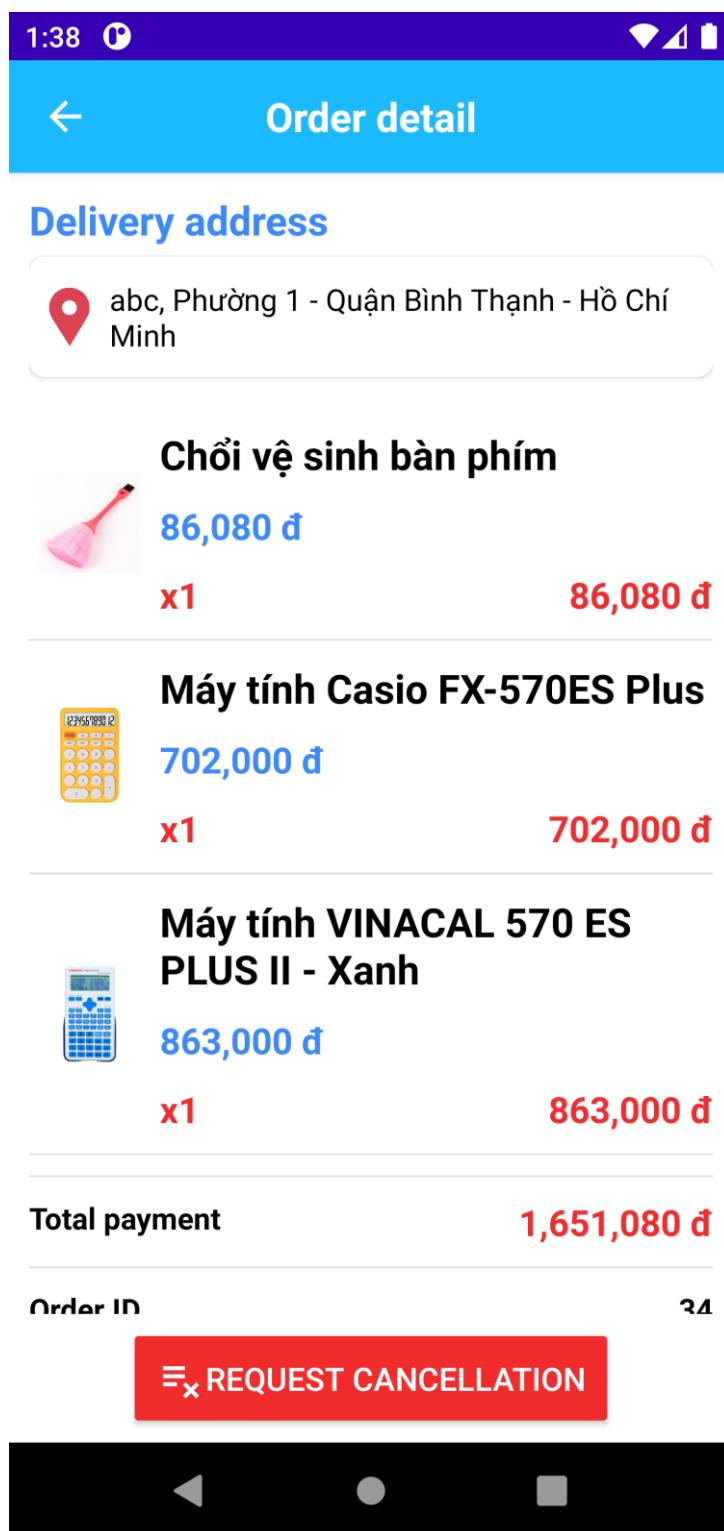
*Hình 4.32. Giao diện hóa đơn (5)*

- Nhiệm vụ: màn hình các đơn hàng đang Đã huỷ

STT	Chức năng	API	Xử lý response

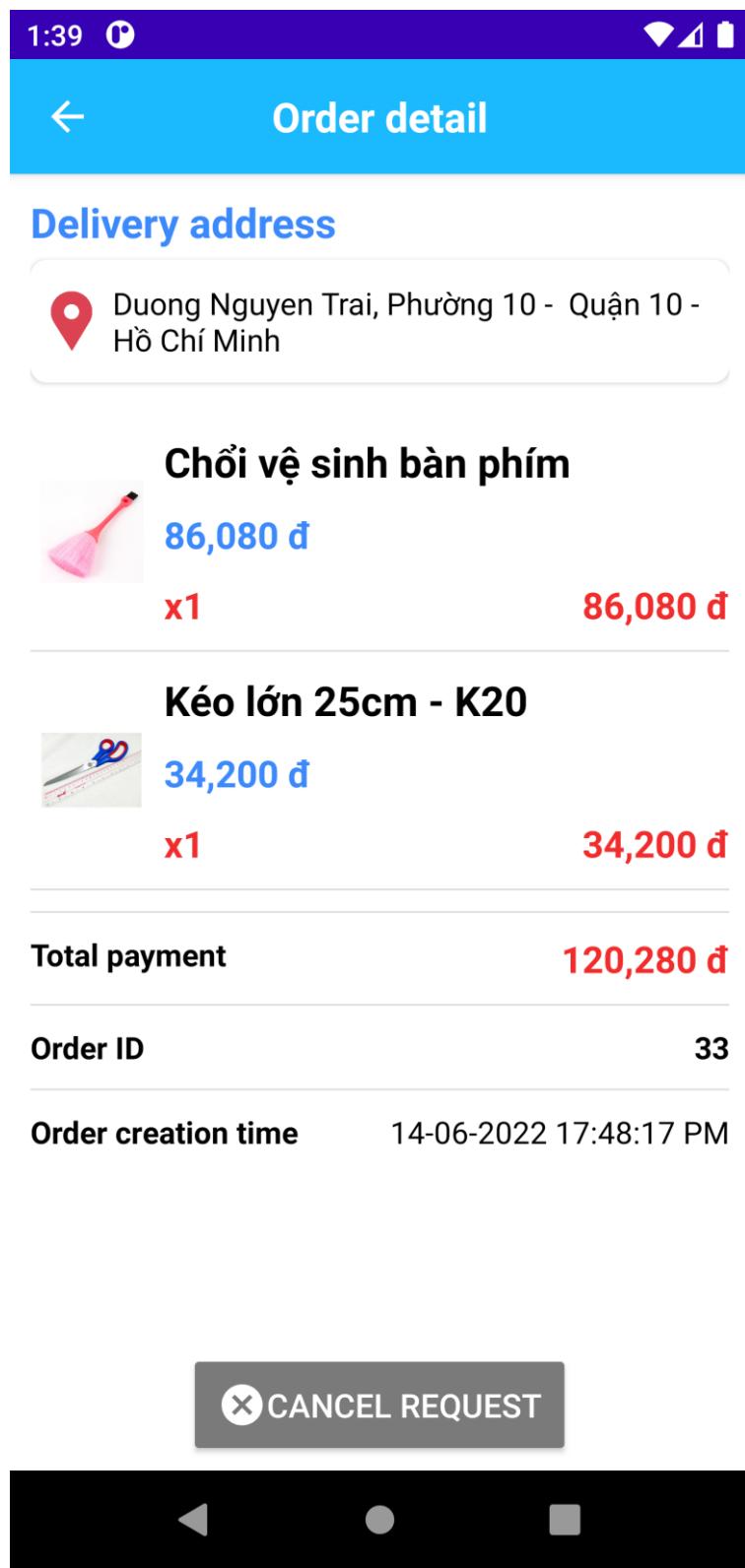
1	Người dùng click vào đơn hàng		Chuyển đến màn hình chi tiết đơn hàng
2	Người dùng click vào Repurchase	Thêm sản phẩm vào giỏ hàng	Refesh danh sách

4.1.2.11. Chi tiết hóa đơn (Order detail)

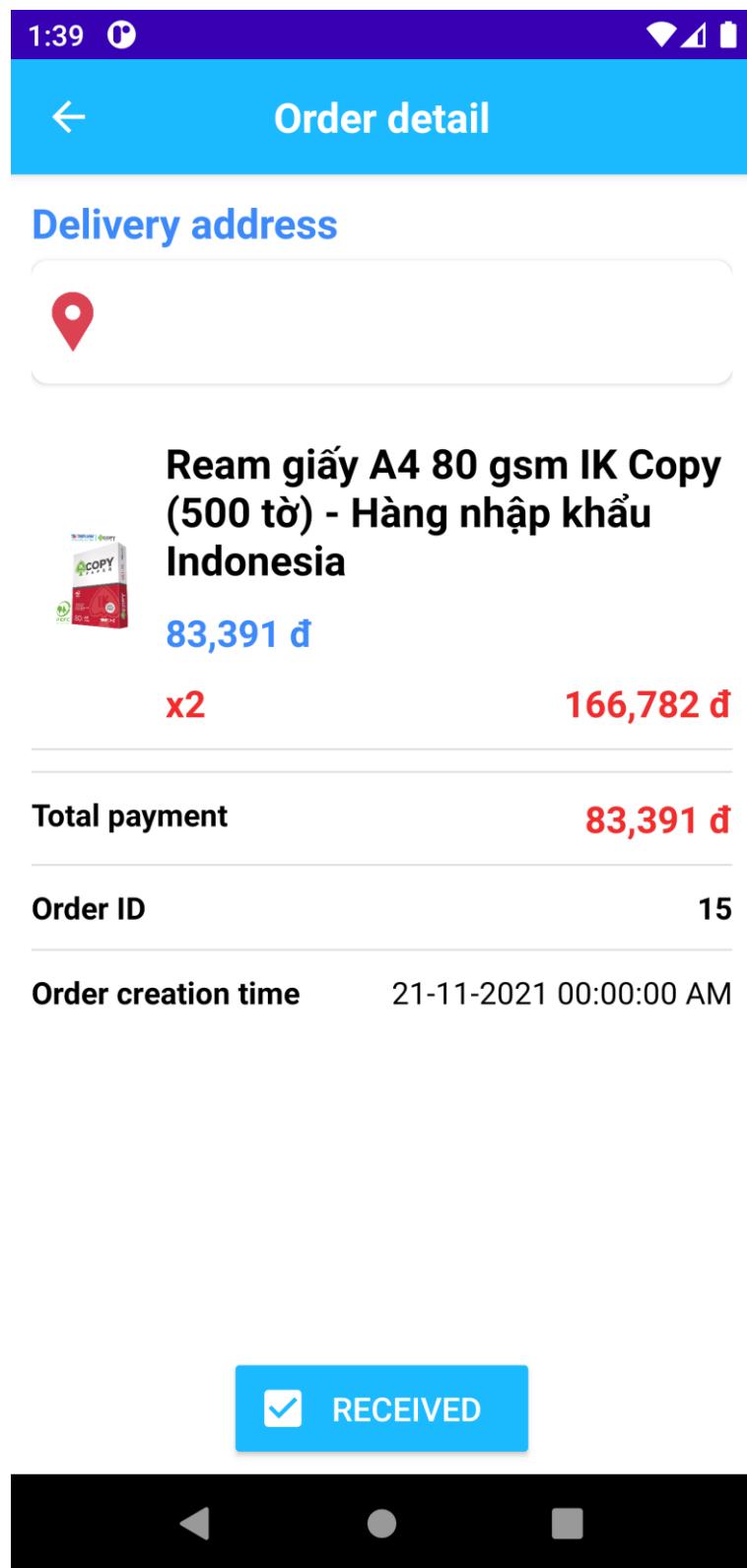


Hình 4.33. Giao diện chi tiết hóa đơn (1)

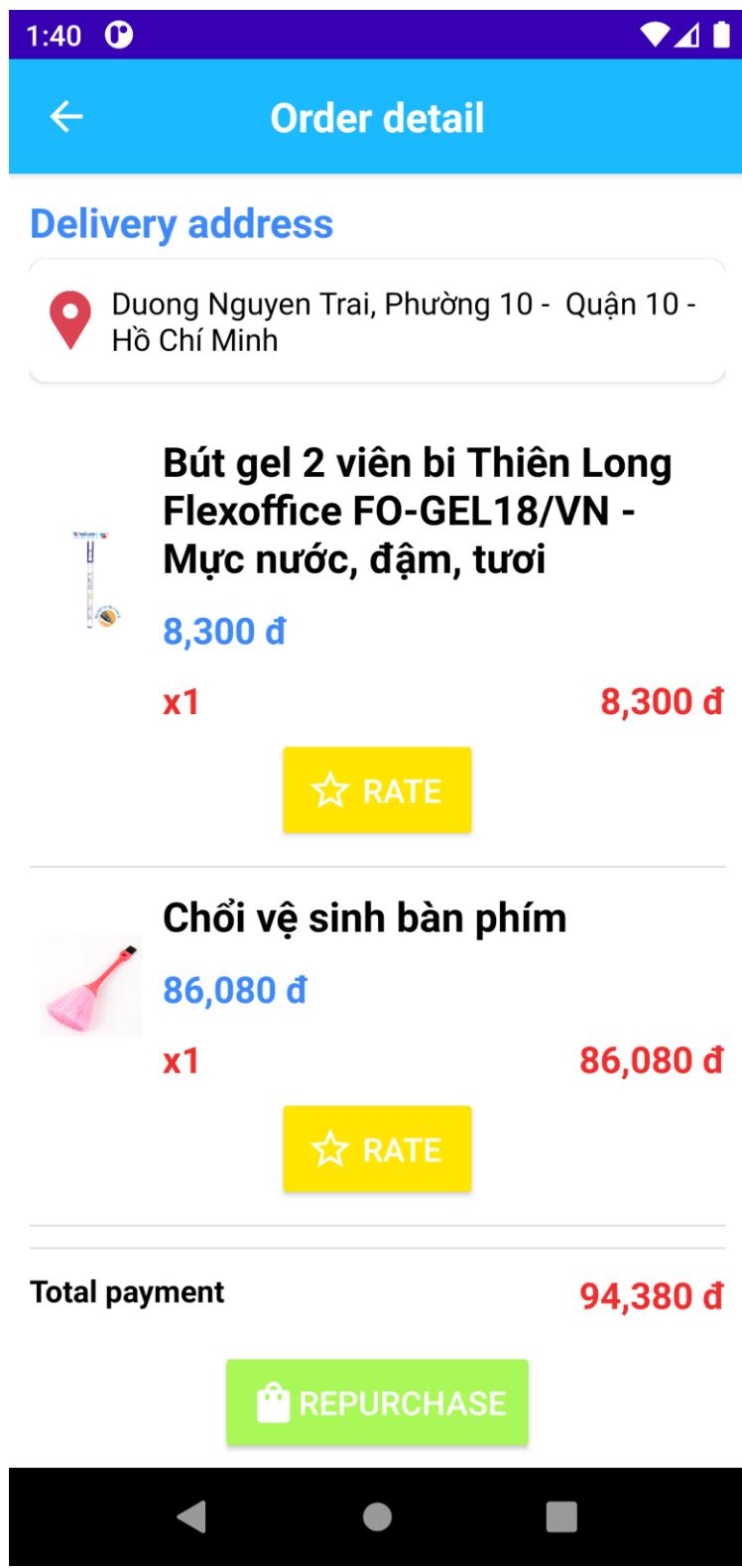
- Nhiệm vụ: màn hình chi tiết đơn hàng



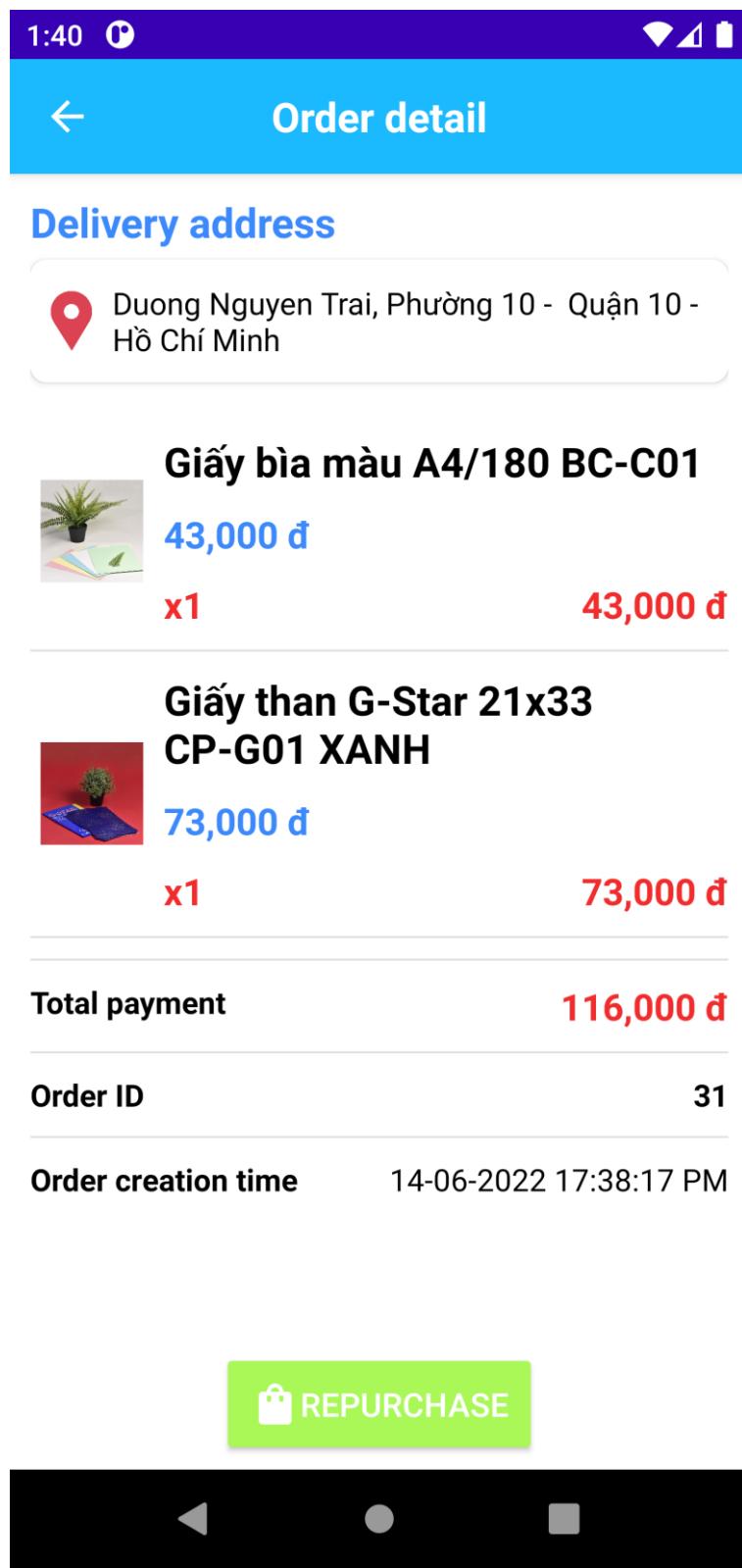
Hình 4.34. Giao diện chi tiết hóa đơn (2)



Hình 4.35. Giao diện chi tiết hóa đơn (3)

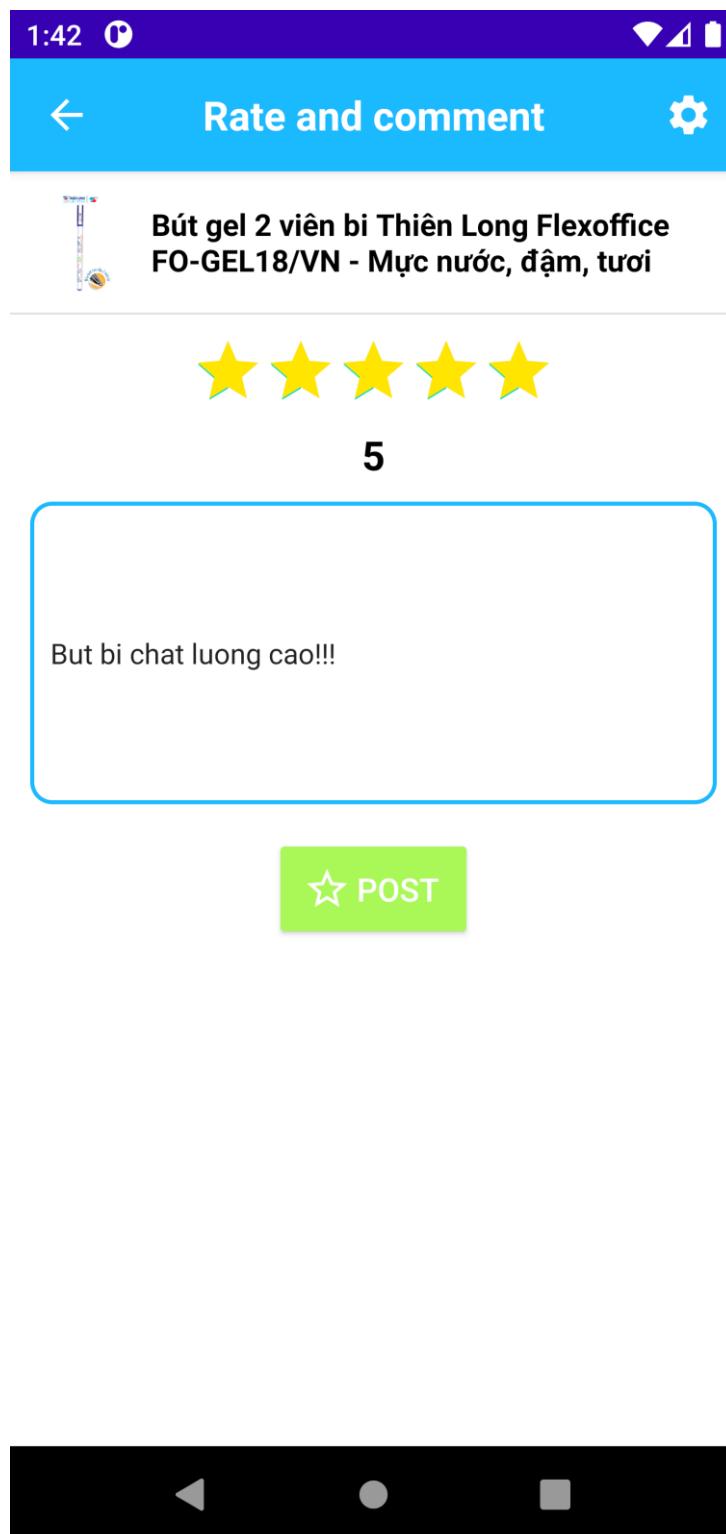


Hình 4.36. Giao diện chi tiết hóa đơn (4)



Hình 4.37. Giao diện chi tiết hóa đơn (5)

4.1.2.12. Đánh giá sản phẩm (Rate)



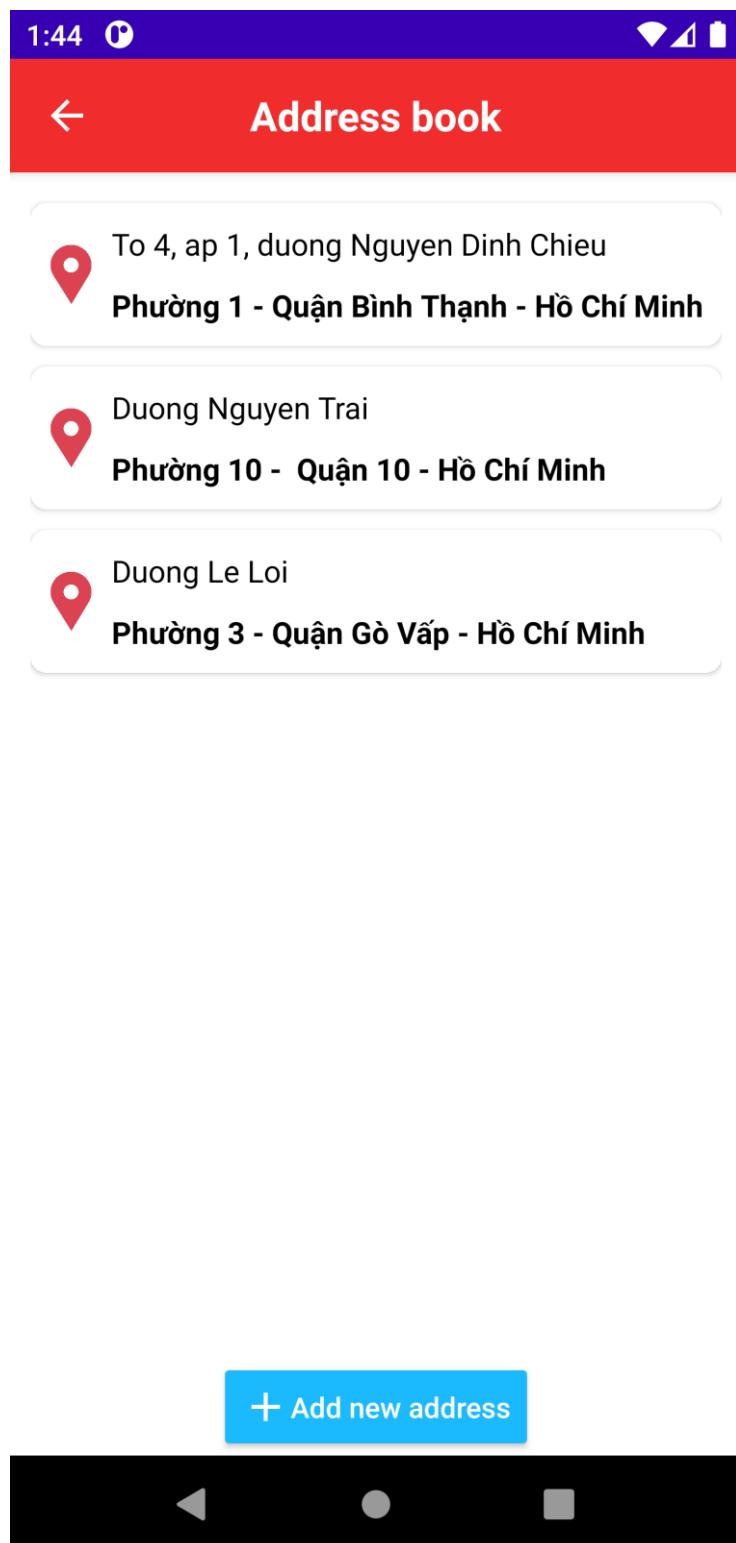
Hình 4.38. Giao diện đánh giá đơn hàng

- Nhiệm vụ: màn hình đánh giá sản phẩm

STT	Chức năng	API	Xử lý response

1	Người dùng click chọn rate, nhập comment → click Post	Thêm mới feedback	Thông báo kết quả
---	---	-------------------	-------------------

4.1.2.13. Sổ địa chỉ (Address book)



Hình 4.39. Giao diện sổ địa chỉ

- Nhiệm vụ: màn hình quản lý địa chỉ

STT	Chức năng	API	Xử lý response
-----	-----------	-----	----------------

1	Người dùng click vào địa chỉ		Chuyển đến màn hình chỉnh sửa địa chỉ
2	Người dùng long-click vào địa chỉ		Hiển thị popup xác nhận xoá địa chỉ
3	Người dùng click Add new address		Chuyển đến màn hình Thêm mới địa chỉ

4.1.2.14. Chính sửa địa chỉ (Edit address)



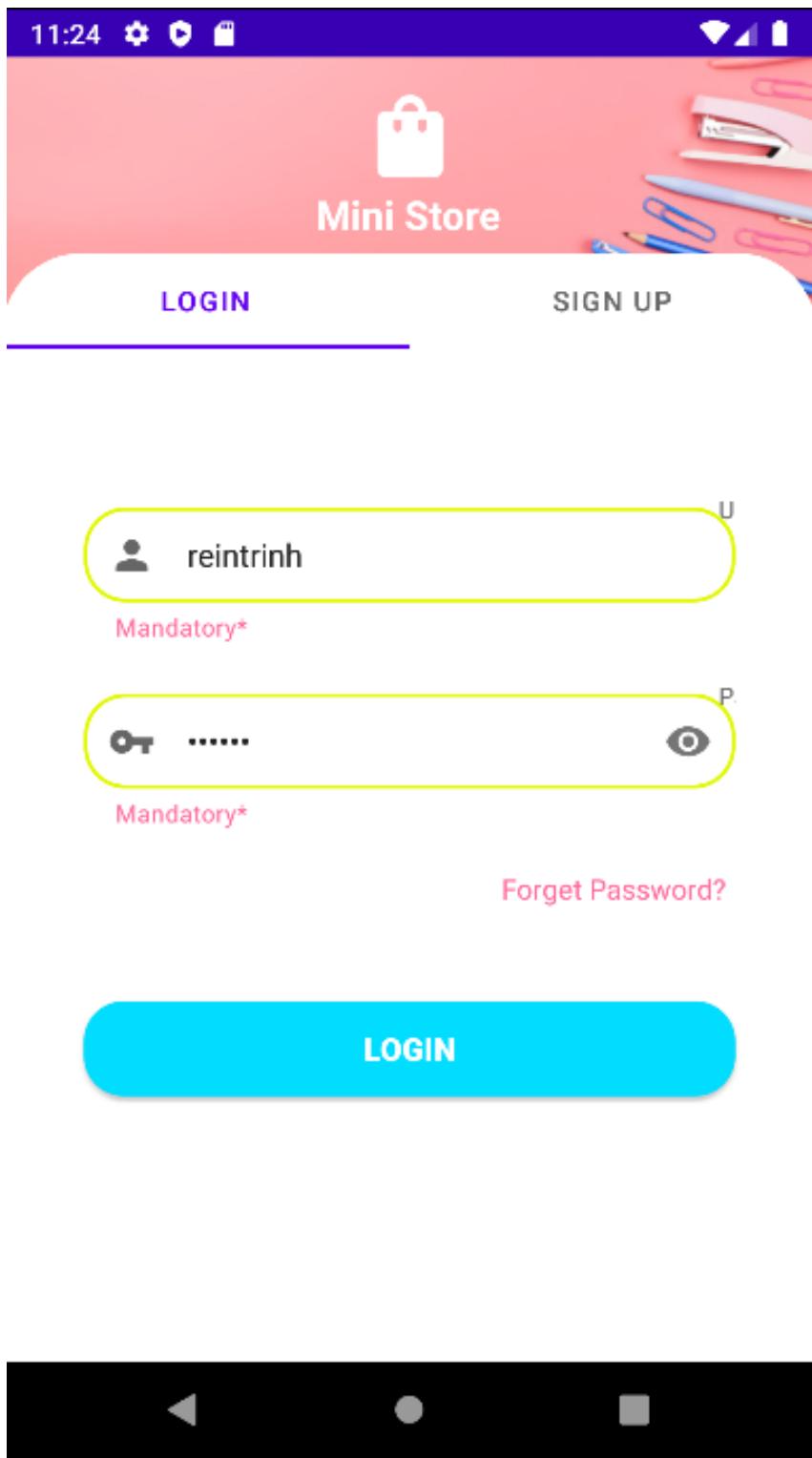
Hình 4.40. Giao diện chỉnh sửa địa chỉ

- Nhiệm vụ: màn hình Chính sửa địa chỉ

STT	Chức năng	API	Xử lý response

1	Người dùng sửa các field → click Confirm	Chỉnh sửa địa chỉ	Thông báo kết quả
---	--	-------------------	-------------------

4.1.2.15. Đăng nhập (Login)

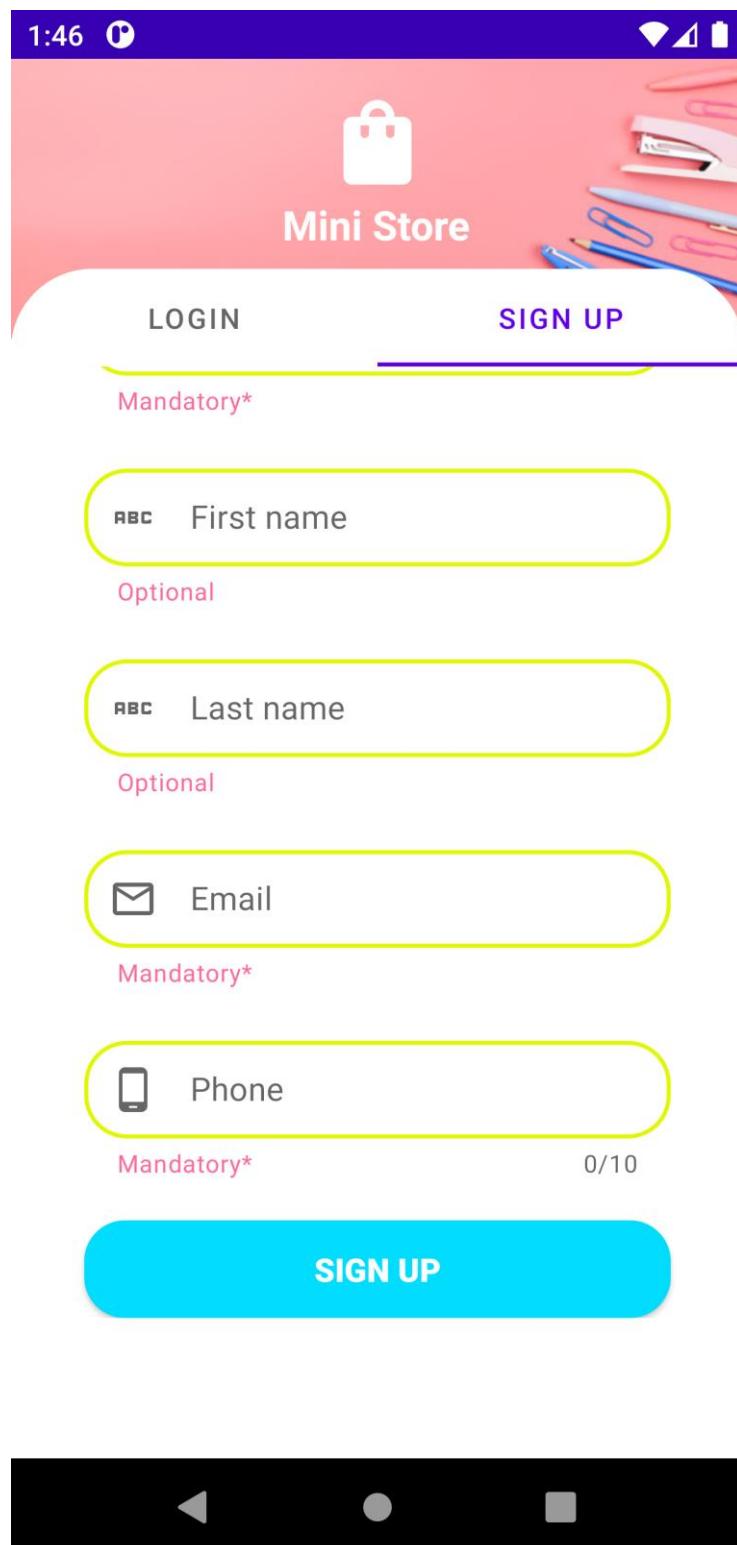


Hình 4.41. Giao diện đăng nhập

- Nhiệm vụ: màn hình thực hiện chức năng đăng nhập

STT	Chức năng	API	Xử lý response
1	Người dùng nhập thông tin → click Login	Lấy thông tin người dùng	Chuyển đến màn hình chính nếu đăng nhập thành công, báo lỗi nếu đăng nhập thất bại
2	Người dùng click vào tab Đăng ký		Chuyển đến tab đăng ký

4.1.2.16. Đăng ký (Sign up)



Hình 4.42. Giao diện đăng ký

- Nhiệm vụ: màn hình thực hiện chức năng đăng ký

STT	Chức năng	API	Xử lý response

1	Người dùng nhập thông tin → click Sign up	Thêm mới người dùng	Chuyển đến màn hình chính nếu đăng ký thành công, báo lỗi nếu đăng ký thất bại
2	Người dùng click vào tab Đăng nhập		Chuyển đến tab đăng nhập

4.2. Back-End (APIs)

4.2.1. Authentication API

Tiêu đề ngữ của API URL: /api/auth

4.2.1.1. Sign up

URL: /api/auth/signup

Method: POST

Body: Chứa thông tin đăng ký gồm

Field	Type	Description	Constraint
username	String	Tên người dùng để đăng ký	Not Null
Email	String	Email dùng để đăng ký	Not Null
password	String	Mật khẩu dùng để đăng ký	Not Null
role	array	Vai trò khi đăng ký	Chỉ gồm các giá trị “admin” và “user”

Response Body

Success Response:

- Code: 200 OK
- Body: gồm message thông báo và thông tin đã đăng ký

Error Response:

- Code 400 Bad Request – Username, Email hoặc phone đã được sử dụng

4.2.1.2. Sign in

URL: /api/auth/signin

Method: POST

Body: Chứa thông tin đăng nhập gồm

Field	Type	Description	Constraint
username	String	Tên người dùng để đăng nhập	Not Null
password	String	Mật khẩu dùng để đăng nhập	Not Null

Response Body**Success Response:**

- Code: 200 OK
- Body: Mã token lưu tại client dùng để xác thực

Error Response:

- Code 401 UNAUTHORIZED – Username hoặc password không đúng

4.2.2. User API**4.2.2.1. Address**

Tiếp đầu ngữ của API URL: /api/address

4.2.2.1.1. Lấy danh sách tỉnh thành

URL: /api/address/province

Method: GET

Response Body

- Code: 200 OK
- Body: Danh sách tỉnh thành

4.2.2.1.2. Lấy danh sách quận huyện thuộc một tỉnh thành

URL: /api/address/district/{id} – id là mã tỉnh thành

Method: GET

Response Body

- Code: 200 OK
- Body: Danh sách quận huyện thuộc một tỉnh thành

Success Response:**4.2.2.1.3. Lấy danh sách phường xã thuộc một quận huyện**

URL: /api/address/ward/{id} – id là mã quận huyện

Method: GET

Response Body

- Code: 200 OK
- Body: Danh sách phường xã thuộc một quận huyện

4.2.2.1.4. Lấy danh sách địa chỉ thuộc một người dùng

URL: /api/address/address/{id} – id là mã người dùng

Method: GET

Response Body

- Code: 200 OK
- Body: Danh sách địa chỉ thuộc một người dùng

4.2.2.1.5. Thêm địa chỉ cho một người dùng

URL: /api/address/address/{id} – id là mã người dùng

Method: POST

Body: Chứa thông tin đăng nhập gồm

Field	Type	Description	Constraint
specificAddress	String	Địa chỉ cụ thể	Not Null
ward	Object	Đối tượng phường, xã	Not Null

Response Body

Success Response:

- Code: 200 OK
- Body: message thông báo và thông tin địa chỉ

Error Response:

- Code 400 Bad Request – Thông tin không đúng hoặc user không tồn tại

4.2.2.1.6. Sửa địa chỉ của một người dùng

URL: /api/address/address/{id} – id là mã người dùng

Method: PUT

Request Body

Field	Type	Description	Constraint
specificAddress	String	Địa chỉ cụ thể	Not Null
ward	Object	Đối tượng phường, xã	Not Null

Response Body

Success Response:

- Code: 200 OK
- Body: message thông báo và thông tin địa chỉ

Error Response:

- Code 400 Bad Request – Thông tin không đúng hoặc user không tồn tại

4.2.2.1.7. Xóa một địa chỉ

URL: /api/address/address/{id} – id của địa chỉ

Method: DELETE

Response Body

Success Response:

- Code: 200 OK
- Body: message thông báo xóa thành công

Error Response:

- Code 400 Bad Request – Địa chỉ không tồn tại

4.2.2.2. Product

Tiếp đầu ngữ của API URL: /api/product

4.2.2.2.1. Lấy danh sách sản phẩm

URL: /api/product

Method: GET

Request params: thông tin truy vấn

Field	Type	Description	Constraint
pageNo	int	Số thứ tự trang	Null
pageSize	int	Kích cỡ trang	Null
sortField	int	Field sắp xếp	Null
sortDirection	String	Hướng sắp xếp	Null

Response Body

Success Response:

- Code: 200 OK
- Body: Danh sách sản phẩm

4.2.2.2.2. Lấy thông tin chi tiết một sản phẩm

URL: /api/product/{id} – id là mã sản phẩm

Method: GET

Response Body

Success Response:

- Code: 200 OK
- Body: thông tin sản phẩm

Error Response:

- Code 400 Bad Request – Sản phẩm không tồn tại

4.2.2.2.3. Lấy hình ảnh của một sản phẩm

URL: /api/product/image/{name} – name là tên hình của sản phẩm

Method: GET

Response Body

Success Response:

- Code: 200 OK
- Body: hình ảnh sản phẩm

Error Response:

- Code 400 Bad Request – Hình ảnh không tồn tại

4.2.2.3. Cart

Tiếp đầu ngữ của API URL: /api/cart

4.2.2.3.1. Lấy thông tin giỏ hàng của user

URL: /api/cart

Method: GET

Request params: thông tin truy vấn

Field	Type	Description	Constraint
userId	int	Id của user	Not Null

Response Body**Success Response:**

- Code: 200 OK
- Body: Danh sách sản phẩm đang lưu trong giỏ hàng

4.2.2.3.2. Thêm sản phẩm vào giỏ hàng**URL:** /api/cart**Method:** POST**Body:** thông tin giỏ hàng

Field	Type	Description	Constraint
userId	int	Id của user	Not Null
productId	int	Id của sản phẩm	Not Null
quantity	int	Số lượng muốn thêm vào giỏ	Lớn hơn 0

Response Body**Success Response:**

- Code: 200 OK
- Body: message thông báo thêm thành công

Error Response:

- Code 400 Bad Request – Sản phẩm, người dùng không tồn tại, số lượng nhỏ hơn hoặc bằng 0, số lượng vượt quá số lượng hiện có của sản phẩm

4.2.2.3.3. Chính sửa số lượng sản phẩm trong giỏ hàng**URL:** /api/cart**Method:** PUT**Body:** thông tin giỏ hàng

Field	Type	Description	Constraint
userId	int	Id của user	Not Null
productId	int	Id của sản phẩm	Not Null
quantity	int	Số lượng muốn chỉnh sửa	Lớn hơn 0

Response Body**Success Response:**

- Code: 200 OK
- Body: message thông báo chỉnh sửa thành công

Error Response:

- Code 400 Bad Request – Sản phẩm, người dùng không tồn tại, số lượng nhỏ hơn hoặc bằng 0

4.2.2.3.4. Xóa sản phẩm trong giỏ hàng**URL:** /api/cart/{id} – id là mã giỏ hàng**Method:** DELETE**Response Body****Success Response:**

- Code: 200 OK
- Body: message thông báo xóa thành công

Error Response:

- Code 400 Bad Request – không tìm thấy cart

4.2.2.4. Category

Tiếp đầu ngữ của API URL: /api/category

4.2.2.4.1. Lấy danh sách thể loại**URL:** /api/category**Method:** GET**Request params:** thông tin truy vấn

Field	Type	Description	Constraint
pageNo	int	Số thứ tự trang	Null
pageSize	int	Kích cỡ trang	Null
sortField	int	Field sắp xếp	Null
sortDirection	String	Hướng sắp xếp	Null

Response Body**Success Response:**

- Code: 200 OK
- Body: Danh sách sản phẩm

4.2.2.4.2. Lấy thông tin chi tiết một thể loại**URL:** /api/category/{id} – id là mã thể loại**Method:** GET**Response Body****Success Response:**

- Code: 200 OK
- Body: thông tin thể loại

Error Response:

- Code 400 Bad Request – Thể loại không tồn tại

4.2.2.4.3. Lấy hình ảnh của một thể loại

URL: /api/category/image/{name} – name là tên hình của thể loại

Method: GET

Response Body

Success Response:

- Code: 200 OK
- Body: hình ảnh thể loại

Error Response:

- Code 400 Bad Request – Hình ảnh không tồn tại

4.2.2.5. Brand

Tiếp đầu ngữ của API URL: /api/brand

4.2.2.5.1. Lấy danh sách thương hiệu

URL: /api/brand

Method: GET

Request params: thông tin truy vấn

Field	Type	Description	Constraint
pageNo	int	Số thứ tự trang	Null
pageSize	int	Kích cỡ trang	Null
sortField	int	Field sắp xếp	Null
sortDirection	String	Hướng sắp xếp	Null

Response Body

Success Response:

- Code: 200 OK
- Body: Danh sách thương hiệu

4.2.2.5.2. Lấy thông tin chi tiết một thương hiệu

URL: /api/brand/{id} – id là mã thương hiệu

Method: GET

Response Body

Success Response:

- Code: 200 OK
- Body: thông tin thương hiệu

Error Response:

- Code 400 Bad Request – Thương hiệu không tồn tại

4.2.2.6. Feedback

Tiếp đầu ngữ của API URL: /api/feedbacks

4.2.2.6.1. Lấy danh sách đánh giá theo Product ID

URL: /api/feedbacks

Method: GET

Request params: thông tin truy vấn

Field	Type	Description	Constraint
productId	int	Mã sản phẩm	Null

Response Body

Success Response:

- Code: 200 OK
- Body: Danh sách đánh giá

4.2.2.6.2. Lấy danh sách đánh giá của người dùng theo Product ID

URL: /api/feedbacks/{userId}

Method: GET

Request params: thông tin truy vấn

Field	Type	Description	Constraint
productId	int	Mã sản phẩm	Null

Response Body

Success Response:

- Code: 200 OK
- Body: Danh sách đánh giá

4.2.2.6.3. Thêm đánh giá

URL: /api/feedbacks

Method: POST

Body: Chứa thông tin đánh giá gồm

Field	Type	Description	Constraint
feedbackId	FeedbackId	Mã đánh giá	Unique
user	User	Người dùng	Not Null
product	Product	Sản phẩm	Not Null
date	Date	Ngày đánh giá	Null
comment	String	Đánh giá	Null
vote	int	Bình chọn	Null

Response Body

Success Response:

- Code: 200 OK

- Body: message thông báo và thông tin đánh giá

Error Response:

- Code 400 Bad Request – Không tìm thấy người dùng
- Code 400 Bad Request – Không tìm thấy sản phẩm

4.2.2.7. File Upload

Tiếp đầu ngữ của API URL: /api/image

4.2.2.7.1. Thêm hình ảnh poster

URL: /api/image/poster

Method: POST

Body: Chứa thông tin đánh giá gồm

Field	Type	Description	Constraint
file	MultipartFile	Hình ảnh poster	Not Null

Response Body***Success Response:***

- Code: 200 OK
- Body: Chuỗi tên file hình ảnh

4.2.2.7.2. Thêm hình ảnh product

URL: /api/image/product

Method: POST

Body: Chứa thông tin đánh giá gồm

Field	Type	Description	Constraint
file	MultipartFile	Hình ảnh product	Not Null

Response Body***Success Response:***

- Code: 200 OK
- Body: Chuỗi tên file hình ảnh

4.2.2.7.3. Thêm hình ảnh category

URL: /api/image/category

Method: POST

Body: Chứa thông tin đánh giá gồm

Field	Type	Description	Constraint
file	MultipartFile	Hình ảnh category	Not Null

Response Body***Success Response:***

- Code: 200 OK

- Body: Chuỗi tên file hình ảnh

4.2.2.7.4. Thêm hình ảnh user

URL: /api/image/user

Method: POST

Body: Chứa thông tin đánh giá gồm

Field	Type	Description	Constraint
file	MultipartFile	Hình ảnh user	Not Null

Response Body

Success Response:

- Code: 200 OK
- Body: Chuỗi tên file hình ảnh

4.2.2.8. Order

Tiếp đầu ngữ của API URL: /api/order

4.2.2.8.1. Lấy thông tin 1 đơn hàng

URL: /api/order/{id} – id là mã đơn hàng

Method: GET

Response Body

Success Response:

- Code: 200 OK
- Body: thông tin đơn hàng

Error Response:

- Code 400 Bad Request – Đơn hàng không tồn tại

4.2.2.8.2. Lấy thông tin các đơn hàng theo trạng thái

URL: /api/order

Method: GET

Request params: thông tin truy vấn

Field	Type	Description	Constraint
statusId	long	Mã trạng thái	Not Null

Response Body

Success Response:

- Code: 200 OK
- Body: Danh sách đơn hàng

4.2.2.8.3. Lấy thông tin chi tiết 1 đơn hàng

URL: /api/order/order-detail

Method: GET

Request params: thông tin truy vấn

Field	Type	Description	Constraint
orderId	long	Mã đơn hàng	Not Null

Response Body**Success Response:**

- Code: 200 OK
- Body: Danh sách chi tiết đơn hàng

4.2.2.8.4. Lấy số lượng đơn hàng theo từng trạng thái**URL:** /api/order/report**Method:** GET**Response Body**

- Code: 200 OK
- Body: Thông kê các trạng thái

4.2.2.8.5. Thay đổi trạng thái của 1 đơn hàng**URL:** /api/order/{id} – id là mã đơn hàng**Method:** PUT**Response Body****Success Response:**

- Code: 200 OK
- Body: message thông báo

Error Response:

- Code 400 Bad Request – Đơn đặt hàng có sản phẩm đã được bán hết

4.2.2.8.6. Lấy các đơn hàng của 1 người dùng theo trạng thái**URL:** /api/order/user2/{id} – id là mã người dùng**Method:** GET**Request params:** thông tin truy vấn

Field	Type	Description	Constraint
statusId	long	Mã trạng thái	Not Null

Response Body**Success Response:**

- Code: 200 OK
- Body: Danh sách đơn hàng

Error Response:

- Code 400 Bad Request – Không tìm thấy người dùng
- Code 400 Bad Request – Mã trạng thái không hợp lệ

4.2.2.8.7. Lấy số lượng sản phẩm trong 1 đơn hàng và sản phẩm đầu tiên của đơn hàng đó

URL: /api/order/order-summary

Method: GET

Request params: thông tin truy vấn

Field	Type	Description	Constraint
orderId	long	Mã đơn hàng	Not Null

Response Body

Success Response:

- Code: 200 OK
- Body: Danh sách đơn hàng

4.2.2.8.8. Thêm 1 đơn hàng cho 1 người dùng

URL: /api/order/add/{id} – id là mã người dùng

Method: POST

Body: thông tin đơn hàng

Field	Type	Description	Constraint
date	Date	Ngày đặt hàng	Not Null
address	String	Địa chỉ nhận hàng	Not Null
totalPrice	float	Tổng tiền thanh toán	Not Null
statusId	OrderStatus	Mã trạng thái	Not Null
user	User	Người dùng đặt hàng	Not Null

Response Body

Success Response:

- Code: 200 OK
- Body: message thông báo và thông tin đơn hàng

Error Response:

- Code 400 Bad Request – Người dùng không tồn tại

4.2.2.8.9. Người dùng thêm đơn hàng

URL: /api/order/add

Method: POST

Headers: tokenType + accessToken

Body: thông tin đơn hàng

Field	Type	Description	Constraint
date	Date	Ngày đặt hàng	Not Null

address	String	Địa chỉ nhận hàng	Not Null
totalPrice	float	Tổng tiền thanh toán	Not Null
statusId	OrderStatus	Mã trạng thái	Not Null
user	User	Người dùng đặt hàng	Not Null

Response Body**Success Response:**

- Code: 200 OK
- Body: message thông báo và thông tin đơn hàng

4.2.2.8.10. Mua lại 1 đơn hàng đã hủy**URL:** /api/order/repurchase/{id} – id là mã đơn hàng**Method:** POST**Response Body****Success Response:**

- Code: 200 OK
- Body: message thông báo và danh sách sản phẩm trong đơn hàng đã hủy

Error Response:

- Code 400 Bad Request – Đơn hàng không tồn tại

4.2.2.9. Order Detail

Tiếp đầu ngữ của API URL: /api/order-detail

4.2.2.9.1. Thêm chi tiết đơn hàng vào 1 đơn hàng**URL:** /api/order-detail/{id}**Method:** POST**Body:** danh sách chi tiết đơn đặt hàng

Field	Type	Description	Constraint
product	Product	Sản phẩm	Not Null
order	Order	Đơn đặt hàng	Not Null
quantity	int	Số lượng sản phẩm	Not Null
price	float	Đơn giá	Not Null

Response Body**Success Response:**

- Code: 200 OK
- Body: message thông báo

Error Response:

- Code 400 Bad Request – Đơn hàng không tồn tại

4.2.2.10. Poster

Tiếp đầu ngữ của API URL: /api/posters

4.2.2.10.1. Lấy tất cả poster

URL: /api/posters

Method: GET

Response Body

- Code: 200 OK
- Body: Danh sách poster

4.2.2.10.2. Lấy hình ảnh của 1 poster

URL: /api/posters/{imageName} – imageName là tên hình ảnh poster

Method: GET

Response Body

- Code: 200 OK
- Body: Hình ảnh poster

4.2.2.11. Account

Tiếp đầu ngữ của API URL: /api/users

4.2.2.11.1. Lấy hình ảnh của 1 người dùng

URL: /api/users/image/{imageName} – imageName là tên hình của người dùng

Method: GET

Response Body

Success Response:

- Code: 200 OK
- Body: hình ảnh tài khoản người dùng

4.2.2.11.2. Lấy danh sách người dùng

URL: /api/users

Method: GET

Request params: thông tin truy vấn

Field	Type	Description	Constraint
pageNo	int	Số thứ tự trang	Null
pageSize	int	Kích cỡ trang	Null
sortField	int	Field sắp xếp	Null
sortDirection	String	Hướng sắp xếp	Null

Response Body

Success Response:

- Code: 200 OK

- Body: Danh sách người dùng

4.2.2.11.3. Lấy danh sách người dùng kèm số lượng đơn hàng và địa chỉ của người dùng đó

URL: /api/users/numorders

Method: GET

Request params: thông tin truy vấn

Field	Type	Description	Constraint
numorders	Boolean	Trạng thái tài khoản	Null
pageNo	int	Số thứ tự trang	Null
pageSize	int	Kích cỡ trang	Null
sortField	int	Field sắp xếp	Null
sortDirection	String	Hướng sắp xếp	Null

Response Body

Success Response:

- Code: 200 OK
- Body: Danh sách người dùng

4.2.2.11.4. Lấy thông tin 1 người dùng

URL: /api/users/{id} – id là mã người dùng

Method: GET

Response Body

Success Response:

- Code: 200 OK
- Body: Thông tin người dùng

4.2.2.11.5. Lấy số lượng đơn hàng của 1 người dùng

URL: /api/users/numorder/{id} – id là mã người dùng

Method: GET

Response Body

Success Response:

- Code: 200 OK
- Body: Số lượng đơn hàng

4.2.2.11.6. Chính sửa thông tin cá nhân

URL: /api/users/edit-profile

Method: PUT

Body: thông tin cá nhân mới

Field	Type	Description	Constraint

id	Long	Id của người dùng	Not Null
username	String	Username của người dùng	Not Null
email	String	Email	Not Null
firstName	String	Họ và tên lót	Not Null
lastName	String	Tên	Not Null
phone	String	Số điện thoại	Not Null
image	String	Tên hình ảnh của tài khoản người dùng	Not Null

Response Body**Success Response:**

- Code: 200 OK
- Body: message thông báo

Error Response:

- Code 400 Bad Request – Username, email, số điện thoại đã tồn tại
- 4.2.2.11.7. Thay đổi mật khẩu**

URL: /api/users/change-password**Method:** PUT**Body:** thông tin cá nhân mới

Field	Type	Description	Constraint
id	Long	Id của người dùng	Not Null
oldPassword	String	Mật khẩu cũ	Not Null
newPassword	String	Mật khẩu mới	Not Null

Response Body**Success Response:**

- Code: 200 OK
- Body: message thông báo

Error Response:

- Code 400 Bad Request – Mật khẩu cũ không đúng, độ dài mật khẩu nhỏ hơn 6 ký tự

4.2.2.11.8. Thay đổi trạng thái tài khoản người dùng

URL: /api/users/setStatus**Method:** PUT**Body:** thông tin cá nhân mới

Field	Type	Description	Constraint

id	Long	Id của người dùng	Not Null
username	String	Username	Not Null
email	String	Email	Not Null
firstName	String	Họ và tên đệm	Not Null
lastName	String	Tên	Not Null
phone	String	Số điện thoại	Not Null
image	String	Tên file hình ảnh của tài khoản người dùng	Not Null
numord	int	Số lượng đơn đặt hàng	Not Null
addresses	List<Address>	Danh sách địa chỉ	Not Null

Response Body**Success Response:**

- Code: 200 OK
- Body: message thông báo

4.2.3. Admin API**4.2.3.1. Admin Product API**

Tiếp đầu ngữ của API URL: /api/admin/product

4.2.3.1.1. Thêm sản phẩm**URL:** /api/admin/product**Method:** POST**Body:** Chứa thông tin sản phẩm gồm

Field	Type	Description	Constraint
name	String	Tên sản phẩm	Unique
description	String	Mô tả sản phẩm	Null
image	String	Hình ảnh sản phẩm	Null
price	double	Giá sản phẩm	Not Null, > 0
specification	String	Quy cách	Null
discount	String	Chiết khấu	Null
soldQuantity	String	Số lượng đã bán	Null
category	Category	Thể loại	Not Null
brand	Brand	Thương hiệu	Not Null

status	boolean	Trạng thái	Null
quantity	int	Số lượng	Not Null

Response Body**Success Response:**

- Code: 200 OK
- Body: message thông báo và thông tin sản phẩm

Error Response:

- Code 400 Bad Request – Thông tin không đúng

4.2.3.1.2. Cập nhật sản phẩm**URL:** /api/admin/product/{id} – id sản phẩm cần cập nhật**Method:** PUT**Response Body****Success Response:**

- Code: 200 OK
- Body: message thông báo và thông tin sản phẩm

Error Response:

- Code 400 Bad Request – Thông tin không đúng

4.2.3.1.3. Xóa sản phẩm**URL:** /api/admin/product/{id} – id sản phẩm cần xóa**Method:** DELETE**Response Body****Success Response:**

- Code: 200 OK
- Body: message thông báo

Error Response:

- Code 400 Bad Request – Sản phẩm đang được sử dụng
- Code 404 Not Found – Không tìm thấy sản phẩm

4.2.3.2. Admin Category API

Tiếp đầu ngữ của API URL: /api/admin/category

4.2.3.2.1. Thêm thể loại**URL:** /api/admin/category**Method:** POST**Body:** Chứa thông tin thể loại gồm

Field	Type	Description	Constraint
name	String	Tên thể loại	Unique
note	String	Mô tả thể loại	Null

image	String	Hình ảnh thể loại	Null
-------	--------	-------------------	------

Response Body**Success Response:**

- Code: 200 OK
- Body: message thông báo và thông tin sản phẩm

Error Response:

- Code 400 Bad Request – Thông tin không đúng

4.2.3.2.2. Cập nhật thể loại**URL:** /api/admin/category/{id} – id thể loại cần cập nhật**Method:** PUT**Response Body****Success Response:**

- Code: 200 OK
- Body: message thông báo và thông tin thể loại

Error Response:

- Code 400 Bad Request – Thông tin không đúng

4.2.3.2.3. Xóa thể loại**URL:** /api/admin/category/{id} – id thể loại cần xóa**Method:** DELETE**Response Body****Success Response:**

- Code: 200 OK
- Body: message thông báo

Error Response:

- Code 400 Bad Request – Thể loại đang được sử dụng
- Code 404 Not Found – Không tìm thấy thể loại

4.2.3.3. Admin Brand API

Tiếp đầu ngữ của API URL: /api/admin/brand

4.2.3.3.1. Thêm thương hiệu**URL:** /api/admin/brand**Method:** POST**Body:** Chứa thông tin thương hiệu gồm

Field	Type	Description	Constraint
name	String	Tên thương hiệu	Unique
description	String	Mô tả thể loại	Null

Response Body

Success Response:

- Code: 200 OK
- Body: message thông báo và thông tin thương hiệu

Error Response:

- Code 400 Bad Request – Thông tin không đúng

4.2.3.3.2. Cập nhật thương hiệu**URL:** /api/admin/brand/{id} – id thương hiệu cần cập nhật**Method:** PUT**Response Body*****Success Response:***

- Code: 200 OK
- Body: message thông báo và thông tin thương hiệu

Error Response:

- Code 400 Bad Request – Thông tin không đúng

4.2.3.3.3. Xóa thương hiệu**URL:** /api/admin/brand/{id} – id thương hiệu cần xóa**Method:** DELETE**Response Body*****Success Response:***

- Code: 200 OK
- Body: message thông báo

Error Response:

- Code 400 Bad Request – Thương hiệu đang được sử dụng
- Code 404 Not Found – Không tìm thấy thương hiệu

4.2.3.4. Admin Poster

Tiếp đầu ngữ của API URL: /api/admin/poster

4.2.3.4.1. Thêm 1 poster**URL:** /api/admin/poster**Method:** POST**Body:** Thông tin poster

Field	Type	Description	Constraint
name	String	Tên file hình ảnh poster	Not Null
type	int	Loại poster	Not Null
isActive	boolean	Có đang sử dụng không	Not Null

Response Body***Success Response:***

- Code: 200 OK
- Body: message thông báo và thông tin poster

4.2.3.4.2. Xóa 1 poster

URL: /api/admin/poster/{ id } – id là mã poster

Method: DELETE

Response Body

Success Response:

- Code: 200 OK
- Body: message thông báo

4.2.3.5. Admin User

Tiếp đầu ngữ của API URL: /api/admin/user

4.2.3.5.1. Thêm, cập nhật 1 tài khoản người dùng

URL: /api/admin/user/add

Method: PUT

Body: Chứa thông tin sản phẩm gồm

Field	Type	Description	Constraint
id	Long	Mã người dùng	Unique
username	String	Username	Not Null
email	String	Email	Not Null
password	String	Mật khẩu	Not Null
firstName	String	Họ và tên đệm	Null
lastName	String	Tên	Null
phone	String	Số điện thoại	Not Null
image	String	Tên file hình ảnh tài khoản	Null
status	boolean	Trạng thái tài khoản	Not Null

Response Body

Success Response:

- Code: 200 OK
- Body: message thông báo

Error Response:

- Code 400 Bad Request – Username đã tồn tại, email, số điện thoại đã được sử dụng

CHƯƠNG 5. KẾT LUẬN VÀ KIẾN NGHỊ

5.1. Kết quả đạt được

- Phần mềm có thể triển khai ở cửa hàng bán đồ dùng văn phòng phẩm
- Phần mềm có giao diện dễ nhìn, thân thiện với người sử dụng.
- Sau khi nghiên cứu, em đã xây dựng được một phần mềm quản lý bán đồ dùng văn phòng phẩm, thay thế cách quản lý thủ công bằng tay, mất rất nhiều thời gian và tốn công sức của người quản lý.

5.2. Hạn chế của đề tài

Phần mềm xây dựng website bán đồ dùng văn phòng phẩm tuy đã hoàn thành song vẫn không tránh khỏi những thiếu sót cũng như những hạn chế nhất định:

- Chưa thanh toán online qua nhiều loại ví điện tử, tài khoản ngân hàng (hiện tại chỉ thanh toán được trên MoMo).
- Chưa có nhiều các loại thống kê doanh số, bán hàng.
- Kỹ năng phân tích hệ thống còn nhiều hạn chế.

5.3. Kiến nghị

- Tiếp tục bổ sung những chức năng mà phần mềm chưa có để phần mềm có thể ứng dụng rộng rãi trong mọi cửa hàng (chức năng thanh toán online qua tài khoản ngân hàng,...)
- Tiếp tục hoàn thiện chương trình hi vọng có thể đáp ứng được mọi nhu cầu của người sử dụng.
- Mở rộng ra thêm nhiều ứng dụng liên kết với Web Service như Win App, Web App,... giúp cho người dùng có thêm nhiều sự lựa chọn

TÀI LIỆU THAM KHẢO

- 1. Java**, <https://topdev.vn/blog/tong-quan-ve-ngon-ngu-lap-trinh-java/>
- 2. Spring**, <https://spring.io/projects>
- 3. Retrofit**, <https://topdev.vn/blog/retrofit-trong-android/>
- 4. MySQL**, <http://dbahire.com/mysql-workbench-la-gi-tai-sao-phai-cai-dat-mysql-workbench>
- 5. RESTful**, <https://topdev.vn/blog/restful-api-la-gi/>
- 6. XML trong Android**, <https://developer.android.com/codelabs/basic-android-kotlin-training-xml-layouts>