



BÁO CÁO TỐT NGHIỆP ĐẠI HỌC

**Đề tài: “Xây dựng hệ thống quản lý mua bán thực phẩm tươi sống
và nhu yếu phẩm”**

Người hướng dẫn	:	ThS. HUỖNH TRUNG TRỤ
Lớp	:	D18CQCP02-N
Khóa	:	2018-2023
Hệ	:	ĐẠI HỌC CHÍNH QUY

Nhóm 8

Nguyễn Ngọc Minh Tú – N18DCCN192

Trần Minh Phúc – N18DCCN156

TP.HCM, tháng 11/2023

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



BÁO CÁO TỐT NGHIỆP
ĐẠI HỌC

**Đề tài: “Xây dựng hệ thống quản lý mua bán thực phẩm tươi sống
và nhu yếu phẩm”**

Người hướng dẫn	:	ThS. HUỖNH TRUNG TRỤ
Lớp	:	D18CQCP02-N
Khóa	:	2018-2023
Hệ	:	ĐẠI HỌC CHÍNH QUY

Nhóm 8

Nguyễn Ngọc Minh Tú – N18DCCN192

Trần Minh Phúc – N18DCCN156

TP.HCM, tháng 11/2023

MỤC LỤC

Chương 1: GIỚI THIỆU ĐỀ TÀI.....	12
1.1 Tổng quan đề tài.....	12
1.2 Mục đích.....	12
1.3 Mục tiêu.....	12
Chương 2: KHẢO SÁT HIỆN TRẠNG	13
Chương 3: CƠ SỞ KHOA HỌC CỦA ĐỀ TÀI.....	18
3.1 Công nghệ hỗ trợ.....	18
3.1.1 Xây dựng Website với NextJS.....	18
3.1.2 Ánh xạ dữ liệu với TypeORM.....	20
3.1.3 Cơ sở dữ liệu PostgreSQL	22
3.1.4 Bản đồ với MapBox API	23
3.1.5 Thanh toán trực tuyến với Paypal.....	26
3.1.6 Thông tin địa chỉ với Province-API.....	29
Chương 4: PHÂN TÍCH HỆ THỐNG.....	31
4.1 Sơ đồ phân rã cấp chức năng mức đỉnh.....	31
4.2 Sơ đồ phân rã chức năng	31
4.3 Định nghĩa yêu cầu và ràng buộc đối với phần mềm.....	35
4.3.1 Các yêu cầu từ môi trường nghiệp vụ.....	35
4.3.2 Các yêu cầu từ môi trường vận hành	37
4.3.3 Các yêu cầu từ môi trường phát triển	38
4.4 Mô hình luồng dữ liệu.....	39
4.5 Mô hình Use case.....	39
Chương 5: THIẾT KẾ PHẦN MỀM.....	62
5.1 Mô hình công nghệ.....	62
5.2 Sơ đồ ứng dụng.....	65
5.3 Thiết kế giao diện chức năng.....	67
5.4 Thiết kế cơ sở dữ liệu cho phần mềm.....	68
5.4.1 Thiết kế ERD	68
5.4.2 Thiết kế Lược đồ dạng chuẩn 3	69
Chương 6: Kịch BẢN KIỂM THỬ	84

Chương 7: KẾT LUẬN	85
7.1 Kết quả đạt được	85
7.2 Các hạn chế	85
7.3 Hướng phát triển	85
Chương 8: PHỤ LỤC	86

KÍ HIỆU CÁC CỤM TỪ VIẾT TẮT

Từ viết tắt	Từ đầy đủ	Giải thích / Dịch
API	Application Programming Interface	Giao diện lập trình ứng dụng
UUID	Universally Unique Identifier	Định danh duy nhất toàn cầu, 1 chuỗi ký tự gồm 128 bit, thường dùng trong ID để định danh độc nhất 1 thực thể nào đó
SSR	Server-side Rendering	Server gửi trực tiếp nội dung web (HTML/CSS) cho client
SSG	Static Site Generation	Nói về việc chuẩn bị sẵn trang web tĩnh cho client để tối ưu về tốc độ, thường sử dụng chung với SSR
SPA	Single Page Application	Ứng dụng một trang, mọi thao tác của người dùng đều thực hiện trên một trang duy nhất, mọi cấu trúc trang sẽ chỉ tải một lần và không tải lại khi chuyển trang. Giúp đem lại trải nghiệm mượt mà.
CRM	Customer relationship management	Hệ thống giúp doanh nghiệp quản lý và phân tích tương tác với khách hàng, hỗ trợ trong việc theo dõi thông tin liên lạc, quản lý giao dịch, cải thiện dịch vụ khách hàng và thúc đẩy chiến lược tiếp thị dựa trên dữ liệu khách hàng.
UI	User Interface	Giao diện người dùng
UX	User experience	Trải nghiệm người dùng
HT	Hệ thống	
KH	Khách hàng	
QLCH	Quản lý cửa hàng	
QLK	Quản lý kho	
NVK	Nhân viên kho	
NVGH	Nhân viên giao hàng	
TN	Thu ngân	

CHÚ GIẢI HÌNH

Hình 2.1 Mô hình cộng tác trước khi sử dụng phần mềm.....	14
Hình 2.2 Mô hình cộng tác sau khi có sự hỗ trợ của phần mềm	15
Hình 2.3 Sơ đồ cơ cấu tổ chức của dự án	16
Hình 4.1 Sơ đồ cơ cấu tổ chức của dự án	31
Hình 4.2 Sơ đồ phân rã chức năng Mua hàng	31
Hình 4.3 Sơ đồ phân rã chức năng Quản lý thông tin cửa hàng.....	32
Hình 4.4 Sơ đồ phân rã chức năng Quản lý mặt hàng.....	33
Hình 4.5 Sơ đồ phân rã chức năng Thống kê	33
Hình 4.6 Sơ đồ phân rã chức năng Thống kê	34
Hình 4.7 Sơ đồ phân rã chức năng Truy cập hệ thống	35
Hình 4.8 Sơ đồ tuần tự Use case Xem thông tin sản phẩm	43
Hình 4.9 Use case Thêm sản phẩm vào giỏ hàng.....	44
Hình 4.10 Sơ đồ tuần tự Use case thêm sản phẩm vào giỏ hàng	46
Hình 4.11 Use case Đặt hàng trực tuyến	47
Hình 4.12 Sơ đồ tuần tự Use case đặt hàng.....	50
Hình 4.13 Sơ đồ tuần tự Thanh toán PayPal cho Use case đặt hàng.....	51
Hình 4.14 Use case Hủy đơn hàng	52
Hình 4.15 Đặc tả Use case Hủy đơn hàng.....	53
Hình 4.16 Sơ đồ tuần tự Use case Hủy đơn hàng.....	54
Hình 4.17 Use case Thêm mới thông tin sản phẩm.....	55
Hình 4.18 Sơ đồ tuần tự Use case Thêm mới thông tin sản phẩm	57
Hình 4.19 Use case Thêm mới thông tin nhân viên	57
Hình 4.20 Sơ đồ tuần tự Use case Thêm mới thông tin nhân viên.....	59
Hình 4.21 Use case Nhập thêm sản phẩm từ nhà cung cấp.....	60
Hình 4.22 Sơ đồ tuần tự Use case Nhập thêm sản phẩm từ nhà cung cấp	61
Hình 5.1 Mô hình kiến trúc lớp (layered architecture).....	62
Hình 5.2 Các lớp trong Mô hình kiến trúc lớp (layered architecture).....	63
Hình 5.3 Sơ đồ ứng dụng.....	65
Hình 5.4 Lược đồ ERD (https://shorturl.at/hjCO1).....	68

Hình 5.5 Lược đồ Database (https://shorturl.at/dlwJ6)	69
---	----

CHÚ GIẢI BẢNG

Bảng 4.1 Các yêu cầu từ môi trường nghiệp vụ.....	36
Bảng 4.2 Các yêu cầu từ môi trường vận hành - Yêu cầu chức năng	37
Bảng 4.3 Các yêu cầu từ môi trường vận hành - Yêu cầu phi chức năng	38
Bảng 4.4 Sơ đồ phân rã chức năng Truy cập hệ thống.....	39
Bảng 4.5 Use case chung cho tất cả actor	40
Bảng 4.6 Use case tổng quát cho khách hàng online	40
Bảng 4.7 Use case tổng quát cho nhân viên thu ngân	41
Bảng 4.8 Use case tổng quát cho nhân viên giao hàng	41
Bảng 4.9 Use case tổng quát cho Quản lý cửa hàng	42
Bảng 4.10 Đặc tả Use case xem thông tin sản phẩm.....	43
Bảng 4.11 Đặc tả Use case Thêm sản phẩm vào giỏ hàng.....	45
Bảng 4.12 Đặc tả Use case Đặt hàng trực tuyến	49
Bảng 4.13 Đặc tả Use case Thêm mới thông tin sản phẩm.....	56
Bảng 4.14 Đặc tả Use case Thêm mới thông tin nhân viên	59
Bảng 4.15 Đặc tả Use case Nhập thêm sản phẩm từ nhà cung cấp.....	61
Bảng 5.1 Table product_category.....	70
Bảng 5.2 Bảng supplier	71
Bảng 5.3 Table product	72
Bảng 5.4 Table product	72
Bảng 5.5 Table product_image.....	72
Bảng 5.6 Table import_order	73
Bảng 5.7 Table export_order	74
Bảng 5.8 Table account	75
Bảng 5.9 Table employee	76
Bảng 5.10 Table customer.....	77
Bảng 5.11 Table customer_address.....	78
Bảng 5.12 Table customer_order	80
Bảng 5.13 Table customer_order_item	81
Bảng 5.14 Table cart_item	81

Bảng 5.15 Table store.....	82
Bảng 5.16 Table store_work_time	83

TRÁCH NHIỆM TRONG ĐỀ TÀI

TT	CÔNG VIỆC	BẮT ĐẦU	KẾT THÚC	THÀNH VIÊN	TÌNH TRẠNG
1	Phân tích yêu cầu khách hàng				
1,1	Vẽ sơ đồ tổng quan hệ thống	25/9/2023	01/10/2023	Cả nhóm	Hoàn thành
1,2	Vẽ sơ đồ use case	25/9/2023	01/10/2023	Cả nhóm	Hoàn thành
1,3	Xây dựng bản đặc tả yêu cầu hệ thống	25/9/2023	01/10/2023	Cả nhóm	Hoàn thành
1,4	Mô tả các quy trình nghiệp vụ	25/9/2023	01/10/2023	Cả nhóm	Hoàn thành
2	Thiết kế hệ thống				
2,1	Phác thảo mô hình công nghệ ứng dụng	02/10/2023	11/10/2023	Phúc	Hoàn thành
2,2	Thiết kế giao diện	02/10/2023	11/10/2023		
2.2.1	Phác thảo sơ đồ ứng dụng	02/10/2023	11/10/2023	Phúc	Hoàn thành
2.2.2	Phác thảo layout	02/10/2023	11/10/2023	Phúc	Hoàn thành
2.2.3	Phác thảo các giao diện chức năng	02/10/2023	11/10/2023	Tú	Hoàn thành
2,3	Thiết kế dữ liệu	02/10/2023	11/10/2023	Cả nhóm	Hoàn thành
2.3.1	Thiết kế sơ đồ quan hệ thực thể (ERD)	02/10/2023	11/10/2023	Phúc	Hoàn thành
2.3.2	Thiết kế chi tiết thực thể	02/10/2023	11/10/2023	Tú	Hoàn thành
2.3.3	Thiết kế sơ đồ lớp (Class Diagram)	02/10/2023	11/10/2023	Tú	Hoàn thành
3	Thực hiện dự án				
3,1	Thiết kế cơ sở dữ liệu	11/10/2023	25/10/2023	Cả nhóm	Hoàn thành
3,2	Thiết kế giao diện	11/10/2023	25/10/2023	Tú	Hoàn thành
3,3	Xây dựng tiện ích cho dự án	11/10/2023	25/10/2023	Tú	Hoàn thành
3,4	Xây dựng các lớp thực thể Entity	11/10/2023	25/10/2023	Cả nhóm	Hoàn thành
3,5	Xây dựng các lớp truy xuất dữ liệu (DAO)	11/10/2023	25/10/2023	Cả nhóm	Hoàn thành
3,6	Lập trình liên kết các trang web chức năng	11/10/2023	25/10/2023	Tú	Hoàn thành
3,7	Lập trình cho các chức năng nghiệp vụ	11/10/2023	25/10/2023	Phúc	Hoàn thành
3,8	Lập trình tổng hợp - thống kê	11/10/2023	25/10/2023	Cả nhóm	Hoàn thành
4	Kiểm thử				
4,1	Xây dựng kịch bản kiểm thử	25/10/2023	01/11/2023	Phúc	Hoàn thành
4,2	Thực hiện kiểm thử	25/10/2023	01/11/2023	Tú	Hoàn thành
4,3	Lập trình sửa lỗi	25/10/2023	01/11/2023	Tú	Hoàn thành
5	Đóng gói và triển khai				
5,1	Đóng gói sản phẩm	10/11/2023	12/11/2023	Tú	Hoàn thành
5,2	Viết tài liệu hướng dẫn sử dụng	01/11/2023	12/11/2023	Cả nhóm	Hoàn thành

LỜI MỞ ĐẦU

Nhóm em xin gửi lời cảm ơn chân thành đến các thầy cô của Khoa Công nghệ Thông tin 2 vì đã tạo điều kiện thuận lợi cho nhóm em thực hiện bài báo cáo thực tập này. Bên cạnh đó, nhóm em cũng muốn gửi lời cảm ơn đến thầy hướng dẫn đã luôn hỗ trợ và giúp đỡ chúng em hoàn thành báo cáo tốt nhất có thể.

Nhóm em nhận thức được rằng bài báo cáo của mình có thể còn nhiều thiếu sót, vì vậy nhóm em mong muốn nhận được sự tiếp tục hỗ trợ, đóng góp ý kiến, phản hồi và hướng dẫn từ các thầy cô trong khoa và đơn vị thực tập để bài báo cáo của nhóm em được hoàn thiện hơn.

TP.HCM, tháng 03 năm 2023

Nhóm sinh viên thực hiện

Nhóm 8

Nguyễn Ngọc Minh Tú
Trần Minh Phúc

CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI**1.1 Tổng quan đề tài**

Đề tài “*Xây dựng hệ thống quản lý mua bán thực phẩm tươi sống và nhu yếu phẩm*” là loại đề tài xây dựng ra một phần mềm ứng dụng phục vụ vào 2 việc chính:

- Bán thực phẩm tươi sống , nhu yếu phẩm trực tuyến và ngoại tuyến
- Kiểm soát việc thu mua, nhập sản phẩm tồn kho

1.2 Mục đích

- Phát triển một ứng dụng thương mại điện tử linh hoạt và đáng tin cậy để cung cấp các sản phẩm chất lượng đến tay khách hàng một cách thuận tiện và nhanh chóng, thông qua cả kênh trực tuyến và trực tiếp.
- Ứng dụng khoa học công nghệ để tối ưu hóa quy trình quản lý hàng tồn kho, đơn đặt hàng và giao hàng, nhằm nâng cao hiệu suất của cửa hàng, qua đó tăng sự hài lòng của cả nhân viên và khách hàng.

1.3 Mục tiêu

Xây dựng một website thân thiện, dễ sử dụng với người dùng để phục vụ cho cả khách hàng lẫn nhân viên cửa tiệm, cụ thể như sau:

- Nhân viên quản lý cửa hàng có một bộ công cụ để quản lý, hiển thị, báo cáo và phân tích đầy đủ các thông tin cần thiết nhằm hỗ trợ trong quá trình điều hành cửa tiệm.
- Nhân viên quản lý kho dễ dàng kiểm soát các thông tin, nghiệp vụ quản lý kho, như nhập và xuất hàng, quản lý nhà thông tin nhà cung cấp,... hạn chế các lỗi do tính toán thủ công.
- Nhân viên giao hàng có thể dễ dàng rà soát các đơn hàng cần giao và cập nhật trạng thái nhanh chóng.
- Hỗ trợ nhân viên thu ngân kiểm soát thông tin sản phẩm trong quá trình bán hàng tại quầy và cập nhật đơn hàng trực tuyến.
- Khách hàng có thể tìm kiếm, chọn mua các sản phẩm uy tín nhờ hệ thống đánh giá sản phẩm và phản hồi, quản lý tình trạng đơn hàng một cách dễ dàng và nhanh chóng

CHƯƠNG 2: KHẢO SÁT HIỆN TRẠNG**2.1 Hiện trạng thực tế:**

Hiện nay, nhu cầu mua thực phẩm tươi sống và nhu yếu phẩm trực tuyến đã trở thành một xu hướng phổ biến. Người tiêu dùng có nhu cầu mua sắm trực tuyến với mục tiêu tiết kiệm thời gian, thuận tiện và đảm bảo chất lượng sản phẩm.

- **Chất lượng và nguồn gốc sản phẩm:** Người tiêu dùng quan tâm đến chất lượng và nguồn gốc của sản phẩm mà họ mua. Họ mong muốn được cung cấp thông tin chi tiết về nguồn gốc, quy trình sản xuất, hạn sử dụng và các tiêu chuẩn về an toàn thực phẩm.
- **Dễ dàng tìm kiếm và đặt hàng:** Người tiêu dùng mong muốn có một dịch vụ mua sắm dễ sử dụng, có tính tương tác cao và hỗ trợ tiện ích tìm kiếm sản phẩm. Họ muốn có khả năng tìm kiếm nhanh chóng, so sánh giá cả và đặt hàng một cách dễ dàng.
- **Tính linh hoạt trong lựa chọn:** Người tiêu dùng mong muốn có sự linh hoạt trong việc lựa chọn loại sản phẩm, số lượng và thời gian giao hàng. Họ có nhu cầu tùy chỉnh đơn hàng và được cung cấp nhiều phương thức thanh toán linh hoạt.

Có thể thấy được, người tiêu dùng hiện nay đang tìm kiếm một trải nghiệm mua sắm trực tuyến tiện lợi, đáng tin cậy và đảm bảo chất lượng sản phẩm. Tuy nhiên, nhiều Nhà bán hàng thực phẩm hiện nay vẫn chưa thực sự đáp ứng được nhu cầu mua trực tuyến của người tiêu dùng hiện nay với những phương thức bán hàng trực tuyến kém hiệu quả như bán trực tuyến thông qua số điện thoại, fanpage hay quản lý sổ sách bằng các bảng tính Excel/Google sheet. Điều này mang lại nhiều bất lợi lớn cho nhà bán hàng lẫn khách hàng:

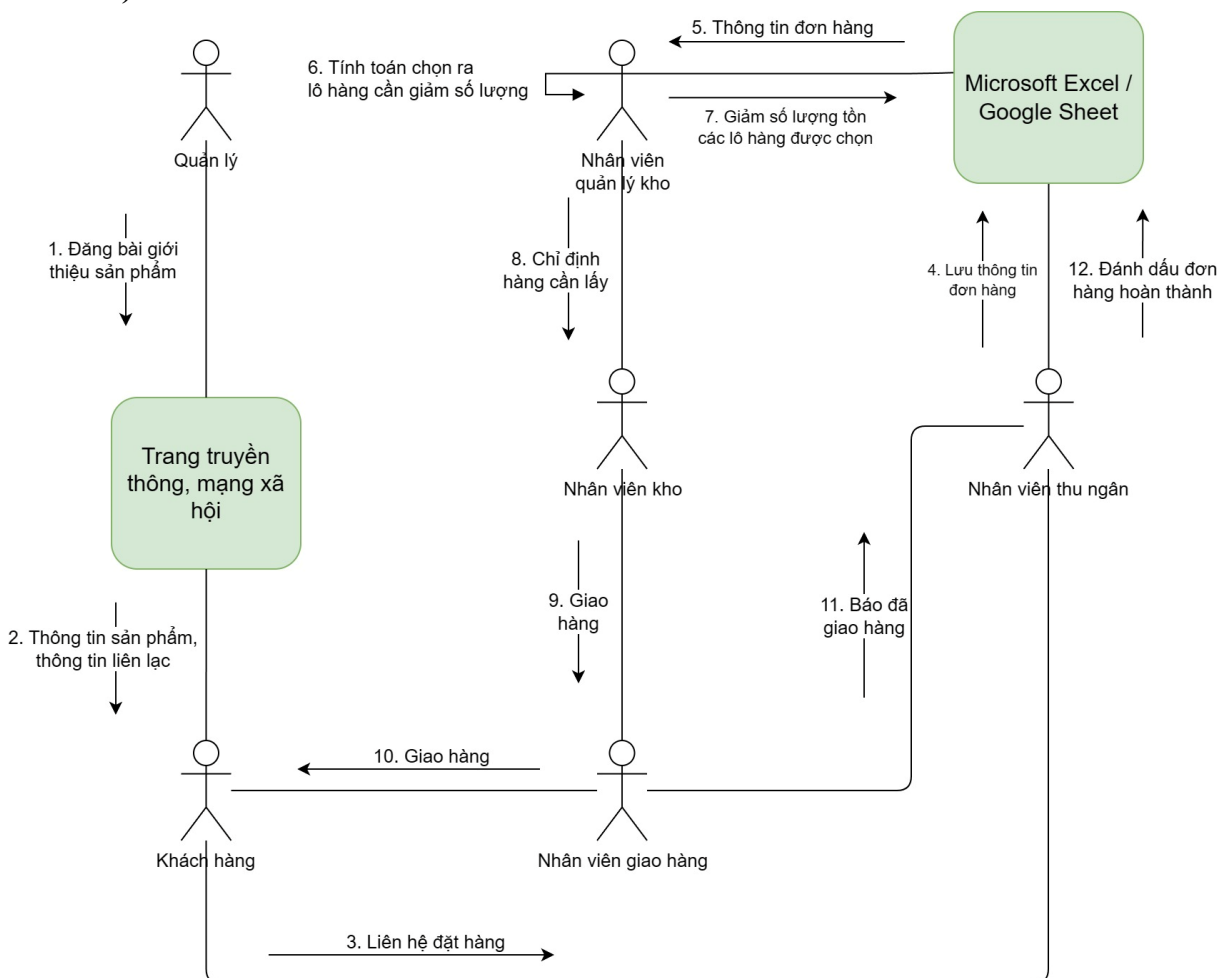
- **Giới hạn khả năng xử lý đơn hàng:** Phương thức đặt hàng qua điện thoại hay fanpage có thể gây ra hiểu nhầm hoặc sai sót trong việc ghi lại thông tin đơn hàng. Điều này có thể dẫn đến sự nhầm lẫn trong việc giao hàng hoặc thanh toán.
- **Khả năng giao tiếp hạn chế:** Giao tiếp qua điện thoại hoặc fanpage có thể gây ra sự hiểu lầm hoặc thiếu rõ ràng trong việc trao đổi thông tin giữa khách hàng và nhân viên của cửa hàng. Điều này có thể ảnh hưởng đến trải nghiệm mua hàng của khách hàng.
- **Quản lý sổ sách không hiệu quả:** Sử dụng bảng tính để quản lý thông tin cửa hàng, đơn hàng của khách hàng, thông tin sản phẩm, nhà cung cấp,... có thể gây khó khăn trong việc tìm kiếm, cập nhật hoặc theo dõi thông tin. Điều này có thể dẫn đến việc mất mát dữ liệu hoặc sai sót trong quá trình quản lý.
- **Thiếu tính tự động hóa:** Với cách thức bán hàng qua số điện thoại, fanpage hay quản lý sổ sách bằng các bảng tính Excel/Google sheet yêu cầu nhân viên phải

thực hiện mọi giao dịch và công việc một cách thủ công, từ việc tiếp nhận đơn hàng đến quản lý thông tin và cập nhật kho hàng. Điều này có thể làm tăng thời gian và công sức cần thiết để hoàn thành các tác vụ.

Như vậy để cải thiện trải nghiệm mua sắm và quản lý, cửa hàng cần đầu tư vào trang web, quảng cáo trực tuyến, tích hợp hệ thống quản lý mua bán để cải thiện dịch vụ khách hàng, cũng như tối ưu vận hành cho Nhà bán hàng.

2.2 Giải pháp

a) Vấn đề



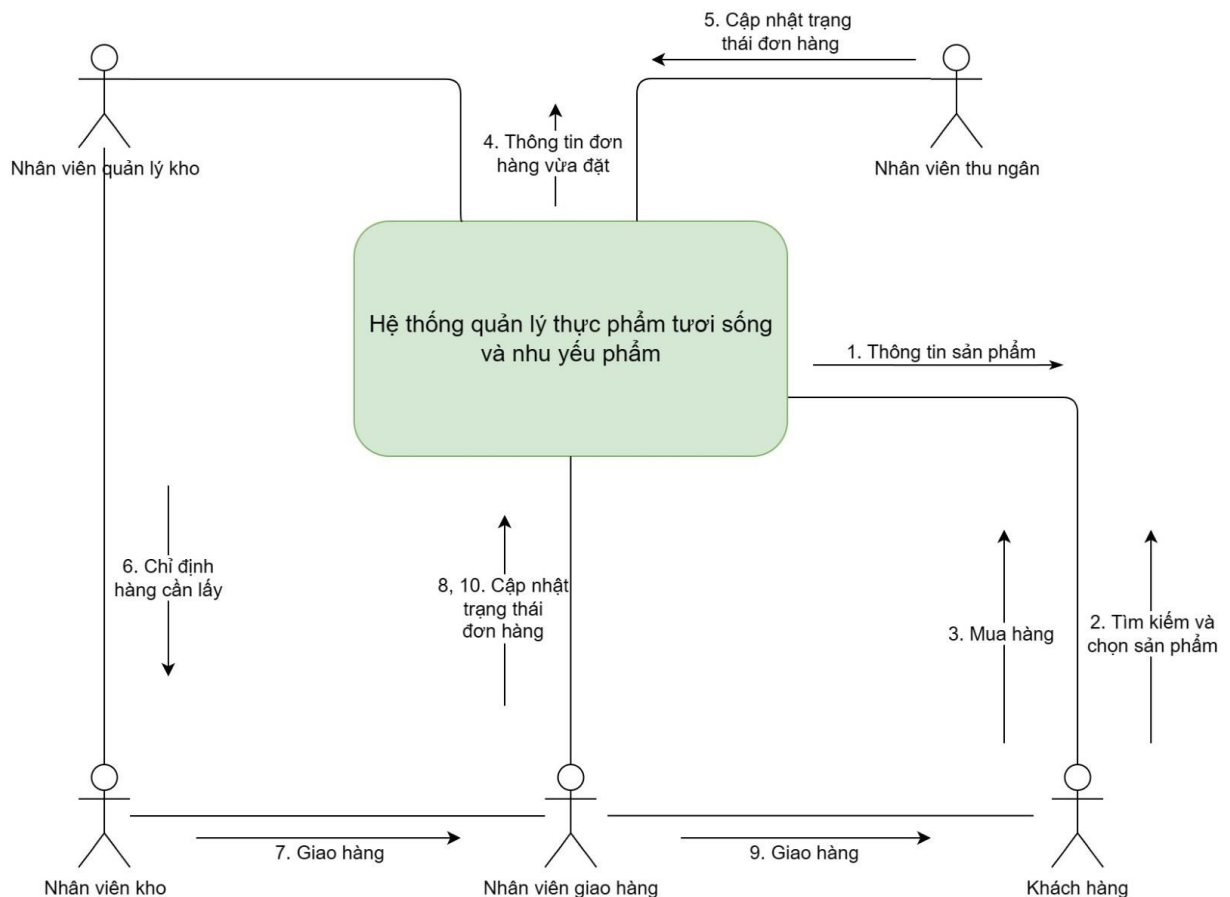
Hình 2.1 Mô hình cộng tác trước khi sử dụng phần mềm

Mô tả:

1. Nhân viên quản lý đăng bài giới thiệu cửa hàng, sản phẩm trên các trang truyền thông, mạng xã hội
2. Khách hàng nhận được các thông tin về sản phẩm, thông tin cửa hàng
3. Khách hàng tìm kiếm thủ công các sản phẩm ưa thích và liên hệ cửa hàng qua thông tin được hiển thị để đặt hàng

4. Nhân viên thu ngân tiếp nhận đơn hàng và lưu thông tin vào bảng tính
5. Quản lý kho kiểm tra thông tin bảng tính
6. Quản lý kho thực hiện các tính toán thủ công cần thiết để chọn ra các lô hàng nhập cần xuất để bán cho khách hàng
7. Quản lý kho thực hiện điều chỉnh dữ liệu trong bảng tính theo kết quả tính toán được
8. Quản lý kho chỉ định nhân viên kho đi lấy các món hàng theo yêu cầu
9. Nhân viên kho giao hàng cho nhân viên giao hàng
10. Nhân viên giao hàng đi giao sản phẩm cho khách
11. Nhân viên giao hàng liên hệ thu ngân thông báo đã hoàn thành đơn hàng
12. Nhân viên thu ngân đánh dấu đơn hàng đã hoàn thành trong bảng tính

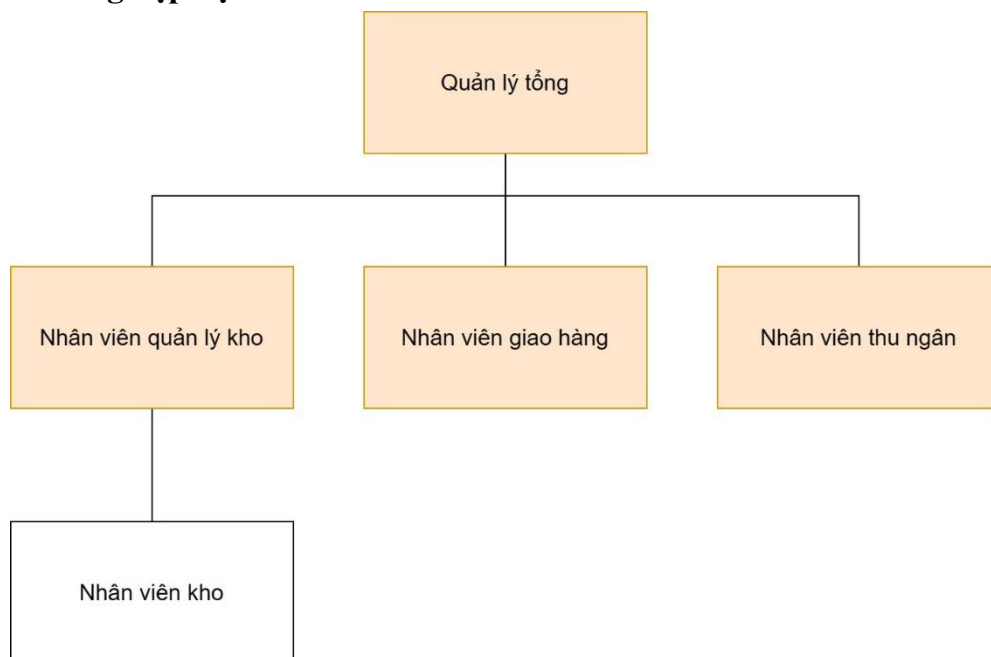
b) Giải pháp



Hình 2.2 Mô hình cộng tác sau khi có sự hỗ trợ của phần mềm

Mô tả:

1. Khách hàng truy cập trang web, nhận được thông tin các sản phẩm của cửa hàng
2. Khách hàng tìm kiếm và thêm sản phẩm vào giỏ hàng
3. Khách hàng gửi yêu cầu mua hàng/thanh toán giỏ hàng
4. Quản lý kho và thu ngân nhận được thông tin đơn hàng vừa đặt
5. Thu ngân tiến hành xác nhận và cập nhật trạng thái đơn hàng lên “Đang xử lý”
6. Quản lý kho chỉ định nhân viên hàng cần lấy
7. Nhân viên kho đưa hàng cần lấy cho nhân viên giao hàng
8. Nhân viên giao hàng cập nhật trạng thái đơn hàng lên “Đang vận chuyển”
9. Nhân viên giao hàng tiến hành vận chuyển và giao hàng cho khách
10. Nhân viên giao hàng cập nhật trạng thái đơn hàng lên “Đã giao hàng”

2.3 Mô hình nghiệp vụ

Hình 2.3 Sơ đồ cơ cấu tổ chức của dự án

Sơ lược về các vai trò trong tổ chức:

- **Quản lý tổng:** Người điều hành quản lý cửa hàng, quản lý các thông tin trọng yếu như các mặt hàng muốn bán, thông tin liên lạc và giờ làm việc, thông tin nhân viên, phản hồi đánh giá của khách hàng,...giám sát tình hình kinh doanh và các thông tin khác của cửa hàng, nhưng ít có quyền chỉnh sửa trực tiếp.

- **Nhân viên quản lý kho:** Chịu trách nhiệm về số lượng tồn kho của các mặt hàng, quản lý thông tin nhà cung cấp, chỉ đạo nhân viên kho lấy các mặt hàng cần thiết.
- **Nhân viên kho:** Lấy hàng theo chỉ đạo của quản lý kho và giao cho nhân viên giao hàng, không có vai trò trong hệ thống website.
- **Nhân viên thu ngân:** Xác nhận đơn hàng trực tuyến
- **Nhân viên giao hàng:** Chịu trách nhiệm trực tiếp việc giao hàng và cập nhật tình trạng đơn hàng cho khách.

CHƯƠNG 3: CƠ SỞ KHOA HỌC CỦA ĐỀ TÀI**3.1 Công nghệ hỗ trợ****3.1.1 Xây dựng Website với NextJS****a) Giới thiệu**

Next.js là một framework phát triển ứng dụng web full-stack, bao gồm cả phần frontend và backend.

b) Lý do lựa chọn

Với khả năng cung cấp cả phần frontend và backend, Next.js giúp giảm sự phân tách giữa các phần của ứng dụng web và cung cấp một cách tiếp cận tốt để phát triển ứng dụng full-stack. Điều này giúp tối ưu nguồn lực, tăng tốc độ phát triển, giảm độ phức tạp và cung cấp sự linh hoạt cho việc xây dựng các ứng dụng web đáng tin cậy và mạnh mẽ.

Front end với NextJS

- Hỗ trợ hệ thống routing đơn giản và dễ sử dụng, cho phép bạn tạo các trang và định nghĩa tuyến đường (route) dễ dàng thông qua tệp tin trong thư mục `pages`.
- Tối ưu hóa SEO: Sử dụng các tính năng SSR và SSG của Next.js để render trang web từ phía máy chủ hoặc tạo các trang tĩnh trước đó, cải thiện hiệu suất và khả năng tìm kiếm của ứng dụng web.

Back end với NextJS

- Next.js được xây dựng trên nền tảng NodeJS. Điều này cho phép NextJS kế thừa toàn bộ điểm mạnh của Node như khả năng xử lý mạnh mẽ, tính mở rộng.
- Next.js cung cấp khả năng tạo API Routes, cho phép bạn xây dựng các endpoint API trên cùng máy chủ Next.js của bạn.
- NextJS xử lý các yêu cầu từ phía client, gọi đến cơ sở dữ liệu, xử lý logic và trả về dữ liệu thông qua API Routes.
- API Routes trong Next.js giúp xây dựng các chức năng phía server và tương tác với các dịch vụ bên ngoài.

c) Sử dụng NextJS

Tài liệu tham khảo: <https://nextjs.org/docs>

Bước 1: Khởi tạo và cấu hình dự án với NextJS

```
npx create-next-app@latest my-next-app  
cd my-next-app
```

Bước 2: Sử Dụng `getStaticProps` để lấy dữ liệu tĩnh

Trong `pages/index.js`, sử dụng `getStaticProps` để fetching dữ liệu:

```
// pages/index.js
export async function getStaticProps() {
  // Fetch data từ một API, file, etc.
  const data = { /* ... */ };

  // Truyền dữ liệu qua props
  return { props: { data } };
}

function HomePage({ data }) {
  return <div>Welcome to NextJS! Data:
{JSON.stringify(data)}</div>;
}

export default HomePage;
```

Bước 3: Sử Dụng `getServerSideProps` cho Server-side Rendering

Trong một trang khác, ví dụ `pages/about.js`, sử dụng `getServerSideProps`:

```
// pages/about.js
export async function getServerSideProps() {
  // Fetch data từ một API, database, etc.
  const data = { /* ... */ };

  // Truyền dữ liệu qua props
  return { props: { data } };
}

function AboutPage({ data }) {
  return <div>About NextJS! Data: {JSON.stringify(data)}</div>;
}

export default AboutPage;
```

Bước 4: Tạo API Endpoint

Trong thư mục `pages/api`, tạo file `hello.js`:

```
// pages/api/hello.js
```

```
export default function handler(req, res) {  
  res.status(200).json({ message: 'Hello from NextJS!' });  
}
```

Truy cập API này qua đường dẫn `/api/hello`.

3.1.2 Ánh xạ dữ liệu với TypeORM

a) Giới thiệu

TypeORM là một ORM (Object-Relational Mapping) cho TypeScript và JavaScript, được thiết kế để làm việc với cơ sở dữ liệu quan hệ.

b) Lý do sử dụng

- ORM cho TypeScript/JavaScript: TypeORM cung cấp một cách tiếp cận đơn giản và linh hoạt để tương tác với cơ sở dữ liệu quan hệ từ TypeScript hoặc JavaScript. Từ đó có thể sử dụng các khái niệm của ngôn ngữ lập trình mà không cần phải viết trực tiếp các truy vấn SQL.
- Hỗ trợ cho nhiều cơ sở dữ liệu: TypeORM hỗ trợ nhiều hệ quản trị cơ sở dữ liệu quan hệ như MySQL, PostgreSQL, SQLite, SQL Server, Oracle và nhiều hơn nữa. Điều này cho phép làm việc với các hệ thống cơ sở dữ liệu khác nhau mà không cần thay đổi mã nguồn của ứng dụng.
- Mô hình dữ liệu và quan hệ: TypeORM cho phép bạn định nghĩa các mô hình dữ liệu và quan hệ thông qua các class và metadata. Bạn có thể xác định các mối quan hệ giữa các bảng, thuộc tính và ràng buộc dữ liệu.
- Tích hợp tạo tự động cơ sở dữ liệu: TypeORM cung cấp các công cụ để tạo tự động cơ sở dữ liệu từ các mô hình dữ liệu được định nghĩa trong mã nguồn, giúp tự động tạo bảng, chỉnh sửa cấu trúc cơ sở dữ liệu và quản lý phiên bản.
- Query Builder và truy vấn linh hoạt: TypeORM cung cấp Query Builder cho phép bạn tạo truy vấn dễ dàng và linh hoạt. Bạn có thể xây dựng truy vấn phức tạp bằng cách sử dụng các phương thức và điều kiện chuỗi.
- Xử lý tương tranh và quản lý transaction: TypeORM hỗ trợ xử lý tương tranh và quản lý transaction một cách dễ dàng. Từ đó có thể thực hiện các thao tác cơ sở dữ liệu có tính tương tranh mà đảm bảo tính nhất quán và an toàn.

c) Sử dụng TypeORM trong dự án

Tài liệu tham khảo: <https://typeorm.io>

Bước 1: Cài đặt TypeORM và các dependencies cần thiết

Bước 2: Cấu hình TypeORM cho dự án bằng ormconfig.json tại thư mục gốc của dự án

```
{
  "type": "postgres",
  "host": "localhost",
  "port": 5432,
  "username": "your_username",
  "password": "your_password",
  "database": "your_database",
  "entities": ["src/entities/*.ts"],
  "migrations": ["src/migrations/*.ts"],
  "cli": {
    "migrationsDir": "src/migrations"
  }
}
```

Bước 3: Tạo entity (thực thể) ánh xạ dữ liệu từ cơ sở dữ liệu

```
import { Entity, Column, PrimaryGeneratedColumn } from
'typeorm';

@Entity()
export class User {
  @PrimaryGeneratedColumn()
  id: number;

  @Column()
  name: string;

  @Column()
  email: string;
}
```

Bước 4: Thao tác với CSDL thông qua entity

```
import { getRepository } from 'typeorm';
import { User } from './entities/User';
// Tạo user mới
const user = new User();
user.name = 'John Doe';
user.email = 'john@example.com';
const userRepository = getRepository(User);
await userRepository.save(user);
```

```
// Lấy danh sách users
const users = await userRepository.find();
console.log(users);
```

3.1.3 Cơ sở dữ liệu PostgreSQL

a) Giới thiệu

PostgreSQL là một hệ thống quản lý cơ sở dữ liệu quan hệ (RDBMS) mã nguồn mở và mạnh mẽ. Nó được biết đến với tính mạnh mẽ, khả năng mở rộng và tuân thủ các tiêu chuẩn SQL. Được phát triển tại Đại học California, Berkeley, PostgreSQL có lịch sử lâu đời và được sử dụng rộng rãi trong nhiều ứng dụng và ngành công nghiệp khác nhau.

b) Lý do sử dụng

- Cơ sở dữ liệu quan hệ: PostgreSQL tuân theo mô hình cơ sở dữ liệu quan hệ, cho phép lưu trữ và quản lý dữ liệu có cấu trúc.
- Bảo mật: PostgreSQL cung cấp các tính năng bảo mật mạnh mẽ, bao gồm các phương thức xác thực, mã hóa SSL, cơ chế kiểm soát truy cập và các đặc quyền chi tiết.
- Tuân thủ ACID: PostgreSQL đảm bảo tính toàn vẹn và độ tin cậy của dữ liệu thông qua hỗ trợ các thuộc tính ACID (Atomicity (nguyên tử), Consistency (nhất quán), Isolation (cô lập), và Durability (lâu bền)) đảm bảo rằng các xử lý cơ sở dữ liệu được xử lý một cách đáng tin cậy và nhất quán.
- Hiệu quả về chi phí: Là cơ sở dữ liệu nguồn mở, PostgreSQL được sử dụng miễn phí. Ngoài ra, khả năng mở rộng và tối ưu hóa hiệu suất của nó có thể giúp giảm chi phí cơ sở hạ tầng bằng cách sử dụng hiệu quả tài nguyên phần cứng. PostgreSQL vượt trội hơn MySQL khi hỗ trợ hầu hết các tính năng cơ sở dữ liệu nâng cao.

c) Sử dụng PostgreSQL

- **Bước 1:** Cài đặt: Tải và cài đặt PostgreSQL từ trang web chính thức (<https://www.postgresql.org/download/>). Làm theo hướng dẫn cài đặt phù hợp với hệ điều hành.
- **Bước 2:** Kết nối với cơ sở dữ liệu: Sử dụng thư mục "bin" của PostgreSQL để truy cập vào Command Prompt hoặc Terminal. Sử dụng lệnh "psql -U username

-d database_name" để kết nối với cơ sở dữ liệu. Thay "username" bằng tên người dùng có quyền truy cập và "database_name" bằng tên cơ sở dữ liệu muốn kết nối.

- **Bước 3a:** Thao tác cơ bản: Sử dụng lệnh SQL như "CREATE," "SELECT," "UPDATE," và "DELETE" để thao tác dữ liệu trong các bảng. Sử dụng lệnh "CREATE TABLE" để tạo bảng mới.
- **Bước 3b:** Câu lệnh SQL nâng cao: PostgreSQL cung cấp nhiều tính năng và câu lệnh mạnh mẽ khác như JOIN, INDEX, FUNCTION, TRIGGER, và nhiều hơn nữa. Tìm hiểu thêm về các câu lệnh SQL và tính năng của PostgreSQL trong tài liệu chính thức (<https://www.postgresql.org/docs>).

3.1.4 Bản đồ với MapBox API

a) Giới thiệu

API MapBox là tập hợp các công cụ và dịch vụ mạnh mẽ, cung cấp cho nhà phát triển khả năng tích hợp chức năng lập bản đồ động, mã hóa địa lý và dịch vụ định vị vào các ứng dụng và trang web. MapBox cung cấp quyền truy cập vào một loạt dữ liệu địa lý và dịch vụ, bao gồm bản đồ tùy chỉnh, mã hóa địa lý, mã hóa địa lý ngược và các tính năng dựa trên vị trí khác.

b) Lý do lựa chọn

- **Độ phủ và dữ liệu chi tiết:** MapBox cung cấp dữ liệu bản đồ đa dạng và chi tiết với khả năng tùy chỉnh cao, bao gồm thông tin địa lý, điểm quan trọng, tuyến đường, và hướng dẫn đi lại. Người dùng có thể tạo bản đồ theo phong cách riêng, đảm bảo thông tin địa lý chính xác và độc đáo cho ứng dụng.
- **Tích hợp dễ dàng và linh hoạt:** MapBox cung cấp các công cụ, tài liệu và ví dụ rõ ràng giúp dễ dàng tích hợp vào các ứng dụng. Cộng đồng phát triển của MapBox hỗ trợ mạnh mẽ, giúp giải quyết vấn đề và tối ưu hóa việc sử dụng bản đồ trong dự án.
- **Tính năng và công nghệ tiên tiến:** MapBox cung cấp nhiều tính năng và công nghệ tiên tiến như tìm kiếm địa điểm, định vị GPS, hướng dẫn đi lại, tích hợp dữ liệu thời tiết và giao thông, cùng với khả năng tùy chỉnh bản đồ mạnh mẽ. Điều này giúp tạo ra trải nghiệm người dùng tốt hơn và cung cấp các tính năng tiện ích, cũng như khả năng thích ứng cao cho ứng dụng.
- **Tốc độ và hiệu suất:** MapBox được tối ưu hóa cho tốc độ và hiệu suất cao, đáp ứng nhanh chóng yêu cầu của người dùng. Hệ thống cơ sở hạ tầng đáng tin cậy và mạnh mẽ giúp đảm bảo rằng bản đồ và dữ liệu liên quan được tải nhanh chóng và mượt mà trên các nền tảng và thiết bị khác nhau.

c) Tích hợp MapBox API vào dự án**Bước 1:** Lấy secret key API từ MapBox**Bước 2:** Sử dụng MapBox API dự án

- Xác định tọa độ dựa trên địa chỉ

```
async function geocodeAddress(address: string): Promise<string>
{
  const endpoint =
'https://api.mapbox.com/geocoding/v5/mapbox.places/';
  const params = {
    access_token: process.env.MAPBOX_SECRET_KEY,
    limit: 1,
  };

  try {
    const response = await axios.get(
      `${endpoint}${encodeURIComponent(address)}.json`,
      { params },
    );
    if (response.data && response.data.features &&
response.data.features[0]) {
      const [longitude, latitude] =
response.data.features[0].center;
      return `${longitude},${latitude}`;
    } else {
      throw new Error('Unable to geocode address.');
```

- Tính khoảng cách giữa hai địa chỉ

```
export async function getEstimatedDeliveryInfo(
  fromAddress: string,
  toAddress: string,
): Promise<EstimatedDeliveryInfo> {
```

```
// Mapbox requires coordinates (longitude, latitude). So,
first, we'll geocode the addresses to get these coordinates.
const [fromCoords, toCoords] = await Promise.all([
  geocodeAddress(fromAddress),
  geocodeAddress(toAddress),
]);

const endpoint =
`https://api.mapbox.com/directions/v5/mapbox/driving/${fromCoords};${toCoords}`;
const params = {
  geometries: 'geojson',
  access_token: process.env.MAPBOX_SECRET_KEY,
};

try {
  const response = await axios.get(endpoint, { params });

  if (response.data && response.data.routes[0]) {
    const route = response.data.routes[0];

    const distance = parseFloat(route.distance) / 1000; //
    Convert distance from meters to kilometers
    const durationInTraffic = parseFloat(route.duration) /
    60; // Convert duration from seconds to minutes

    const deliveryTime = addMinutes(
      new Date(),
      durationInTraffic + PREPARATION_TIME,
    );

    return {
      distance,
      durationInTraffic,
      deliveryTime,
      heavyTraffic: false,
    };
  } else {
    throw new Error('Unable to retrieve route details.');
```

```
  }
} catch (error) {
  console.error(`Error: ${error.message}`);
  throw error;
}
```

}

3.1.5 Thanh toán trực tuyến với Paypal

a) Giới thiệu

PayPal là một hệ thống thanh toán trực tuyến phổ biến trên toàn thế giới. PayPal cho phép người dùng gửi và nhận tiền, thanh toán trực tuyến và chấp nhận thanh toán từ các khách hàng trên Internet

b) Lý do lựa chọn

- **Tính tiện lợi:** PayPal giúp người dùng gửi và nhận tiền một cách dễ dàng và nhanh chóng chỉ với một địa chỉ email. Người dùng có thể kết nối tài khoản PayPal với thẻ tín dụng, thẻ ghi nợ hoặc tài khoản ngân hàng để thực hiện các giao dịch.
- **An toàn và bảo mật:** PayPal sử dụng các biện pháp bảo mật cao như mã hóa SSL để bảo vệ thông tin tài khoản và giao dịch của bạn. Người dùng không cần tiết lộ thông tin tài chính của mình trực tiếp cho người bán, giúp giảm nguy cơ lộ thông tin cá nhân.
- **Tích hợp linh hoạt:** PayPal cung cấp các công cụ và API mạnh mẽ cho phép tích hợp thanh toán vào các dự án phát triển đa nền tảng (website, điện thoại, ...) và ngôn ngữ lập trình khác nhau.
- **Giao dịch quốc tế:** PayPal hỗ trợ giao dịch quốc tế và có thể chuyển đổi tiền tệ tự động. Giúp người dùng và doanh nghiệp có thể gửi và nhận thanh toán từ khách hàng trên toàn thế giới một cách thuận tiện.
- **PayPal Sandbox:** PayPal cung cấp môi trường Sandbox cho phép thử nghiệm và kiểm tra tích hợp thanh toán trước khi triển khai vào môi trường sản phẩm thực tế. Điều này giúp đảm bảo tính ổn định và chính xác của tích hợp PayPal.

c) Tích hợp Paypal vào dự án

Bước 1: Khai báo và cài đặt gói Paypal trong NextJS

```
npm install @paypal/react-paypal-js --save-dev
```

Bước 2: Tạo tài khoản Paypal SandBox

- Truy cập trang PayPal Developer (<https://developer.paypal.com>) và đăng nhập vào tài khoản của bạn.
- Tạo một ứng dụng Sandbox và lấy Client ID từ trang cấu hình ứng dụng.

Bước 3: Tích hợp Paypal vào giao diện (Front end)

- Tạo component React PayPalButton, để hiển thị thanh toán PayPal

```
import { PayPalScriptProvider, PayPalButtons } from
 '@paypal/react-paypal-js';

const PayPalButton = () => {
  const paypalOptions = {
    'client-id': 'YOUR_SANDBOX_CLIENT_ID',
    currency: 'USD',
  };
  const createOrder = (data: any, actions: any) => {
    // Gửi yêu cầu tạo đơn hàng tới backend Sandbox và trả về
    order ID
    // ...
  };
  const onApprove = (data: any, actions: any) => {
    // Xử lý khi người dùng chấp nhận thanh toán
    // ...
  };

  return (
    <PayPalScriptProvider options={paypalOptions}>
      <PayPalButtons createOrder={createOrder}
onApprove={onApprove} />
    </PayPalScriptProvider>
  );
};
export default PayPalButton;
```

- Sử dụng component đã tạo

```
import PayPalButton from './PayPalButton';
const CheckoutPage = () => {
  return (
    <div>
      {/* Các phần tử khác trong trang */}
      <PayPalButton />
    </div>
  );
};
```

```
    );  
};  
export default CheckoutPage;
```

Bước 4: Tích hợp Paypal phía server (Backend)

```
const express = require('express');  
const paypal = require('@paypal/checkout-server-sdk');  
const app = express();  
  
// Cấu hình PayPal SDK  
const clientId = 'YOUR_SANDBOX_CLIENT_ID';  
const clientSecret = 'YOUR_SANDBOX_CLIENT_SECRET';  
const environment = new  
paypal.core.SandboxEnvironment(clientId, clientSecret);  
const client = new paypal.core.PayPalHttpClient(environment);  
  
// Endpoint để tạo đơn hàng  
app.post('/api/create-order', async (req, res) => {  
    const request = new paypal.orders.OrdersCreateRequest();  
    // Cấu hình thông tin đơn hàng ...  
    try {  
        const response = await client.execute(request);  
        // Trả về order ID cho frontend  
        res.json({ orderId: response.result.id });  
    } catch (error) {  
        // Xử lý lỗi  
        res.status(500).json({ error: 'Something went wrong' });  
    }  
});  
  
// Endpoint để xác nhận thanh toán  
app.post('/api/complete-order', async (req, res) => {  
    const orderId = req.body.orderId;  
    const request = new  
paypal.orders.OrdersCaptureRequest(orderId);  
    try {  
        const response = await client.execute(request);  
        // Xử lý khi thanh toán thành công  
        res.json({ status: 'success' });  
    } catch (error) {  
        // Xử lý lỗi  
        res.status(500).json({ error: 'Something went wrong' });  
    }  
});
```

});

3.1.6 Thông tin địa chỉ với Province-API

a) Giới thiệu

Province Open API là một API HTTP cho phép lấy danh sách tỉnh thành, quận huyện, phường xã Việt Nam.

b) Lý do sử dụng

- **Dữ liệu được cung cấp và duy trì:** provinces.open-api là một API công cộng cung cấp thông tin về các tỉnh thành. Bạn không cần phải tự lưu trữ và duy trì dữ liệu này, mà có thể truy cập vào dữ liệu được cập nhật và quản lý bởi nhà cung cấp API. Điều này giảm công sức và trách nhiệm của bạn trong việc duy trì và cập nhật dữ liệu.
- **Dữ liệu được chuẩn hóa và đáng tin cậy:** provinces.open-api đảm bảo rằng dữ liệu về các tỉnh thành được chuẩn hóa và đáng tin cậy. Đảm bảo rằng bạn có dữ liệu chính xác và nhất quán để sử dụng trong ứng dụng của mình, mà không cần phải tự kiểm tra và xử lý dữ liệu không đáng tin cậy.
- **Tiết kiệm thời gian và công sức:** Bằng cách sử dụng provinces.open-api, bạn không cần phải xây dựng và triển khai một hệ thống lưu trữ riêng để lưu trữ dữ liệu về các tỉnh thành. Điều này giúp tiết kiệm thời gian và công sức trong việc phát triển và quản lý hệ thống của bạn.
- **Tích hợp dễ dàng:** provinces.open-api cung cấp một giao diện API sẵn sàng để bạn có thể tích hợp vào ứng dụng của mình một cách dễ dàng. Bạn chỉ cần gửi các yêu cầu API đến endpoints cung cấp thông tin về các tỉnh thành và nhận lại dữ liệu theo định dạng được chỉ định.

Tuy nhiên, cũng có một số hạn chế khi sử dụng provinces.open-api. Trang web phải phụ thuộc vào nhà cung cấp API và đảm bảo rằng dịch vụ này luôn hoạt động và cung cấp dữ liệu. Nếu dịch vụ bị gián đoạn hoặc dữ liệu không khả dụng, ứng dụng sẽ bị ảnh hưởng

c) Sử dụng Province-API

Tài liệu tham khảo: <https://provinces.open-api.vn/redoc>

- *Lấy danh sách Tỉnh/thành phố*

GET <https://provinces.open-api.vn/api/?depth=1>

- Lấy danh sách Quận/Huyện thuộc một Tỉnh/Thành phố. Ví dụ Thành phố Hà Nội ($id = 1$)

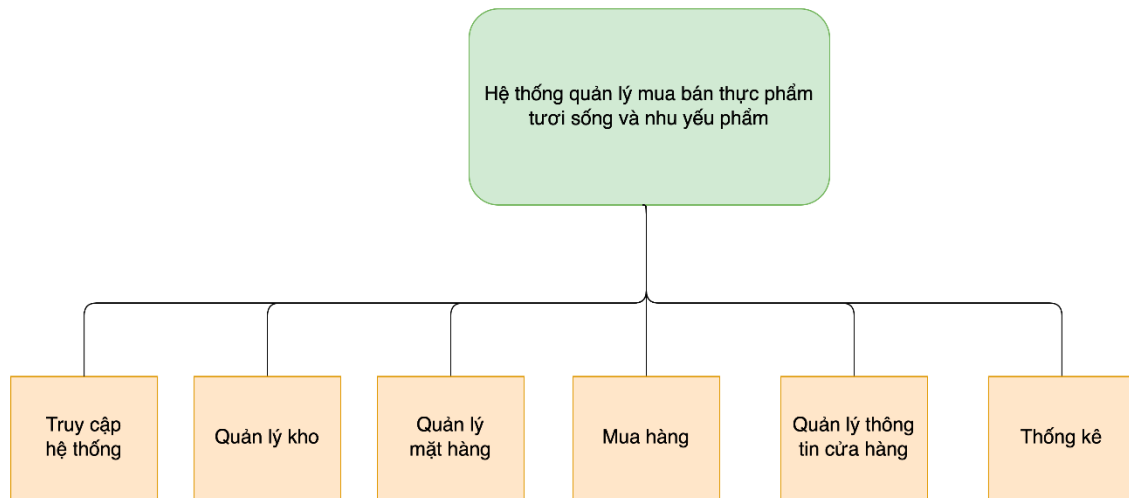
GET <https://provinces.open-api.vn/api/p/1?depth=2>

- Lấy danh sách Phường/Xã thuộc một Quận/Huyện. Ví dụ Quận Tây Hồ ($id = 3$)

GET <https://provinces.open-api.vn/api/d/3?depth=2>

CHƯƠNG 4: PHÂN TÍCH HỆ THỐNG

4.1 Sơ đồ phân rã cấp chức năng mức đỉnh

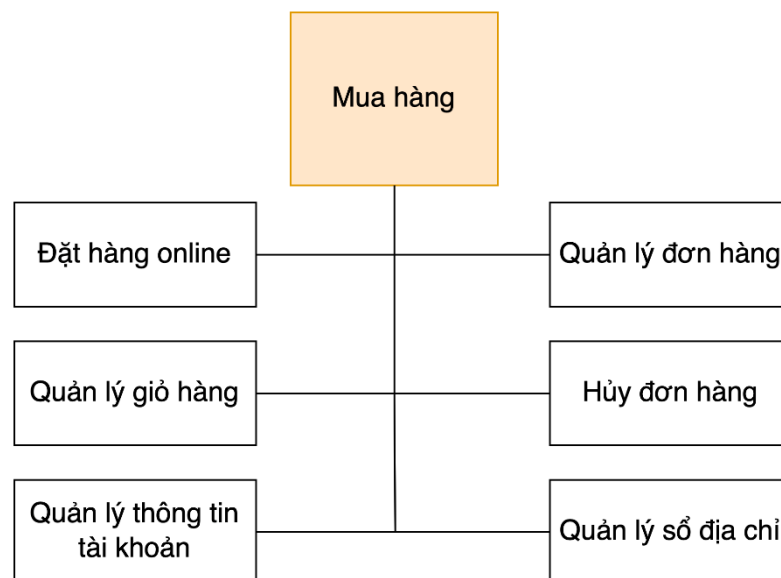


Hình 4.1 Sơ đồ cơ cấu tổ chức của dự án

Hệ thống nghiệp vụ được xét gồm các chức năng chính: Quản lý kho, quản lý mặt hàng, mua hàng, quản lý thông tin cửa hàng và thống kê. Mỗi chức năng này được phân rã thành một số chức năng nhỏ hơn mà thao tác thực hiện nó đủ đơn giản cho việc lập trình.

4.2 Sơ đồ phân rã chức năng

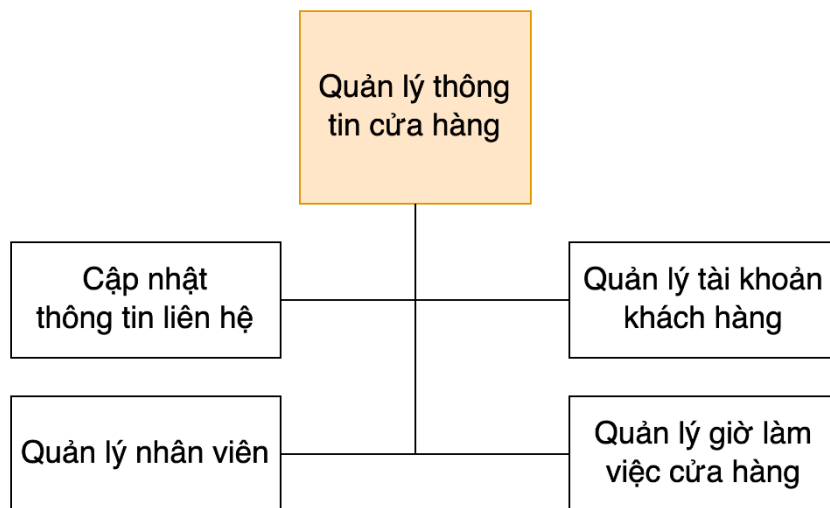
4.2.1 Chức năng Mua hàng



Hình 4.2 Sơ đồ phân rã chức năng Mua hàng

- Đặt hàng online: Khách hàng tiến hành đặt hàng với các món hàng đã chọn và cung cấp thông tin liên quan như địa chỉ giao hàng, thông tin thanh toán.
- Quản lý giỏ hàng: Khách hàng có thể thêm, xóa, cập nhật số lượng sản phẩm trong giỏ hàng
- Quản lý thông tin tài khoản: Cho phép Khách hàng thay đổi các thông tin như Họ, tên số điện thoại. Thông tin tài khoản sẽ được dùng để phục vụ trong quá trình đặt hàng
- Quản lý đơn hàng: Khách hàng có thể xem thông tin tất cả các đơn hàng đã đặt, cũng như tìm kiếm
- Hủy đơn hàng: Khách hàng hoặc nhân viên thu ngân có thể hủy đơn hàng nếu đơn hàng vẫn còn đang trong quá trình chờ xác nhận, hoặc đang được xử lý
- Quản lý sở địa chỉ: Hệ thống cung cấp cơ chế thêm, xóa, sửa địa chỉ giao hàng, giúp khách hàng dễ dàng lưu, chọn địa chỉ giao hàng.

4.2.2 Chức năng quản lý thông tin cửa hàng

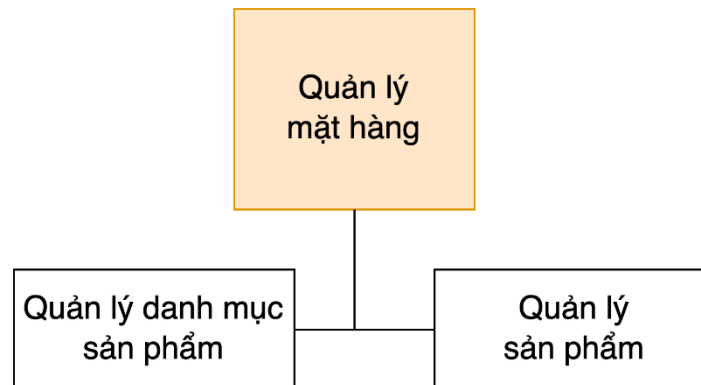


Hình 4.3 Sơ đồ phân rã chức năng Quản lý thông tin cửa hàng

- Cập nhật thông tin liên hệ: Cho phép Quản lý tổng cập nhật thông tin liên hệ của cửa hàng: Địa chỉ, email, số điện thoại. Thông tin liên hệ sẽ được hiển thị tại footer của trang web.
- Quản lý giờ làm việc cửa hàng: Cho phép cài đặt khung giờ hoạt động một cách linh động của cửa hàng theo các ngày trong tuần. Khi ngoài giờ làm việc, khách hàng sẽ không được mua hàng.

- Quản lý nhân viên: Cho phép Quản lý tổng dễ dàng thêm, cập nhật thông tin nhân viên trong cửa hàng bao gồm: Nhân viên quản lý kho, Nhân viên thu ngân, nhân viên kho, nhân viên giao hàng.
- Quản lý tài khoản khách hàng: Cho phép Quản lý tổng xem danh sách khách hàng, xem chi tiết thông tin của từng khách hàng cũng như tìm kiếm khách hàng theo tên.

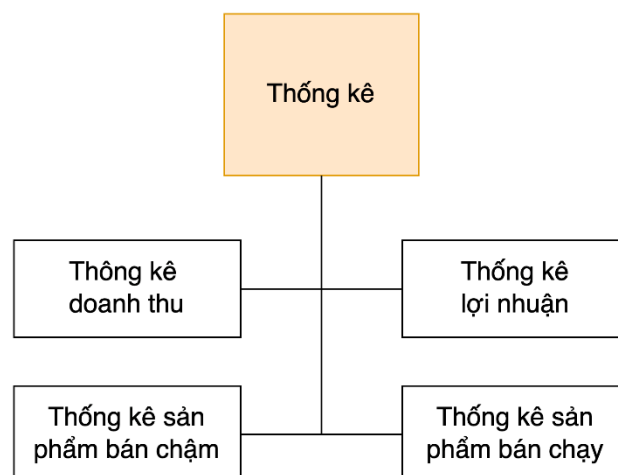
4.2.3 Chức năng quản lý Mặt hàng



Hình 4.4 Sơ đồ phân rã chức năng Quản lý mặt hàng

- Quản lý danh mục sản phẩm: Cho phép Quản lý tổng thêm danh mục sản phẩm để phục vụ quản lý thông tin sản phẩm. Mỗi sản phẩm sẽ thuộc một danh mục sản phẩm
- Quản lý sản phẩm: Cho phép Quản lý tổng quản lý thông tin sản phẩm

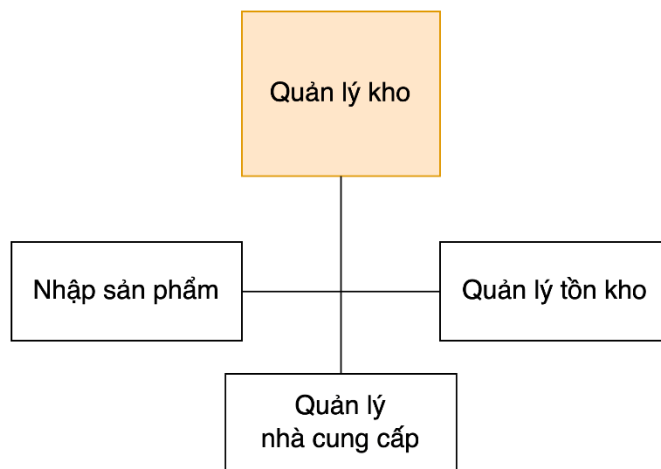
4.2.4 Chức năng thống kê



Hình 4.5 Sơ đồ phân rã chức năng Thống kê

- Thống kê doanh thu: Cho phép Quản lý tổng xem và theo dõi doanh thu của cửa hàng theo các tiêu chí: Doanh thu theo ngày, doanh thu theo tháng.
- Thống kê lợi nhuận: Cho phép Quản lý tổng xem và theo dõi lợi nhuận của cửa hàng theo các tiêu chí: Doanh thu theo ngày, doanh thu theo tháng.
- Thống kê sản phẩm bán chạy: Hệ thống tự động tổng hợp dữ liệu mua hàng và đưa ra danh sách các sản phẩm bán chạy trên web.
- Thống kê sản phẩm bán chậm: Hệ thống tự động tổng hợp dữ liệu mua hàng và đưa ra danh sách các sản phẩm bán chậm trên web.

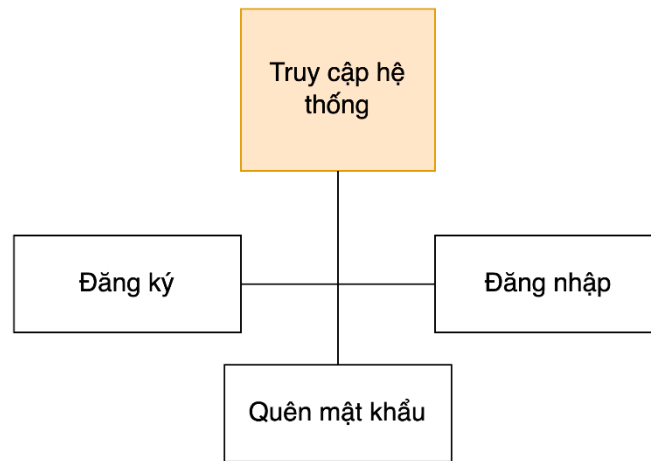
4.2.4 Chức năng Quản lý kho



Hình 4.6 Sơ đồ phân rã chức năng Thống kê

- Nhập sản phẩm: Cho phép nhân viên Quản lý kho nhập thêm một số lượng sản phẩm từ nhà cung cấp. Đồng thời ghi lại lịch sử nhập sản phẩm.
- Quản lý tồn kho: Nhân viên Quản lý kho sẽ kiểm tra định kỳ các sản phẩm đang còn tồn trong kho, Hệ thống sẽ liệt kê danh sách các sản phẩm sắp hoặc đang hết hàng
- Quản lý nhà cung cấp: Cho phép Quản lý tổng được xem - thêm - cập nhật thông tin các nhà cung cấp sản phẩm của cửa hàng.

4.2.5 Chức năng Truy cập hệ thống



Hình 4.7 Sơ đồ phân rã chức năng Truy cập hệ thống

- Cho phép khách hàng đăng ký tài khoản để mua hàng tại website của cửa hàng. Sau khi đăng ký khách hàng phải xác thực tài khoản thông qua email mà hệ thống đã gửi thông qua email của khách hàng.
- Đăng nhập: Khách hàng, nhân viên cửa hàng có thể đăng nhập vào hệ thống khi đã có tài khoản
- Quên mật khẩu: Hỗ trợ khách hàng và nhân viên cửa hàng khôi phục mật khẩu trong trường hợp quên mật khẩu

4.3 Định nghĩa yêu cầu và ràng buộc đối với phần mềm

4.3.1 Các yêu cầu từ môi trường nghiệp vụ

Req-ID	Nội dung yêu cầu	Stack-Holder
B01	Cửa hàng cần có cam kết về chính sách giao hàng <ul style="list-style-type: none"> • Bán kính giao hàng tối đa: 15km • Thời gian giao hàng tối đa: 60 phút • Chỉ giao trong giờ làm việc, mỗi ngày có thể có giờ làm việc khác nhau, thông tin này phải được nêu rõ ở website 	Quản lý tổng / Khách hàng
B02	Khi lấy hàng nhân viên kho theo đơn đặt hàng, phải lấy hàng có ngày hết hạn gần nhất	Quản lý kho
B03	Cam kết về chính sách huỷ đơn của cửa hàng <ul style="list-style-type: none"> - Đơn hàng chỉ có thể huỷ khi đang ở trạng thái chờ xác nhận từ nhân viên cửa hàng hoặc đang xử lý được xử lý. 	Quản lý tổng

	- Đơn hàng không được phép hủy nếu đơn hàng đang trong quá trình vận chuyển hoặc đã xác nhận nhận hàng đúng yêu cầu	
B04	Khách hàng khi hủy đơn hàng phải cung cấp lý do hủy đơn	Thu ngân
B05	Cửa hàng cam kết hoàn tiền khi khách hàng hủy đơn thành công.	Quản lý kho

Bảng 4.1 Các yêu cầu từ môi trường nghiệp vụ

4.3.2 Các yêu cầu từ môi trường vận hành

a) Các yêu cầu chức năng

ID	Nội dung yêu cầu	Stack-Holder
FR.1	Khi đặt hàng thành công, hệ thống phải trừ số lượng của các sản phẩm tồn kho có ngày hết hạn gần nhất, để tránh lãng phí nguyên liệu	Quản lý tổng, Quản lý kho
FR.2	Hệ thống phải tự động cập nhật số lượng sản phẩm khi khách hàng hủy đơn thành công.	Quản lý tổng
FR.3	Hệ thống phải tự động cập nhật số lượng sản phẩm khi khách hàng hủy đơn thành công.	Quản lý tổng / Quản lý kho
FR.4	Hệ thống phải tự động hoàn tiền đầy đủ cho khách hàng khi khách hàng hủy đơn, khi đơn hàng của khách thanh toán trực tuyến	Quản lý tổng
Fr.7	Khi hiển thị thông tin sản phẩm phải thể hiện được sản phẩm còn hàng hay hết hàng	
FR.8	Trước khi thêm sản phẩm vào giỏ hàng. Hệ thống cần kiểm tra số lượng tồn kho của sản phẩm.	Quản lý kho
FR.9	Trong trường hợp số lượng tồn kho không đáp ứng đủ số lượng cần mua. Hệ thống phải thông báo cho khách hàng	Quản lý tổng
FR.10	Khi thêm sản phẩm vào giỏ hàng nếu sản phẩm đã có trong giỏ hàng. Hệ thống phải cập nhật số lượng của sản phẩm đã có trong giỏ hàng	Quản lý tổng

Bảng 4.2 Các yêu cầu từ môi trường vận hành - Yêu cầu chức năng

b) Các yêu cầu phi chức năng

ID	Nội dung yêu cầu	Stack-Holder
NFF.1	Thời gian ghi dữ liệu ≤ 2 giây	Quản lý tổng
NFF.2	Thời gian đọc dữ liệu ≤ 1.5 giây, áp dụng phương pháp phân trang với các tập dữ liệu lớn (Thời gian được tính sau khi deploy)	Quản lý tổng

NFF.3	Thời gian kết xuất dữ liệu và trả về ≤ 3 giây, do cần tính toán phức tạp trên các tập dữ liệu lớn. (Thời gian được tính sau khi deploy)	Quản lý tổng / Quản lý kho
NFF.4	Thời gian ghi/xử lý dữ liệu và trả về ≤ 3 giây, do cần xử lý phức tạp, liên quan đến nhiều bảng. (Thời gian được tính sau khi deploy)	Quản lý kho

Bảng 4.3 Các yêu cầu từ môi trường vận hành - Yêu cầu phi chức năng

4.3.3 Các yêu cầu từ môi trường phát triển

ID	Đối tượng	Nội dung yêu cầu	Stack-Holder
TA01	Application	Website phải chạy tương thích trên các trình duyệt phổ biến: <ul style="list-style-type: none"> - Chrome: version 88+ - Edge: version 88+ - Safari: version 16+ - Firefox: version 89+ 	Khách hàng / Dev team
TD02	Software	Viết code trên Visual Studio Code.	Dev team
TD03	Software	Dùng ReactJS 18, NextJS 13	Dev team

Bảng 4.3 Các yêu cầu từ môi trường phát triển

4.4 Mô hình luồng dữ liệu



Bảng 4.4 Sơ đồ phân rã chức năng Truy cập hệ thống

4.5 Mô hình Use case

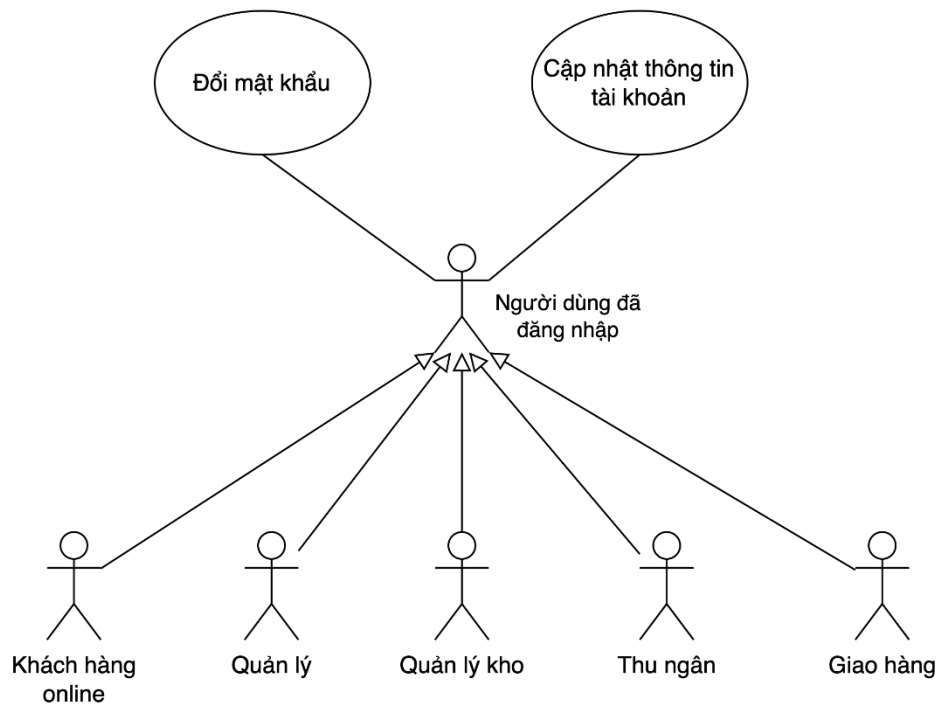
Nhắc lại một số cụm từ viết tắt:

- HT: Hệ thống
- KH: Khách hàng
- TN: Thu ngân
- QLCH: Quản lý cửa hàng (Quản lý tổng)

- QLK: Quản lý kho

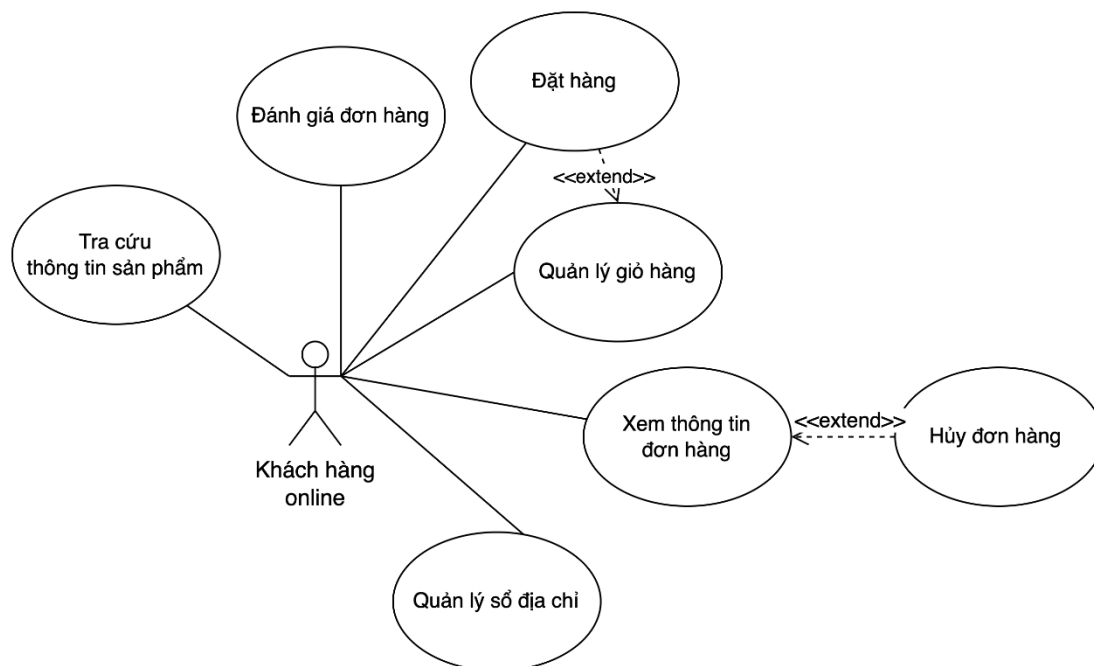
4.5.1 Mô hình Use case tổng quát

a) Usecase chung cho tất cả các Actor



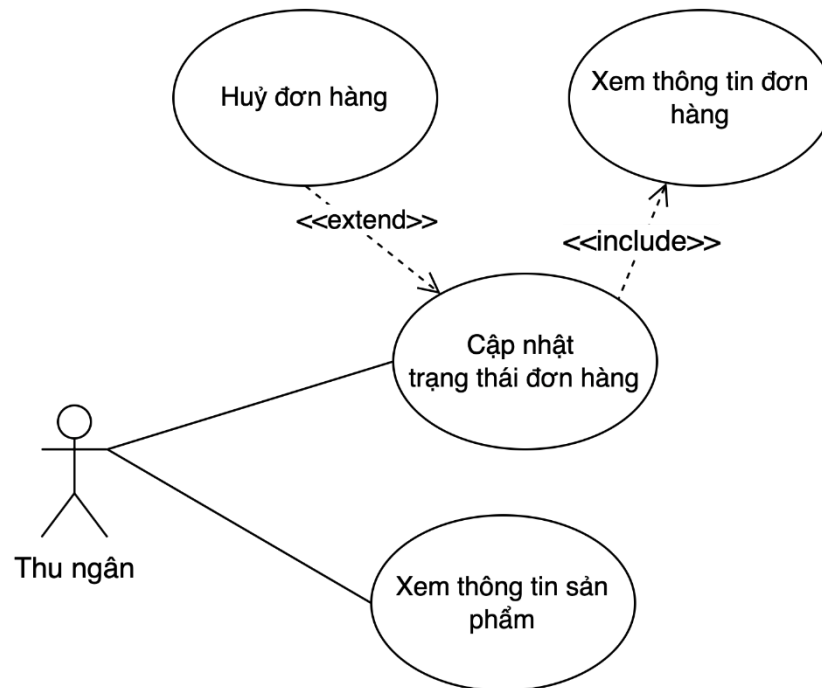
Bảng 4.5 Use case chung cho tất cả actor

b) Usecase tổng quát cho khách hàng online



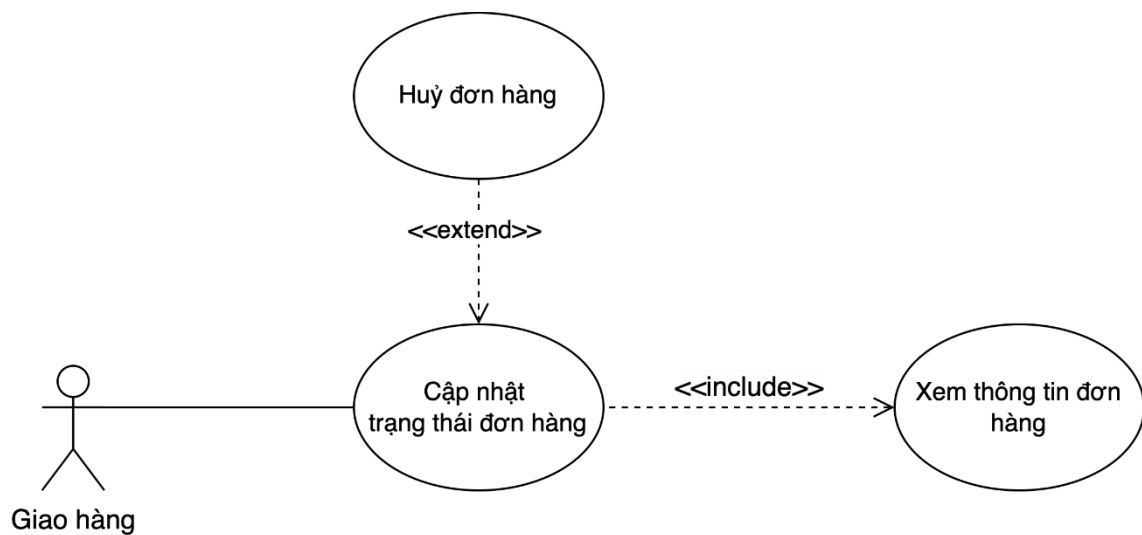
Bảng 4.6 Use case tổng quát cho khách hàng online

c) Usecase tổng quát cho nhân viên thu ngân



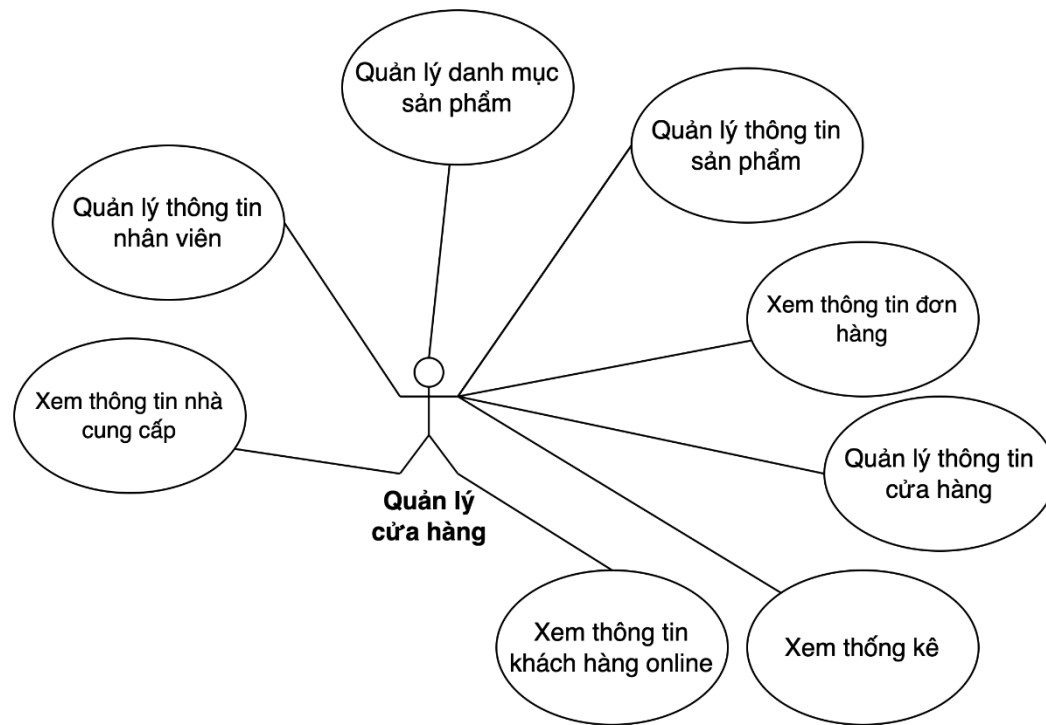
Bảng 4.7 Use case tổng quát cho nhân viên thu ngân

d) Usecase tổng quát cho nhân viên giao hàng



Bảng 4.8 Use case tổng quát cho nhân viên giao hàng

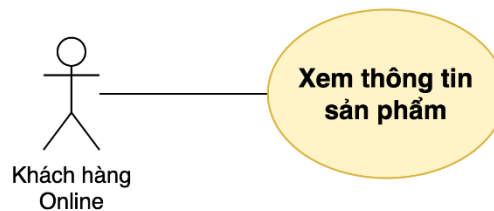
d) Usecase tổng quát cho Quản lý tổng



Bảng 4.9 Use case tổng quát cho Quản lý cửa hàng

4.5.2 Đặc tả các Usecase

a) Use case xem thông tin sản phẩm

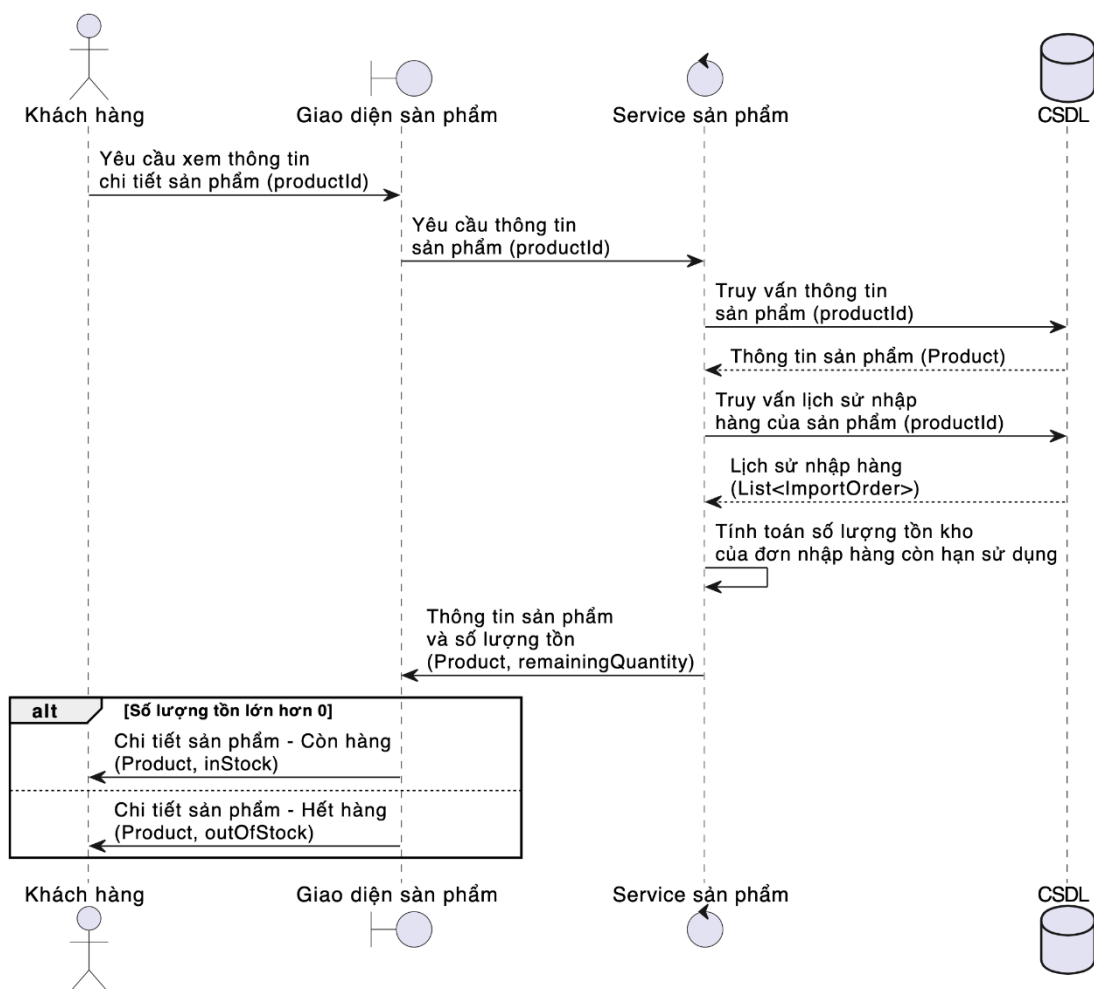


Hình 4.11 Use case Xem thông tin sản phẩm

Tên	Xem thông tin sản phẩm
Tác nhân chính	KH
Mô tả	KH có nhu cầu xem thông tin chi tiết sản phẩm
Điều kiện kích hoạt	KH chọn xem thông tin chi tiết của một sản phẩm
Điều kiện trước	
Điều kiện sau	- KH xem được đầy đủ thông tin chi tiết của sản phẩm

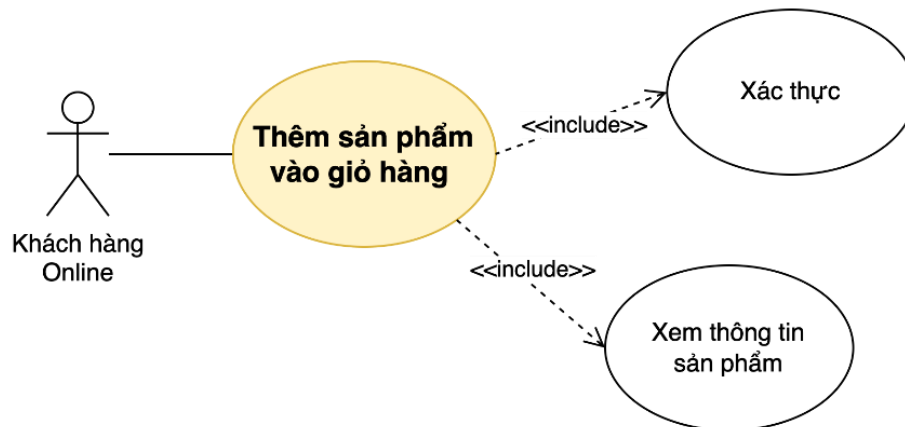
	- Nếu sản phẩm hết hàng, phải đánh nhãn thể hiện cho KH biết
Luồng chính	<ol style="list-style-type: none"> 1. KH chọn một sản phẩm để xem thông tin chi tiết 2. HT lấy thông tin sản phẩm phẩm và số lượng tồn kho 3. HT hiển thị thông tin chi tiết sản phẩm cho KH 3.1 Nếu sản phẩm hết hàng, đánh nhãn hết hàng cho sản phẩm
Luồng thay thế	
Luồng ngoại lệ	

Bảng 4.10 Đặc tả Use case xem thông tin sản phẩm



Hình 4.8 Sơ đồ tuần tự Use case Xem thông tin sản phẩm

b) Use case thêm sản phẩm vào giỏ hàng

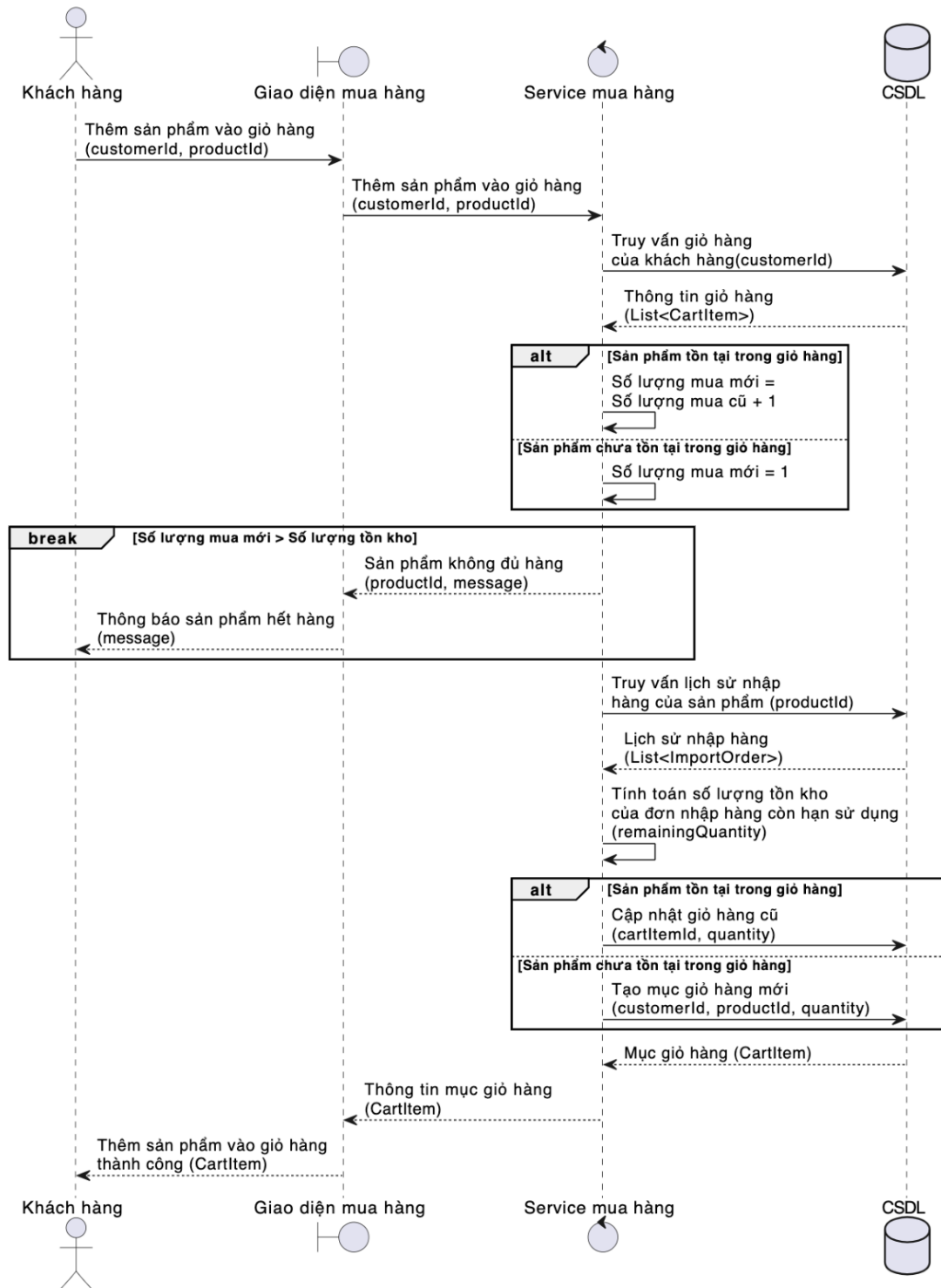


Hình 4.9 Use case Thêm sản phẩm vào giỏ hàng

Tên	Thêm sản phẩm vào giỏ hàng
Tác nhân chính	KH
Mô tả	KH có yêu cầu thêm sản phẩm vào giỏ hàng
Điều kiện kích hoạt	KH thêm vào giỏ hàng sản phẩm mong muốn
Điều kiện trước	<ul style="list-style-type: none"> - Tài khoản KH đã xác thực thành công - Thông tin sản phẩm tồn tại trong hệ thống
Điều kiện sau	<ul style="list-style-type: none"> - Tạo và thêm mới mục giỏ hàng mới cho sản phẩm đối với sản phẩm chưa có trong giỏ hàng của KH - Cập nhật số lượng của mục giỏ hàng tương ứng khi thêm vào sản phẩm đã tồn tại trong giỏ hàng của KH
Luồng chính	<ol style="list-style-type: none"> 1. KH chọn thêm một sản phẩm vào giỏ hàng 2. HT kiểm tra sản phẩm cần thêm vào đã tồn tại trong giỏ hàng của KH hay chưa 3. HT tính số lượng mua mới của sản phẩm <ol style="list-style-type: none"> 3.1 Nếu sản phẩm chưa tồn tại trong giỏ hàng, số lượng mua mới của sản phẩm là 1 3.2 Nếu sản phẩm đã tồn tại trong giỏ hàng, số lượng mua mới sẽ bằng số lượng mua cũ cộng thêm 1 4. HT kiểm tra số lượng tồn kho của sản phẩm đáp ứng được số lượng mua của sản phẩm

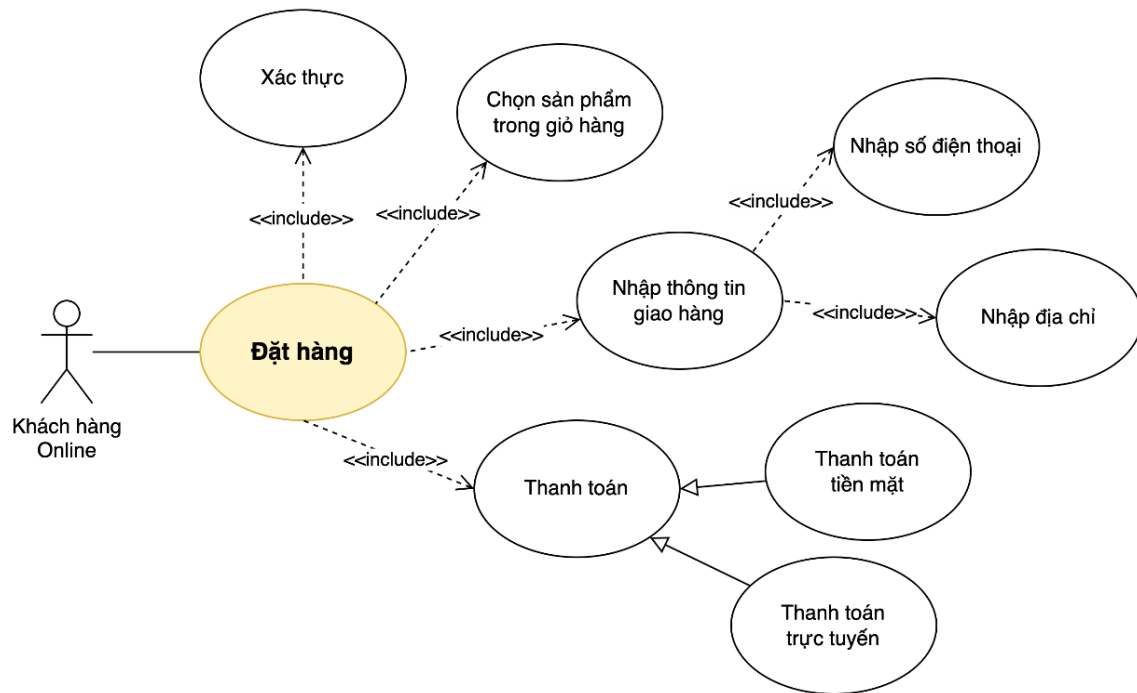
	<p>5. HT thêm sản phẩm vào giỏ hàng</p> <p>5.1 Nếu sản phẩm chưa tồn tại trong giỏ hàng, HT thêm sản phẩm mới vào giỏ hàng, với số lượng mua là 1</p> <p>5.2 Nếu sản phẩm tồn tại trong giỏ hàng, HT cập nhật lại số lượng mua của sản phẩm đó trong giỏ hàng</p> <p>6. HT thông báo cho KH thêm sản phẩm vào giỏ hàng thành công.</p>
Luồng thay thế	4.1 Khi Số lượng tồn kho của sản phẩm nhỏ hơn số lượng cần mua của KH. HT thông báo cho KH số lượng sản phẩm không đủ đáp ứng và kết thúc xử lý.
Luồng ngoại lệ	

Bảng 4.11 Đặc tả Use case Thêm sản phẩm vào giỏ hàng



Hình 4.10 Sơ đồ tuần tự Use case thêm sản phẩm vào giỏ hàng

c) Use case Đặt hàng



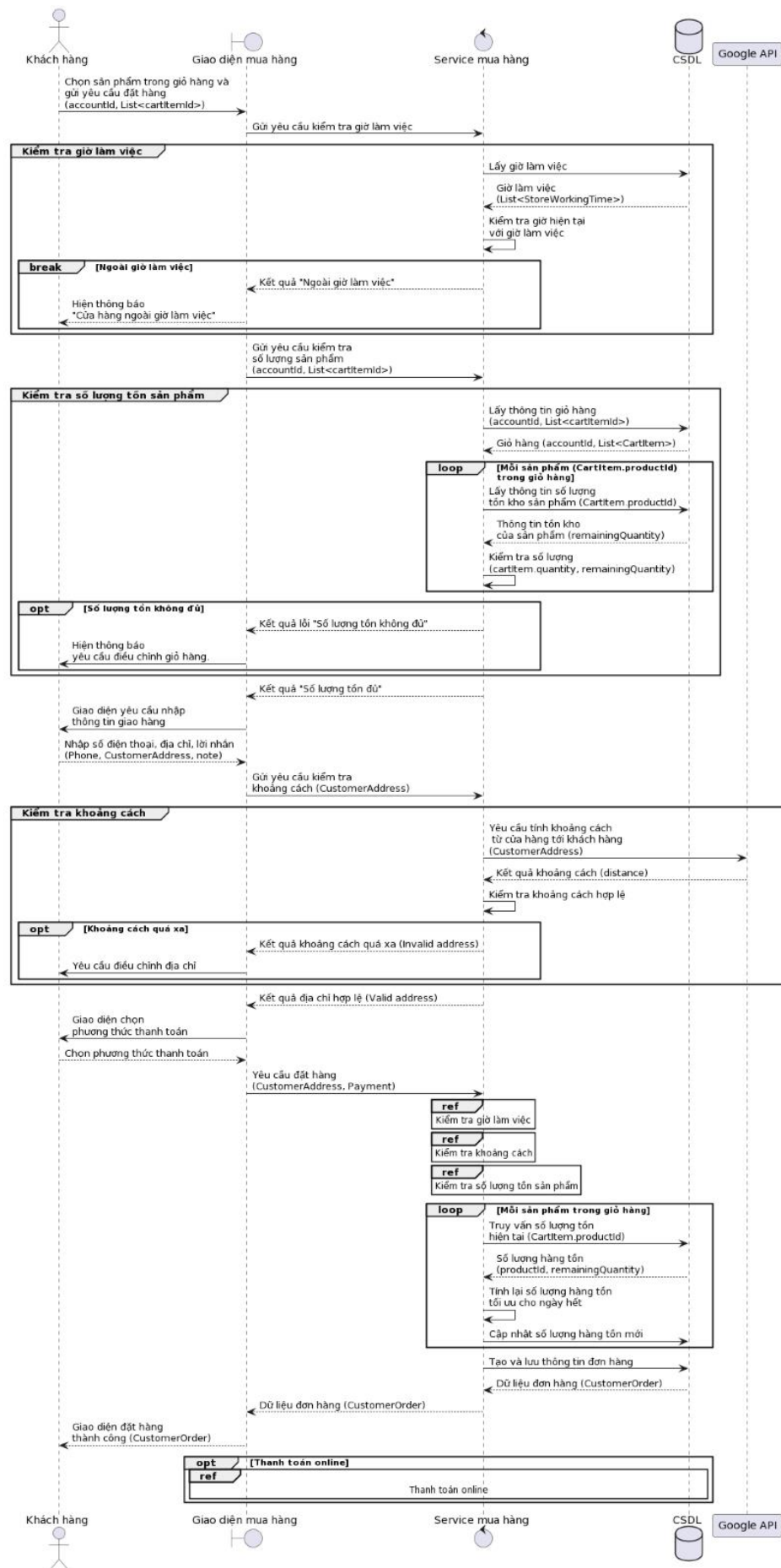
Hình 4.11 Use case Đặt hàng trực tuyến

Tên	Đặt hàng trực tuyến
Tác nhân chính	KH
Mô tả	KH muốn mua hàng trên website
Điều kiện kích hoạt	KH yêu cầu thanh toán giỏ hàng
Điều kiện trước	<ul style="list-style-type: none"> - KH đã xác thực thành công - Giỏ hàng của KH phải có sản phẩm - Giờ đặt hàng phải nằm trong giờ hoạt động của cửa hàng
Điều kiện sau	<ul style="list-style-type: none"> - KH mua hàng thành công - KH nhận được thông tin đơn hàng khi đặt hàng thành công - Số lượng sản phẩm trong kho phải được cập nhật lại theo đơn hàng vừa đặt - Nếu KH thanh toán trực tuyến, HT phải tự động xử lý thanh toán

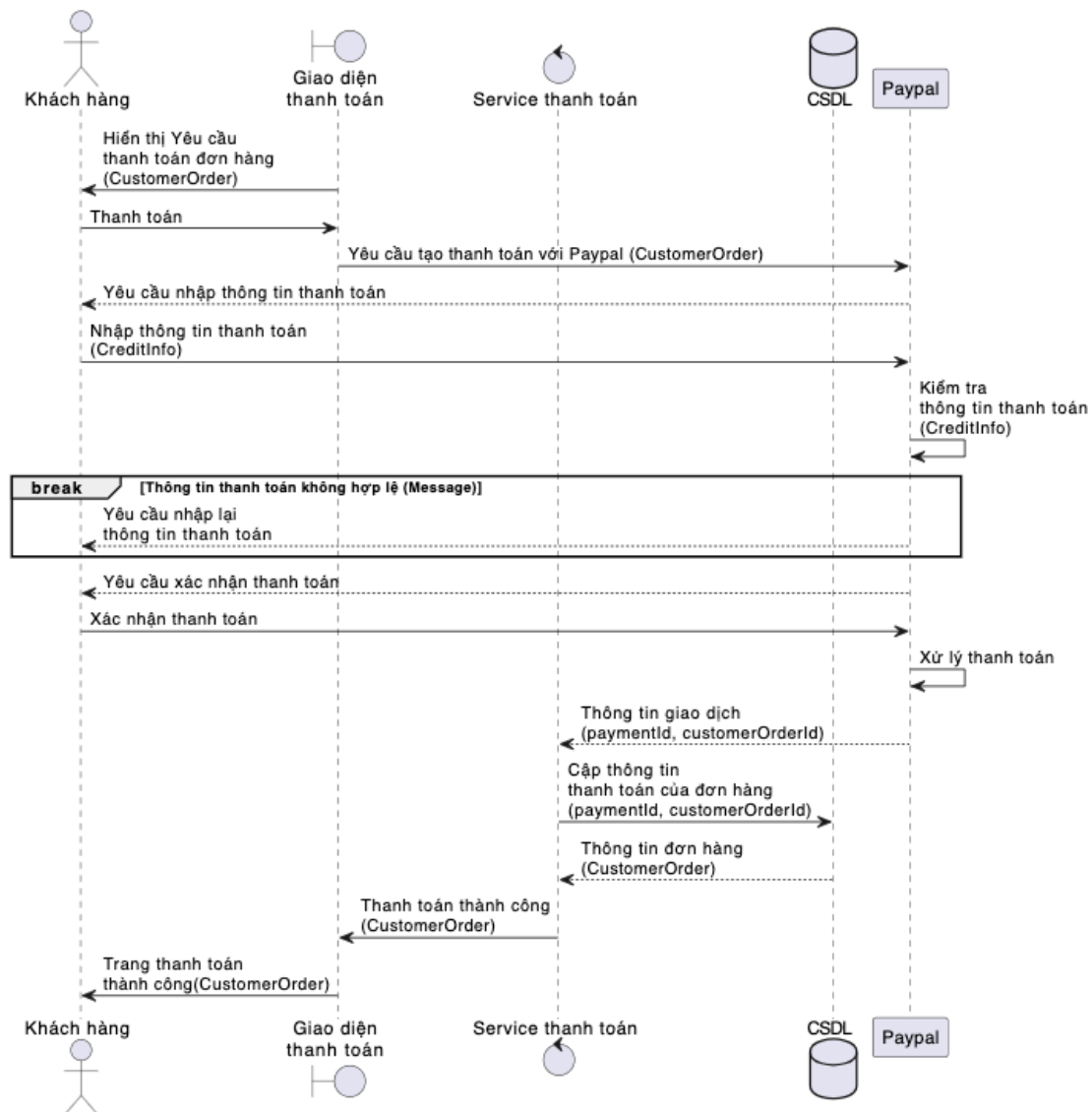
Luồng chính	<ol style="list-style-type: none"> 1. KH tiến hành đặt hàng với các sản phẩm kèm số lượng trong giỏ hàng 2. HT kiểm tra giờ làm việc của cửa hàng 3. HT kiểm tra số lượng tồn kho của các sản phẩm trong giỏ hàng với số lượng cần mua của KH 4. Hệ thống yêu cầu KH cung cấp thông tin giao hàng bao gồm số điện thoại, lời nhắn, địa chỉ 5. KH cung cấp thông tin giao hàng cho hệ thống 6. HT tính khoảng cách từ địa chỉ giao hàng đến cửa hàng 7. HT kiểm tra điều kiện khoảng cách 8. HT yêu cầu KH cung cấp thông tin về phương thức thanh toán: tiền mặt hoặc thanh toán trực tuyến. 9. KH cung cấp thông tin phương thanh toán. 10. HT kiểm tra lại toàn bộ thông tin: giờ làm việc, khoảng cách giao hàng, số lượng tồn của sản phẩm trong giỏ hàng 11. HT tính và cập nhật số lượng tồn trong kho của mỗi sản phẩm trong hoá đơn 12. HT tạo đơn hàng 13. HT thông báo thông tin đơn đặt hàng thành công cho KH 14. HT yêu cầu thông tin thanh toán trực tuyến 15. KH cung cấp thông tin thanh toán. 16. HT xác thực thông tin thanh toán của KH 17. HT yêu cầu KH xác nhận thanh toán 18. KH xác nhận thanh toán 19. HT xử lý thanh toán cho đơn hàng 20. HT cập nhật thông tin thanh toán của đơn hàng 21. HT thông báo KH thanh toán đơn hàng thành công
--------------------	--

Luồng thay thế	14.1 Nếu KH thanh toán tiền mặt: Use case dừng lại tại bước 13
Luồng ngoại lệ	<p>2.1 Cửa hàng ngoài giờ làm việc</p> <p>2.1.1 HT thông báo cho KH cửa hàng ngoài giờ làm việc vui lòng đặt lại sau.</p> <p>3.1 Sản phẩm hết hàng hoặc không đủ</p> <p>3.1.1 HT thông báo sản phẩm hết hàng hoặc không đủ, vui lòng điều chỉnh số lượng sản phẩm trong giỏ hàng</p> <p>3.1.2 KH Điều chỉnh số lượng sản phẩm trong giỏ hàng.</p> <p>3.1.3 Tiếp tục tại bước 3</p> <p>7.1 Khoảng cách giao hàng quá xa</p> <p>7.1.1 HT thông báo khoảng cách quá xa, vui lòng chỉnh sửa địa chỉ</p> <p>7.1.2 KH nhập lại địa chỉ giao hàng phù hợp</p> <p>7.1.3 Tiếp tục tại bước 7</p>

Bảng 4.12 Đặc tả Use case Đặt hàng trực tuyến

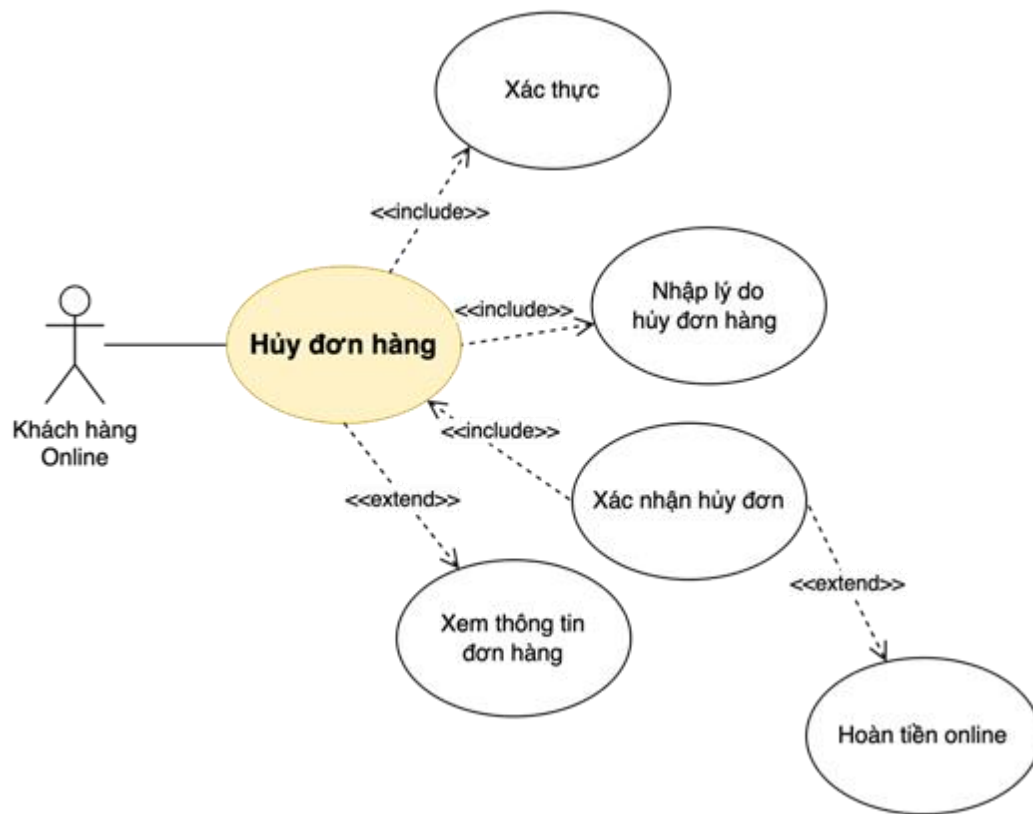


Hình 4.12 Sơ đồ tuần tự Use case đặt hàng



Hình 4.13 Sơ đồ tuần tự Thanh toán PayPal cho Use case đặt hàng

c) Use case Hủy đơn hàng

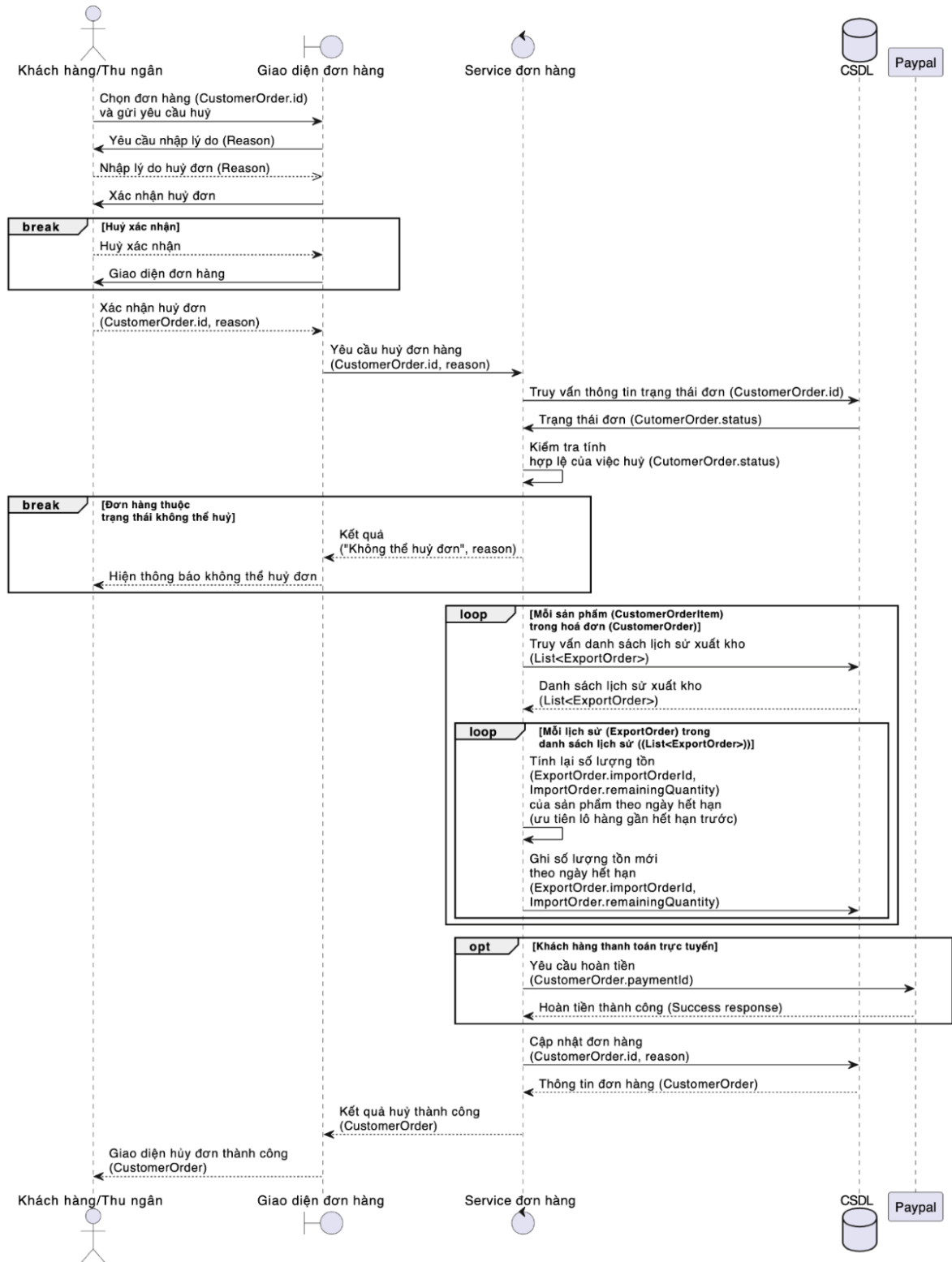


Hình 4.14 Use case Hủy đơn hàng

Tên	Hủy đơn hàng
Tác nhân chính	KH
Mô tả	KH muốn huỷ đơn đặt hàng cho đơn đặt trước đó
Điều kiện kích hoạt	KH yêu cầu hệ thống hủy đơn hàng đã đặt
Điều kiện trước	<ul style="list-style-type: none"> - Tài khoản KH đã xác thực thành công - Đơn hàng chỉ có thể hủy được khi đang chờ nhân viên xác nhận, hoặc đang chờ nhân viên xử lý
Điều kiện sau	<ul style="list-style-type: none"> - KH/TN huỷ đơn hàng thành công. - Thông tin của đơn hàng được cập nhật trạng thái hủy - Số lượng hàng tồn được hoàn lại - Nếu KH thanh toán trực tuyến, tiền sẽ được hoàn lại

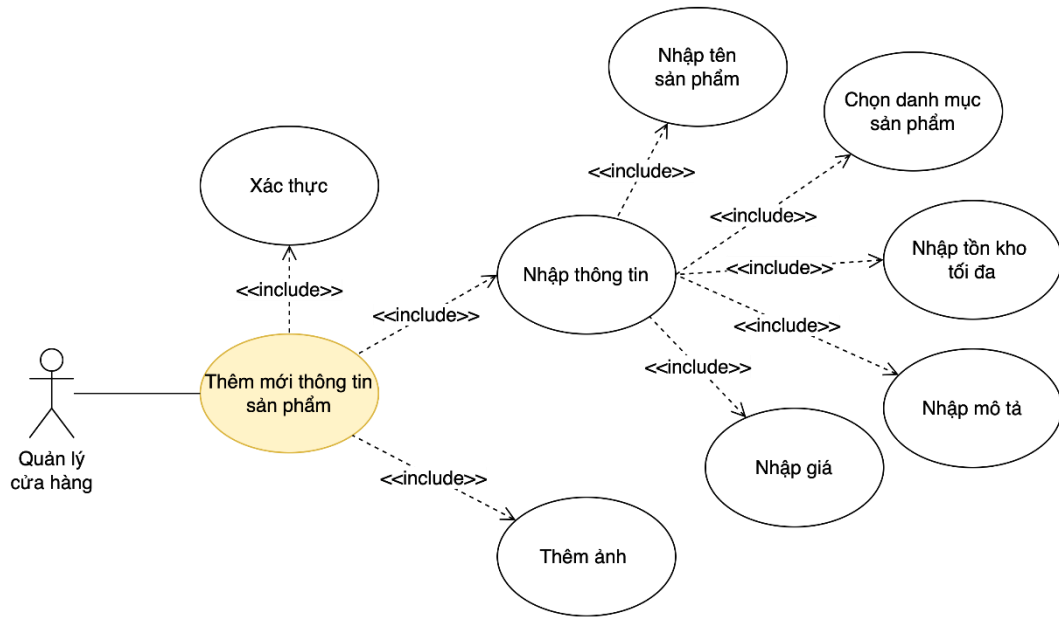
Luồng chính	<ol style="list-style-type: none"> 1. KH yêu cầu hủy đơn hàng đã đặt 2. HT yêu cầu lý do hủy đơn hàng 3. KH cung cấp lý do hủy đơn hàng 4. HT yêu cầu KH xác nhận việc hủy đơn 5. KH xác nhận hủy đơn hàng 6. HT xác minh đơn hàng đủ điều kiện để hủy theo chính sách của cửa hàng 7. HT cập nhật lại số lượng hàng tồn trong kho 8. Nếu KH thanh toán trực tuyến, tiền sẽ được HT hoàn lại 9. HT cập nhật thông tin đơn hàng đã hủy 10. HT thông báo KH đơn hàng được hủy thành công
Luồng thay thế	<ol style="list-style-type: none"> 5.1 KH không xác nhận hủy đơn <ol style="list-style-type: none"> 5.1.1 Kết thúc use case 8.1 Nếu KH thanh toán tiền mặt <ol style="list-style-type: none"> 8.1.1 Bỏ qua bước 8, tiếp tục tại bước 9
Luồng ngoại lệ	<ol style="list-style-type: none"> 4.1 Đơn hàng không hợp lệ để hủy <ol style="list-style-type: none"> 4.1 HT thông báo KH đơn hàng không thể hủy bỏ (lý do) 4.2 Kết thúc use case

Hình 4.15 Đặc tả Use case Hủy đơn hàng



Hình 4.16 Sơ đồ tuần tự Use case Hủy đơn hàng

d) Use case Thêm mới thông tin sản phẩm

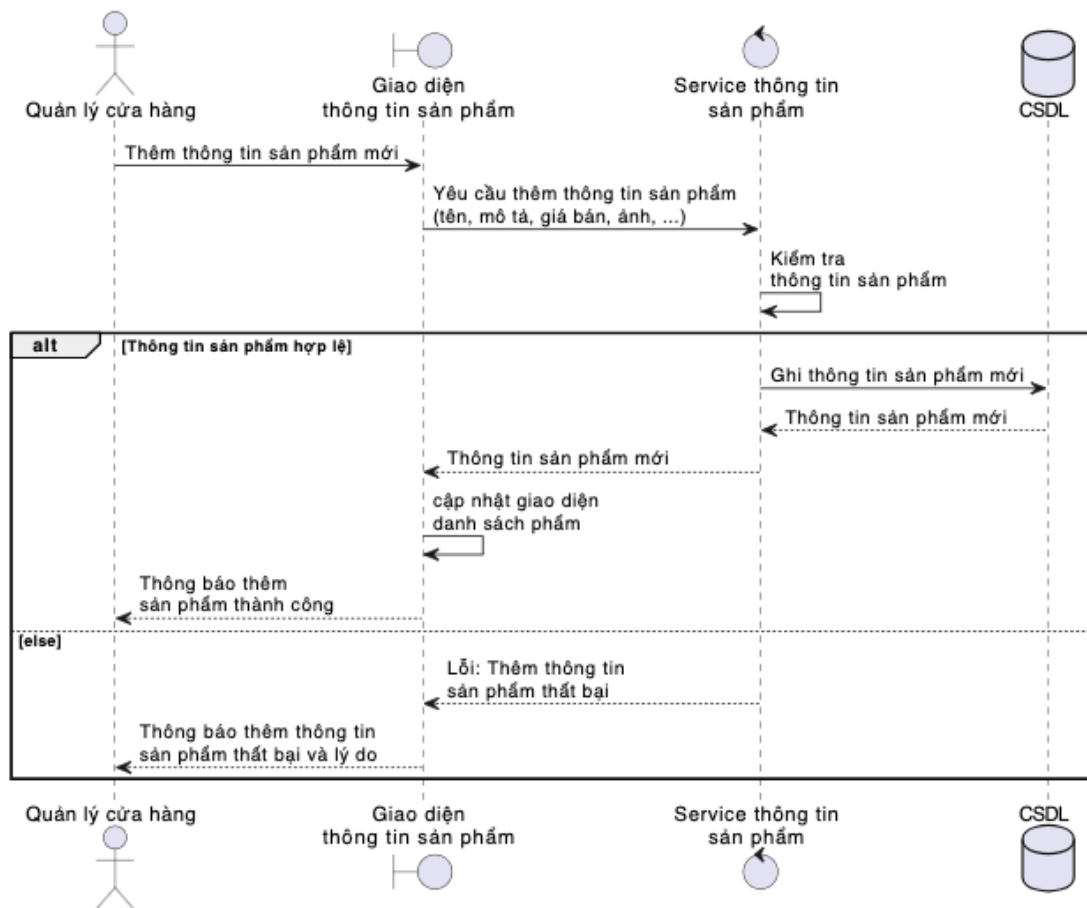


Hình 4.17 Use case Thêm mới thông tin sản phẩm

Tên	Thêm mới thông tin sản phẩm
Tác nhân chính	QLCH
Mô tả	QLCH muốn thêm mới thông tin sản phẩm cần bán
Điều kiện kích hoạt	QLCH chọn lệnh thêm mới thông tin sản phẩm từ trang quản lý sản phẩm
Điều kiện trước	<ul style="list-style-type: none"> - Tài khoản QLCH đã đăng nhập thành công - Hệ thống hoạt động ổn định
Điều kiện sau	<ul style="list-style-type: none"> - Thông tin sản phẩm sau khi thêm phải đủ tên, loại sản phẩm, giá bán, giá nhập, thời hạn sử dụng, ảnh - Danh sách sản phẩm phải hiện thông tin sản phẩm vừa thêm
Luồng chính	<ol style="list-style-type: none"> 1. QLCH chọn lệnh thêm sản phẩm mới trong trang quản lý sản phẩm 2. QLCH nhập thông tin sản phẩm mới như: tên sản phẩm, giá bán, số ngày sử dụng, ảnh, danh mục sản phẩm, tình trạng bày bán.

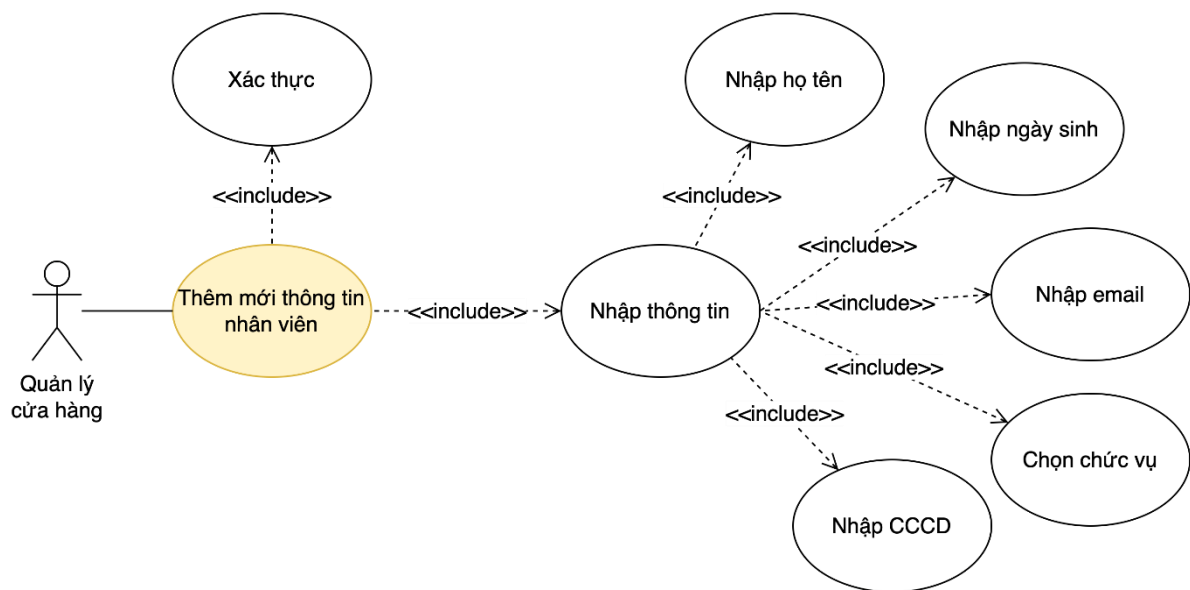
	<ul style="list-style-type: none"> 3. QLCH lưu thông tin sản phẩm mới 4. HT xác thực thông tin sản phẩm mới đảm bảo thỏa mãn các yêu cầu và ràng buộc 5. HT lưu thông tin sản phẩm mới 6. HT cập nhật giao diện danh sách sản phẩm có sản phẩm mới. 7. HT thông báo cho QLCH thêm thông tin sản phẩm mới thành công
Luồng ngoại lệ	<p>4.1 HT xác thực thông tin sản phẩm không thành công và hiển thị thông báo lỗi</p> <p>4.1.1. QLCH chọn huỷ nhập thông tin</p> <p>4.1.1.1 Use case dừng lại</p> <p>4.1.2 QLCH chỉnh sửa thông tin sản phẩm dựa trên thông báo lỗi.</p> <p>4.1.2.1 Quay lại bước 4</p>

Bảng 4.13 Đặc tả Use case Thêm mới thông tin sản phẩm



Hình 4.18 Sơ đồ tuần tự Use case Thêm mới thông tin sản phẩm

e) Use case Thêm mới thông tin nhân viên

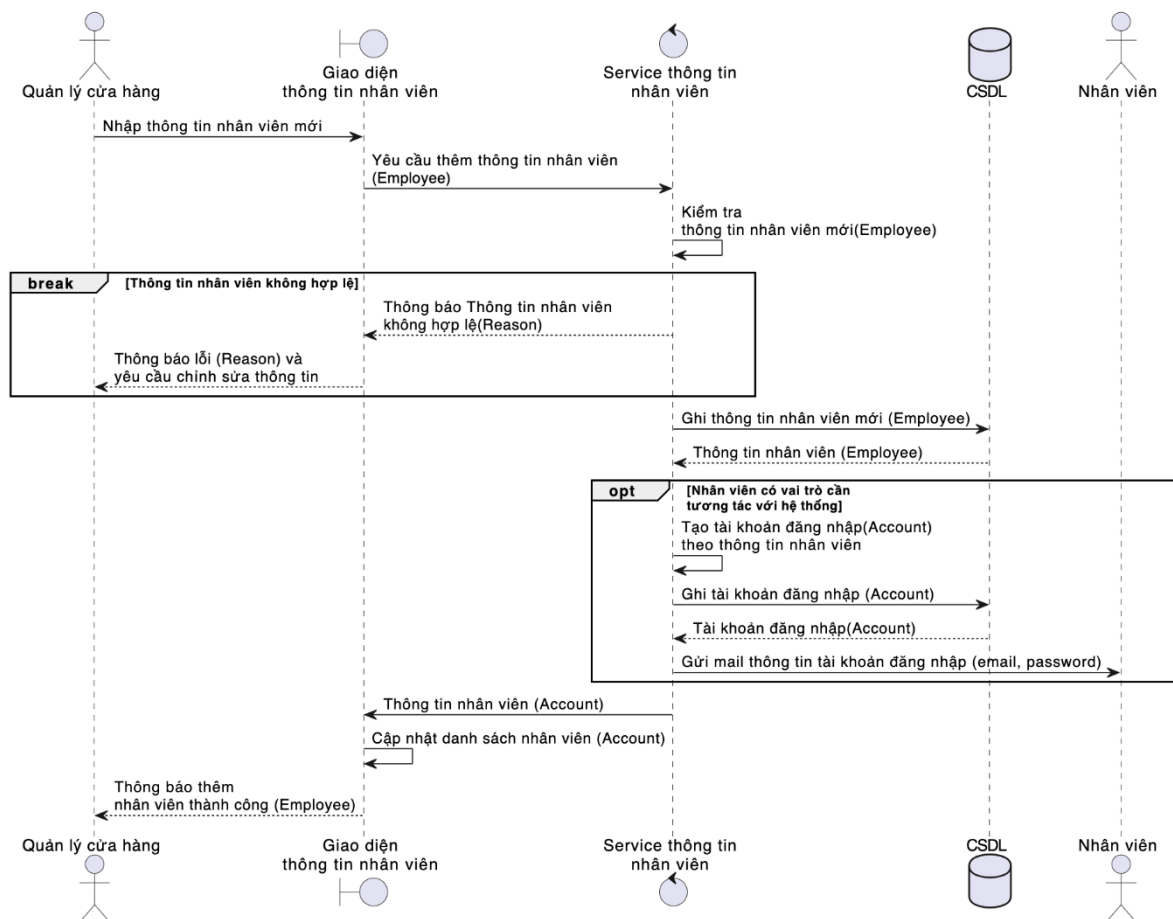


Hình 4.19 Use case Thêm mới thông tin nhân viên

Tên	Thêm mới thông tin nhân viên
Tác nhân chính	QLCH
Mô tả	QLCH có nhu cầu thêm thông tin nhân viên để quản lý nhân sự trong cửa hàng
Điều kiện kích hoạt	- QLCH chọn lệnh thêm mới thông tin nhân viên từ trang quản lý nhân viên
Điều kiện trước	- Tài khoản QLCH đã đăng nhập thành công
Điều kiện sau	- Thông tin nhân viên sau khi thêm phải đủ tên, ngày sinh, giới tính, số điện thoại, email, căn cước công dân, chức vụ
Luồng chính	<ol style="list-style-type: none"> 1. QLCH chọn lệnh mới thông tin nhân viên trong trang quản lý nhân viên 2. QLCH nhập các thông tin của nhân viên bao gồm tên, ngày sinh, giới tính, số điện thoại, email, căn cước công dân, chức vụ 3. QLCH lưu thông tin nhân viên mới 4. HT xác thực thông tin nhân viên mới đảm bảo thỏa mãn các yêu cầu và ràng buộc 5. HT lưu thông tin nhân viên mới 6. HT kiểm tra chức vụ của nhân viên 7. Nếu chức vụ của nhân viên thuộc chức vụ Quản lý kho, Giao hàng, Bán hàng <ul style="list-style-type: none"> - HT tạo tài khoản đăng nhập cho nhân viên - HT gửi thông tin tài khoản qua email của nhân viên 8. HT cập nhật giao diện danh sách nhân viên có nhân viên mới. 9. HT thông báo cho QLCH việc thêm thông tin mới thành công
Luồng thay thế	<p>7.1 Nhân viên mới không thuộc chức vụ Quản lý kho, Giao hàng, Bán hàng</p> <p>7.1.1 Bỏ qua xử lý bước 7</p>

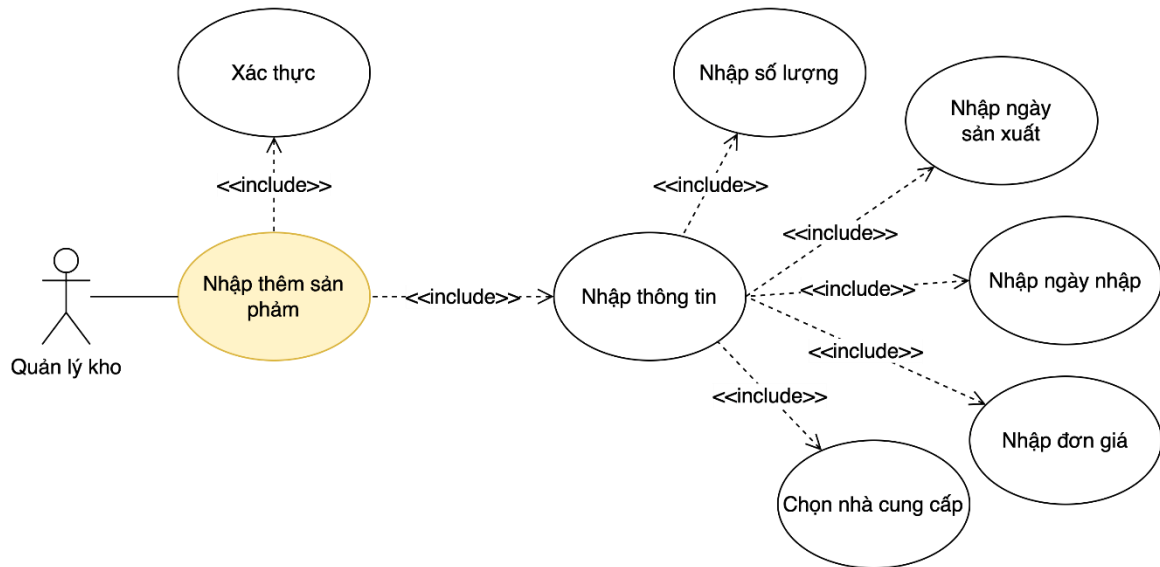
	7.1.2 Tiếp tục xử lý ở bước 8
Luồng ngoại lệ	<p>4.1 HT xác thực thông tin nhân viên không thành công và hiển thị thông báo lỗi</p> <p>4.1.1 QLCH chọn huỷ nhập thông tin</p> <p>4.1.1.1 Use case dừng lại</p> <p>4.1.2 QLCH chỉnh sửa thông tin nhân viên dựa trên thông báo lỗi.</p> <p>4.1.2.1 Use case tiếp tục tại bước 4</p>

Bảng 4.14 Đặc tả Use case Thêm mới thông tin nhân viên



Hình 4.20 Sơ đồ tuần tự Use case Thêm mới thông tin nhân viên

e) Use case Nhập thêm sản phẩm từ nhà cung cấp

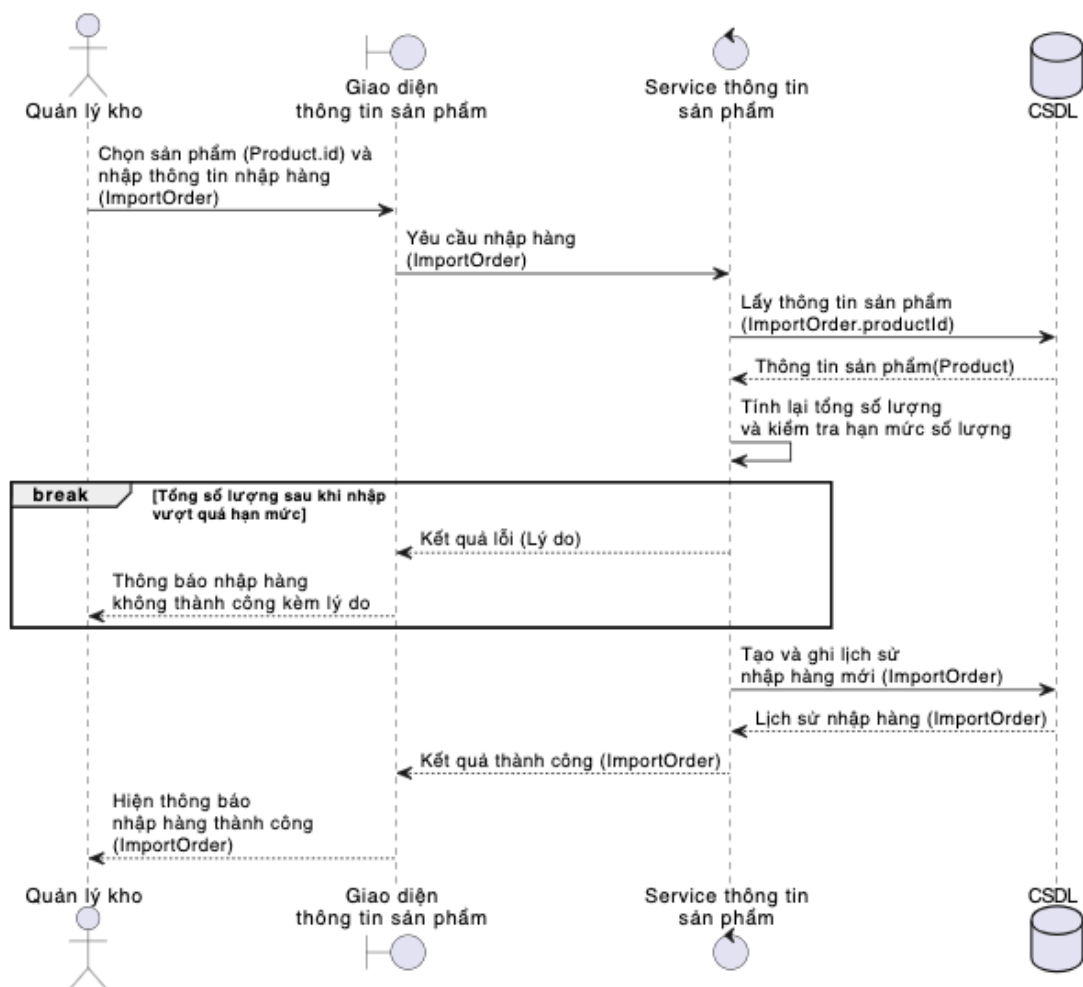


Hình 4.21 Use case Nhập thêm sản phẩm từ nhà cung cấp

Tên	Nhập thêm sản phẩm từ nhà cung cấp
Tác nhân chính	QLK
Mô tả	QLK cần nhập thêm sản phẩm từ nhà cung cấp để phục vụ nhu cầu mua bán của cửa hàng
Điều kiện kích hoạt	- QLK chọn lệnh nhập thêm sản phẩm từ trang chi tiết sản phẩm
Điều kiện trước	- Tài khoản QLK đã đăng nhập thành công - Nhà cung cấp sản phẩm đã có thỏa thuận mua bán với cửa hàng
Điều kiện sau	- Lịch sử nhập hàng phải được ghi lại.
Luồng chính	<ol style="list-style-type: none"> 1. QLK chọn xem thông tin chi tiết một sản phẩm trong trang quản lý sản phẩm 2. QLK lựa chọn nhập sản phẩm 3. HT hiển thị hộp thoại yêu cầu điền thông tin nhập hàng bao gồm nhà cung cấp, số lượng, ngày nhập và ngày sản xuất 4. QLK nhập thông tin và tạo yêu cầu nhập hàng 5. HT kiểm tra thông tin nhập hàng đảm bảo các yêu cầu và ràng buộc

	6. HT lưu lại thông tin nhập hàng 7. HT cập nhật lại trang thông tin sản phẩm với số lượng tồn mới
Luồng ngoại lệ	5.1 Thông tin nhập hàng không hợp lệ và hiển thị thông báo lỗi 5.1.1 QLK huỷ đơn nhập hàng 5.1.1.1 Use case dừng lại 5.2.1 QLK chỉnh sửa thông tin nhập hàng dựa trên thông báo lỗi 5.2.1.1 Tiếp tục tại bước 5

Bảng 4.15 Đặc tả Use case Nhập thêm sản phẩm từ nhà cung cấp

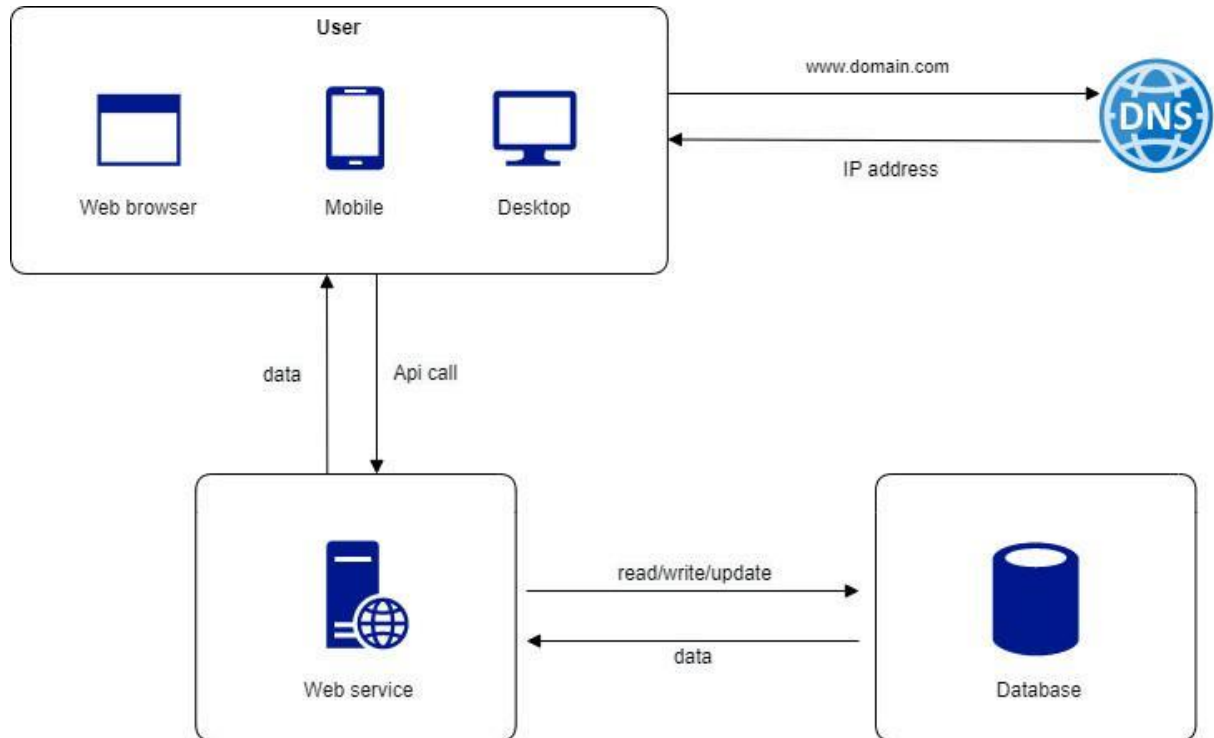


Hình 4.22 Sơ đồ tuần tự Use case Nhập thêm sản phẩm từ nhà cung cấp

CHƯƠNG 5: THIẾT KẾ PHẦN MỀM

5.1 Mô hình công nghệ

Kiến trúc phân tầng giao tiếp qua RESTful API

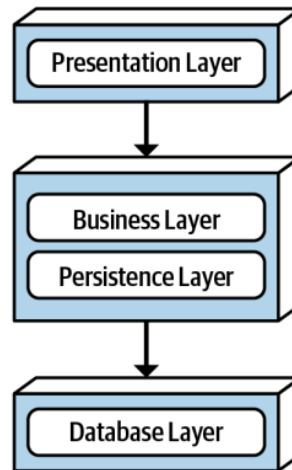


Hình 5.1 Mô hình kiến trúc lớp (layered architecture)

Loại kiến trúc: Kiến trúc lớp (layered architecture).

Mô tả:

- Hệ thống được xây dựng theo mỗi lớp đảm nhận chức năng riêng biệt.
- Kiến trúc không có ràng buộc về số lượng các lớp trên hệ thống, thông thường gồm 4 lớp quan trọng: *presentation*, *business*, *persistence* và *database*.
- Trong một số trường hợp, *business* và *persistence* layer có thể kết hợp lại thành 1 lớp chung



Hình 5.2 Các lớp trong Mô hình kiến trúc lớp (layered architecture)

- **Tầng presentation** là tầng giao diện người dùng, có trách nhiệm hiển thị giao diện người dùng, thúc đẩy sự tương tác giữa người dùng với ứng dụng, chuyển tiếp dữ liệu nhập từ người dùng cho tầng business, hiển thị thông tin từ tầng Business. Tầng này gồm: trình duyệt web, ứng dụng điện thoại, ứng dụng máy tính.
- **Tầng business** là Tầng nghiệp vụ, còn được gọi là tầng ứng dụng hoặc tầng logic, chứa các quy tắc và logic nghiệp vụ cốt lõi của ứng dụng. Nó xử lý các yêu cầu của người dùng từ tầng presentation, thực hiện các tính toán cần thiết và điều phối chức năng của ứng dụng. Cấp này chịu trách nhiệm triển khai các quy tắc kinh doanh, xác thực dữ liệu, điều phối quy trình công việc và đảm bảo tính toàn vẹn của ứng dụng. Nó hoạt động như một trung gian hòa giải giữa tầng presentation và tầng persistence.
- **Tầng persistence** là tầng truy cập dữ liệu hoặc tầng dữ liệu, chịu trách nhiệm lưu trữ và truy xuất dữ liệu từ cơ sở dữ liệu. Nó xử lý việc lưu trữ và truy xuất dữ liệu ứng dụng, đảm bảo dữ liệu luôn ổn định và sẵn sàng để sử dụng trong tương lai. Tầng này tương tác với hệ thống quản lý cơ sở dữ liệu (DBMS) để thực hiện các hoạt động như truy vấn, cập nhật và xóa dữ liệu. Nó trừu tượng hóa việc lưu trữ dữ liệu cơ bản và cung cấp giao diện (interface) nhất quán cho tầng business, tránh khỏi sự phức tạp của việc lưu trữ dữ liệu.
- **Tầng database** là tầng cơ sở dữ liệu đề cập đến hệ thống cơ sở dữ liệu thực tế nơi lưu trữ dữ liệu của ứng dụng. Nó có thể là một hệ thống quản lý cơ sở dữ liệu quan hệ (RDBMS) như MySQL, Oracle, PostgreSQL hoặc có thể là cơ sở dữ liệu NoSQL như MongoDB hoặc Cassandra, tùy thuộc vào yêu cầu của ứng dụng. Tầng cơ sở dữ liệu chịu trách nhiệm quản lý việc lưu trữ, truy xuất và thao tác dữ liệu dựa trên các hoạt động được thực hiện bởi tầng lưu giữ. Tầng này đảm bảo

tính toàn vẹn dữ liệu, thực thi các ràng buộc dữ liệu và cung cấp các cơ chế để truy cập và truy xuất dữ liệu hiệu quả.

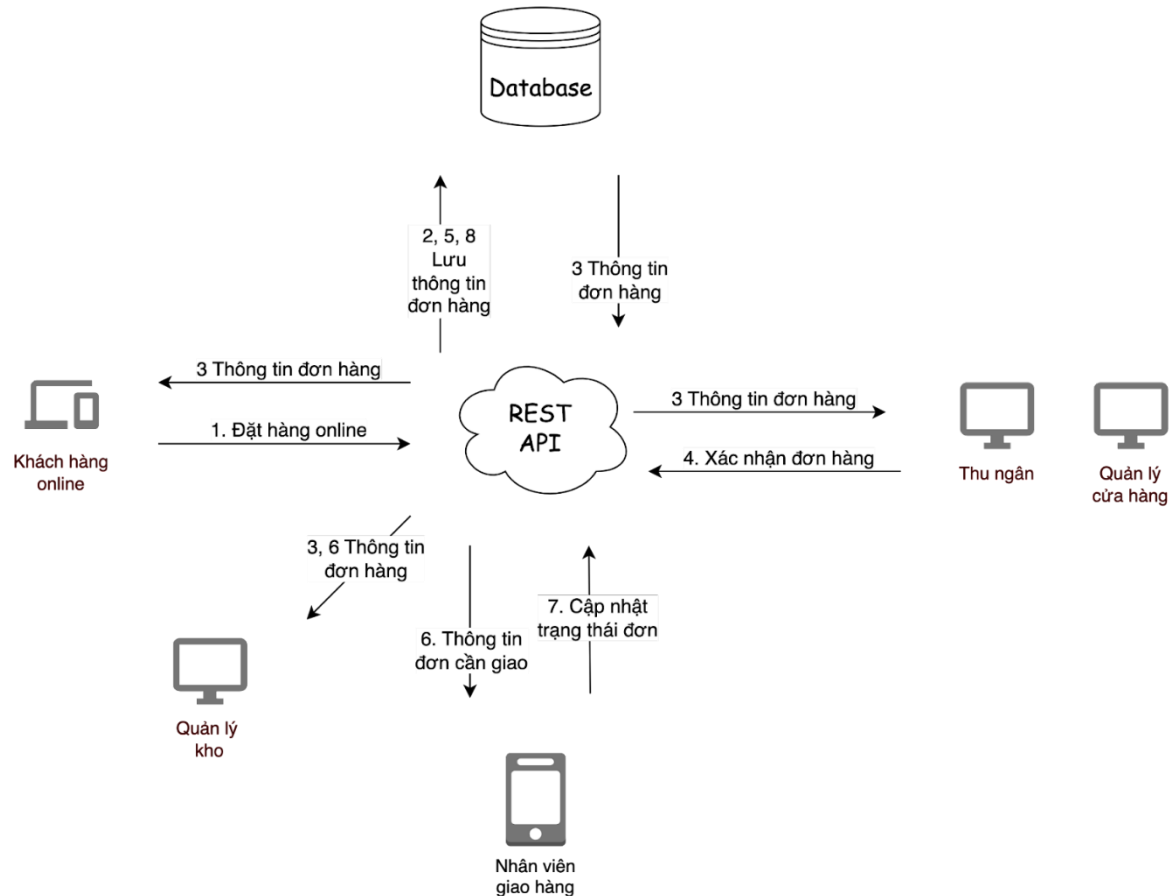
Lý do chọn kiến trúc lớp trong việc phát triển ứng dụng:

- **Khả năng mở rộng:** Kiến trúc ba tầng cho phép khả năng mở rộng bằng cách tách ứng dụng thành các tầng riêng biệt. Mỗi tầng có thể được mở rộng độc lập dựa trên nhu cầu cụ thể của hệ thống.
- **Khả năng bảo trì:** Mỗi tầng có thể được phát triển, thử nghiệm và duy trì độc lập, giúp cập nhật hoặc thay thế một tầng cụ thể dễ dàng hơn mà không ảnh hưởng đến toàn bộ ứng dụng.
- **Khả năng sử dụng lại:** Bản chất mô-đun của kiến trúc ba tầng tạo điều kiện cho việc tái sử dụng mã. Các thành phần trong mỗi tầng có thể được thiết kế để có thể tái sử dụng trên các phần khác nhau của ứng dụng hoặc thậm chí trong các ứng dụng khác, giúp tăng hiệu quả phát triển và giảm nỗ lực.
- **Bảo mật:** Với kiến trúc ba tầng, các biện pháp bảo mật có thể được triển khai ở mỗi tầng. Tầng trình bày có thể xử lý xác thực và ủy quyền của người dùng, tầng doanh nghiệp có thể thực thi các quy tắc kinh doanh và áp dụng kiểm tra bảo mật, còn tầng lưu giữ có thể bảo vệ cơ sở dữ liệu khỏi bị truy cập trái phép.

Hạn chế của kiến trúc:

- **Độ phức tạp:** Kiến trúc ba tầng có độ phức tạp cao hơn so với các kiến trúc đơn giản hơn. Việc tách biệt các mối quan tâm và liên lạc giữa các cấp đòi hỏi phải có sự thiết kế và phối hợp cẩn thận. Do đó có thể dẫn đến tăng thời gian và nỗ lực phát triển.
- **Đòi hỏi nhiều nỗ lực phát triển:** Phát triển một ứng dụng dựa trên kiến trúc ba tầng có thể đòi hỏi nhiều nỗ lực phát triển hơn so với các kiến trúc đơn giản hơn. Việc thiết kế, triển khai và thử nghiệm nhiều tầng có thể tốn nhiều thời gian và tài nguyên hơn.
- **Độ phức tạp của việc triển khai:** Việc triển khai và quản lý kiến trúc ba tầng có thể phức tạp hơn việc triển khai một ứng dụng nguyên khối (monolithic). Mỗi tầng có thể yêu cầu triển khai và cấu hình riêng, điều này có thể làm tăng chi phí hoạt động.

5.2 Sơ đồ ứng dụng



Hình 5.3 Sơ đồ ứng dụng

Sơ đồ mô tả tổng quan mô tả cách ứng dụng vận hành khi có khách hàng đặt hàng online theo trình tự:

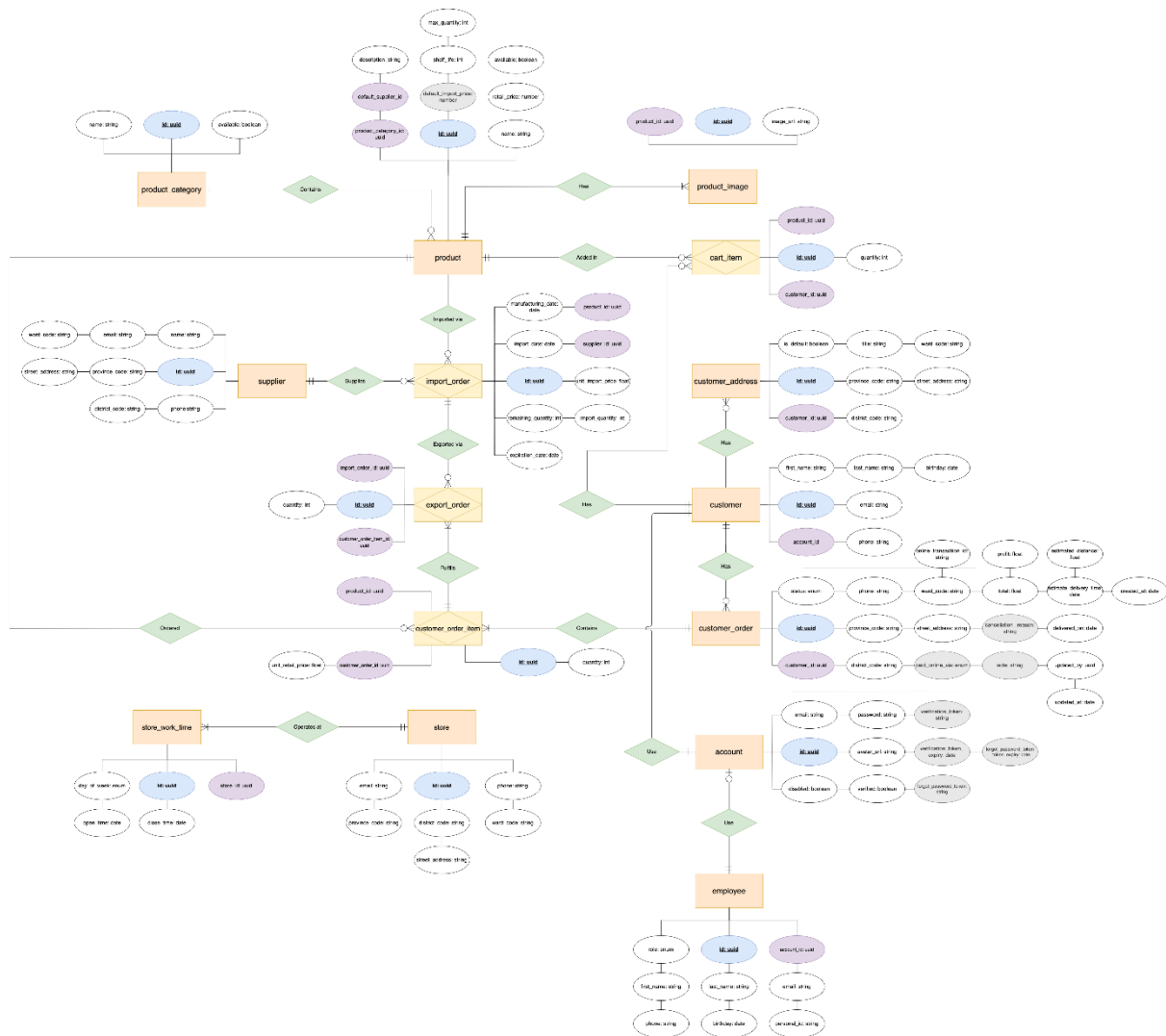
1. Khách hàng tiến hành đặt hàng online thông qua Website của hàng trên máy tính hoặc trên điện thoại
2. REST API sẽ nhận và xử lý yêu cầu đặt hàng của khách, sau đó tạo đơn hàng tương ứng. Thông tin đơn hàng sẽ được lưu lại trong cơ sở dữ liệu
3. Thông tin đơn hàng mới sẽ được thông báo cho các bên liên quan bao gồm Khách hàng, Thu ngân, Quản lý cửa hàng và Quản lý kho.
4. Thu ngân hoặc quản lý cửa hàng xem thông tin đơn hàng và xác nhận đơn
- 5, 6. Sau khi đơn hàng được xác nhận, REST API sẽ cập nhật thông tin đơn hàng vào cơ sở dữ liệu, sau đó thông tin đơn hàng sẽ được gửi cho Quản lý kho để chuẩn bị hàng và Nhân viên giao hàng để tiến hành giao hàng cho khách.

7, 8. Nhân viên giao hàng sau khi giao hàng thành công sẽ chuyển trạng thái đơn hàng. REST API tiếp nhận và cập nhật trạng thái của đơn hàng trên cơ sở dữ liệu.

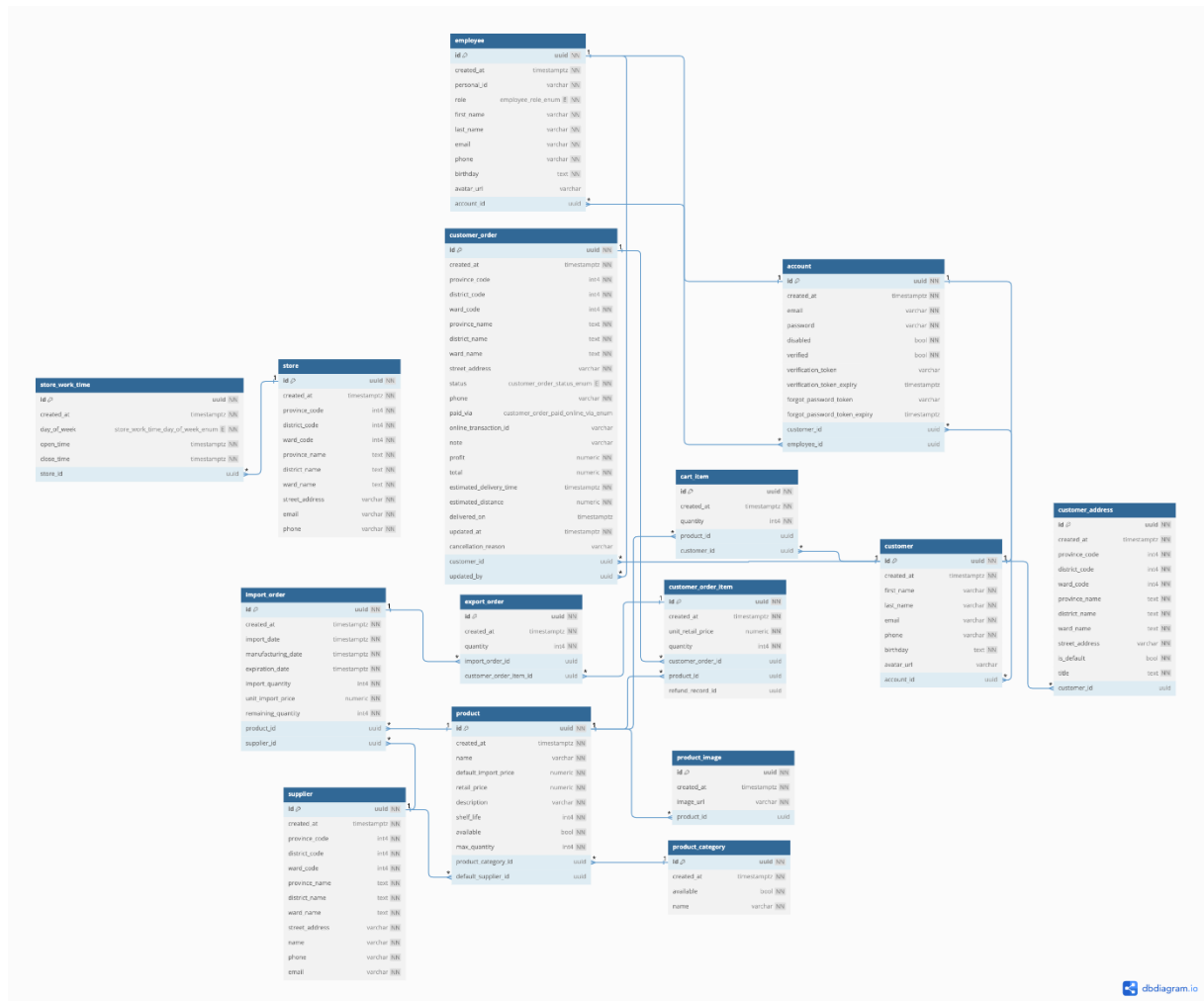
5.3 Thiết kế giao diện chức năng

5.4 Thiết kế cơ sở dữ liệu cho phần mềm

5.4.1 Thiết kế ERD

Hình 5.4 Lược đồ ERD (<https://shorturl.at/hjCO1>)

5.4.2 Thiết kế Lược đồ dạng chuẩn 3



Hình 5.5 Lược đồ Database (<https://shorturl.at/dlwJ6>)

- Mỗi bảng đều có 2 trường mặc định là
 - o **id**: Tránh việc dùng cặp khóa ngoại làm khóa chính (Bảng sinh ra từ quan hệ nhiều – nhiều) do đặc thù của dữ liệu có nhiều nhiều bảng khác cần tham chiếu đến các bảng bảng sinh ra từ quan hệ nhiều – nhiều, nếu dùng cặp khóa ngoại làm khóa chính sẽ dẫn đến rườm rà trong việc tham chiếu. ⇒ Vẫn cần đảm bảo rằng buộc cặp khóa ngoại đó là độc nhất trong bảng
 - o **created_at**: Dùng để phục vụ việc sắp xếp trong chương trình, khi user có nhu cầu sắp xếp theo thứ tự giảm dần/tăng dần của ngày tạo.
- **province_code, district_code, ward_code** là các mã tỉnh thành do Nhà nước định nghĩa

1. product_category (**id**, name, available)

Tên	Kiểu	Mô tả	Khoá	Ràng buộc
id	uuid	Mã danh mục	- Khoá chính	- Không null - Độc nhất
created_at	timestampz	Thời điểm tạo		- Không null
name	varchar	Tên danh mục		- Độc nhất
available	bool	Tình trạng kinh doanh của danh mục sản phẩm		- Không null - Default: true

Bảng 5.1 Table product_category

2. supplier (**id**, name, email, phone, province_code, district_code, ward_code, street_address)

Tên	Kiểu	Mô tả	Khoá	Ràng buộc
id	uuid	Mã nhà cung cấp	- Khoá chính	- Không null - Độc nhất
created_at	timestampz	Thời điểm tạo		- Không null
name	varchar	Tên nhà cung cấp		- Không null - Độc nhất
email	varchar	Email nhà cung cấp		- Không null - Độc nhất
phone	varchar	SDT nhà cung cấp		- Không null - Độc nhất
province_code	int4	Mã Thành phố / Tỉnh		- Không null
district_code	int4	Mã Quận / Huyện		- Không null
ward_code	int4	Mã Phường / Xã		- Không null
province_name	varchar	Tên Thành phố / Tỉnh, được lưu để tránh việc phải gọi API ngoài thường xuyên để tra cứu tên dựa theo mã khi hiển thị ở giao diện		- Không null
district_name	varchar	Tên Quận / Huyện được lưu để tránh việc phải gọi API ngoài thường xuyên		- Không null

		để tra cứu tên dựa theo mã khi hiển thị ở giao diện		
ward_name	varchar	Tên Phường / Xã, được lưu để tránh việc phải gọi API ngoài thường xuyên để tra cứu tên dựa theo mã khi hiển thị ở giao diện		- Không null
street_address	varchar	Địa chỉ cụ thể (Tên đường, số nhà, tầng,...)		- Không null

Bảng 5.2 Bảng supplier

3. product (**id**, **product_category_id**, **default_supplier_id**, name, default_import_price, retail_price, description, available, shelf_life, max_quantity)

Tên	Kiểu	Mô tả	Khoá	Ràng buộc
id	uuid	Mã sản phẩm	- Khoá chính	- Không null - Độc nhất
created_at	timestampz	Thời điểm tạo		- Không null
product_category_id	uuid	Danh mục mà sản phẩm thuộc về	- Khoá ngoại	- Không null
name	varchar	Tên sản phẩm		- Không null - Độc nhất
default_import_price	numeric	Giá nhập sản phẩm (Giá sỉ)		- Lớn hơn 0
retail_price	numeric	Giá bán sản phẩm (Giá lẻ)		- Không null - Lớn hơn giá nhập
description	varchar	Mô tả sản phẩm		- Tối đa 500 ký tự
shelf_life	int4	Số ngày có thể sử dụng của sản phẩm, khoảng cách giữa		- Không null - Lớn hơn 0

		importDate và expirationDate		
available	bool	Tình trạng kinh doanh của sản phẩm (Khác với tình trạng còn hàng của sản phẩm)		- Mặc định: true
default_supplier_id	uuid	Nhà cung cấp sản phẩm này ở lần gần nhất, dùng làm giá trị mặc định khi nhập sản phẩm nếu không có thay đổi từ user	- Khoá ngoại	
max_quantity	int4	Số lượng tồn kho tối đa của sản phẩm		- Không null - Lớn hơn 0

Bảng 5.3 Table product

Bảng 5.4 Table product

4. product_image (id, product_id, image_url)

Tên	Kiểu	Mô tả	Khoá	Ràng buộc
id	uuid	Mã ảnh sản phẩm	- Khoá chính	- Không null - Độc nhất
created_at	timestampz	Thời điểm tạo		- Không null
product_id	uuid	Sản phẩm mà ảnh này thuộc về	- Khoá ngoại	- Không null
image_url	varchar	Đường dẫn tới ảnh sản phẩm		- Không null - Độc nhất

Bảng 5.5 Table product_image

5. import_order (id, product_id, supplier_id, import_quantity, import_date, unit_import_price, remaining_quantity, expiration_date, manufacturing_date)

Tên	Kiểu	Mô tả	Khoá	Ràng buộc
id	uuid	Mã đơn nhập sản phẩm	- Khoá chính	- Không null - Độc nhất

created_at	timestampz	Thời điểm tạo		- Không null
product_id	uuid	Mã sản phẩm nhập	- Khoá ngoại	- Không null
supplier_id	uuid	Mã nhà cung cấp sản phẩm tại đơn nhập này	- Khoá ngoại	- Không null
import_quantity	int4	Số lượng nhập của đơn hàng		- Không null - Lớn hơn 0
import_date	timestampz	Ngày nhập sản phẩm		- Không null
unit_import_price	numeric	Giá nhập ngay tại thời điểm nhập, trường này tồn tại để tránh sự thay đổi của giá nhập trong tương lai ảnh hưởng đến lịch sử		- Không null - Lớn hơn 0
remaining_quantity	int4	Số lượng sản phẩm còn lại của đơn nhập này		- Không null - Lớn hơn 0 - Nhỏ hơn hoặc bằng import quantity
expiration_date	timestampz	Ngày hết hạn của sản phẩm thuộc đơn nhập này		- Không null - Lớn hơn ngày nhập và ngày sản xuất
manufacturing_date	timestampz	Ngày sản xuất của sản phẩm thuộc đơn nhập này		- Nhỏ hơn ngày nhập và ngày hết hạn

Bảng 5.6 Table import_order

6. export_order (id, import_order_id, customer_order_item_id, quantity)

Tên	Kiểu	Mô tả	Khoá	Ràng buộc
-----	------	-------	------	-----------

id	uuid	Mã lịch sử xuất hàng	- Khoá chính	- Không null - Độc nhất
created_at	timestampz	Thời điểm tạo		- Không null
import_order_id	uuid	Mã đơn nhập sản phẩm	- Khoá ngoại	- Không null
customer_order_item_id	uuid	Mã chi tiết đơn hàng	- Khoá Ngoại	- Không null
quantity	int4	Số lượng xuất trong 1 đơn nhập		- Không null - Lớn hơn 0

Bảng 5.7 Table export_order

7. account (**id**, email, verified, password, disabled, verification_token, verification_token_expiry, forgot_password_token, forgot_password_token_expiry, avatar_url)

Tên	Kiểu	Mô tả	Khoá	Ràng buộc
id	uuid	Mã tài khoản	- Khoá chính	- Không null - Độc nhất
created_at	timestampz	Thời điểm tạo		- Không null
customer_id	uuid	Mã khách hàng sử dụng tài khoản	- Khoá ngoại	
employee_id	uuid	Mã nhân viên sử dụng tài khoản		
email	varchar	Email tài khoản, dùng để đăng nhập		- Không null - Độc nhất
verified	bool	Tài khoản đã được xác thực hay chưa		- Default: 0

password	varchar	Mật khẩu đã được mã hoá của tài khoản		- Không null - Mật khẩu (Khi chưa mã hoá) phải ít nhất 10 ký tự, bao gồm ký tự đặc biệt và số
disabled	bool	Tình trạng hoạt động của tài khoản		- Default: true - Không null
verification_token	varchar	Token xác thực tài khoản		
verification_token_expiry	timestampz	HSD của token xác thực tài khoản		
forgot_password_token	varchar	Token xác thực khi quên mật khẩu		
forgot_password_token_expiry	timestampz	HSD của token xác thực khi quên mật khẩu		
avatar_url	varchar	Mã đường dẫn ảnh đại diện của tài khoản		

Bảng 5.8 Table account

8. employee (**id**, **account_id**, first_name, last_name, email, phone, birthday, personal_id, role)

Tên	Kiểu	Mô tả	Khoá	Ràng buộc
id	uuid	Mã nhân viên	- Khoá chính	- Không null - Độc nhất
created_at	timestampz	Thời điểm tạo		- Không null
account_id	uuid	Mã tài khoản của nhân viên	- Khóa ngoại	- Không null - Độc nhất
first_name	varchar	Tên nhân viên		
last_name	varchar	Họ và tên lót nhân viên		
email	varchar	Email của nhân viên		- Không null - Độc nhất
phone	varchar	SDT của khách hàng		- Không null - Độc nhất
birthday	timestampz	Ngày sinh của nhân viên		
personal_id	varchar	Mã CCCD của nhân viên		- Không null - Độc nhất
role	enum	Chức vụ của nhân viên: MANAGER, CASHIER, WAREHOUSE_MANAGER, WAREHOUSE_STAFF, SHIPPER		- Không null

Bảng 5.9 Table employee

9. customer (**id**, **account_id**, first_name, last_name, email, phone, birthday, avatar_url)

Tên	Kiểu	Mô tả	Khoá	Ràng buộc
id	uuid	Mã khách hàng	- Khoá chính	- Không null - Độc nhất
created_at	timestampz	Thời điểm tạo		- Không null
account_id	uuid	Mã tài khoản của khách hàng	- Khóa ngoại	- Không null - Độc nhất

first_name	varchar	Tên khách hàng		
last_name	varchar	Họ và tên lót khách hàng		
email	varchar	Email của khách hàng		- Không null - Độc nhất
phone	varchar	SDT của khách hàng		- Không null - Độc nhất
birthday	timestampz	Ngày sinh của khách hàng		

Bảng 5.10 Table customer

10. customer_address (**id**, customer_id, province_code, district_code, ward_code, street_address, is_default, title)

Tên	Kiểu	Mô tả	Khoá	Ràng buộc
id	uuid	Mã địa chỉ tài khoản	- Khoá chính	- Không null - Độc nhất
created_at	timestampz	Thời điểm tạo		- Không null
account_id	uuid	Mã tài khoản mà địa chỉ thuộc về	- Khoá ngoại	- Không null
province_code	int4	Mã Thành Phố / Tỉnh		- Không null
district_code	int4	Mã Quận / Huyện		- Không null
ward_code	int4	Mã Phường / Xã		- Không null
province_name	varchar	Tên Thành phố / Tỉnh, được lưu để tránh việc phải gọi API ngoài thường xuyên để tra cứu tên dựa theo mã khi hiển thị ở giao diện		- Không null
district_name	varchar	Tên Quận / Huyện được lưu để tránh việc phải gọi API ngoài thường		- Không null

		xuên để tra cứu tên dựa theo mã khi hiển thị ở giao diện		
ward_name	varchar	Tên Phường / Xã, được lưu để tránh việc phải gọi API ngoài thường xuên để tra cứu tên dựa theo mã khi hiển thị ở giao diện		Không null
street_address	varchar	Địa chỉ cụ thể (Tên đường, số nhà)		- Không null
is_default	bool	Địa chỉ mặc định dùng để điền tự động thông tin khi thanh toán		- Không null
title	varchar	Tên địa chỉ		- Không null - Độc nhất

Bảng 5.11 Table customer_address

11. customer_order (**id**, **customer_id**, status, phone, province_code, district_code, ward_code, street_address, paid_via, online_transaction_id, profit, total, cancellation_reason, note, estimated_distance, estimated_delivery_time, delivered_on, updated_by, updated_at, created_at)

Tên	Kiểu	Mô tả	Khoá	Ràng buộc
id	uuid	Mã đơn hàng	- Khoá chính	- Không null - Độc nhất
created_at	timestamptz	Thời điểm tạo		- Không null
customer_id	uuid	Mã của khách hàng sở hữu đơn hàng này	- Khóa ngoại	- Không null
status	enum	Trạng thái đơn hàng: PENDING, PROCESSING, SHIPPING,		- Không null

		SHIPPED, CANCELLED,		
phone	varchar	Số điện thoại liên lạc khi giao hàng.		- Không null
province_code	int4	Mã Thành phố / Tỉnh		- Không null
district_code	int4	Mã Quận / Huyện		- Không null
ward_code	int4	Mã Phường / Xã		- Không null
province_name	varchar	Tên Thành phố / Tỉnh, được lưu để tránh việc phải gọi API ngoài thường xuyên để tra cứu tên dựa theo mã khi hiển thị ở giao diện		- Không null
district_name	varchar	Tên Quận / Huyện được lưu để tránh việc phải gọi API ngoài thường xuyên để tra cứu tên dựa theo mã khi hiển thị ở giao diện		- Không null
ward_name	varchar	Tên Phường / Xã, được lưu để tránh việc phải gọi API ngoài thường xuyên để tra cứu tên dựa theo mã khi hiển thị ở giao diện		Không null
street_address	varchar	Địa chỉ cụ thể (Tên đường, số nhà)		- Không null
paid_via	enum	Chọn phương thức thanh toán: COD, PayPal		- Không null
online_transacti on_id	varchar	Mã giao dịch của Paypal		

note	varchar	Ghi chú đơn hàng		- Tối đa 500 ký tự
total	numeric	Tổng tiền đơn hàng, tính bằng tổng của các retail_price x quantity trong từng customer_order_items		- Không null - Lớn hơn 0
profit	numeric	Lợi nhuận đơn hàng, tính bằng tổng tiền hàng trừ tổng của import_price x quantity trong từng customer_order_item		- Không null - Lớn hơn 0
estimated_delivery_time	timestamptz	Thời gian giao hàng dự kiến, sử dụng Google Maps API để tính toán		- Không null
estimated_distance	numeric	Quãng đường giao hàng ngắn nhất dự kiến, đơn vị km, sử dụng Google Maps API để tính toán		- Không null
delivered_on	timestamptz	Thời điểm đã giao hàng thành công		
updated_at	timestamptz	Thời điểm cập nhật đơn hàng		- Không null
updated_by	uuid	Người cuối cùng cập nhật đơn hàng (Khách hàng hoặc nhân viên)		- Không null
cancellation_reason	varchar	Lý do hủy đơn		- Tối đa 500 ký tự

Bảng 5.12 Table customer_order

12. customer_order_item (**id**, customer_order_id, product_id, refund_record_id, quantity, unit_retail_price,)

Tên	Kiểu	Mô tả	Khoá	Ràng buộc
-----	------	-------	------	-----------

id	uuid	Mã chi tiết đơn hàng	- Khoá chính	- Không null - Độc nhất
created_at	timestampz	Thời điểm tạo		- Không null
customer_order_id	uuid	Mã đơn hàng mà chi tiết này thuộc về	- Khoá ngoại	- Không null
product_id	uuid	Mã sản phẩm của chi tiết đơn hàng này	- Khoá ngoại	- Không null
refund_record_id	uuid	Mã phiếu hoàn trả nếu có	- Khoá ngoại	
quantity	int4	Số lượng sản phẩm		- Không null - Lớn hơn 0
unit_retail_price	numeric	Giá bán lẻ của sản phẩm, dùng chung với unit_import_price để tính lợi nhuận (profit). Trường này tồn tại để tránh sự thay đổi của giá nhập trong tương lai ảnh hưởng đến lịch sử		- Không null - Lớn hơn giá nhập

Bảng 5.13 Table customer_order_item

13. cart_item (id, product_id, customer_id, quantity)

Tên	Kiểu	Mô tả	Khoá	Ràng buộc
id	uuid	Mã chi tiết giỏ hàng	- Khoá chính	- Không null - Độc nhất
created_at	timestampz	Thời điểm tạo		- Không null
product_id	uuid	Mã sản phẩm của chi tiết giỏ hàng này	- Khoá ngoại	- Không null
customer_id	uuid	Mã khách hàng sở hữu chi tiết giỏ hàng này	- Khoá ngoại	- Không null
quantity	int4	Số lượng sản phẩm của chi tiết giỏ hàng		- Không null - Lớn hơn 0

Bảng 5.14 Table cart_item

14. store (**id**, phone, email, province_code, district_code, ward_code, street_address)

Tên	Kiểu	Mô tả	Khoá	Ràng buộc
id	uuid	Mã cửa hàng	- Khoá chính	- Không null - Độc nhất
created_at	timestampz	Thời điểm tạo		- Không null
phone	varchar	Số điện thoại cửa hàng		- Không null - Độc nhất
email	varchar	Email cửa hàng		- Không null - Độc nhất
province_code	int4	Mã Thành phố / Tỉnh		- Không null
district_code	int4	Mã Quận / Huyện		- Không null
ward_code	int4	Mã Phường / Xã		- Không null
province_name	varchar	Tên Thành phố / Tỉnh, được lưu để tránh việc phải gọi API ngoài thường xuyên để tra cứu tên dựa theo mã khi hiển thị ở giao diện		- Không null
district_name	varchar	Tên Quận / Huyện được lưu để tránh việc phải gọi API ngoài thường xuyên để tra cứu tên dựa theo mã khi hiển thị ở giao diện		- Không null
ward_name	varchar	Tên Phường / Xã, được lưu để tránh việc phải gọi API ngoài thường xuyên để tra cứu tên dựa theo mã khi hiển thị ở giao diện		- Không null
street_address	varchar	Địa chỉ cụ thể (Tên đường, số nhà)		- Không null

Bảng 5.15 Table store

15. store_work_time (**id**, **store_id**, day_of_week, open_time, close_time)

Tên	Kiểu	Mô tả	Khoá	Ràng buộc
id	uuid	Mã giờ làm việc	- Khoá chính	- Không null - Độc nhất
created_at	timestampz	Thời điểm tạo		- Không null
store_id	uuid	Mã cửa hàng mà giờ làm việc này thuộc về		- Không null
day_of_week	enum	Cho biết giờ làm việc này thuộc về thứ nào trong tuần. Thuộc 1 trong 7 giá trị: MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY, SUNDAY		- Không null
open_time	timestampz	Giờ mở cửa		- Không null
close_time	timestampz	Giờ đóng cửa		- Không null - Trễ hơn giờ mở cửa

Bảng 5.16 Table store_work_time

CHƯƠNG 6: KỊCH BẢN KIỂM THỬ

CHƯƠNG 7: KẾT LUẬN

7.1 Kết quả đạt được

- **Xây dựng hệ thống theo yêu cầu của đề tài căn bản hoàn thiện:** Hệ thống đã được xây dựng hoàn chỉnh, tích hợp các chức năng cần thiết như quản lý hàng tồn kho, đơn hàng, thông tin khách hàng, và quá trình mua bán.
- **Giao diện trực quan và dễ sử dụng:** Giao diện người dùng được thiết kế thân thiện, giúp việc quản lý và mua bán thực phẩm trở nên dễ dàng và hiệu quả hơn.
- **Tích hợp hệ thống thanh toán trực tuyến:** Hệ thống hỗ trợ các phương thức thanh toán trực tuyến như PayPal, giúp tối ưu hóa quá trình thanh toán và tăng cường sự tiện lợi cho người dùng.
- **Tối ưu hoá quản lý kho và dữ liệu:** Cơ sở dữ liệu và quy trình quản lý kho được thiết kế hiệu quả, giúp theo dõi hàng tồn kho một cách chính xác và kịp thời.

7.2 Các hạn chế

- **Chưa có khả năng phân tích dữ liệu:** Hệ thống hiện tại chưa hỗ trợ một số tính năng nâng cao như dự báo nhu cầu mua hàng hay phân tích xu hướng thị trường.
- **Giới hạn về khả năng mở rộng quy mô:** Thiết kế của hệ thống chỉ phù hợp cho mô hình một chi nhánh đơn lẻ, chưa thể mở rộng, áp dụng rộng rãi cho mô hình chuỗi cửa hàng.
- **Hạn chế về tích hợp với các hệ thống khác:** Hiện tại, hệ thống chưa thể tích hợp linh hoạt với các hệ thống quản lý cung ứng hoặc CRM (Customer Relationship Management) của bên thứ ba.
- **Thách thức về bảo mật dữ liệu:** Mặc dù đã có các biện pháp bảo mật cơ bản, nhưng hệ thống vẫn cần được cải thiện về khả năng bảo vệ dữ liệu khách hàng và thông tin giao dịch mới có thể áp dụng vào môi trường thực tế.

7.3 Hướng phát triển

- **Phát triển tính năng phân tích dữ liệu:** Phát triển các công cụ phân tích dữ liệu nâng cao để dự báo nhu cầu và xu hướng mua hàng, giúp quyết định kinh doanh trở nên thông minh và chính xác hơn.
- **Nâng cấp kiến trúc hệ thống:** Phát triển kiến trúc hệ thống hiện tại để hỗ trợ mô hình đa chi nhánh, cho phép quản lý và tích hợp dữ liệu từ nhiều cửa hàng một cách linh hoạt và hiệu quả.

CHƯƠNG 8: PHỤ LỤC

1. NextJS Document

<https://nextjs.org/docs>

2. TypeORM Document

<https://typeorm.io/>

3. PostgreSQL Tutorial

<https://www.postgresqltutorial.com/>

4. Google Maps Platform Documentation

<https://developers.google.com/maps/documentation/routes>

5. PayPal Documentation

<https://developer.paypal.com/home>

6. Province-API Documentation

<https://provinces.open-api.vn/redoc>