

ĐẠI HỌC BÁCH KHOA HÀ NỘI

# ĐỒ ÁN TỐT NGHIỆP

Hệ thống quản lý bán hàng POS cho mô hình bán lẻ  
lưu động

NGUYỄN DUY THÀNH  
thanh.nd204691@sis.hust.edu.vn

Ngành Khoa học máy tính

Giảng viên hướng dẫn: ThS. Nguyễn Mạnh Tuấn

Chữ kí GVHD

Khoa: Khoa học máy tính

Trường: Công nghệ Thông tin và Truyền thông

HÀ NỘI, 1/2025

# LỜI CAM KẾT

Họ và tên sinh viên: Nguyễn Duy Thành

MSSV: 20204691

Điện thoại liên lạc: 0333139967

Email: thanh.nd204691@sis.hust.edu.vn

Lớp: KHMT 02-K65

Hệ đào tạo: cử nhân

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

*Hà Nội, ngày      tháng      năm*

Tác giả ĐATN

*Họ và tên sinh viên*

# LỜI CẢM ƠN

Lời đầu tiên, em xin gửi lời cảm ơn chân thành và sự tri ân sâu sắc đến thầy ThS. Nguyễn Mạnh Tuấn, người đã tận tình hướng dẫn, chỉ bảo và đóng góp nhiều ý kiến quý báu giúp em hoàn thành đồ án tốt nghiệp. Sự tận tâm và chỉ dẫn của thầy là nguồn động lực lớn lao giúp em vượt qua những khó khăn trong suốt quá trình thực hiện đồ án.

Em cũng xin gửi lời cảm ơn sâu sắc đến các thầy cô trong Trường Đại học Công nghệ Thông tin và Truyền thông, cũng như các thầy cô trong Đại học Bách khoa Hà Nội, những người đã tận tâm giảng dạy và trang bị cho em những kiến thức chuyên môn cũng như kỹ năng cần thiết, tạo nền tảng vững chắc để em hoàn thiện đồ án và phục vụ công việc sau này.

Cuối cùng, em xin chân thành cảm ơn gia đình và bạn bè đã luôn quan tâm, động viên và tạo mọi điều kiện tốt nhất cho em trong suốt quá trình học tập và thực hiện đồ án tốt nghiệp.

Trong quá trình thực hiện đồ án, do còn hạn chế về kiến thức và kinh nghiệm, em rất mong nhận được sự góp ý từ các thầy cô để đồ án của em được hoàn thiện hơn.

Em xin chân thành cảm ơn!

# TÓM TẮT NỘI DUNG ĐỒ ÁN

Hiện nay, các cửa hàng bán lẻ mô hình lưu động gặp phải nhiều khó khăn trong việc quản lý giao dịch, theo dõi tồn kho và cập nhật dữ liệu giữa các điểm bán hàng cố định và lưu động. Các giải pháp hiện tại chủ yếu tập trung vào các hệ thống POS truyền thống, nhưng vẫn còn hạn chế trong việc quản lý đa điểm bán và tính linh hoạt khi mở rộng mô hình kinh doanh. Các hướng tiếp cận bao gồm việc sử dụng các hệ thống POS trực tuyến hoặc phần mềm quản lý lưu động, tuy nhiên, nhiều hệ thống thiếu tính tương thích và đồng bộ giữa các cửa hàng cố định và các điểm bán hàng lưu động.

Hướng tiếp cận mà em lựa chọn là xây dựng một hệ thống POS hỗ trợ việc quản lý đồng bộ giữa các cửa hàng cố định và lưu động, giúp giảm thiểu rủi ro trong việc kiểm soát tồn kho và giao dịch bán hàng. Em chọn hướng tiếp cận này vì nó có thể giải quyết hiệu quả các vấn đề về quản lý và mở rộng trong môi trường bán lẻ lưu động, đồng thời tối ưu hóa quá trình vận hành và báo cáo của hệ thống.

Giải pháp mà em triển khai bao gồm một ứng dụng web-based POS, tích hợp các tính năng như quản lý tồn kho, xử lý giao dịch bán hàng, báo cáo doanh thu, và chuyển giao hàng hóa giữa các điểm bán giúp các nhân viên bán hàng, quản lý điểm bán và chủ cửa hàng có thể truy cập thông tin một cách nhanh chóng và chính xác.

Đóng góp chính của đồ án là phát triển một hệ thống POS hoàn chỉnh và linh hoạt cho mô hình bán lẻ lưu động, giúp các cửa hàng này nâng cao hiệu quả hoạt động và cải thiện chất lượng quản lý bán hàng. Sau khi triển khai, hệ thống đã được kiểm tra và đánh giá, cho thấy kết quả đáng khích lệ về tính hiệu quả trong việc tối ưu hóa quy trình bán hàng.

Sinh viên thực hiện

(Ký và ghi rõ họ tên)

## MỤC LỤC

<b>CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI.....</b>	<b>1</b>
1.1 Đặt vấn đề.....	1
1.2 Mục tiêu và phạm vi đề tài.....	2
1.3 Định hướng giải pháp.....	2
1.4 Bố cục đồ án .....	3
<b>CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU.....</b>	<b>4</b>
2.1 Khảo sát hiện trạng .....	4
2.2 Tổng quan chức năng .....	6
2.2.1 Biểu đồ use case tổng quát .....	7
2.2.2 Biểu đồ use case phân rã Xử lý giao dịch bán hàng.....	8
2.2.3 Biểu đồ use case phân rã Quản lý luân chuyển hàng hóa .....	9
2.2.4 Biểu đồ use case phân rã Quản lý tồn kho tại điểm bán.....	10
2.2.5 Biểu đồ use case phân rã Quản lý thông tin hàng hóa.....	10
2.2.6 Biểu đồ use case phân rã Quản lý kế hoạch điểm bán .....	11
2.2.7 Biểu đồ use case phân rã Quản lý thông tin điểm bán .....	12
2.2.8 Biểu đồ use case phân rã Quản lý nhân viên .....	12
2.2.9 Quy trình nghiệp vụ .....	13
2.3 Đặc tả chức năng .....	16
2.3.1 Đặc tả use case Xử lý giao dịch bán hàng .....	16
2.3.2 Đặc tả use case Quản lý luân chuyển hàng hóa.....	17
2.3.3 Đặc tả use case Quản lý thông tin sản phẩm.....	18
2.3.4 Đặc tả use case Quản lý tồn kho điểm bán.....	19
2.3.5 Đặc tả use case Quản lý thông tin điểm bán .....	20
2.3.6 Đặc tả use case Quản lý nhân viên .....	21

2.3.7 Đặc tả use case Quản lý kế hoạch điểm bán .....	22
2.4 Yêu cầu phi chức năng .....	22
2.4.1 Tính dễ sử dụng .....	22
2.4.2 Tính dễ bảo trì .....	22
2.4.3 Độ tin cậy.....	23
2.4.4 Tính bảo mật .....	23
<b>CHƯƠNG 3. CÔNG NGHỆ SỬ DỤNG.....</b>	<b>24</b>
3.1 API (Application Programming Interface) .....	24
3.2 Spring Boot.....	24
3.3 React .....	25
3.4 MySQL .....	26
<b>CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG ....</b>	<b>28</b>
4.1 Thiết kế kiến trúc.....	28
4.1.1 Lựa chọn kiến trúc phần mềm .....	28
4.1.2 Thiết kế tổng quan.....	30
4.1.3 Thiết kế chi tiết gói .....	34
4.2 Thiết kế chi tiết.....	36
4.2.1 Thiết kế giao diện .....	36
4.2.2 Thiết kế lớp .....	40
4.2.3 Thiết kế cơ sở dữ liệu .....	44
4.3 Xây dựng ứng dụng.....	51
4.3.1 Thư viện và công cụ sử dụng.....	51
4.3.2 Kết quả đạt được .....	51
4.3.3 Minh họa các chức năng chính .....	52
4.4 Kiểm thử.....	56
4.4.1 Kiểm tra chức năng bán hàng .....	56

4.4.2 Kiểm tra chức năng luân chuyển hàng hóa .....	57
4.5 Triển khai .....	57
<b>CHƯƠNG 5. CÁC GIẢI PHÁP VÀ ĐÓNG GÓP NỔI BẬT .....</b>	<b>59</b>
5.1 Quản lý luân chuyển hàng hóa giữa các điểm bán .....	59
5.1.1 Đặt vấn đề .....	59
5.1.2 Giải pháp .....	59
5.1.3 Kết quả đạt được .....	63
5.2 Tạo kế hoạch bán hàng .....	65
5.2.1 Đặt vấn đề .....	65
5.2.2 Giải pháp .....	65
5.2.3 Kết quả đạt được .....	67
<b>CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN .....</b>	<b>69</b>
6.1 Kết luận .....	69
6.2 Hướng phát triển .....	69
<b>TÀI LIỆU THAM KHẢO .....</b>	<b>71</b>
<b>PHỤ LỤC .....</b>	<b>73</b>
<b>A. ĐẶC TẢ BẢNG TRONG CƠ SỞ DỮ LIỆU .....</b>	<b>73</b>
A.1 Bảng <b>balance_variant</b> .....	73
A.2 Bảng <b>base_product</b> .....	73
A.3 Bảng <b>customer</b> .....	74
A.4 Bảng <b>customer_group</b> .....	74
A.5 Bảng <b>feedback</b> .....	75
A.6 Bảng <b>import_item</b> .....	75
A.7 Bảng <b>order_line</b> .....	76
A.8 Bảng <b>return_order</b> .....	76
A.9 Bảng <b>return_order_line</b> .....	77

A.10 Bảng <b>payment</b> .....	77
A.11 Bảng <b>role</b> .....	78
A.12 Bảng <b>user_role</b> .....	78
A.13 Bảng <b>user_shop_access</b> .....	78
A.14 Bảng <b>shift</b> .....	79
A.15 Bảng <b>vendor</b> .....	79
A.16 Bảng <b>transfer_item</b> .....	80



## DANH MỤC HÌNH VẼ

Hình 2.1	Lightspeed [1] . . . . .	4
Hình 2.2	Zettle [2] . . . . .	5
Hình 2.3	Biểu đồ use case tổng quan . . . . .	7
Hình 2.4	Biểu đồ use case phân rã Xử lý giao dịch bán hàng . . . . .	8
Hình 2.5	Biểu đồ use case phân rã Quản lý luân chuyển hàng hóa . . . . .	9
Hình 2.6	Biểu đồ use case phân rã Quản lý tồn kho tại điểm bán . . . . .	10
Hình 2.7	Biểu đồ use case phân rã Quản lý thông tin hàng hóa . . . . .	10
Hình 2.8	Biểu đồ use case phân rã Quản lý kế hoạch điểm bán . . . . .	11
Hình 2.9	Biểu đồ use case phân rã Quản lý thông tin điểm bán . . . . .	12
Hình 2.10	Biểu đồ use case phân rã Quản lý nhân viên . . . . .	12
Hình 2.11	Biểu đồ quy trình bán hàng . . . . .	13
Hình 2.12	Biểu đồ quy trình luân chuyển hàng hóa . . . . .	15
Hình 3.1	Biểu đồ cách hoạt động của API [3] . . . . .	24
Hình 3.2	Spring Boot [4] . . . . .	25
Hình 3.3	React[5] . . . . .	25
Hình 3.4	Biểu đồ cách hoạt động của Virtual DOM trong React[6] . . . . .	26
Hình 3.5	MySQL [7] . . . . .	27
Hình 4.1	Mô hình kiến trúc 3 tầng [8] . . . . .	28
Hình 4.2	Thiết kế kiến trúc MVC [9] . . . . .	29
Hình 4.3	Biểu đồ phụ thuộc gói backend . . . . .	30
Hình 4.4	Biểu đồ phụ thuộc gói frontend . . . . .	32
Hình 4.5	Biểu đồ chi tiết gói chức năng Luân chuyển hàng hóa . . . . .	34
Hình 4.6	Thiết kế màn hình tổng quan . . . . .	36
Hình 4.7	Thiết kế màn hình bán hàng . . . . .	37
Hình 4.8	Thiết kế màn hình đơn hàng . . . . .	37
Hình 4.9	Thiết kế màn hình sản phẩm . . . . .	38
Hình 4.10	Thiết kế màn hình khách hàng . . . . .	38
Hình 4.11	Thiết kế màn hình kho hàng . . . . .	39
Hình 4.12	Thiết kế màn hình thống kê . . . . .	39
Hình 4.13	Thiết kế màn hình kế hoạch bán hàng . . . . .	40
Hình 4.14	Thiết kế lớp TransferController . . . . .	40
Hình 4.15	Thiết kế lớp TransferService . . . . .	41
Hình 4.16	Thiết kế lớp ProductController . . . . .	42
Hình 4.17	Thiết kế lớp ProductService . . . . .	43

Hình 4.18	Sơ đồ thực thể liên kết . . . . .	44
Hình 4.19	Giao diện bán hàng . . . . .	52
Hình 4.20	Giao diện chi tiết đơn hàng . . . . .	53
Hình 4.21	Giao diện chi tiết sản phẩm . . . . .	53
Hình 4.22	Giao diện luân chuyển hàng hóa . . . . .	54
Hình 4.23	Giao diện thống kê . . . . .	54
Hình 4.24	Giao diện quản lý nhân viên . . . . .	55
Hình 4.25	Giao diện kế hoạch bán hàng . . . . .	55
Hình 5.1	Giao diện tạo đơn chuyển hàng . . . . .	60
Hình 5.2	Sơ đồ chuyển đổi trạng thái . . . . .	60
Hình 5.3	Giao diện đơn khi đang tạo . . . . .	61
Hình 5.4	Giao diện đơn khi đang chuyển hàng . . . . .	62
Hình 5.5	Giao diện đơn khi đã chuyển hàng . . . . .	62
Hình 5.6	Giao diện đơn khi đã hủy . . . . .	63
Hình 5.7	Giao diện quản lý chuyển hàng . . . . .	63
Hình 5.8	Giao diện chi tiết tồn kho điểm bán . . . . .	64
Hình 5.9	Giao diện tạo phiếu kiểm kho . . . . .	65
Hình 5.10	Bản mẫu kế hoạch bán hàng . . . . .	66
Hình 5.11	Giao diện điều chỉnh kế hoạch bán hàng . . . . .	66
Hình 5.12	Giao diện tạo kế hoạch bán hàng . . . . .	67
Hình 5.13	Giao diện theo dõi kế hoạch bán hàng . . . . .	67
Hình 5.14	Giao diện chi tiết kế hoạch bán hàng . . . . .	68

## DANH MỤC BẢNG BIỂU

Bảng 2.1	Đặc tả use case Xử lý giao dịch bán hàng . . . . .	16
Bảng 2.2	Đặc tả use case Quản lý luân chuyển hàng hóa . . . . .	17
Bảng 2.3	Đặc tả use case Quản lý thông tin sản phẩm . . . . .	18
Bảng 2.4	Đặc tả use case Quản lý tồn kho điểm bán . . . . .	19
Bảng 2.5	Đặc tả use case Quản lý thông tin điểm bán . . . . .	20
Bảng 2.6	Đặc tả use case Quản lý nhân viên . . . . .	21
Bảng 2.7	Đặc tả use case Quản lý kế hoạch điểm bán . . . . .	22
Bảng 4.1	Thông tin thiết kế giao diện . . . . .	36
Bảng 4.2	Các phương thức của lớp TransferController . . . . .	40
Bảng 4.3	Các phương thức của lớp TransferService . . . . .	41
Bảng 4.4	Các phương thức của lớp ProductController . . . . .	42
Bảng 4.5	Các phương thức của lớp ProductService . . . . .	43
Bảng 4.6	Danh sách thuộc tính của bảng <b>shop</b> . . . . .	46
Bảng 4.7	Danh sách thuộc tính của bảng <b>user</b> . . . . .	47
Bảng 4.8	Danh sách thuộc tính của bảng <b>work_schedule</b> . . . . .	47
Bảng 4.9	Danh sách thuộc tính của bảng <b>variant</b> . . . . .	48
Bảng 4.10	Danh sách thuộc tính của bảng <b>order</b> . . . . .	49
Bảng 4.11	Danh sách thuộc tính của bảng <b>import_order</b> . . . . .	49
Bảng 4.12	Danh sách thuộc tính của bảng <b>transfer</b> . . . . .	50
Bảng 4.13	Danh sách thuộc tính của bảng <b>inventory</b> . . . . .	50
Bảng 4.14	Danh sách thư viện và công cụ sử dụng . . . . .	51
Bảng 4.15	Thống kê thông tin hệ thống . . . . .	52
Bảng 4.16	Kết quả kiểm thử chức năng bán hàng . . . . .	56
Bảng 4.17	Kết quả kiểm thử chức năng luân chuyển hàng hóa . . . . .	57
Bảng 4.18	Thông số cấu hình . . . . .	58
Bảng 5.1	Mô tả các trạng thái chuyển hàng và quy tắc chuyển đổi . . . . .	61
Bảng A.1	Danh sách thuộc tính của bảng <b>balance_variant</b> . . . . .	73
Bảng A.2	Danh sách thuộc tính của bảng <b>base_product</b> . . . . .	73
Bảng A.3	Danh sách thuộc tính của bảng <b>customer</b> . . . . .	74
Bảng A.4	Danh sách thuộc tính của bảng <b>customer_group</b> . . . . .	74
Bảng A.5	Danh sách thuộc tính của bảng <b>feedback</b> . . . . .	75
Bảng A.6	Danh sách thuộc tính của bảng <b>import_item</b> . . . . .	75
Bảng A.7	Danh sách thuộc tính của bảng <b>order_line</b> . . . . .	76

Bảng A.8	Danh sách thuộc tính của bảng <b>return_order</b>	76
Bảng A.9	Danh sách thuộc tính của bảng <b>return_order_line</b>	77
Bảng A.10	Danh sách thuộc tính của bảng <b>payment</b>	77
Bảng A.11	Danh sách thuộc tính của bảng <b>role</b>	78
Bảng A.12	Danh sách thuộc tính của bảng <b>user_role</b>	78
Bảng A.13	Danh sách thuộc tính của bảng <b>user_shop_access</b>	78
Bảng A.14	Danh sách thuộc tính của bảng <b>shift</b>	79
Bảng A.15	Danh sách thuộc tính của bảng <b>vendor</b>	79
Bảng A.16	Danh sách thuộc tính của bảng <b>transfer_item</b>	80

## DANH MỤC THUẬT NGỮ VÀ TỪ VIẾT TẮT

Thuật ngữ	Ý nghĩa
API	Giao diện lập trình ứng dụng (Application Programming Interface)
DOM	Mô hình đối tượng tài liệu (Document Object Model)
HTML	Ngôn ngữ đánh dấu siêu văn bản (HyperText Markup Language)
HTTP	Giao thức truyền tải siêu văn bản (HyperText Transfer Protocol)
JSON	Định dạng dữ liệu dựa trên JavaScript(JavaScript Object Notation)
POS	Điểm bán hàng (Point of Sale)
RDBMS	Hệ quản trị cơ sở dữ liệu quan hệ (Relational Database Management System)
SQL	Ngôn ngữ truy vấn mạng tính cấu trúc (Structured Query Language)
XML	Ngôn ngữ đánh dấu mở rộng (Extensible Markup Language)

# CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

## 1.1 Đặt vấn đề

Trong thời đại công nghệ số và toàn cầu hóa, ngành bán lẻ đang trải qua những thay đổi sâu sắc. Từ những cửa hàng bán lẻ truyền thống với quy trình bán hàng và giao dịch trực tiếp, các doanh nghiệp hiện nay có nhiều cách tiếp cận khách hàng hơn như bán hàng trực tuyến, qua các sàn thương mại điện tử và đặc biệt là mô hình bán lẻ lưu động. Mô hình bán lẻ lưu động, trong đó doanh nghiệp không chỉ kinh doanh tại một cửa hàng cố định mà còn có thể bán hàng tại các địa điểm khác nhau, đã dần trở thành một xu hướng phổ biến.

Mô hình này mang lại nhiều lợi ích to lớn như tăng cường tính linh hoạt và mở rộng phạm vi thị trường mà không bị giới hạn bởi vị trí địa lý. Đồng thời, nó tối ưu hóa chi phí vận hành so với việc duy trì nhiều cửa hàng cố định. Đối với các doanh nghiệp vừa và nhỏ, bán lẻ lưu động giúp họ tiếp cận lượng khách hàng lớn hơn mà không phải đầu tư nhiều vào cơ sở hạ tầng. Điển hình là các doanh nghiệp có thể tổ chức các điểm bán lưu động tại các khu vực tập trung đông người như chợ phiên, hội chợ, sự kiện đường phố, giúp nhanh chóng tiếp cận khách hàng và tăng trưởng doanh số bán hàng.

Tuy nhiên, cùng với những lợi ích đó, mô hình bán lẻ lưu động cũng đặt ra nhiều thách thức. Một trong những thách thức lớn nhất là làm thế nào để quản lý hiệu quả toàn bộ quy trình vận hành, từ việc quản lý kho hàng, theo dõi bán hàng, kiểm soát dòng tiền, cho đến việc sắp xếp lịch làm việc cho nhân viên tại nhiều địa điểm khác nhau. Khi quy trình vận hành không được kiểm soát tốt, doanh nghiệp dễ gặp phải tình trạng thất thoát hàng tồn kho, không theo dõi được chính xác doanh số, và thiếu công cụ phân tích hiệu quả để đưa ra quyết định kinh doanh.

Để giải quyết những thách thức này, hệ thống quản lý bán hàng POS (Point of Sale) đã được phát triển và áp dụng rộng rãi. Tuy nhiên, các hệ thống POS hiện tại chủ yếu được thiết kế cho các cửa hàng cố định, nơi tất cả hoạt động bán hàng và quản lý diễn ra tại một địa điểm duy nhất. Đối với mô hình bán lẻ lưu động, việc áp dụng POS cần có sự linh hoạt và nhiều cải tiến hơn, bao gồm khả năng quản lý nhiều điểm bán lưu động, quản lý kho hàng lưu động, chuyển giao hàng hóa và phân ca làm việc cho từng điểm bán.

Xuất phát từ nhu cầu này, việc phát triển một hệ thống quản lý bán hàng POS chuyên biệt cho mô hình bán lẻ lưu động trở nên cấp thiết. Hệ thống này không chỉ đảm bảo các chức năng quản lý cơ bản, mà còn phải cung cấp các tính năng nâng

cao như quản lý kho hàng lưu động, lập kế hoạch bán hàng lưu động, và tối ưu hóa quy trình vận hành của doanh nghiệp.

## **1.2 Mục tiêu và phạm vi đề tài**

Mục tiêu của đề tài là phát triển một hệ thống quản lý bán hàng POS, hỗ trợ doanh nghiệp quản lý hiệu quả các điểm bán cố định và lưu động. Hệ thống này sẽ giúp doanh nghiệp tối ưu hóa quy trình bán hàng, quản lý tồn kho, và theo dõi tình hình kinh doanh tại các điểm bán. Cụ thể, đề tài hướng đến việc xây dựng một hệ thống POS bao gồm các tính năng như bán hàng, quản lý tồn kho, quản lý nhân sự và báo cáo tình hình kinh doanh. Tuy nhiên, điều khác biệt là hệ thống sẽ tích hợp các tính năng đặc thù cho mô hình bán lẻ lưu động, bao gồm khả năng điều phối hàng hóa giữa kho cố định và các điểm bán lưu động, theo dõi tình trạng tồn kho tại các điểm bán khác nhau.

Một mục tiêu khác là đảm bảo việc quản lý tồn kho linh hoạt và hiệu quả, đặc biệt trong bối cảnh hàng hóa được lưu trữ và phân bổ tại nhiều địa điểm. Hệ thống cần cho phép theo dõi sự dịch chuyển của hàng hóa từ kho chính đến các điểm bán lưu động, cũng như giữa các điểm bán với nhau. Điều này giúp doanh nghiệp kiểm soát tốt hơn tình trạng tồn kho, tránh thiếu hụt hoặc tồn đọng hàng hóa không cần thiết.

Bên cạnh đó, hệ thống cũng sẽ tích hợp các công cụ quản lý nhân sự và lịch làm việc. Đối với mô hình bán lẻ lưu động, việc quản lý nhân sự và điều phối lịch làm việc là một bài toán quan trọng. Hệ thống cần hỗ trợ doanh nghiệp lập kế hoạch làm việc cho nhân viên tại các điểm bán khác nhau một cách hợp lý và hiệu quả.

Phạm vi của đề tài tập trung vào việc phát triển hệ thống POS cho một doanh nghiệp bán lẻ quy mô vừa, có một cửa hàng cố định và nhiều điểm bán lưu động. Các tính năng chính bao gồm quản lý bán hàng và thanh toán, quản lý tồn kho, quản lý nhân sự và lịch làm việc, cũng như cung cấp báo cáo giúp doanh nghiệp ra quyết định kinh doanh chính xác.

## **1.3 Định hướng giải pháp**

Để xây dựng hệ thống quản lý bán hàng POS cho mô hình bán lẻ lưu động, giải pháp cần không chỉ đáp ứng được các yêu cầu quản lý thông thường, mà còn phải đảm bảo tính linh hoạt, khả năng mở rộng và tích hợp chặt chẽ với hoạt động tại các điểm bán lưu động. Hệ thống sẽ được phát triển dựa trên kiến trúc ba lớp (Three-Tier Architecture) kết hợp với mô hình MVC (Model-View-Controller), nhằm đảm bảo sự phân chia rõ ràng giữa giao diện người dùng, logic xử lý và tầng cơ sở dữ liệu. Kiến trúc này mang lại nhiều lợi ích, bao gồm tính linh hoạt và khả năng mở rộng, dễ bảo trì và phát triển, đồng thời đảm bảo tính tương tác cao.

Giao diện người dùng của hệ thống sẽ được phát triển bằng React.js kết hợp với Material UI, đảm bảo tính thẩm mỹ, nhất quán và hiệu quả trong trải nghiệm người dùng. Phần backend sẽ được phát triển bằng ngôn ngữ Java, với khả năng xử lý mạnh mẽ và bảo mật cao. Hệ thống cơ sở dữ liệu sẽ sử dụng MySQL, cho phép quản lý và đồng bộ dữ liệu giữa các điểm bán lưu động và kho hàng chính.

Bên cạnh các tính năng quản lý bán hàng, thanh toán và tồn kho cơ bản, hệ thống sẽ hỗ trợ quản lý kho hàng lưu động, quản lý nhân sự và lịch làm việc, cũng như cung cấp báo cáo về hiệu suất kinh doanh giúp tối ưu hóa quy trình quản lý và vận hành.

### 1.4 Bố cục đề án

Nội dung của đề án được tổ chức thành các chương như sau

Chương 1, nội dung chính sẽ bao gồm việc trình bày lý do lựa chọn đề tài cùng với mục tiêu, phạm vi nghiên cứu và định hướng giải pháp của hệ thống quản lý bán hàng POS cho mô hình bán lẻ lưu động. Đề tài này xuất phát từ nhu cầu ngày càng tăng của các doanh nghiệp trong việc ứng dụng các hệ thống quản lý linh hoạt và hiện đại để cải thiện quy trình vận hành.

Tiếp theo, chương 2 sẽ tập trung vào việc khảo sát hiện trạng của vấn đề và đưa ra các chức năng cần thiết để giải quyết vấn đề đó. Phần này sẽ cung cấp một cái nhìn tổng quan về các use case, mô tả về vai trò của từng tác nhân trong hệ thống và tập trung vào các tính năng chính cần được phát triển.

Sau khi nắm vững cơ sở lý thuyết, chương 3 sẽ tập trung giới thiệu về các công nghệ và thư viện cụ thể đã được sử dụng trong đề án, đồng thời đánh giá ưu nhược điểm của từng công nghệ khi áp dụng vào đề án.

Trong chương 4, em sẽ trình bày chi tiết về quá trình thiết kế, xây dựng, kiểm thử, triển khai và kết quả đạt được của đề án. Thiết kế kiến trúc và chi tiết: em sẽ giới thiệu về kiến trúc tổng quan của hệ thống, bao gồm thiết kế chi tiết về giao diện người dùng, cơ sở dữ liệu và các lớp. Quá trình xây dựng và kiểm thử: phần này sẽ trình bày về các bước thực hiện để đảm bảo chất lượng của sản phẩm.

Tiếp đó, chương 5 sẽ trình bày các đóng góp nổi bật và những khó khăn đã giải quyết được trong quá trình làm đề án tốt nghiệp.

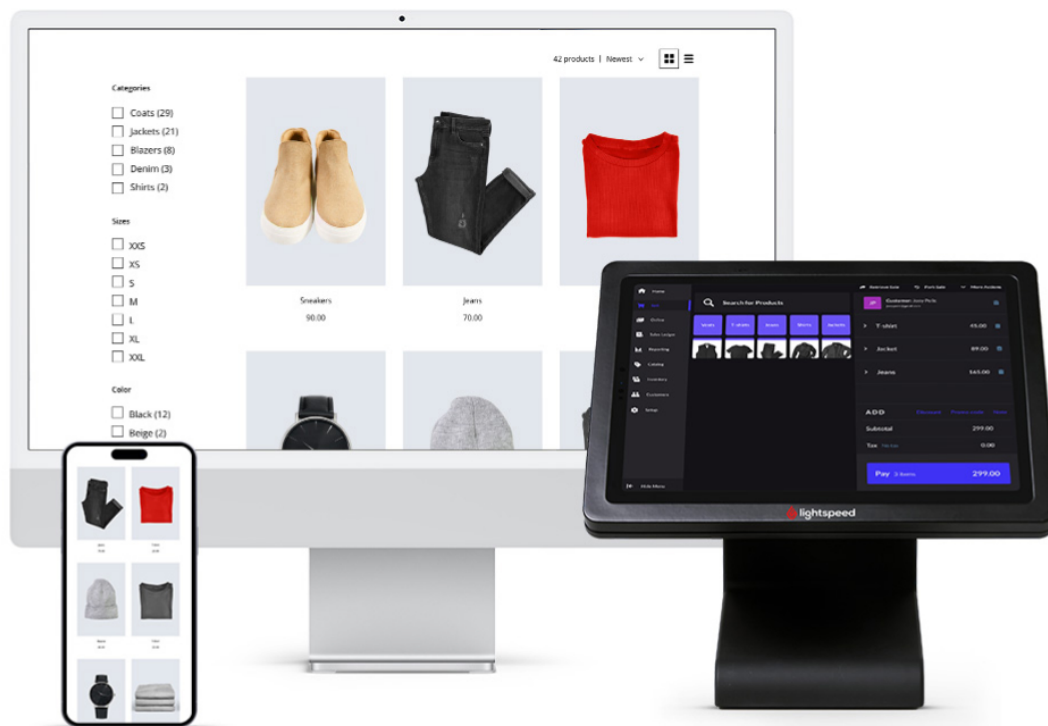
Cuối cùng, chương 6 sẽ tổng kết toàn bộ nội dung đã thực hiện trong đề án. Đồng thời, chương này cũng sẽ đánh giá tổng quan về hiệu quả của hệ thống và đưa ra các đề xuất phát triển trong tương lai nhằm hoàn thiện hơn hệ thống quản lý bán hàng POS cho mô hình bán lẻ lưu động.



## CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU

### 2.1 Khảo sát hiện trạng

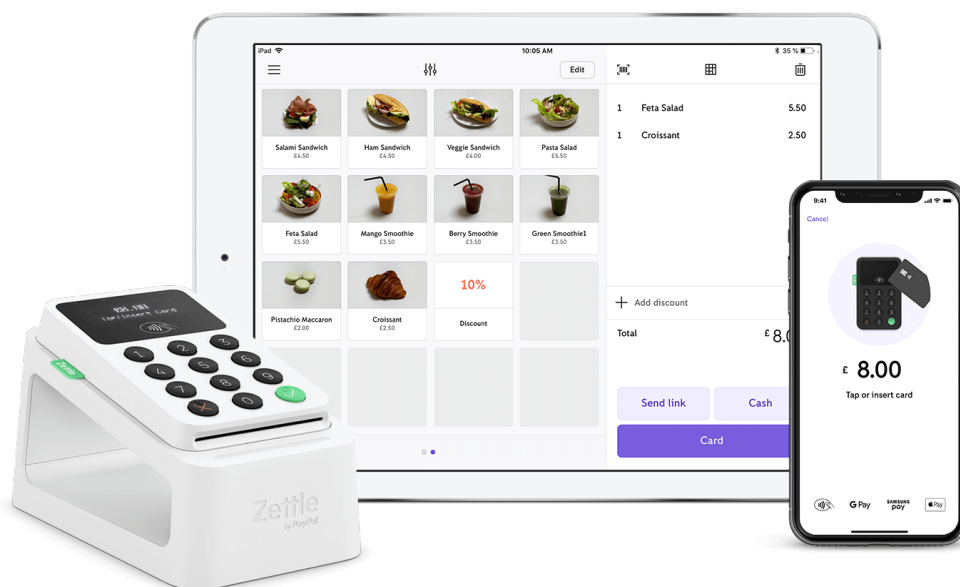
Trong bối cảnh chuyển đổi số đang diễn ra mạnh mẽ, các doanh nghiệp bán lẻ buộc phải tìm kiếm những giải pháp sáng tạo để nâng cao hiệu quả hoạt động và khả năng cạnh tranh trên thị trường. Một trong những xu hướng được nhiều doanh nghiệp nhỏ và vừa quan tâm là mô hình bán lẻ lưu động. Điểm đặc biệt của mô hình này nằm ở khả năng tiếp cận linh hoạt đến khách hàng tại các địa điểm đa dạng như chợ phiên, hội chợ, sự kiện ngoài trời, hay thậm chí là các khu vực dân cư xa trung tâm. Không giống như các cửa hàng cố định, mô hình bán lẻ lưu động yêu cầu một hệ thống quản lý phù hợp với các yếu tố đặc thù như di chuyển thường xuyên, quản lý tồn kho linh hoạt và điều phối nhân sự hiệu quả. Dù thị trường đã xuất hiện một số giải pháp như Lightspeed và Zettle, các hệ thống này vẫn chưa thực sự đáp ứng đầy đủ nhu cầu phức tạp của mô hình bán lẻ lưu động.



Hình 2.1: Lightspeed [1]

Lightspeed là một trong những nền tảng POS được biết đến rộng rãi nhờ khả năng quản lý đa điểm bán hàng một cách tập trung và hiệu quả. Điểm mạnh của Lightspeed nằm ở việc cho phép doanh nghiệp theo dõi tình trạng hàng tồn kho và doanh số bán hàng tại từng địa điểm, đồng thời cung cấp các báo cáo chi tiết về

hiệu quả kinh doanh. Hệ thống này hỗ trợ đồng bộ hóa dữ liệu giữa các điểm bán hàng và trung tâm quản lý, giúp doanh nghiệp tiết kiệm thời gian và giảm thiểu sai sót trong việc ghi nhận giao dịch. Tuy nhiên, Lightspeed vẫn còn nhiều hạn chế khi áp dụng cho mô hình bán lẻ lưu động. Các tính năng hiện có của Lightspeed chủ yếu được thiết kế để phục vụ các cửa hàng cố định, không tập trung vào việc hỗ trợ các điểm bán lưu động. Ví dụ, Lightspeed chưa tích hợp các công cụ hỗ trợ lập kế hoạch, quản lý nhân sự làm việc tại các điểm bán tạm thời, một yếu tố quan trọng đối với mô hình bán lẻ lưu động.



**Hình 2.2:** Zettle [2]

Trong khi đó, Zettle lại được thiết kế như một hệ thống POS lưu động đơn giản và dễ sử dụng, phù hợp với các doanh nghiệp nhỏ hoặc cá nhân kinh doanh cần sự linh hoạt trong xử lý thanh toán tại chỗ. Điểm mạnh nổi bật của Zettle là khả năng thiết lập điểm bán nhanh chóng tại bất kỳ địa điểm nào và hỗ trợ các phương thức thanh toán đa dạng, bao gồm thẻ tín dụng và ví điện tử. Đây là một giải pháp hiệu quả cho các nhà bán lẻ nhỏ, những người không yêu cầu các tính năng quản lý phức tạp. Tuy nhiên, Zettle cũng tồn tại những hạn chế lớn khi áp dụng cho mô hình bán lẻ lưu động có quy mô lớn. Hệ thống này thiếu các tính năng quản lý kho hàng toàn diện, không hỗ trợ theo dõi trạng thái tồn kho tại các điểm bán lưu động hoặc luân chuyển hàng hóa giữa các địa điểm. Ngoài ra, Zettle chưa có công cụ hỗ trợ lập kế hoạch và điều phối nhân sự, điều này gây khó khăn cho các doanh nghiệp cần quản lý đội ngũ nhân viên làm việc tại nhiều điểm bán khác nhau.

Từ những phân tích trên, có thể thấy rằng mặc dù các hệ thống POS hiện có

đã có những bước tiến đáng kể, nhưng vẫn còn tồn tại những khoảng trống khi áp dụng cho mô hình bán lẻ lưu động. Cả Lightspeed và Zettle đều chưa cung cấp giải pháp toàn diện để quản lý hiệu quả các điểm bán hàng lưu động. Sự thiếu vắng các tính năng để quản lý kho hàng linh hoạt giữa các địa điểm và công cụ lập kế hoạch nhân sự khiến các hệ thống này khó có thể đáp ứng đầy đủ nhu cầu của mô hình kinh doanh này. Điều này đặt ra một yêu cầu cấp thiết đối với việc phát triển một hệ thống POS mới, được thiết kế riêng cho mô hình bán lẻ lưu động với các tính năng tối ưu hóa toàn diện.

Một hệ thống POS mới, chuyên biệt cho mô hình bán lẻ lưu động, cần tập trung vào việc giải quyết các hạn chế nêu trên. Trước hết, hệ thống này cần tích hợp các công cụ hỗ trợ quản lý nhân sự làm việc tại các điểm bán lưu động. Điều này bao gồm lập lịch làm việc và phân bổ nguồn lực hợp lý theo nhu cầu của từng điểm bán. Bên cạnh đó, hệ thống cần cung cấp công cụ quản lý kho hàng chi tiết, cho phép theo dõi số lượng hàng hóa xuất nhập tại từng điểm bán và hỗ trợ cảnh báo khi hàng tồn kho đạt ngưỡng tối thiểu. Tính năng luân chuyển hàng hóa giữa kho trung tâm và các điểm bán lưu động cũng là một yếu tố không thể thiếu để đảm bảo nguồn cung ổn định.

Ngoài ra, hệ thống hỗ trợ quản lý trạng thái của các điểm bán lưu động. Tính năng này không chỉ giúp doanh nghiệp kiểm soát hiệu quả các hoạt động tại điểm bán mà còn hỗ trợ việc lập kế hoạch phù hợp với các sự kiện hoặc khu vực có lượng khách hàng tiềm năng cao. Đồng thời, việc cung cấp các báo cáo chi tiết về doanh số, hiệu quả kinh doanh và xu hướng mua sắm sẽ giúp doanh nghiệp có cái nhìn toàn diện để đưa ra các quyết định chiến lược. Cuối cùng, hệ thống cần đảm bảo hỗ trợ đa phương thức thanh toán như tiền mặt và thanh toán qua mã QR, nhằm đáp ứng sự đa dạng trong hành vi tiêu dùng của khách hàng hiện đại.

Nhìn chung, việc phát triển một hệ thống POS mới với các tính năng tập trung vào quản lý toàn diện cho mô hình bán lẻ lưu động không chỉ giúp doanh nghiệp tối ưu hóa vận hành mà còn tạo ra nền tảng bền vững cho sự phát triển trong tương lai. Đây chính là cơ hội để các doanh nghiệp nhỏ và vừa gia tăng sức cạnh tranh, đồng thời tận dụng tối đa các lợi ích mà công nghệ số mang lại.

## **2.2 Tổng quan chức năng**

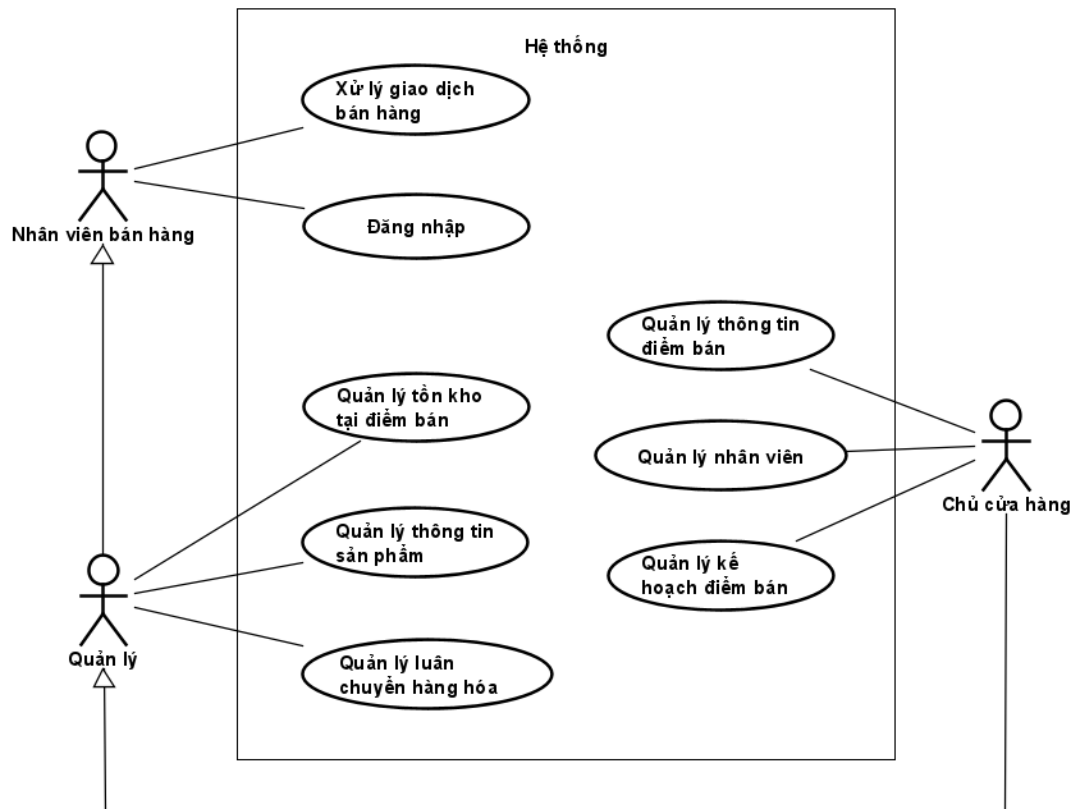
Hệ thống bao gồm 3 tác nhân chính là Nhân viên bán hàng, Quản lý và Chủ cửa hàng.

Đối với Nhân viên bán hàng, mỗi tài khoản người dùng chịu trách nhiệm xử lý các giao dịch bán hàng và tương tác với khách hàng.

Đối với Quản lý, mỗi tài khoản người dùng thừa kế các chức năng của người bán và có thêm vai trò là quản lý tồn kho và thông tin sản phẩm, cùng với việc điều phối luân chuyển hàng hóa.

Tài khoản của Chủ cửa hàng thừa kế các chức năng của quản lý và chịu trách nhiệm thêm về việc quản lý thông tin các điểm bán và nhân viên, cũng như quản lý kế hoạch làm việc của các điểm bán.

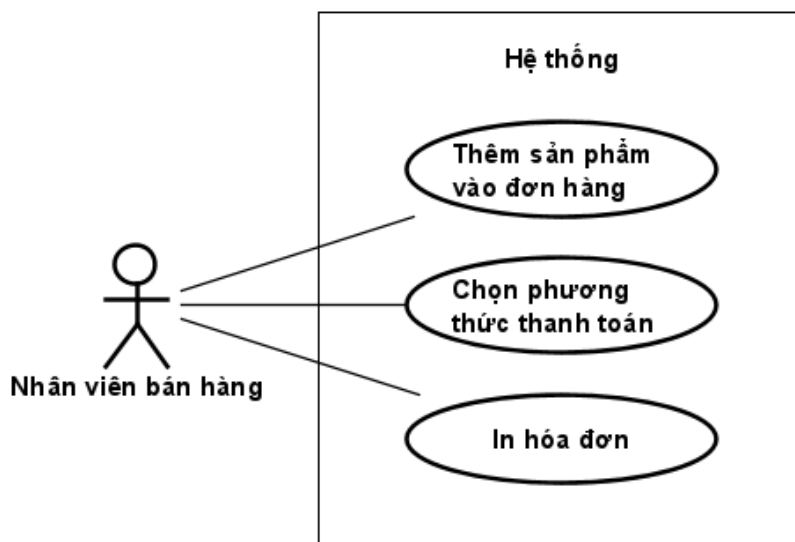
### 2.2.1 Biểu đồ use case tổng quát



**Hình 2.3:** Biểu đồ use case tổng quan

Trên hình là danh sách các usecase tổng quan của hệ thống, tiếp theo em sẽ phân rã chi tiết các usecase này để thấy được chi tiết hơn về các chức năng của hệ thống.

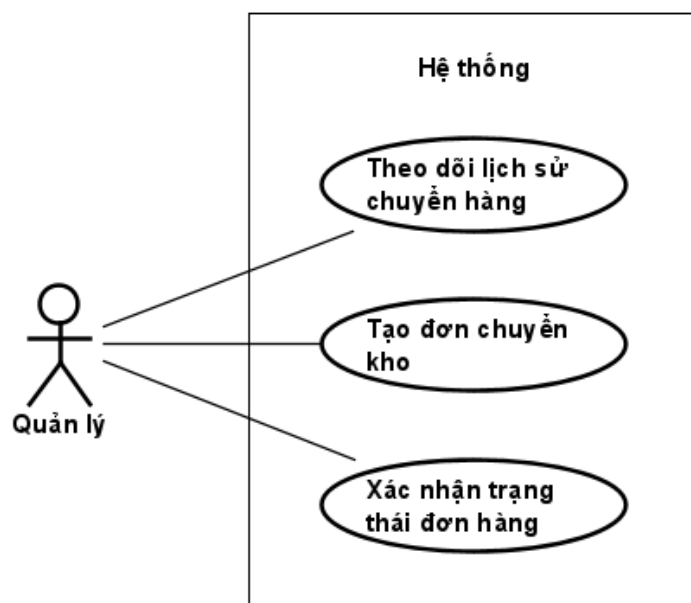
### 2.2.2 Biểu đồ use case phân rã Xử lý giao dịch bán hàng



**Hình 2.4:** Biểu đồ use case phân rã Xử lý giao dịch bán hàng

Biểu đồ use case phân rã Xử lý giao dịch bán hàng liên quan đến tác nhân Nhân viên bán hàng có các chức năng thêm sản phẩm vào đơn hàng, chọn phương thức thanh toán và in hóa đơn. Nhân viên bán hàng có thể thêm sản phẩm vào đơn bằng cách dùng thanh tìm kiếm theo tên hoặc mã sản phẩm, có thể tăng giảm số lượng sản phẩm hoặc loại bỏ sản phẩm khỏi đơn khi cần thiết. Ngoài ra, nhân viên bán hàng có thể thay đổi phương thức thanh toán mặc định là tiền mặt sang chuyển khoản. Phần mềm sẽ tạo mã QR để người mua hàng có thể thanh toán tiện lợi hơn. Nhân viên bán hàng có thể chọn in hóa đơn sau khi thanh toán hoàn tất. Hệ thống cũng hỗ trợ tính toán tiền thừa nếu khách thanh toán bằng tiền mặt. Quy trình này tối ưu việc giao dịch thanh toán tại điểm bán hàng lưu động, cung cấp sự thuận tiện trong giao dịch và tránh sai sót phát sinh.

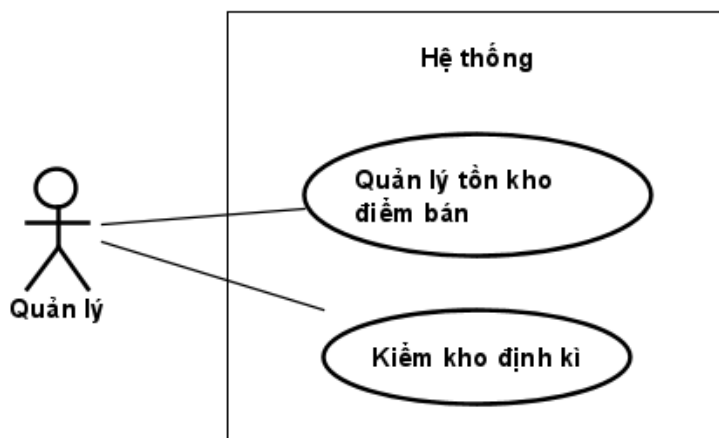
### 2.2.3 Biểu đồ use case phân rã Quản lý luân chuyển hàng hóa



**Hình 2.5:** Biểu đồ use case phân rã Quản lý luân chuyển hàng hóa

Biểu đồ use case phân rã Quản lý luân chuyển hàng hóa của Quản lý liên quan đến việc theo dõi và điều phối quá trình vận chuyển hàng từ nơi này đến nơi khác. Mỗi điểm bán có tối thiểu một quản lý đảm nhiệm việc theo dõi lịch sử, trạng thái các đơn vận chuyển, có thể tạo mới đơn chuyển kho và xác nhận, thay đổi trạng thái của từng đơn hàng. Quản lý có thể chuyển đổi giữa các trạng thái "Đang tạo", "Đang xuất hàng", "Đã chuyển hàng" và "Đã hủy". Quy trình này đảm bảo hàng hóa được vận chuyển đúng thời gian và địa điểm, giảm thiểu rủi ro và tăng hiệu quả hoạt động. Ngoài ra, quá trình này cũng giúp tối ưu việc cung ứng hàng hóa cho các điểm bán, hàng hóa sẽ được bổ sung nhanh hơn từ các điểm bán lân cận thay vì phải vận chuyển từ điểm bán chính.

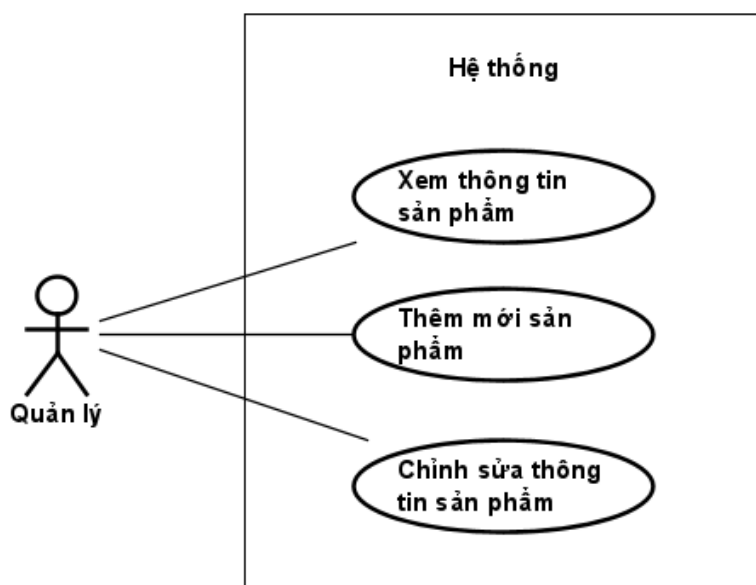
#### 2.2.4 Biểu đồ use case phân rã Quản lý tồn kho tại điểm bán



**Hình 2.6:** Biểu đồ use case phân rã Quản lý tồn kho tại điểm bán

Biểu đồ use case phân rã Quản lý tồn kho tại điểm bán của Quản lý bao gồm quản lý tồn kho điểm bán và kiểm kho định kì. Quản lý thực hiện các hoạt động theo dõi, kiểm đếm và cập nhật thông tin tồn kho, như ghi nhận các sản phẩm nhập và xuất kho, kiểm tra số lượng sản phẩm hiện có và thực hiện cân bằng kho định kì. Nếu có chênh lệch so với thực tế, sau khi thực hiện cân bằng kho, hệ thống sẽ cập nhật lại tồn kho theo số liệu thực tế được ghi nhận. Mục tiêu của use case này là đảm bảo mức tồn kho luôn ổn định và sẵn sàng đáp ứng nhu cầu của khách hàng.

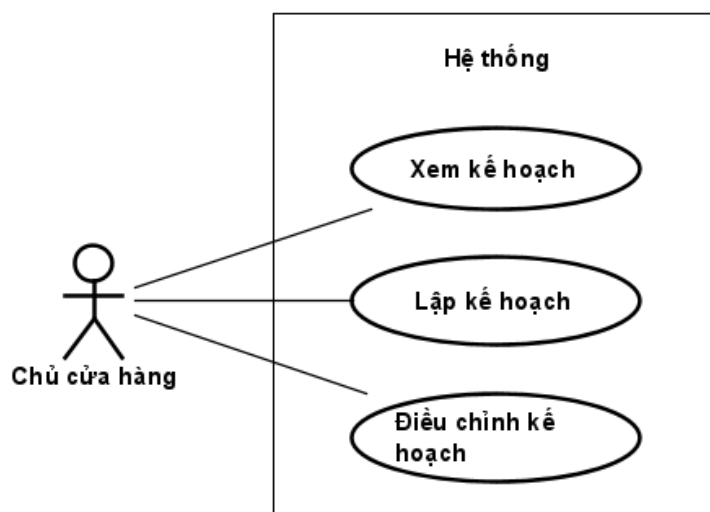
#### 2.2.5 Biểu đồ use case phân rã Quản lý thông tin hàng hóa



**Hình 2.7:** Biểu đồ use case phân rã Quản lý thông tin hàng hóa

Biểu đồ usecase phân rã Quản lý thông tin hàng hóa của Quản lý điểm bán có các chức năng chính bao gồm xem, thêm, chỉnh sửa sản phẩm. Quản lý có thể thêm sản phẩm mới bằng cách nhập thông tin như tên, nhãn hiệu, giá nhập, giá bán và mã sản phẩm, mã này là duy nhất và sẽ tự sinh nếu không được nhập thủ công. Với sản phẩm có nhiều biến thể, quản lý có thể thêm các biến thể mới với các thuộc tính về màu sắc, kích cỡ, chất liệu và tải lên ảnh sản phẩm liên quan. Ngoài ra, quản lý có thể sửa đổi các thông tin sản phẩm khi cần thiết. Quá trình này giúp đảm bảo thông tin sản phẩm luôn được cập nhật và chính xác trong hệ thống.

### 2.2.6 Biểu đồ use case phân rã Quản lý kế hoạch điểm bán

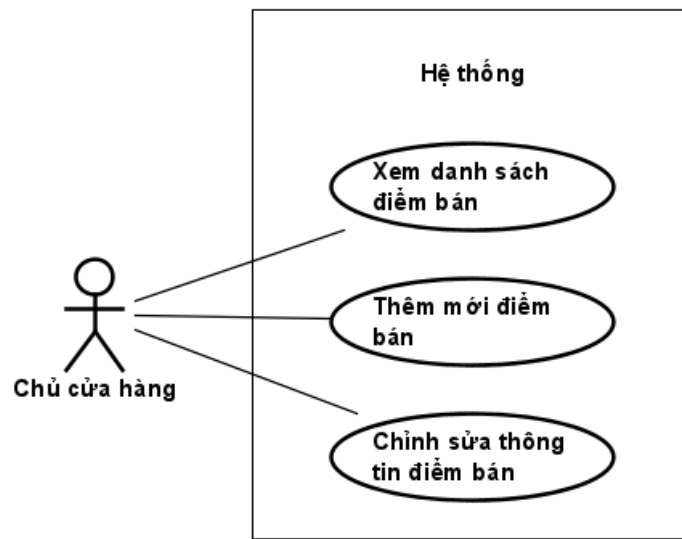


**Hình 2.8:** Biểu đồ use case phân rã Quản lý kế hoạch điểm bán

Use case phân rã Quản lý kế hoạch điểm bán mô tả các chức năng mà chủ cửa hàng có thể thực hiện bao gồm: xem kế hoạch, lập kế hoạch, điều chỉnh kế hoạch bán hàng. Chủ cửa hàng có thể quản lý về địa điểm, thời gian và phân bổ ca làm việc cho mỗi nhân viên tại các điểm bán tùy theo chiến lược kinh doanh. Chủ cửa hàng sẽ tải về bản mẫu kế hoạch bán hàng, chỉnh sửa thời gian và ca làm của mỗi nhân viên theo kế hoạch thực tế, sau đó tải lên hệ thống. Quy trình này hỗ trợ việc thiết kế và quản lý kế hoạch, ca làm việc của nhân viên tại các điểm bán hàng thuận tiện và dễ dàng hơn.



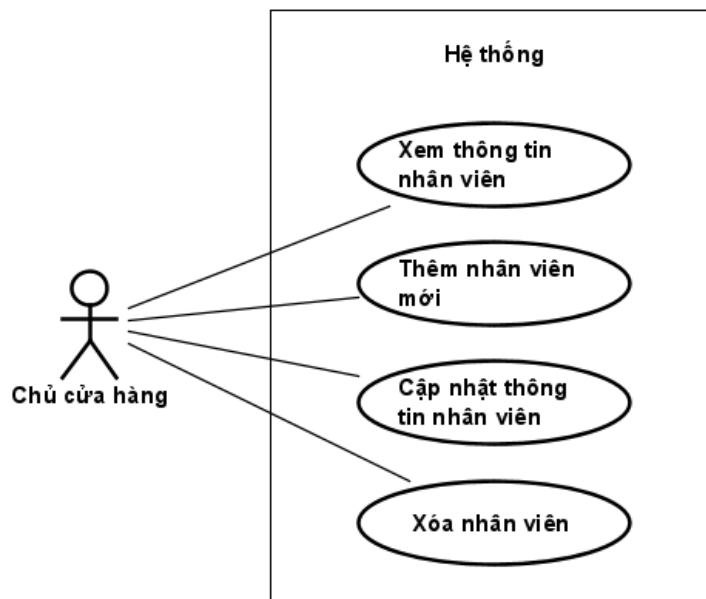
### 2.2.7 Biểu đồ use case phân rã Quản lý thông tin điểm bán



**Hình 2.9:** Biểu đồ use case phân rã Quản lý thông tin điểm bán

Biểu đồ phân rã use case trên mô tả các chức năng Quản lý thông tin điểm bán của chủ cửa hàng. Chủ cửa hàng có thể xem danh sách điểm bán, thêm mới điểm bán và chỉnh sửa thông tin điểm bán. Các thông tin liên quan tới điểm bán bao gồm tên cửa hàng, số điện thoại, email, địa chỉ và lĩnh vực sản phẩm kinh doanh.

### 2.2.8 Biểu đồ use case phân rã Quản lý nhân viên

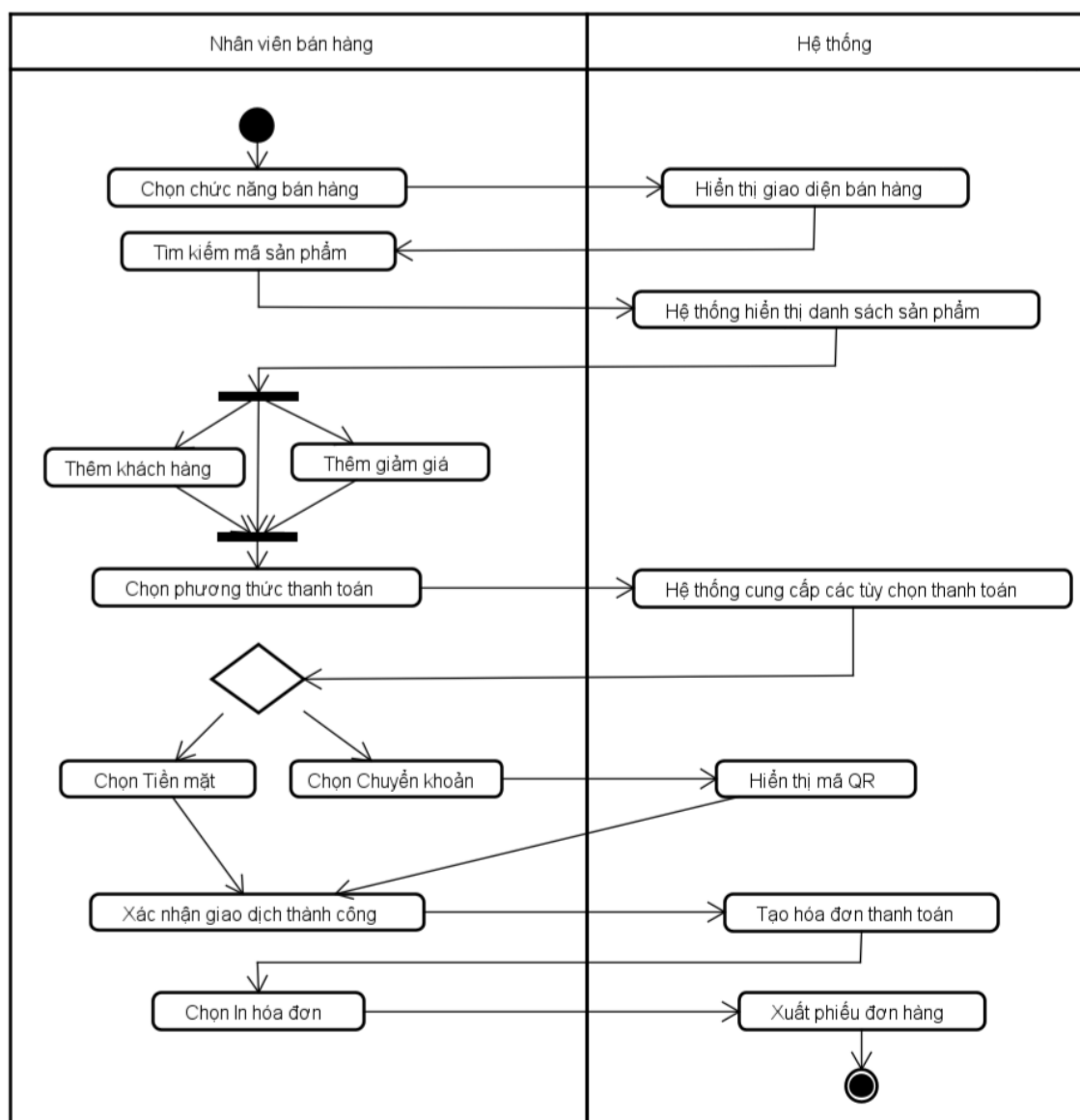


**Hình 2.10:** Biểu đồ use case phân rã Quản lý nhân viên

Biểu đồ phân rã use case trên mô tả các chức năng Quản lý nhân viên của chủ cửa hàng. Chủ cửa hàng có thể xem danh sách nhân viên, thêm mới nhân viên và chỉnh sửa thông tin nhân viên. Các thông tin liên quan tới nhân viên bao gồm tên, ảnh, vai trò, thông tin liên lạc, thông tin tài khoản. Các tài khoản hoạt động sẽ hiển thị đang hoạt động còn các nhân viên không còn làm việc sẽ bị chủ cửa hàng vô hiệu hóa tài khoản chuyển sang trạng thái đã thôi việc.

### 2.2.9 Quy trình nghiệp vụ

#### a, Biểu đồ quy trình bán hàng



Hình 2.11: Biểu đồ quy trình bán hàng

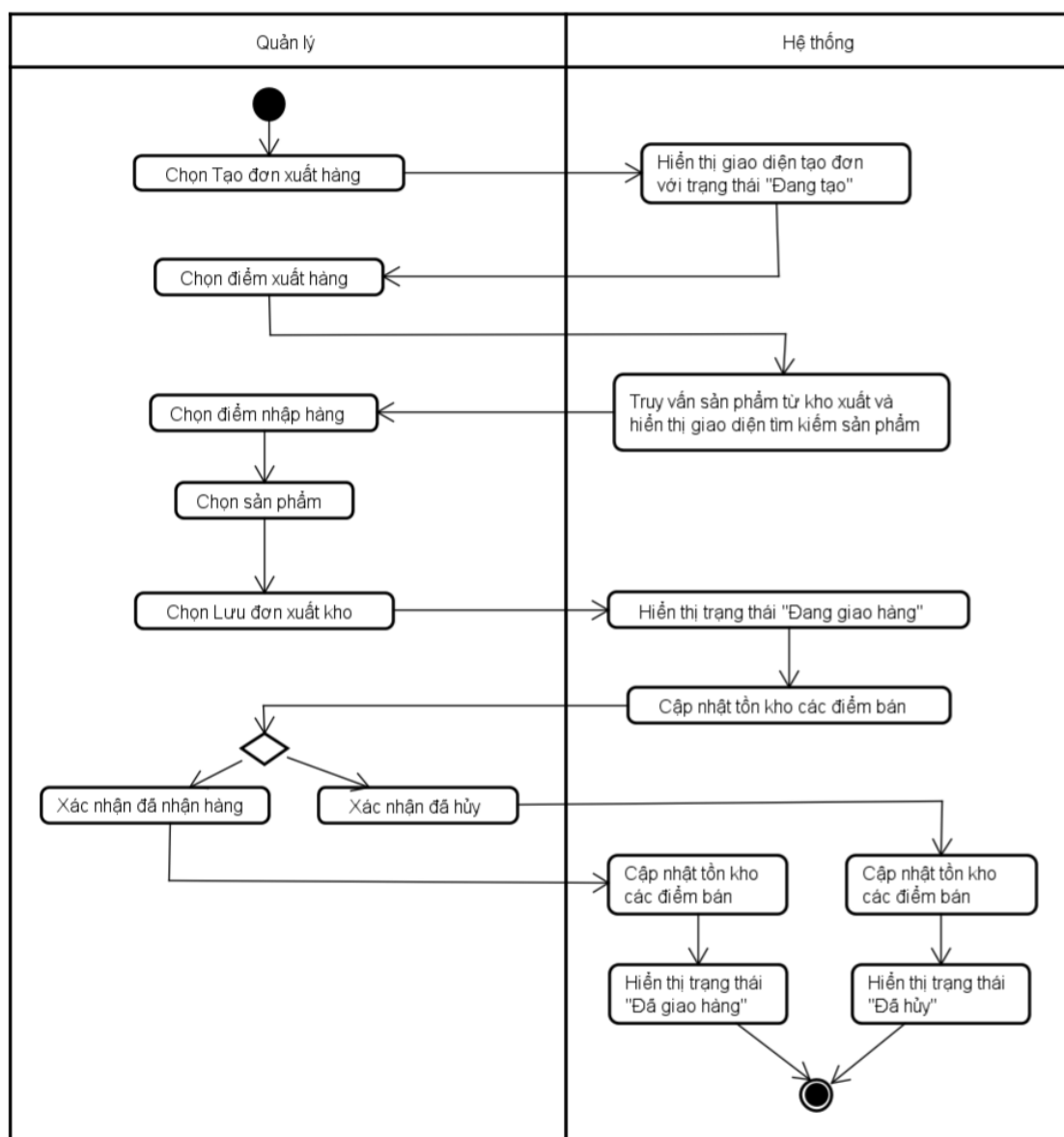
Để bắt đầu một giao dịch bán hàng, nhân viên bán hàng chọn chức năng bán hàng, hệ thống sẽ hiển thị giao diện bán hàng. Nhân viên bán hàng sẽ tìm kiếm, chọn sản phẩm để thêm sản phẩm vào đơn hàng. Tại đây, hệ thống sẽ hiển thị danh

sách sản phẩm đã chọn, giá tiền, số lượng mỗi sản phẩm và tổng giá trị đơn hàng.

Tiếp theo, nhân viên bán hàng kiểm tra thông tin đơn hàng, có thể điều chỉnh để thay đổi số lượng mỗi sản phẩm hoặc xóa sản phẩm khỏi đơn. Ngoài ra, nhân viên bán hàng có thể điều chỉnh chính sách giảm giá hoặc theo phần trăm hoặc theo giá trị cụ thể. Bên cạnh đó, người bán hàng có thể điều chỉnh phương thức thanh toán bao gồm tiền mặt hoặc chuyển khoản. Nếu phương thức thanh toán là tiền mặt, người bán hàng có thể thao tác nhập tiền khách đưa để tính tiền thừa trả khách. Nếu phương thức thanh toán được chọn là chuyển khoản thì hệ thống sẽ cung cấp mã QR để người mua quét và thanh toán.

Cuối cùng, nhân viên bán hàng xác nhận thanh toán thành công để hoàn tất giao dịch và in hóa đơn giao cho người mua.

**b, Quy trình luân chuyển hàng hóa**



**Hình 2.12:** Biểu đồ quy trình luân chuyển hàng hóa

Để bắt đầu quy trình luân chuyển hàng hóa, quản lý chọn tạo đơn xuất hàng, hệ thống sẽ hiển thị giao diện tạo đơn hàng với trạng thái "Đang tạo". Tại đây, quản lý chọn điểm xuất hàng, điểm nhận hàng, sản phẩm cần giao, có thể điều chỉnh số lượng và chọn lưu đơn. Hệ thống sẽ hiển thị đơn trong danh sách với trạng thái "Đang giao hàng" và cập nhật tồn kho các điểm bán.

Trong quá trình xuất và giao hàng, quản lý có thể hủy đơn hàng trong trường hợp cần thiết. Khi đó, trạng thái đơn chuyển sang "Đã hủy" và hệ thống sẽ cập nhật tồn kho các điểm bán. Ngược lại, khi đơn hàng đã được giao tới điểm nhận, quản lý điểm nhận sẽ xác nhận và cập nhật số lượng hàng nhận được, hệ thống sẽ cập nhật tồn kho và chuyển trạng thái đơn giao sang "Đã giao hàng".

## 2.3 Đặc tả chức năng

### 2.3.1 Đặc tả use case Xử lý giao dịch bán hàng

<b>Mã use case</b>	<b>UC01</b>
<b>Tên use case</b>	Xử lý giao dịch bán hàng
<b>Tên tác nhân</b>	Nhân viên bán hàng
<b>Mô tả</b>	Nhân viên bán hàng tạo đơn bán hàng
<b>Tiền điều kiện</b>	Nhân viên bán hàng đã đăng nhập vào hệ thống.
<b>Luồng sự kiện chính</b>	<ol style="list-style-type: none"> <li>1. Nhân viên bán hàng tìm kiếm và thêm sản phẩm vào đơn bán.</li> <li>2. Hệ thống truy xuất thông tin sản phẩm (ảnh, mã, tên, giá, số lượng) và hiển thị lên màn hình.</li> <li>3. Nhân viên bán hàng có thể thêm thông tin khách hàng hoặc thêm giảm giá vào đơn.</li> <li>4. Nhân viên bán hàng chọn phương thức thanh toán (hoặc tiền mặt, hoặc chuyển khoản).</li> <li>5. Nhân viên bán hàng xác nhận giao dịch thanh toán đã thành công.</li> <li>6. Hệ thống tạo hóa đơn thanh toán.</li> <li>7. Nhân viên bán hàng in hóa đơn và giao cho khách hàng.</li> </ol>
<b>Luồng sự kiện thay thế</b>	<ol style="list-style-type: none"> <li>4a. Nếu chọn phương thức thanh toán là chuyển khoản thì hệ thống sẽ hiển thị mã QR.</li> <li>5a. Nếu khách hàng từ chối thanh toán, nhân viên hủy giao dịch, hệ thống quay lại giao diện bán hàng.</li> <li>6a. Nếu hệ thống gặp lỗi trong quá trình lưu hóa đơn vào cơ sở dữ liệu, nó sẽ thông báo tạo đơn mới thất bại.</li> </ol>
<b>Hậu điều kiện</b>	Hệ thống ghi nhận giao dịch, cập nhật tồn kho.
<b>Ngoại lệ</b>	Không có

**Bảng 2.1:** Đặc tả use case Xử lý giao dịch bán hàng

**2.3.2 Đặc tả use case Quản lý luân chuyển hàng hóa**

<b>Mã use case</b>	<b>UC02</b>
<b>Tên use case</b>	Quản lý luân chuyển hàng hóa
<b>Tên tác nhân</b>	Quản lý
<b>Mô tả</b>	Quản lý tạo đơn chuyển hàng giữa các điểm bán
<b>Tiền điều kiện</b>	Quản lý đã đăng nhập vào hệ thống.
<b>Luồng sự kiện chính</b>	<ol style="list-style-type: none"> <li>1. Quản lý chọn tạo đơn xuất hàng.</li> <li>2. Hệ thống hiển thị giao diện tạo đơn với trạng thái "Đang tạo".</li> <li>3. Quản lý chọn điểm xuất hàng, điểm nhập hàng.</li> <li>4. Hệ thống truy vấn dữ liệu tồn kho và hiển thị giao diện tìm kiếm sản phẩm.</li> <li>5. Quản lý chọn sản phẩm cần giao, điều chỉnh số lượng và nhấn lưu đơn.</li> <li>6. Hệ thống cập nhật tồn kho và hiển thị trạng thái "Đang giao hàng".</li> </ol>
<b>Luồng sự kiện thay thế</b>	<ol style="list-style-type: none"> <li>5a. Nếu chưa chọn điểm xuất hoặc chưa chọn sản phẩm, hệ thống sẽ hiển thị cảnh báo.</li> <li>6a. Nếu Quản lý chọn "Hủy đơn", hệ thống cập nhật tồn kho và hiển thị trạng thái "Đã hủy".</li> </ol>
<b>Hậu điều kiện</b>	Hàng hóa được giao tới điểm bán và xác nhận bởi quản lý (của điểm nhập).
<b>Ngoại lệ</b>	Không có

**Bảng 2.2:** Đặc tả use case Quản lý luân chuyển hàng hóa

### 2.3.3 Đặc tả use case Quản lý thông tin sản phẩm

<b>Mã use case</b>	<b>UC03</b>
<b>Tên use case</b>	Thêm mới sản phẩm
<b>Tên tác nhân</b>	Quản lý
<b>Mô tả</b>	Thêm mới sản phẩm
<b>Tiền điều kiện</b>	Quản lý đã đăng nhập vào hệ thống
<b>Luồng sự kiện chính</b>	<ol style="list-style-type: none"> <li>1. Quản lý chọn thêm sản phẩm mới</li> <li>2. Hệ thống lấy danh sách nhóm thông tin và hiển thị giao diện thêm sản phẩm</li> <li>3. Quản lý nhập các thông tin chung (tên, nhãn hiệu) cho sản phẩm</li> <li>4. Hệ thống sẽ tạo tự động mã SKU cho sản phẩm nếu không được nhập thủ công</li> <li>5. Quản lý thêm thông tin (hình ảnh, giá nhập, thuế, giá bán lẻ, tồn kho)</li> <li>6. Quản lý kiểm tra thông tin và chọn “Lưu”</li> <li>7. Hệ thống kiểm tra các trường bắt buộc nhập</li> <li>8. Hệ thống kiểm tra các trường cần đảm bảo duy nhất</li> <li>9. Hệ thống cập nhật các thông tin và thông báo thêm mới thành công</li> </ol>
<b>Luồng sự kiện thay thế</b>	<ol style="list-style-type: none"> <li>3a. Quản lý chọn tạo thêm biến thể sản phẩm</li> <li>3b. Hệ thống hiển thị giao diện tạo biến thể</li> <li>3c. Quản lý nhập các thuộc tính, giá trị mỗi thuộc tính và chọn các biến thể muốn tạo</li> <li>4a. Hệ thống tự động tạo mã SKU cho mỗi biến thể nếu không được nhập thủ công</li> </ol>
<b>Hậu điều kiện</b>	Sản phẩm mới được tạo thành công, hiển thị trong danh sách sản phẩm và được lưu trong cơ sở dữ liệu
<b>Ngoại lệ</b>	Không có

**Bảng 2.3:** Đặc tả use case Quản lý thông tin sản phẩm

**2.3.4 Đặc tả use case Quản lý tồn kho điểm bán**

<b>Mã use case</b>	<b>UC04</b>
<b>Tên use case</b>	Quản lý tồn kho điểm bán
<b>Tên tác nhân</b>	Quản lý
<b>Mô tả</b>	Theo dõi số lượng, kiểm đếm sản phẩm tồn kho
<b>Tiền điều kiện</b>	Quản lý đã đăng nhập vào hệ thống
<b>Luồng sự kiện chính</b>	<ol style="list-style-type: none"> <li>1. Quản lý chọn thêm kiểm đếm tồn kho</li> <li>2. Hệ thống hiển thị giao diện tạo phiếu kiểm đếm tồn kho</li> <li>3. Quản lý chọn sản phẩm cần kiểm đếm tồn kho</li> <li>4. Quản lý kiểm đếm sản phẩm và nhập số lượng thực tế tồn kho cho tất cả sản phẩm</li> <li>5. Quản lý thêm ghi chú cho đơn kiểm đếm và chọn hoàn thành</li> <li>6. Hệ thống lưu phiếu kiểm đếm tồn kho và cập nhật tồn kho theo số lượng kiểm đếm thực tế</li> </ol>
<b>Luồng sự kiện thay thế</b>	6a. Hệ thống thông báo lỗi nếu không lưu được đơn kiểm đếm
<b>Hậu điều kiện</b>	Không có
<b>Ngoại lệ</b>	Không có

**Bảng 2.4:** Đặc tả use case Quản lý tồn kho điểm bán



### 2.3.5 Đặc tả use case Quản lý thông tin điểm bán

<b>Mã use case</b>	<b>UC05</b>
<b>Tên use case</b>	Quản lý thông tin điểm bán
<b>Tên tác nhân</b>	Chủ cửa hàng
<b>Mô tả</b>	Chủ cửa hàng thêm mới điểm bán
<b>Tiền điều kiện</b>	Chủ cửa hàng đã đăng nhập vào hệ thống
<b>Luồng sự kiện chính</b>	<ol style="list-style-type: none"> <li>1. Chủ cửa hàng thêm điểm bán hàng</li> <li>2. Hiển thị giao diện thêm điểm bán hàng</li> <li>3. Chủ cửa hàng thêm thông tin ( tên điểm bán, số điện thoại, email, lĩnh vực kinh doanh và địa chỉ đăng ký)</li> <li>4. Chủ cửa hàng kiểm tra thông tin và chọn lưu</li> <li>5. Hệ thống kiểm tra các trường bắt buộc nhập</li> <li>6. Hệ thống kiểm tra các trường cần đảm bảo duy nhất</li> </ol>
<b>Luồng sự kiện thay thế</b>	4a. Hệ thống thông báo nếu các trường thông tin đã tồn tại hoặc chưa được nhập
<b>Hậu điều kiện</b>	Hệ thống cập nhật các thông tin và thông báo thêm mới thành công
<b>Ngoại lệ</b>	Không có

**Bảng 2.5:** Đặc tả use case Quản lý thông tin điểm bán

### 2.3.6 Đặc tả use case Quản lý nhân viên

<b>Mã use case</b>	<b>UC06</b>
<b>Tên use case</b>	Quản lý nhân viên
<b>Tên tác nhân</b>	Chủ cửa hàng
<b>Mô tả</b>	Chủ cửa hàng thêm mới nhân viên
<b>Tiền điều kiện</b>	Chủ cửa hàng đã đăng nhập vào hệ thống
<b>Luồng sự kiện chính</b>	<ol style="list-style-type: none"> <li>1. Chủ cửa hàng thêm nhân viên</li> <li>2. Hiển thị giao diện thêm nhân viên mới.</li> <li>3. Chủ cửa hàng thêm thông tin ( tên hiển thị, số điện thoại, email, ngày sinh, địa chỉ)</li> <li>4. Chủ cửa hàng chọn điểm bán làm việc, vai trò và tạo mật khẩu cho tài khoản</li> <li>5. Chủ cửa hàng kiểm tra thông tin và chọn lưu</li> <li>5. Hệ thống kiểm tra các trường bắt buộc nhập</li> <li>6. Hệ thống kiểm tra các trường cần đảm bảo duy nhất</li> </ol>
<b>Luồng sự kiện thay thế</b>	4a. Hệ thống thông báo nếu các trường thông tin đã tồn tại hoặc chưa được nhập
<b>Hậu điều kiện</b>	Hệ thống cập nhật các thông tin và thông báo thêm mới thành công
<b>Ngoại lệ</b>	Không có

**Bảng 2.6:** Đặc tả use case Quản lý nhân viên

### 2.3.7 Đặc tả use case Quản lý kế hoạch điểm bán

<b>Mã use case</b>	<b>UC07</b>
<b>Tên use case</b>	Quản lý kế hoạch điểm bán
<b>Tên tác nhân</b>	Chủ cửa hàng
<b>Mô tả</b>	Chủ cửa hàng thêm mới kế hoạch các điểm bán
<b>Tiền điều kiện</b>	Chủ cửa hàng đã đăng nhập vào hệ thống
<b>Luồng sự kiện chính</b>	<ol style="list-style-type: none"> <li>1. Chủ cửa hàng thêm kế hoạch điểm bán</li> <li>2. Hiển thị giao diện thêm kế hoạch điểm bán</li> <li>3. Chủ cửa hàng tải xuống bản kế hoạch bán hàng mẫu và chỉnh sửa</li> <li>3. Chủ cửa hàng tải lên bản kế hoạch bán hàng</li> <li>4. Chủ cửa hàng kiểm tra thông tin và chọn lưu</li> <li>5. Hệ thống kiểm tra các trường bắt buộc nhập</li> </ol>
<b>Luồng sự kiện thay thế</b>	<ol style="list-style-type: none"> <li>5a. Hệ thống thông báo nếu các trường thông tin chưa được nhập</li> <li>5b. Hệ thống ghi đè dữ liệu cũ nếu các trường thông tin được nhập đã tồn tại</li> </ol>
<b>Hậu điều kiện</b>	Hệ thống cập nhật các thông tin vào cơ sở dữ liệu
<b>Ngoại lệ</b>	Không có

**Bảng 2.7:** Đặc tả use case Quản lý kế hoạch điểm bán

## 2.4 Yêu cầu phi chức năng

### 2.4.1 Tính dễ sử dụng

Hệ thống POS có thiết kế giao diện người dùng thân thiện, dễ hiểu và dễ sử dụng cho cả nhân viên và quản lý. Các thao tác phải mượt mà, có phản hồi nhanh chóng từ hệ thống để mang lại trải nghiệm thuận tiện. Giao diện cần đồng bộ và nhất quán, từ màu sắc, bố cục đến các thành phần như nút bấm (button), ô nhập liệu (input), bảng biểu và thông báo. Hệ thống cũng cần cung cấp các thông báo về trạng thái hệ thống, thông báo thành công hoặc thất bại đối với các thao tác của người dùng. Ví dụ, khi thực hiện chuyển hàng hoặc hoàn thành giao dịch, người dùng sẽ nhận được phản hồi kịp thời từ hệ thống.

### 2.4.2 Tính dễ bảo trì

Hệ thống được phát triển theo hướng module hóa, nghĩa là phân chia thành các thành phần nhỏ với mỗi thành phần chịu trách nhiệm một chức năng cụ thể, như quản lý sản phẩm, giao dịch, lịch trình, và tồn kho. Điều này giúp hệ thống dễ dàng bảo trì và nâng cấp trong tương lai mà không gây ảnh hưởng đến các phần khác. Các thành phần giao diện (UI components) cũng cần được thiết kế theo hướng tái sử dụng, đảm bảo thuận tiện khi có nhu cầu thay đổi hoặc mở rộng tính năng trong tương lai. Ví dụ, khi thêm một tính năng mới như quản lý phiếu giảm giá, module

hiện tại sẽ không bị ảnh hưởng.

### **2.4.3 Độ tin cậy**

Hệ thống POS đảm bảo tính tin cậy cao về dữ liệu, đặc biệt là các thông tin nhạy cảm như mật khẩu người dùng phải được mã hóa khi lưu trữ vào cơ sở dữ liệu để bảo vệ tính riêng tư. Hệ thống phải đảm bảo rằng dữ liệu không bị mất mát hoặc thay đổi do lỗi hệ thống hoặc sự cố. Hệ thống cần ngăn chặn việc truy cập trái phép bằng cách bảo mật quá trình xác thực người dùng, đảm bảo không lộ thông tin người dùng cho đến khi đăng nhập thành công.

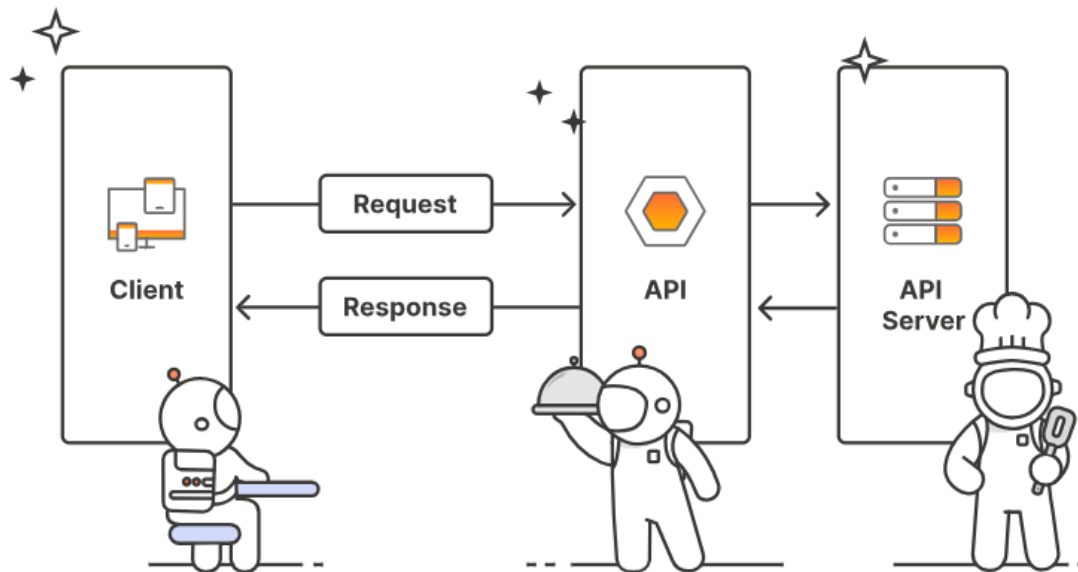
### **2.4.4 Tính bảo mật**

Hệ thống đảm bảo bảo mật cho tất cả thông tin quan trọng, bao gồm việc mã hóa dữ liệu nhạy cảm và hạn chế quyền truy cập dựa trên vai trò của người dùng. Ví dụ, quản lý chỉ có quyền truy cập vào báo cáo doanh thu của điểm bán mình phụ trách và nhân viên bán hàng chỉ có quyền thực hiện giao dịch tại điểm bán mà họ phụ trách. Cơ chế phân quyền sẽ giúp bảo vệ dữ liệu khỏi các hành vi truy cập trái phép hoặc can thiệp sửa đổi không mong muốn.

## CHƯƠNG 3. CÔNG NGHỆ SỬ DỤNG

### 3.1 API (Application Programming Interface)

API (Application Programming Interface) là một tập hợp các quy tắc và giao thức cho phép các ứng dụng khác nhau giao tiếp với nhau. API đóng vai trò như một cầu nối, cho phép frontend và backend của một ứng dụng tương tác một cách liền mạch.



**Hình 3.1:** Biểu đồ cách hoạt động của API [3]

#### Cách hoạt động

**Yêu cầu HTTP:** Frontend gửi yêu cầu HTTP (GET, POST, PUT, DELETE) đến server backend. Các yêu cầu này chứa thông tin về hành động cần thực hiện hoặc dữ liệu cần gửi.

**Xử lý yêu cầu:** Server nhận yêu cầu, xử lý logic nghiệp vụ cần thiết và truy xuất dữ liệu từ cơ sở dữ liệu nếu cần.

**Phản hồi HTTP:** Server trả về phản hồi HTTP, thường dưới dạng JSON hoặc XML, chứa dữ liệu yêu cầu hoặc thông báo trạng thái của hành động đã thực hiện. Các API này đảm bảo rằng dữ liệu được truyền tải một cách an toàn và hiệu quả giữa các phần của ứng dụng.

### 3.2 Spring Boot

Spring Boot là một framework mạnh mẽ, được xây dựng trên nền tảng Spring Framework, nhằm giúp đơn giản hóa quá trình phát triển ứng dụng Java. Nó hỗ trợ các lập trình viên tạo ra các ứng dụng độc lập, có khả năng hoạt động ngay lập tức

mà không cần phải cấu hình phức tạp.



**Hình 3.2:** Spring Boot [4]

Spring Boot cung cấp hệ sinh thái phong phú thông qua Spring Boot Starters, là các bộ thư viện được tối ưu hóa và đóng gói sẵn để hỗ trợ nhanh chóng các tính năng như kết nối cơ sở dữ liệu, bảo mật, REST API, hoặc giao tiếp với các dịch vụ bên ngoài.

Nhờ sự linh hoạt, dễ sử dụng, và khả năng mở rộng cao, Spring Boot đã trở thành một trong những công cụ không thể thiếu đối với các lập trình viên Java hiện nay, đặc biệt trong các dự án phát triển nhanh hoặc yêu cầu sự ổn định và hiệu quả.

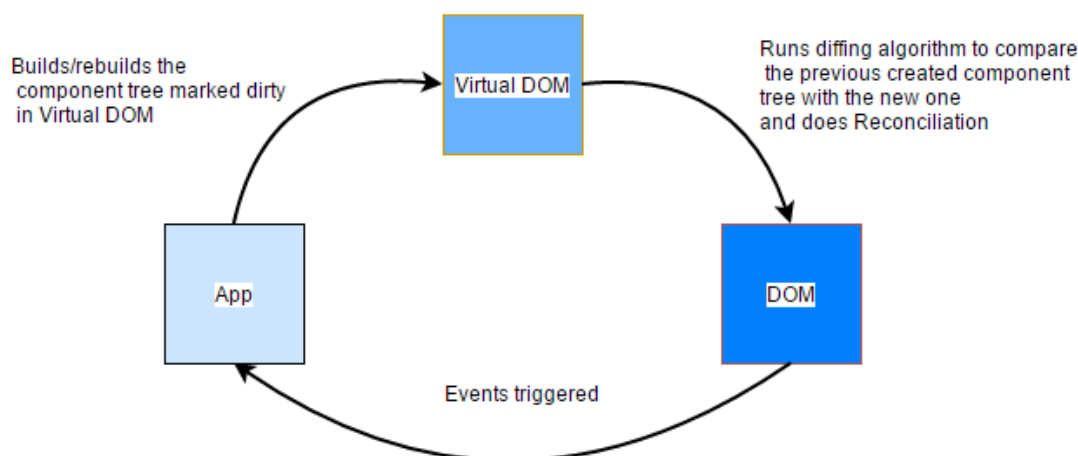
Bên cạnh đó, Spring Boot cũng hỗ trợ mô hình MVC (Model-View-Controller) một cách tự nhiên và hiệu quả. MVC là một kiến trúc phần mềm giúp phân tách rõ ràng giữa dữ liệu, giao diện hiển thị, và logic điều khiển của ứng dụng. Cụ thể cách thức triển khai MVC sẽ được trình bày ở chương 4.

### 3.3 React

React là một thư viện JavaScript mạnh mẽ được phát triển bởi Facebook để xây dựng giao diện người dùng (UI). React cho phép tạo ra các thành phần UI có thể tái sử dụng, dễ dàng quản lý trạng thái, cập nhật giao diện người dùng một cách linh hoạt và tự động khi trạng thái thay đổi mà không cần phải tải lại trang.



**Hình 3.3:** React[5]



**Hình 3.4:** Biểu đồ cách hoạt động của Virtual DOM trong React[6]

Cách hoạt động: React dựa vào Virtual DOM, là bản sao của Actual DOM. Virtual DOM được tải lại ngay lập tức để phản ánh thay đổi mới bất cứ khi nào có thay đổi trong dữ liệu. Sau đó React so sánh Virtual DOM với Actual DOM để tìm ra chính xác những gì đã thay đổi.

Component-Based: Ứng dụng được chia thành các thành phần nhỏ, độc lập, giúp dễ dàng quản lý và tái sử dụng. Mỗi component trong React có thể tự quản lý trạng thái và logic riêng, giúp cấu trúc ứng dụng rõ ràng và dễ bảo trì.

Virtual DOM: React sử dụng Virtual DOM để tăng hiệu suất render. Khi trạng thái của một thành phần thay đổi, chỉ những phần thay đổi sẽ được cập nhật thay vì render lại toàn bộ trang.

### 3.4 MySQL

MySQL là một trong những hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) phổ biến nhất trên thế giới, được sử dụng rộng rãi trong các ứng dụng web và nhiều lĩnh vực khác. MySQL đã trải qua nhiều phiên bản và cải tiến để trở thành một hệ quản trị cơ sở dữ liệu mạnh mẽ và phổ biến.

MySQL sử dụng Structured Query Language (SQL) để quản lý và truy vấn cơ sở dữ liệu. Hoạt động theo mô hình client-server, nơi mà MySQL Server cung cấp dịch vụ lưu trữ và quản lý dữ liệu và MySQL Clients thực hiện các truy vấn và thao tác.



**Hình 3.5:** MySQL [7]

Các tính năng nổi bật của MySQL:

**Hiệu suất cao:** MySQL được tối ưu hóa để xử lý các truy vấn nhanh chóng và hiệu quả. Nó hỗ trợ nhiều kỹ thuật tối ưu hóa và cải tiến để đáp ứng các yêu cầu về hiệu suất từ các ứng dụng web và hệ thống lớn. **Đa nền tảng:** MySQL hỗ trợ nhiều nền tảng hệ điều hành như Linux, Windows, macOS,... Điều này cho phép triển khai MySQL trên nhiều môi trường hệ thống khác nhau.

**Dễ sử dụng:** MySQL cung cấp các công cụ quản lý dữ liệu và quản trị cơ sở dữ liệu giúp người dùng dễ dàng thiết kế, quản lý và giám sát cơ sở dữ liệu một cách hiệu quả.

**Độ bảo mật cao :** MySQL cung cấp các cơ chế bảo mật mạnh mẽ như quản lý người dùng, phân quyền, mã hóa dữ liệu và kiểm soát truy cập để bảo vệ dữ liệu khỏi các mối đe dọa bảo mật.

**Khả năng mở rộng:** MySQL hỗ trợ các tính năng mở rộng để có thể xử lý và lưu trữ dữ liệu lớn hơn.

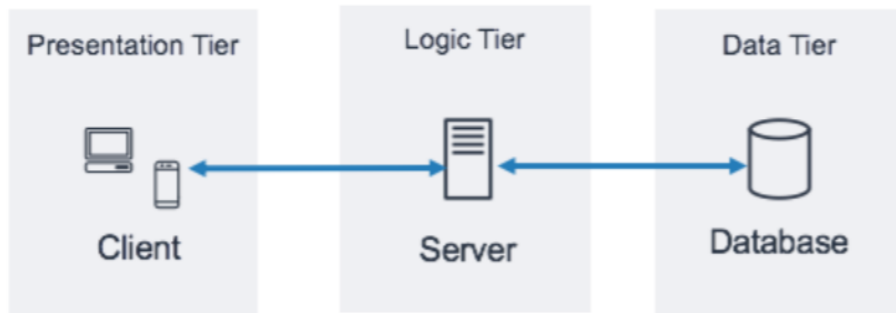


## CHƯƠNG 4. THIẾT KẾ, TRIỂN KHAI VÀ ĐÁNH GIÁ HỆ THỐNG

### 4.1 Thiết kế kiến trúc

#### 4.1.1 Lựa chọn kiến trúc phần mềm

Để phát triển hệ thống, em sử dụng kết hợp kiến trúc 3 tầng (3-tier architecture) và kiến trúc MVC(Model-View-Controller).



**Hình 4.1:** Mô hình kiến trúc 3 tầng [8]

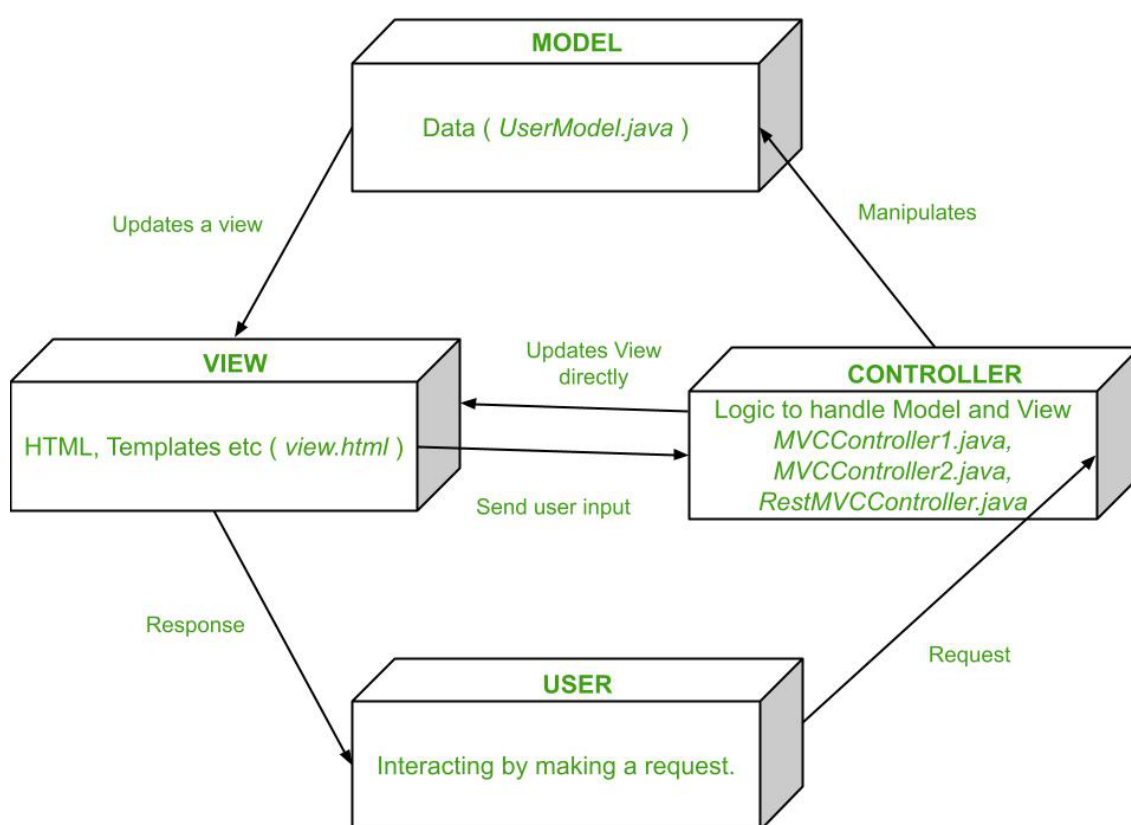
Kiến trúc này được chia thành 3 tầng: tầng trình bày (application tier), tầng ứng dụng (Data tier), tầng dữ liệu (data tier). Mô hình kiến trúc 3 tầng là một kiến trúc kiểu client-server mà trong đó giao diện người dùng (UI-user interface), các xử lý logic, nghiệp vụ (BR-business rule hay BL-business logic), và việc lưu trữ dữ liệu (database) được phát triển như những module độc lập và hầu hết là được duy trì trên các nền tảng độc lập.

Tầng trình bày (presentation tier): Tầng xử lý giao tiếp với người dùng, hiển thị giao diện, nhận yêu cầu từ người dùng, truyền tải yêu cầu cần xử lý đến tầng ứng dụng, đồng thời nhận lại các kết quả trả về từ tầng ứng dụng để hiển thị thông tin đến người dùng. Trong hệ thống, tầng này là Frontend sử dụng Reactjs.

Tầng ứng dụng (application tier): còn được gọi là tầng logic, là trung tâm, lõi xử lý hầu hết hoạt động của hệ thống. Trong tầng này, thông tin được thu thập từ tầng trình bày và được xử lý nghiệp vụ (có thể xử lý dữ liệu từ tầng dữ liệu). Tầng này cũng có thể thêm, sửa, xóa dữ liệu trong tầng dữ liệu. Cụ thể trong hệ thống tầng này là Backend và được chia tầng thành 3 lớp chính (3-layer) bao gồm: Controller, Service, Repository. Controller Layer: Giao tiếp với Frontend thông qua RESTful API, có nhiệm vụ nhận yêu cầu tầng trình bày chuyển xuống cho Service Layer xử

lý, sau khi được xử lý xong trả kết quả về phía Client. Service Layer: Xử lý nghiệp vụ và logic của phần mềm, gọi xuống Repository Layer để thao tác với dữ liệu của hệ thống. Repository Layer: Thao tác với tầng dữ liệu bao gồm cơ sở dữ liệu và các file khác của hệ thống.

Tầng dữ liệu (Data tier): là nơi lưu trữ, quản lý, truy xuất dữ liệu của hệ thống, với hệ thống này là hệ quản trị cơ sở dữ liệu MySQL. Trong hệ thống, tất cả các giao tiếp đều đi qua tầng ứng dụng. Tầng trình bày và tầng dữ liệu không giao tiếp trực tiếp với nhau



**Hình 4.2:** Thiết kế kiến trúc MVC [9]

Mô hình MVC là kiến trúc kiểu client-server, mô tả luồng đi của dữ liệu, chia hệ thống thành ba thành phần chính: Model, View, Controller

Model: chịu trách nhiệm cho việc lưu trữ dữ liệu và logic liên quan của nó. Model biểu diễn các đối tượng thực thể hoặc các dữ liệu của ứng dụng, và cung cấp các phương thức để truy xuất, thay đổi dữ liệu này. Trong một ứng dụng web, Model thường tương tác với cơ sở dữ liệu để lấy và lưu trữ thông tin.

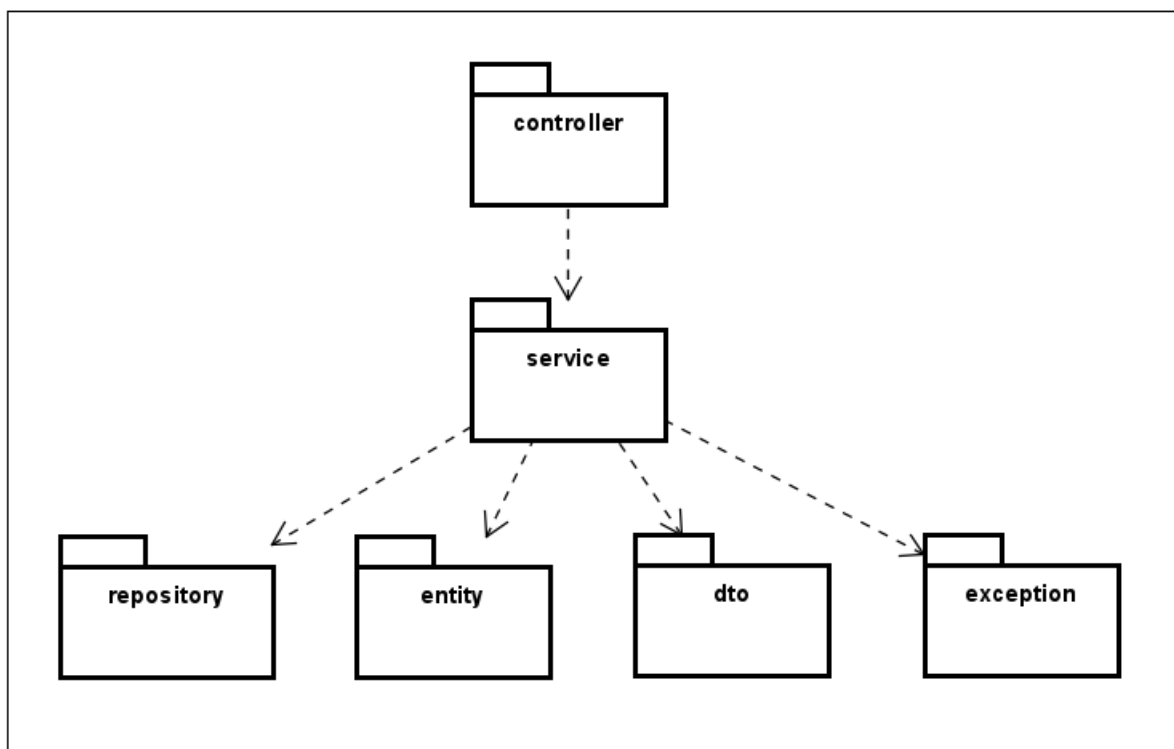
View: là thành phần chịu trách nhiệm hiển thị dữ liệu cho người dùng và tổ chức giao diện người dùng. View nhận dữ liệu từ Model và render nó ra dưới dạng HTML, XML, hoặc một định dạng hiển thị khác. Nó thể hiện trực tiếp các dữ liệu

và thông tin từ Model mà người dùng có thể thấy và tương tác.

Controller: chịu trách nhiệm điều hướng các yêu cầu và sự kiện từ người dùng đến Model và View tương ứng. Controller lắng nghe các yêu cầu từ người dùng thông qua giao diện người dùng (View), xử lý các yêu cầu này và gửi các lệnh cần thiết đến Model để cập nhật dữ liệu. Sau khi Model được cập nhật, Controller sẽ chuyển dữ liệu mới nhất này đến View để hiển thị cho người dùng.

Khi triển khai mô hình MVC với React là phần Frontend và Spring Boot là phần Backend, React đảm nhận vai trò quan trọng trong việc thiết kế và hiển thị giao diện người dùng, xử lý sự kiện từ người dùng như click và nhập liệu, đồng thời gọi các API từ Backend để tương tác với dữ liệu và logic nghiệp vụ của ứng dụng. Spring Boot với vai trò là Backend, đảm nhận các nhiệm vụ chính bao gồm xử lý logic, quản lý và truy xuất dữ liệu từ cơ sở dữ liệu, và cung cấp các endpoint API để cho phép Frontend gửi yêu cầu và nhận phản hồi. Controller làm nhiệm vụ điều hướng các yêu cầu từ Frontend và xử lý logic nghiệp vụ, sử dụng các phương thức của Model để truy xuất và xử lý dữ liệu trước khi trả về kết quả cho Frontend. Sự kết hợp này giúp phân tách rõ ràng trách nhiệm giữa phần hiển thị và xử lý dữ liệu, đồng thời tối ưu hóa khả năng bảo trì và mở rộng của ứng dụng.

#### 4.1.2 Thiết kế tổng quan



**Hình 4.3:** Biểu đồ phụ thuộc gói backend

Gói Controller chịu trách nhiệm tiếp nhận các yêu cầu từ phía Frontend thông qua các API endpoint. Các yêu cầu này có thể bao gồm các thao tác như thêm mới, cập nhật, xóa, hoặc truy vấn dữ liệu. Sau khi tiếp nhận, các Controller sẽ thực hiện việc xác thực và kiểm tra tính hợp lệ của dữ liệu đầu vào. Tiếp đó, các yêu cầu hợp lệ được chuyển tiếp đến các lớp trong gói Service để xử lý. Dữ liệu sau khi xử lý sẽ được trả về Frontend dưới dạng JSON theo yêu cầu, đảm bảo đáp ứng nhanh chóng và đúng mục tiêu.

Gói Service chịu trách nhiệm chính trong việc xử lý nghiệp vụ của hệ thống. Đây là tầng logic trung gian giữa gói Controller và gói Repository. Các lớp trong gói này nhận yêu cầu từ Controller, áp dụng các quy tắc nghiệp vụ và logic cụ thể, đồng thời gọi đến các phương thức trong gói Repository để thao tác với cơ sở dữ liệu. Kết quả sau khi xử lý sẽ được trả về gói Controller, thường dưới dạng các lớp DTO (Data Transfer Object). Để đảm bảo tính tách biệt, gói Service sử dụng các lớp Mapper để chuyển đổi dữ liệu giữa Entity và DTO, giúp dữ liệu trở nên an toàn và dễ kiểm soát hơn.

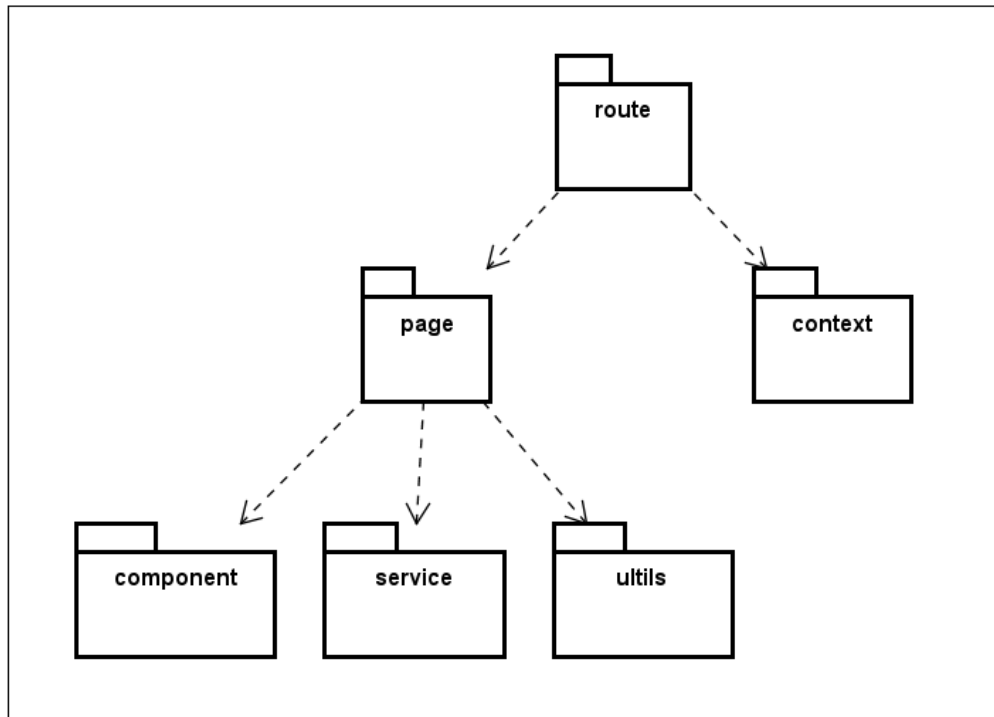
Gói Repository là tầng giao tiếp trực tiếp với cơ sở dữ liệu, chịu trách nhiệm thực hiện các truy vấn CRUD (Create, Read, Update, Delete) thông qua các công cụ như JPA/Hibernate. Ngoài ra, các lớp trong gói này có thể cung cấp các phương thức tùy chỉnh để thực hiện các truy vấn phức tạp hơn, đảm bảo phục vụ tốt nhất cho các yêu cầu từ gói Service. Dữ liệu từ cơ sở dữ liệu sẽ được trả về dưới dạng các đối tượng Entity, cho phép xử lý và lưu trữ dễ dàng.

Gói Entity chứa các lớp đại diện cho các bảng trong cơ sở dữ liệu, mỗi lớp bao gồm các thuộc tính tương ứng với các cột trong bảng. Các lớp này thường được gắn các annotation như @Table, @Column, @Id để ánh xạ trực tiếp với cấu trúc cơ sở dữ liệu. Ngoài ra, các Entity cũng mô tả các mối quan hệ giữa các bảng, như OneToMany, ManyToOne, hay ManyToMany, giúp dễ dàng quản lý và truy vấn dữ liệu phức tạp.

Gói DTO (Data Transfer Object) được thêm vào để chuyển đổi dữ liệu giữa các tầng của hệ thống mà không cần sử dụng trực tiếp các Entity. Các lớp DTO đảm bảo tính bảo mật khi giao tiếp với Frontend, bằng cách ẩn đi các thông tin nhạy cảm và chỉ cung cấp các dữ liệu cần thiết. Gói này bao gồm các lớp kế thừa từ BaseDto, chứa các thuộc tính như id, createdAt, updatedAt để đồng bộ dữ liệu. Ngoài ra, gói này cũng có các lớp hỗ trợ phản hồi chuẩn hóa như ResponseObject, giúp việc giao tiếp giữa Frontend và Backend trở nên thuận tiện hơn.

Gói Exception giúp quản lý và xử lý các lỗi phát sinh trong hệ thống, từ tầng Controller đến Repository. Gói này bao gồm các lớp ngoại lệ tùy chỉnh như Not-

FoundException, BadRequestException, UnauthorizedException và một lớp xử lý ngoại lệ toàn cục GlobalExceptionHandler sử dụng các annotation như @ControllerAdvice và @ExceptionHandler. Các lỗi được chuẩn hóa với mã lỗi và thông điệp rõ ràng, không chỉ hỗ trợ việc debug mà còn cung cấp trải nghiệm người dùng tốt hơn khi gặp sự cố.



**Hình 4.4:** Biểu đồ phụ thuộc gói frontend

Gói Route chịu trách nhiệm quản lý và xác định các định tuyến trong ứng dụng React, cho phép người dùng điều hướng giữa các trang một cách mượt mà và trực quan. Trong React, Route thường được sử dụng với thư viện như React Router để thiết lập các đường dẫn URL tương ứng với các thành phần UI cụ thể.

Gói Page chứa các trang giao diện chính trong ứng dụng, được tổ chức thành từng module để dễ dàng quản lý và bảo trì. Mỗi trang trong gói này là một component lớn, bao gồm nhiều thành phần UI nhỏ hơn (components) được tổ chức theo bố cục cụ thể. Việc tách giao diện thành các trang giúp ứng dụng có cấu trúc rõ ràng, dễ mở rộng và thuận tiện cho việc kiểm thử từng phần.

Gói Component bao gồm các thành phần UI nhỏ, tái sử dụng được, nhằm giảm thiểu sự lặp lại mã và tăng tính linh hoạt trong việc xây dựng giao diện. Các component này có thể là các nút bấm (Button), trường nhập liệu (Input), hộp thoại (Modal), hoặc các thành phần phức tạp hơn như thanh điều hướng (Navbar) hoặc biểu đồ (Chart). Việc tái sử dụng các component không chỉ giúp tiết kiệm thời gian mà còn đảm bảo tính nhất quán trong giao diện người dùng trên toàn bộ ứng dụng.

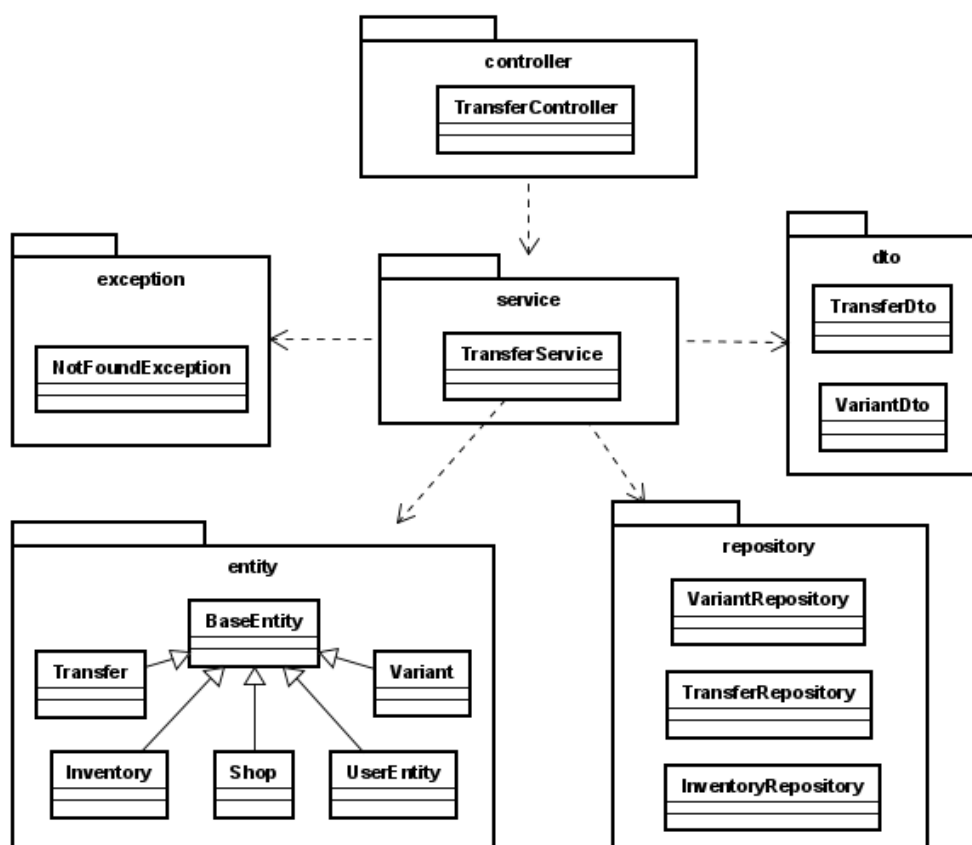
Các component thường được thiết kế theo hướng module và được tùy chỉnh thông qua các props.

Gói Service chịu trách nhiệm xử lý các logic liên quan đến việc giao tiếp với API backend. Các lớp hoặc hàm trong gói này thường được sử dụng để thực hiện các yêu cầu HTTP (GET, POST, PUT, DELETE) và xử lý phản hồi từ máy chủ. Chúng có thể bao gồm các chức năng như xác thực người dùng, lấy dữ liệu từ cơ sở dữ liệu, hoặc gửi thông báo đến máy chủ. Gói Service cũng thường được tích hợp với các công cụ như Axios để đảm bảo việc giao tiếp với máy chủ hiệu quả và an toàn. Ngoài ra, các logic xử lý lỗi và quản lý token cũng có thể được triển khai tại đây.

Gói Context được sử dụng để quản lý trạng thái toàn cục của ứng dụng, thay vì truyền dữ liệu qua nhiều cấp component. Thông qua Context API của React, các trạng thái chung như thông tin người dùng hoặc cài đặt giao diện có thể được lưu trữ và truy cập dễ dàng từ bất kỳ thành phần nào trong ứng dụng. Gói này không chỉ giúp giảm độ phức tạp trong việc truyền dữ liệu mà còn nâng cao hiệu suất bằng cách kiểm soát chính xác các thành phần nào cần cập nhật khi trạng thái thay đổi.

Gói Utils chứa các hàm tiện ích được sử dụng chung trên toàn bộ ứng dụng, nhằm hỗ trợ các tác vụ lặp đi lặp lại hoặc tạo các hàm tiện ích như xử lý chuỗi ký tự, kiểm tra tính hợp lệ của dữ liệu. Gói này đóng vai trò như một thư viện công cụ nhỏ, giúp mã nguồn trở nên gọn gàng, dễ đọc và dễ bảo trì hơn. Việc tách riêng các hàm tiện ích vào gói Utils cũng giúp giảm sự phụ thuộc giữa các thành phần khác trong ứng dụng.

### 4.1.3 Thiết kế chi tiết gói



**Hình 4.5:** Biểu đồ chi tiết gói chức năng Luân chuyển hàng hóa

Biểu đồ trên mô tả kiến trúc chi tiết của chức năng Luân chuyển hàng hóa trong hệ thống quản lý POS, được tổ chức thành các gói (package). Mỗi gói đảm nhiệm một phần cụ thể trong quy trình xử lý luân chuyển hàng hóa, bao gồm từ tiếp nhận yêu cầu từ phía người dùng đến việc tương tác với cơ sở dữ liệu và xử lý logic nghiệp vụ.

Gói controller chứa lớp **TransferController**, đóng vai trò như điểm tiếp nhận yêu cầu từ phía giao diện người dùng (Frontend). Khi người dùng thực hiện các thao tác liên quan đến luân chuyển hàng hóa, chẳng hạn như yêu cầu chuyển hàng giữa kho và các điểm bán hàng, lớp này sẽ tiếp nhận và chuyển các yêu cầu đến tầng dịch vụ để xử lý. Đồng thời, nó cũng chịu trách nhiệm trả về dữ liệu đã được xử lý cho giao diện người dùng, đảm bảo sự giao tiếp mượt mà giữa người dùng và hệ thống.

Gói service chứa lớp **TransferService**, nơi tập trung xử lý toàn bộ logic nghiệp vụ liên quan đến luân chuyển hàng hóa. Lớp này nhận các yêu cầu từ tầng điều khiển, thực hiện các tác vụ như kiểm tra tồn kho, xác minh tính hợp lệ của yêu cầu luân chuyển, và tính toán các thông tin cần thiết. Ngoài ra, nó còn gọi đến các lớp

trong gói repository để thực hiện các thao tác lưu trữ hoặc truy xuất dữ liệu từ cơ sở dữ liệu. Kết quả xử lý sau đó được trả về tầng điều khiển để phản hồi cho người dùng.

Gói repository chứa các lớp **TransferRepository**, **VariantRepository**, và **InventoryRepository**, chịu trách nhiệm thực hiện các thao tác truy vấn và cập nhật cơ sở dữ liệu. **TransferRepository** xử lý thông tin liên quan đến các giao dịch chuyển hàng, **VariantRepository** quản lý thông tin về các biến thể sản phẩm, và **InventoryRepository** đảm bảo dữ liệu về tồn kho được duy trì chính xác. Tầng này đóng vai trò như cầu nối giữa logic nghiệp vụ và cơ sở dữ liệu, đảm bảo việc lưu trữ và truy xuất dữ liệu diễn ra an toàn và hiệu quả.

Gói entity chứa các lớp thực thể đại diện cho các bảng dữ liệu trong cơ sở dữ liệu. Trong đó, **BaseEntity** là lớp cơ sở cung cấp các thuộc tính chung như **id**, **createdAt**, và **updatedAt**. Các lớp khác như **Transfer**, **Variant**, **Inventory**, **Shop**, và **UserEntity** mô hình hóa các thông tin quan trọng trong hệ thống. Ví dụ, lớp **Transfer** quản lý dữ liệu về các giao dịch chuyển hàng, trong khi **Inventory** theo dõi thông tin tồn kho tại các kho và điểm bán hàng.

Gói dto bao gồm các lớp như **TransferDto** và **VariantDto**, được thiết kế để truyền dữ liệu giữa các tầng trong hệ thống. Các lớp này giúp giảm sự phụ thuộc giữa các tầng bằng cách cung cấp các dữ liệu cần thiết dưới dạng cấu trúc đã được tối ưu hóa. Điều này không chỉ nâng cao hiệu suất mà còn giúp hệ thống dễ bảo trì và mở rộng hơn.

Gói exception chứa lớp **NotFoundException**, đảm nhận việc xử lý các lỗi xảy ra khi không tìm thấy dữ liệu mong muốn. Chẳng hạn, nếu không tìm thấy thông tin về một yêu cầu chuyển hàng hoặc một sản phẩm cụ thể, lớp này sẽ được kích hoạt để thông báo lỗi một cách rõ ràng và có kiểm soát.

Trên đây là cách em tổ chức, thiết kế các gói cho hệ thống và ví dụ cụ thể cho gói chức năng về Luân chuyển hàng hóa, các chức năng còn lại được thiết kế tương tự. Tiếp theo em xin giới thiệu về phần thiết kế chi tiết.



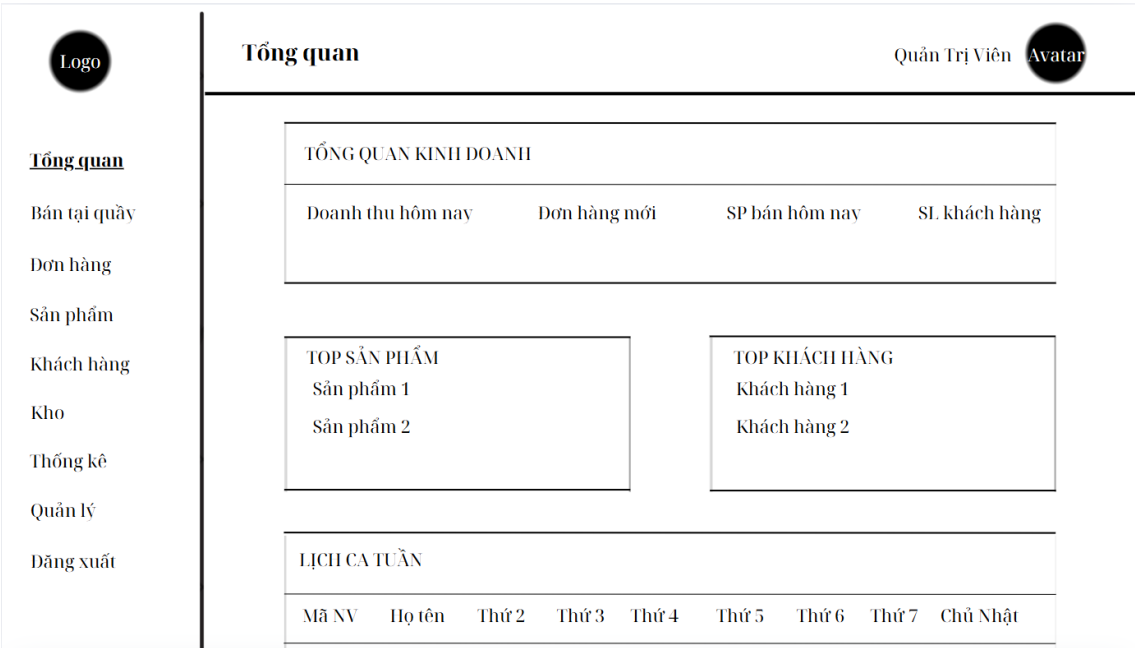
4.2 Thiết kế chi tiết

4.2.1 Thiết kế giao diện

Thông tin màn hình	Thiết kế
Độ phân giải	Hỗ trợ các màn hình có độ phân giải HD trở lên
Số lượng màu sắc hỗ trợ	Toàn bộ dải màu RGBA hoặc HEX
Hiển thị thông báo từ hệ thống	Thành công: Hộp thoại màu xanh lá  Thông tin: Hộp thoại màu xanh dương  Lỗi: Hộp thoại màu đỏ
Ngôn ngữ hiển thị	Tiếng Việt

Bảng 4.1: Thông tin thiết kế giao diện

Dưới đây là một số giao diện minh họa các chức năng chính của hệ thống



Hình 4.6: Thiết kế màn hình tổng quan

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

Thêm sản phẩm vào đơn

×

Fashion Shop

STT	Ảnh SP	Mã SKU	Tên sản phẩm	Số lượng	Đơn giá
Đơn hàng chưa có sản phẩm nào					

Tổng tiền

Chiết khấu

VAT

Hình thức thanh toán

Tiền mặt ▼

In Hóa Đơn

Thanh Toán

**Hình 4.7:** Thiết kế màn hình bán hàng

Logo

Tổng quan

Bán tại quầy

**Đơn hàng**

Sản phẩm

Khách hàng

Kho

Thống kê

Quản lý

Đăng xuất

Danh sách đơn hàng

Quản Trị Viên Avatar

Tạo đơn hàng

Nhập tên hoặc mã SKU

×

Mã đơn hàng	Ngày tạo đơn	Ngày tạo đơn	Trạng thái	Tổng tiền
SON0003				
SON0002				
SON0001				

**Hình 4.8:** Thiết kế màn hình đơn hàng

Logo

Tổng quan

Bán tại quầy

Đơn hàng

**Sản phẩm**

Khách hàng

Kho

Thống kê

Quản lý

Đăng xuất

Danh sách sản phẩm

Quản Trị ViênAvatar

Thoát

Lưu

Thông tin chung	
Tên sản phẩm	Mã SKU
Nhãn hiệu	Tồn kho

Thuộc tính	
Thuộc tính 1	Kích thước
<div>+</div>	

Phiên bản	
-----------	--

Hình 4.9: Thiết kế màn hình sản phẩm

Logo

Tổng quan

Bán tại quầy

Đơn hàng

Sản phẩm

**Khách hàng**

Kho

Thống kê

Quản lý

Đăng xuất

Chi tiết khách hàng

Quản Trị ViênAvatar

Quay lại

Xóa khách hàng

Tạo phản hồi

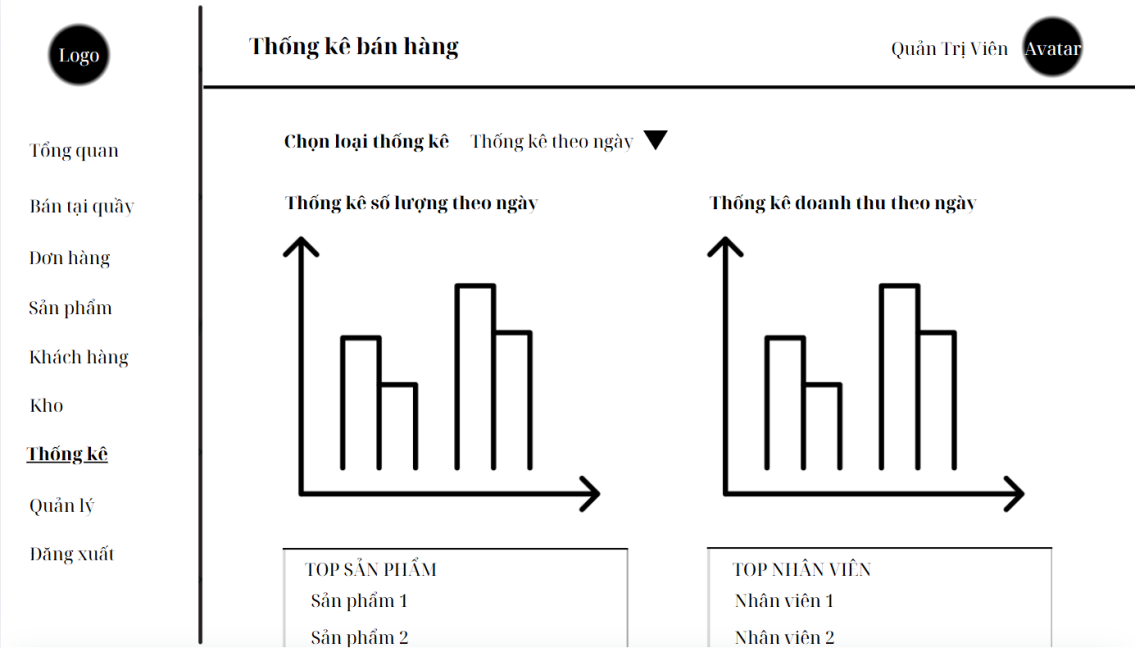
Thông tin cá nhân				
Tên khách hàng	Mã khách hàng			
Giới tính	Email			
Số điện thoại	Địa chỉ			

<b>Lịch sử mua hàng</b>		Lịch sử phản hồi		
Mã đơn hàng	Thời gian thanh toán	Giá trị	Số lượng sản phẩm	Nhân viên xử lý

Hình 4.10: Thiết kế màn hình khách hàng



Hình 4.11: Thiết kế màn hình kho hàng



Hình 4.12: Thiết kế màn hình thống kê

Logo

Tổng quan

Bán tại quầy

Đơn hàng

Sản phẩm

Khách hàng

Kho

Thống kê

**Quản lý**

Đăng xuất

Kế hoạch bán hàng

Quản Trị Viên Avatar

Tạo đơn chuyển kho

Tuần trước

Tuần sau

Mã NV	Họ tên	Vị trí	Thứ 2	Thứ 3	Thứ 4	Thứ 5	Thứ 6	Thứ 7	CN
A0004									
A0003									
A0002									
A0001									

Hình 4.13: Thiết kế màn hình kế hoạch bán hàng

## 4.2.2 Thiết kế lớp

### a, Thiết kế lớp TransferController

TransferController
+ getTransferList() : ResponseEntity<ResponseObject> + getTransferById(id : int) : ResponseEntity<ResponseObject> + createTransfer(transferDto : TransferDto) : ResponseEntity<ResponseObject> + updateTransfer(transferDto : TransferDto, int id : int) : ResponseEntity<ResponseObject>

Hình 4.14: Thiết kế lớp TransferController

**Attribute:** Không

**Operation:**

STT	Tên	Kiểu trả về	Mục đích
1	getTransferList	ResponseEntity<ResponseObject>	Lấy danh sách đơn chuyển hàng
2	getTransferById	ResponseEntity<ResponseObject>	Lấy đơn chuyển hàng theo ID
3	createTransfer	ResponseEntity<ResponseObject>	Tạo mới đơn chuyển hàng
4	updateTransfer	ResponseEntity<ResponseObject>	Cập nhật đơn chuyển hàng

Bảng 4.2: Các phương thức của lớp TransferController

**Parameter:**

- transferDto : Đối tượng chứa thông tin đơn hàng chuyển
- id : ID của đơn hàng chuyển

**b, Thiết kế lớp TransferService**

TransferService
+ findAll() : List<TransferDto> + findById(id : int) : TransferDto + createNewTransfer(transferDto : TransferDto) : TransferDto + updateTransfer(transferDto : TransferDto) : TransferDto + cancelTransfer(transferDto : TransferDto) : void

**Hình 4.15:** Thiết kế lớp TransferService

**Attribute:** Không

**Operation:**

STT	Tên	Kiểu trả về	Mục đích
1	findAll	List<TransferDto>	Lấy danh sách đơn chuyển hàng
2	findById	TransferDto	Lấy đơn chuyển hàng theo ID
3	createNewTransfer	TransferDto	Tạo mới đơn chuyển hàng
4	updateTransfer	TransferDto	Cập nhật đơn chuyển hàng
5	cancelTransfer	void	Hủy đơn chuyển hàng

**Bảng 4.3:** Các phương thức của lớp TransferService

**Parameter:**

- transferDto : Đối tượng chứa thông tin đơn hàng chuyển
- id : ID của đơn hàng chuyển

**c, Thiết kế lớp ProductController**

**Attribute:** Không

**Operation:**

ProductController
<pre> + getAllBaseProduct(page : int, size : int) : ResponseEntity&lt;ResponseObject&gt; + getBaseProductById(baseId : int) : ResponseEntity&lt;ResponseObject&gt; + createBaseProduct(baseProductDto : BaseProductDto) : ResponseEntity&lt;ResponseObject&gt; + updateBaseProduct(baseId : int, baseProduct : BaseProductDto) : ResponseEntity&lt;ResponseObject&gt; + deleteBaseProduct(baseId : int) : ResponseEntity&lt;ResponseObject&gt; </pre>

**Hình 4.16:** Thiết kế lớp ProductController

STT	Tên	Kiểu trả về	Mục đích
1	getAllBaseProduct	ResponseEntity<ResponseObject>	Lấy danh sách sản phẩm
2	getBaseProductById	ResponseEntity<ResponseObject>	Lấy sản phẩm theo ID
3	createBaseProduct	ResponseEntity<ResponseObject>	Tạo sản phẩm mới
4	updateBaseProduct	ResponseEntity<ResponseObject>	Cập nhật sản phẩm
5	deleteBaseProduct	ResponseEntity<ResponseObject>	Xóa sản phẩm và các phiên bản của sản phẩm

**Bảng 4.4:** Các phương thức của lớp ProductController

**Parameter:**

- baseProductDto : Đối tượng chứa thông tin sản phẩm
- baseId : ID của sản phẩm
- page : Số thứ tự trang
- size : Số lượng bản ghi mỗi trang

**d, Thiết kế lớp ProductService**

**Attribute:** Không

**Operation:**

ProductService
<pre> + countBaseProduct() : long + getAll(page : int, size : int) : List&lt;BaseProductDto&gt; + getBaseProductById(baseId : int) : BaseProductDto + createBaseProduct(baseProductDto : BaseProductDto) : BaseProductDto + updateBaseProduct(baseId : int, baseProduct : BaseProductDto) : BaseProductDto + deleteBaseProduct(baseId : int) : void </pre>

**Hình 4.17:** Thiết kế lớp ProductService

STT	Tên	Kiểu trả về	Mục đích
1	countBaseProduct	long	Đếm số lượng sản phẩm
2	getAll	List<BaseProductDto>	Lấy danh sách sản phẩm
3	createBaseProduct	BaseProductDto	Tạo sản phẩm mới
4	updateBaseProduct	BaseProductDto	Cập nhật sản phẩm
4	deleteBaseProduct	void	Xóa sản phẩm và các phiên bản của sản phẩm

**Bảng 4.5:** Các phương thức của lớp ProductService

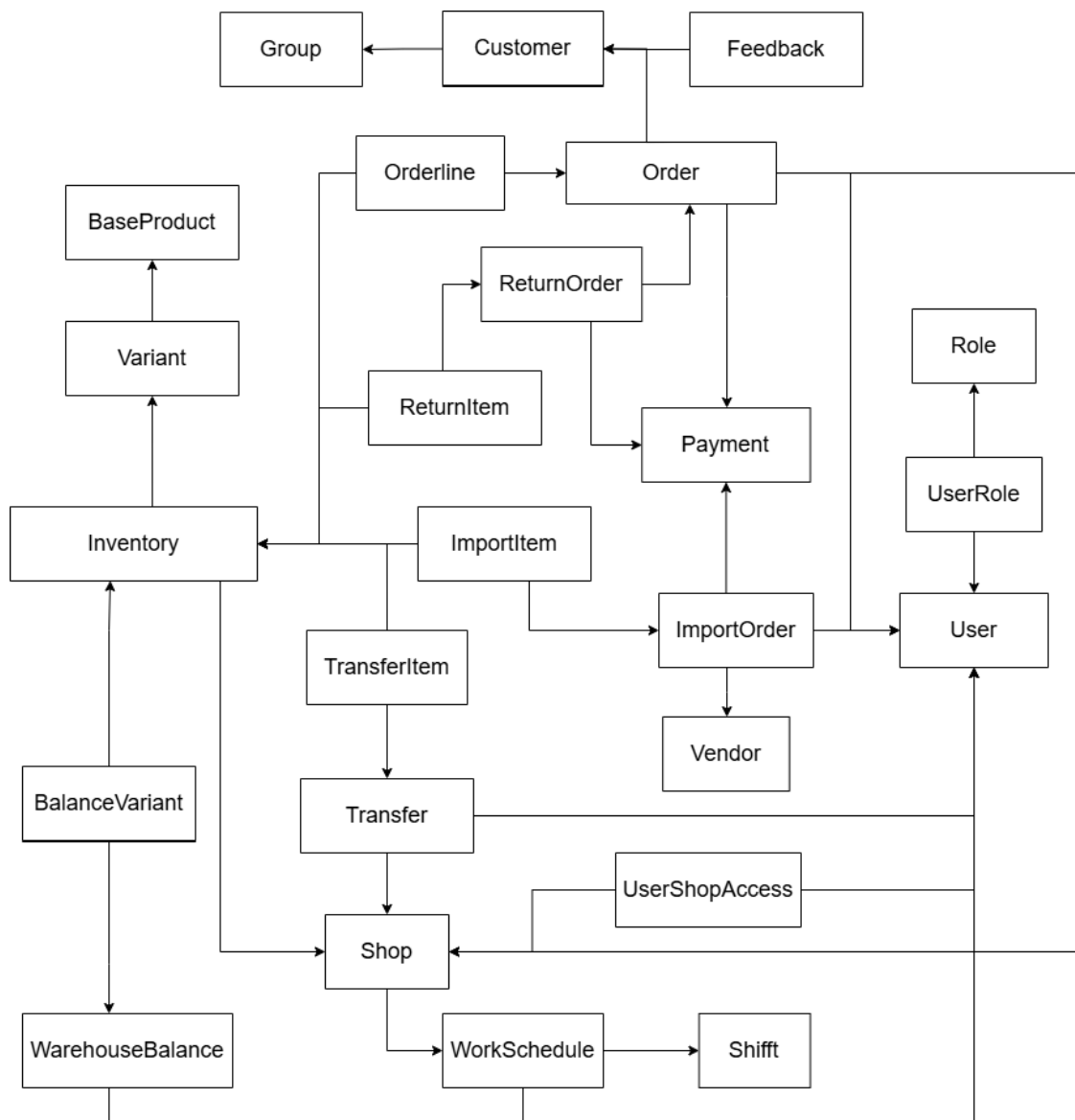
**Parameter:**

- baseProductDto : Đối tượng chứa thông tin sản phẩm
- baseId : ID của sản phẩm
- page : Số thứ tự trang
- size: Số lượng bản ghi mỗi trang



### 4.2.3 Thiết kế cơ sở dữ liệu

#### a, Sơ đồ thực thể liên kết



**Hình 4.18:** Sơ đồ thực thể liên kết

Sơ đồ thực thể liên kết trên cung cấp cái nhìn tổng quan về cấu trúc cơ bản của hệ thống, trong đó các thực thể và mối quan hệ giữa chúng được thể hiện rõ ràng.

Thực thể **Customer** đại diện cho khách hàng, đóng vai trò trung tâm trong hoạt động kinh doanh. Khách hàng là đối tượng chính tham gia vào các giao dịch mua sắm và do đó có mối liên kết chặt chẽ với thực thể **Order**, nơi lưu trữ thông tin chi tiết về các đơn hàng. Một đơn hàng, được định nghĩa bởi thực thể **Order**, có thể bao gồm nhiều mục hàng hóa, được quản lý thông qua thực thể **Orderline**. Mỗi dòng trong đơn hàng (**Orderline**) biểu thị một sản phẩm cụ thể mà khách hàng đặt mua, bao gồm số lượng, giá cả, và các thông tin chi tiết khác. Các sản phẩm này

được tổ chức và quản lý thông qua thực thể **Variant**, đại diện cho các biến thể khác nhau của cùng một sản phẩm cơ bản (**BaseProduct**). Điều này cho phép hệ thống phân biệt các sản phẩm dựa trên các thuộc tính như kích thước, màu sắc hoặc chất liệu, giúp đáp ứng linh hoạt nhu cầu đa dạng của khách hàng.

Bên cạnh đó, hệ thống còn hỗ trợ việc quản lý các đơn hàng trả lại thông qua hai thực thể **ReturnOrder** và **ReturnItem**. Đây là phần không thể thiếu, giúp xử lý các trường hợp khách hàng không hài lòng với sản phẩm và yêu cầu đổi trả. Các thông tin về hoàn tiền hoặc sản phẩm thay thế cũng được liên kết chặt chẽ với thực thể **Payment**, chịu trách nhiệm quản lý quá trình thanh toán và hoàn tiền.

Về phía nhà cung cấp, hệ thống quản lý việc nhập hàng thông qua các thực thể **ImportOrder** và **ImportItem**, thể hiện chi tiết từng đơn hàng nhập và các sản phẩm liên quan. Hai thực thể này được liên kết trực tiếp với thực thể **Vendor**, đại diện cho các nhà cung cấp. Thông qua các liên kết này, hệ thống đảm bảo rằng nguồn hàng hóa được kiểm soát chặt chẽ từ khâu cung cấp cho đến khi nhập kho.

Một trong những thành phần cốt lõi của hệ thống là quản lý kho hàng, được mô hình hóa thông qua thực thể **Inventory**. Thực thể này liên kết với các biến thể sản phẩm **Variant** cho phép hệ thống theo dõi số lượng tồn kho cụ thể cho từng biến thể sản phẩm. Bên cạnh đó, **WarehouseBalance** và **BalanceVariant** cung cấp khả năng cân bằng và điều chỉnh hàng tồn kho tại các kho hàng khác nhau. Để đảm bảo việc phân phối hàng hóa hiệu quả, hệ thống sử dụng thực thể **Transfer** và **TransferItem** để quản lý việc chuyển hàng giữa các kho hoặc từ kho đến các điểm bán lẻ.

Cửa hàng, được định nghĩa bởi thực thể **Shop**, đóng vai trò là trung tâm của các hoạt động kinh doanh. Lịch làm việc của nhân viên tại các cửa hàng được tổ chức thông qua thực thể **WorkSchedule**, giúp hệ thống lên kế hoạch và phân bổ công việc một cách hợp lý. Để đảm bảo phân quyền truy cập, thực thể **UserShopAccess** quản lý quyền truy cập của từng người dùng đối với từng cửa hàng cụ thể. Nhân viên làm việc trong hệ thống được định nghĩa thông qua thực thể **User**, liên kết với các vai trò (**Role**) thông qua thực thể **UserRole**. Đồng thời, các ca làm việc được tổ chức và quản lý chi tiết qua thực thể **Shift**.

Ngoài các thực thể cốt lõi, hệ thống còn cung cấp khả năng phân nhóm khách hàng thông qua thực thể **Group**, giúp cửa hàng dễ dàng triển khai các chương trình chăm sóc khách hàng hiệu quả. Cuối cùng, thực thể **Feedback** thu thập phản hồi từ khách hàng về các đơn hàng đã thực hiện, cung cấp dữ liệu quý giá để cải thiện chất lượng dịch vụ và sản phẩm.

**b, Xây dựng cơ sở dữ liệu**

Trong phạm vi chính của đề án sẽ mô tả các bảng **shop**, **user**, **work\_schedule**, **variant**, **order**, **import\_order**, **transfer**, **inventory**. Các bảng khác trên cơ sở dữ liệu của hệ thống được đặc tả ở phần phụ lục.

STT	Tên cột	Kiểu dữ liệu	Mô tả
1	id	int	Khoá chính, tự động tăng (Auto Increment), đại diện duy nhất cho từng cửa hàng.
2	created_at	datetime(6)	Thời gian tạo bản ghi.
3	updated_at	datetime(6)	Thời gian cập nhật bản ghi gần nhất.
4	address	varchar(255)	Địa chỉ cụ thể của cửa hàng.
5	avatar	varchar(255)	Đường dẫn tới ảnh đại diện của cửa hàng.
6	business	varchar(255)	Loại hình kinh doanh của cửa hàng.
7	city	varchar(255)	Thành phố nơi cửa hàng đặt địa chỉ.
8	district	varchar(255)	Quận/huyện nơi cửa hàng đặt địa chỉ.
9	email	varchar(255)	Địa chỉ email của cửa hàng, dùng cho liên lạc.
10	name	varchar(255)	Tên của cửa hàng.
11	phone	varchar(255)	Số điện thoại liên lạc của cửa hàng.
12	vat	double	Thuế giá trị gia tăng (VAT) áp dụng cho cửa hàng.
13	wards	varchar(255)	Phường/xã nơi cửa hàng đặt địa chỉ.
14	schedule_id	int	Khoá ngoại tham chiếu đến lịch làm việc của cửa hàng.

**Bảng 4.6:** Danh sách thuộc tính của bảng **shop**

Bảng **shop** lưu trữ thông tin của từng cửa hàng trong hệ thống. Mỗi cửa hàng có một mã định danh duy nhất **id** và các trường như tên cửa hàng, địa chỉ, số điện thoại, và loại hình kinh doanh. Thông tin về địa lý, bao gồm thành phố, quận/huyện, và phường/xã, giúp xác định vị trí của cửa hàng. Cùng với đó, bảng này cũng chứa trường **schedule\_id**, liên kết với bảng lịch làm việc, cho phép quản lý ca làm việc của cửa hàng và phân công công việc cho nhân viên.

STT	Tên cột	Kiểu dữ liệu	Mô tả
1	id	int	Khoá chính, tự động tăng (Auto Increment).
2	created_at	datetime(6)	Thời gian tạo bản ghi.
3	updated_at	datetime(6)	Thời gian cập nhật bản ghi gần nhất.
4	address	varchar(255)	Địa chỉ của người dùng.
5	dob	date	Ngày sinh của người dùng.
6	full_name	varchar(255)	Họ và tên đầy đủ của người dùng.
7	gender	enum	Giới tính của người dùng 'FEMALE', 'MALE', 'OTHER'.
8	is_active	tinyint(1)	Trạng thái hoạt động của người dùng.
9	password	varchar(255)	Mật khẩu của người dùng.
10	phone	varchar(255)	Số điện thoại của người dùng.
11	work_status	enum	Trạng thái công việc của người dùng 'QUIT', 'WORKING'.

**Bảng 4.7:** Danh sách thuộc tính của bảng **user**

Bảng **user** lưu trữ thông tin người dùng trong hệ thống. Các trường quan trọng bao gồm tên, trạng thái công việc **work\_status**, giới tính, và **is\_active**, giúp xác định người dùng có còn hoạt động trong hệ thống hay không. Trường **work\_status** cho biết người dùng đang làm việc hay đã nghỉ việc.

STT	Tên cột	Kiểu dữ liệu	Mô tả
1	id	int	Khoá chính, tự động tăng (Auto Increment).
2	created_at	datetime(6)	Thời gian tạo bản ghi.
3	updated_at	datetime(6)	Thời gian cập nhật bản ghi gần nhất.
4	date	datetime(6)	Ngày áp dụng lịch làm việc.
5	shift_id	int	Khoá ngoại tham chiếu đến bảng <b>shift</b> , xác định ca làm việc.
6	user_id	int	Khoá ngoại tham chiếu đến bảng <b>user</b> , xác định nhân viên được phân công.

**Bảng 4.8:** Danh sách thuộc tính của bảng **work\_schedule**

Bảng **work\_schedule** mô tả chi tiết các lịch làm việc của nhân viên trong cửa

hàng. Mỗi bản ghi trong bảng tương ứng với một ca làm việc của một nhân viên, được chỉ định thông qua các khoá ngoại **shift\_id** và **user\_id**. Trường **date** lưu trữ thông tin về ngày áp dụng lịch, trong khi các trường **created\_at** và **updated\_at** giúp quản lý lịch sử thay đổi của dữ liệu. Bảng này là phần cốt lõi trong việc tổ chức và quản lý ca làm việc, đảm bảo sự vận hành hiệu quả cho cửa hàng.

STT	Tên cột	Kiểu dữ liệu	Mô tả
1	id	int	Khoá chính, tự động tăng (Auto Increment).
2	created_at	datetime(6)	Thời gian tạo bản ghi.
3	updated_at	datetime(6)	Thời gian cập nhật bản ghi gần nhất.
4	barcode	varchar(255)	Mã vạch sản phẩm.
5	image	varchar(255)	Đường dẫn đến ảnh của sản phẩm.
6	import_price	int	Giá nhập sản phẩm.
7	is_deleted	tinyint(1)	Trạng thái xóa của sản phẩm.
8	name	varchar(255)	Tên sản phẩm.
9	quantity	int	Số lượng sản phẩm tồn kho.
10	retail_price	int	Giá bán lẻ của sản phẩm.
11	sku	varchar(255)	Mã SKU của sản phẩm.
12	value1	varchar(255)	Giá trị của thuộc tính 1.
13	value2	varchar(255)	Giá trị của thuộc tính 2.
14	value3	varchar(255)	Giá trị của thuộc tính 3.
15	wholesale_price	int	Giá bán sỉ của sản phẩm.
16	base_id	int	Mã sản phẩm cơ bản (nếu có).

**Bảng 4.9:** Danh sách thuộc tính của bảng **variant**

Bảng **variant** lưu trữ các biến thể của sản phẩm như màu sắc, kích thước và các đặc tính khác. Các sản phẩm có thể có nhiều biến thể, và bảng này giúp phân biệt chúng một cách rõ ràng. Mỗi biến thể sản phẩm được gán mã vạch (**barcode**) và có thể có giá bán sỉ khác nhau, lưu tại trường **wholesale\_price**. Thông qua bảng này, việc quản lý tồn kho và giá bán của từng biến thể sản phẩm trở nên dễ dàng hơn.

STT	Tên cột	Kiểu dữ liệu	Mô tả
1	id	int	Khoá chính, tự động tăng (Auto Increment).
2	created_at	datetime(6)	Thời gian tạo đơn hàng.
3	updated_at	datetime(6)	Thời gian cập nhật đơn hàng.
4	discount	int	Mức giảm giá của đơn hàng.
5	customer_id	int	Khoá ngoại tham chiếu đến bảng khách hàng.
6	shop_id	int	Khoá ngoại tham chiếu đến bảng cửa hàng.
7	person_in_charge	int	Người phụ trách đơn hàng.

**Bảng 4.10:** Danh sách thuộc tính của bảng **order**

Bảng **order** lưu trữ thông tin về đơn hàng của khách hàng, bao gồm sản phẩm, số lượng, giá trị và thông tin khách hàng. Trường **customer\_id** liên kết với bảng khách hàng, còn **discount** quản lý các chương trình giảm giá.

STT	Tên cột	Kiểu dữ liệu	Mô tả
1	id	int	Khoá chính, tự động tăng (Auto Increment).
2	created_at	datetime(6)	Thời gian tạo đơn nhập hàng.
3	updated_at	datetime(6)	Thời gian cập nhật đơn nhập hàng.
4	shipment_status	tinyint	Trạng thái vận chuyển của đơn nhập hàng.
5	person_in_charge	int	Người phụ trách đơn nhập hàng.
6	vendor_id	int	Khoá ngoại tham chiếu đến nhà cung cấp.

**Bảng 4.11:** Danh sách thuộc tính của bảng **import\_order**

Bảng **import\_order** ghi nhận các đơn nhập hàng từ nhà cung cấp, bao gồm thông tin người phụ trách, trạng thái vận chuyển và nhà cung cấp. Trường **shipment\_status** theo dõi tiến độ vận chuyển, còn **vendor\_id** xác định nhà cung cấp.

STT	Tên cột	Kiểu dữ liệu	Mô tả
1	id	int	Khoá chính, tự động tăng (Auto Increment).
2	created_at	datetime(6)	Thời gian tạo phiếu chuyển hàng.
3	updated_at	datetime(6)	Thời gian cập nhật phiếu chuyển hàng.
4	status	enum	Trạng thái của phiếu chuyển hàng 'CANCELLED', 'COMPLETED', 'PENDING'.
5	from_shop_id	int	Khoá ngoại tham chiếu đến cửa hàng xuất hàng.
6	to_shop_id	int	Khoá ngoại tham chiếu đến cửa hàng nhận hàng.
7	person_in_charge	int	Người phụ trách phiếu chuyển hàng.

**Bảng 4.12:** Danh sách thuộc tính của bảng **transfer**

Bảng **transfer** lưu trữ thông tin về các phiếu chuyển hàng giữa các cửa hàng, bao gồm cửa hàng xuất và cửa hàng nhận. Trường **status** cho biết tình trạng của phiếu, từ lúc chờ xử lý đến khi hoàn tất. Bảng này giúp theo dõi việc di chuyển sản phẩm chính xác và kịp thời.

STT	Tên cột	Kiểu dữ liệu	Mô tả
1	id	int	Khoá chính, tự động tăng (Auto Increment).
2	created_at	datetime(6)	Thời gian tạo bản ghi trong kho.
3	updated_at	datetime(6)	Thời gian cập nhật bản ghi trong kho.
4	quantity	int	Số lượng sản phẩm tồn kho tại cửa hàng.
5	shop_id	int	Khoá ngoại tham chiếu đến bảng cửa hàng.
6	variant_id	int	Khoá ngoại tham chiếu đến bảng sản phẩm (variant).

**Bảng 4.13:** Danh sách thuộc tính của bảng **inventory**

Bảng **inventory** theo dõi số lượng tồn kho tại cửa hàng. Trường **quantity** xác định số lượng sản phẩm hiện có, trong khi **shop\_id** và **variant\_id** liên kết với bảng cửa hàng và sản phẩm (variant) để xác định sản phẩm cụ thể. Bảng này giúp quản lý hàng hóa và đảm bảo không thiếu sản phẩm.

### 4.3 Xây dựng ứng dụng

#### 4.3.1 Thư viện và công cụ sử dụng

Mục đích	Công cụ/Thư viện	Địa chỉ URL
IDE lập trình	IntelliJ	<a href="https://www.jetbrains.com/idea/">https://www.jetbrains.com/idea/</a>
Thư viện lập trình frontend	ReactJS	<a href="https://react.dev/">https://react.dev/</a>
Thư viện hỗ trợ lập trình frontend	Material UI	<a href="https://mui.com">https://mui.com</a>
Framework backend	Spring Boot	<a href="https://spring.io/projects/spring-boot">https://spring.io/projects/spring-boot</a>
Kiểm thử API	Postman	<a href="https://www.postman.com/">https://www.postman.com/</a>
Tương tác với API	Axios	<a href="https://axios-http.com/">https://axios-http.com/</a>
Phân tích, thiết kế hệ thống	Astah UML	<a href="https://astah.net/">https://astah.net/</a>
ORM framework	Hibernate	<a href="https://hibernate.org/">https://hibernate.org/</a>
Thư viện lập trình giao diện người dùng	FontAwesome	<a href="https://fontawesome.com/">https://fontawesome.com/</a>
Quản lý trạng thái	Lodash	<a href="https://lodash.com/">https://lodash.com/</a>
Xử lý ngày giờ	Moment.js	<a href="https://momentjs.com/">https://momentjs.com/</a>
Thư viện thông báo	React Toastify	<a href="https://fkhadra.github.io/react-toastify/introduction">https://fkhadra.github.io/react-toastify/introduction</a>
Tạo mã QR thanh toán	VietQR	<a href="https://www.npmjs.com/package/vietqr">https://www.npmjs.com/package/vietqr</a>
Thư viện gửi OTP	Twilio	<a href="https://www.twilio.com/">https://www.twilio.com/</a>
Quản lý file Excel	XLSX	<a href="https://www.npmjs.com/package/xlsx">https://www.npmjs.com/package/xlsx</a>
Thư viện JWT	JWT	<a href="https://github.com/jwt/jwt">https://github.com/jwt/jwt</a>
Hỗ trợ kiểm tra	Jest	<a href="https://jestjs.io/">https://jestjs.io/</a>
Phân tích và hiển thị biểu đồ	Recharts	<a href="https://recharts.org/">https://recharts.org/</a>
Xử lý ảnh trên cloud	Cloudinary	<a href="https://cloudinary.com/">https://cloudinary.com/</a>

**Bảng 4.14:** Danh sách thư viện và công cụ sử dụng

#### 4.3.2 Kết quả đạt được

Hệ thống quản lý bán hàng POS cho mô hình bán lẻ lưu động đã đáp ứng được các yêu cầu cơ bản như: hỗ trợ người dùng có thể xem và tìm kiếm các sản phẩm có sẵn tại cửa hàng hoặc các điểm bán lưu động, cho phép nhân viên thêm sản phẩm vào giỏ hàng và thực hiện ghi nhận giao dịch bán hàng. Hệ thống hỗ trợ theo dõi tình trạng tồn kho, nhập và luân chuyển hàng hóa giữa kho cố định và các điểm bán lưu động. Ngoài ra, hệ thống cũng hỗ trợ quản lý thông tin hàng hóa, nhà cung cấp, khách hàng, nhân viên. Với các biểu đồ tổng quan, hệ thống giúp chủ cửa hàng nắm bắt hiệu quả hoạt động kinh doanh và quản lý hệ thống một cách hiệu quả hơn.



Dưới đây là bảng thống kê thông tin của hệ thống cung cấp các thông tin chi tiết về hệ thống như số file mã nguồn, số bảng cơ sở dữ liệu, dung lượng mã nguồn:

Thông số	Chi tiết
Tổng số file mã nguồn	424
Số bảng cơ sở dữ liệu	25
Dung lượng mã nguồn	21 MB

**Bảng 4.15:** Thống kê thông tin hệ thống

### 4.3.3 Minh họa các chức năng chính

#### a, Giao diện bán hàng

The screenshot displays the 'Fashion Shop' sales interface. It features a search bar at the top with the text 'Thêm mỗi sản phẩm vào đơn'. Below the search bar is a table with columns: STT, Ảnh SP, Mã SKU, Tên sản phẩm, Số lượng, Đơn giá, and Thành tiền. The table contains four rows of product data. To the right of the table is a summary panel with a search bar 'Thêm khách hàng vào đơn' and a plus icon. The summary panel lists: Tổng tiền: (4 sản phẩm) 3,700,000, VAT(10%): 370,000, Chiết khấu: 100,000, KHÁCH PHẢI TRẢ: 3,970,000, Tiền khách đưa: 4,000,000, Tiền thừa trả khách: 30,000. At the bottom of the summary panel is a dropdown menu for 'Hình thức thanh toán' with 'Tiền mặt' selected. At the bottom of the interface are two buttons: 'IN HÓA ĐƠN' and 'THANH TOÁN'.

STT	Ảnh SP	Mã SKU	Tên sản phẩm	Số lượng	Đơn giá	Thành tiền
1		SKU4100010	Đồng hồ nam - PB1	1	2,000,000	2,000,000
2		SKU2100005	Giày thể thao nam - PB1	2	400,000	800,000
3		SKU1100003	Áo thun nam - PB2	2	300,000	600,000
4		SKU1100002	Áo thun nam - PB1	1	300,000	300,000

Summary Panel:

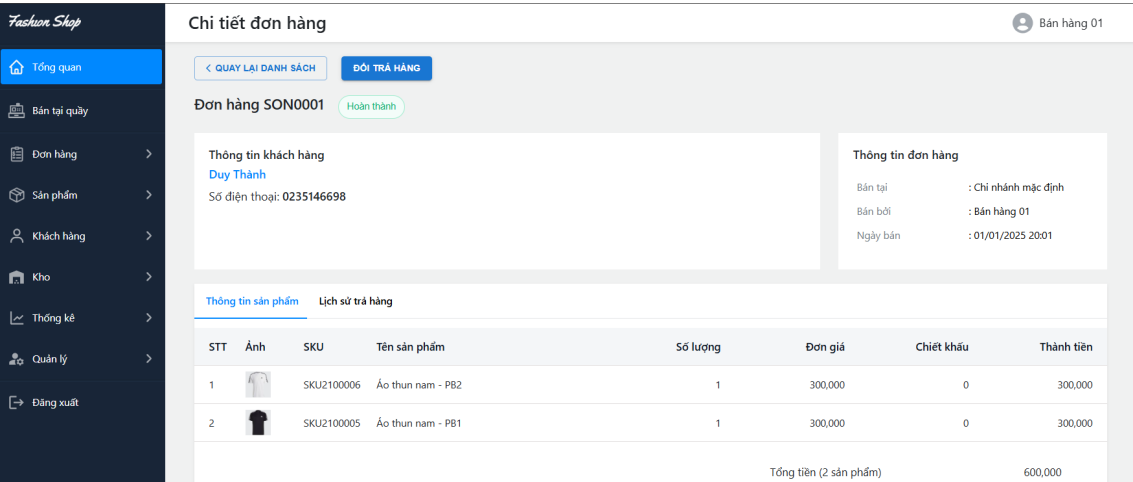
- Tổng tiền: (4 sản phẩm) 3,700,000
- VAT(10%): 370,000
- Chiết khấu: 100,000
- KHÁCH PHẢI TRẢ: 3,970,000**
- Tiền khách đưa: 4,000,000
- Tiền thừa trả khách: 30,000
- Hình thức thanh toán: Tiền mặt

Buttons: IN HÓA ĐƠN, THANH TOÁN

**Hình 4.19:** Giao diện bán hàng

Trên đây là màn hình bán hàng, màn hình này hiển thị khi người bán hàng thực hiện tạo đơn bán hàng tại quầy hàng. Ở màn hình này người bán hàng có thể thêm, hoặc loại bỏ sản phẩm cho đơn hàng. Sản phẩm trong đơn sẽ bao gồm các thông tin như tên, hình ảnh, số lượng, giá bán. Ngoài ra người bán hàng có thể thêm chiết khấu, chọn phương thức thanh toán và chọn lưu, xuất hóa đơn để hoàn thành giao dịch.

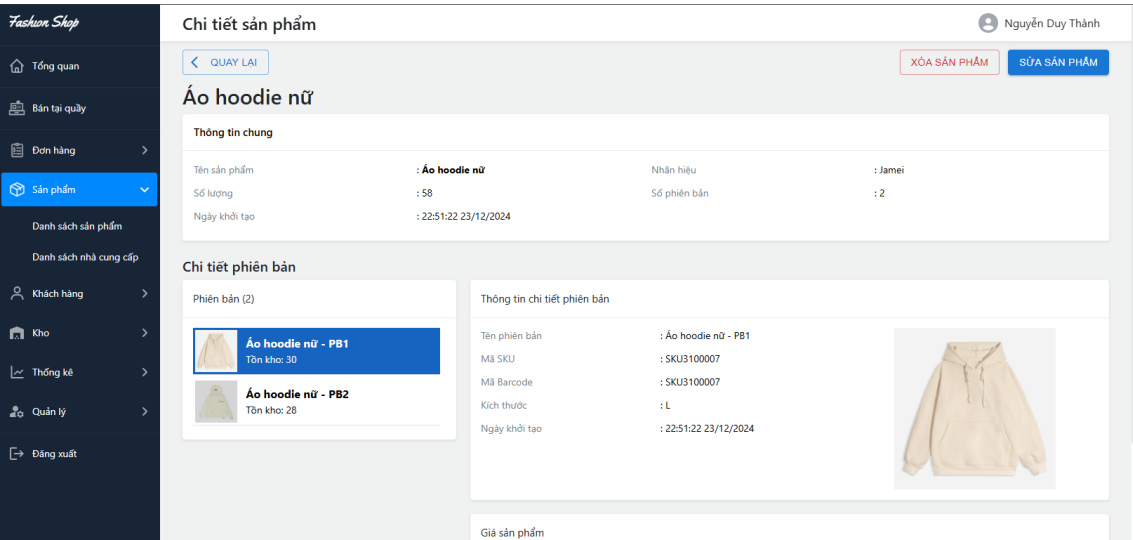
b, Giao diện chi tiết đơn hàng



Hình 4.20: Giao diện chi tiết đơn hàng

Đây là giao diện chi tiết đơn hàng, hiển thị đầy đủ thông tin về đơn hàng như mã đơn, trạng thái, thông tin khách hàng (tên, số điện thoại), nhân viên bán hàng, thời gian giao dịch và chi tiết các sản phẩm trong đơn (hình ảnh, mã SKU, tên, số lượng, đơn giá, chiết khấu, thành tiền). Giao diện cũng cung cấp tổng tiền đơn hàng và các chức năng như Đổi trả hàng và Xem lịch sử trả hàng, hỗ trợ nhân viên quản lý đơn hàng một cách toàn diện và hiệu quả.

c, Giao diện chi tiết sản phẩm

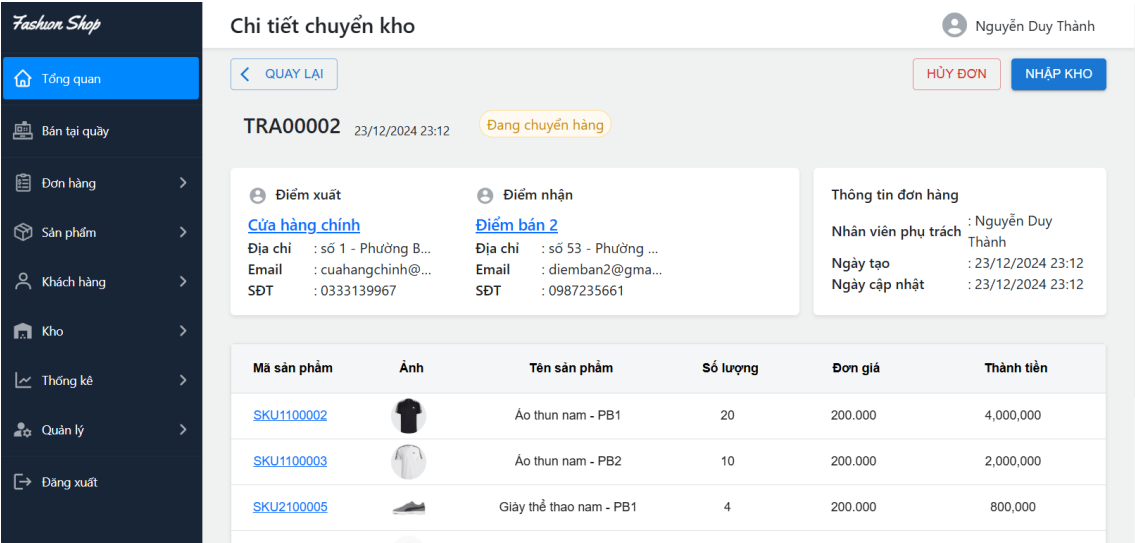


Hình 4.21: Giao diện chi tiết sản phẩm

Đây là giao diện màn hình xem chi tiết sản phẩm, nó sẽ hiển thị ra khi người dùng click vào sản phẩm. Ở màn hình này người dùng có thể xem được thông tin chi tiết của sản phẩm, bao gồm các thông tin: tên, giá sản phẩm, thương hiệu, kích

thước, số lượng và mô tả của sản phẩm. Ở màn hình này người dùng có thể chọn sửa thông tin sản phẩm khi chọn nút Sửa sản phẩm.

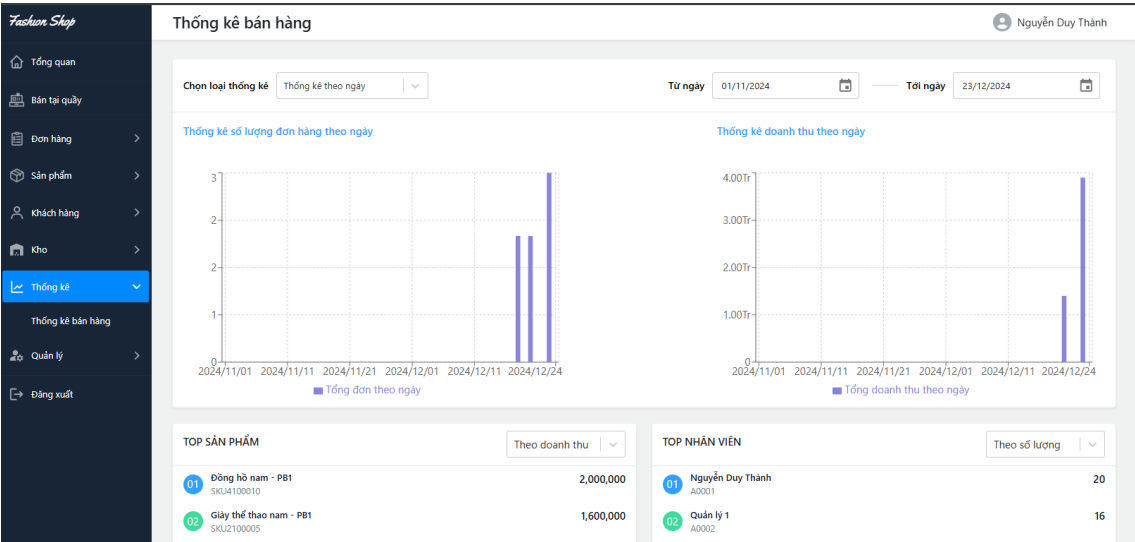
d, Giao diện luân chuyển hàng hóa



Hình 4.22: Giao diện luân chuyển hàng hóa

Đây là giao diện luân chuyển hàng hóa, sau khi quản lý tạo đơn xuất, hệ thống sẽ tạo đơn chuyển hàng giữa các điểm bán cho phép theo dõi trạng thái đơn, thông tin các sản phẩm cần chuyển hàng, các kho xuất và nhận hàng. Quản lý có thể xác nhận nhận hàng sau khi đã nhập hàng vào kho nhận bằng cách nhấn Nhập kho.

e, Giao diện thống kê



Hình 4.23: Giao diện thống kê

Đây là giao diện thống kê, chủ cửa hàng có thể lựa chọn theo dõi thống kê theo thời gian tùy chọn hoặc theo bộ lọc ngày tháng năm. Đơn vị thống kê tập trung vào doanh thu và số lượng đơn đặt hàng. Dựa theo kết quả thống kê, chủ cửa hàng

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

sẽ nắm bắt được các mùa mua sắm cũng như lưu lượng mua sắm tại các thời gian trong năm, từ đó đưa ra kế hoạch phân bổ nhân lực và bổ sung hàng hóa cần thiết.

f, Giao diện quản lý nhân viên

Fashion Shop

Tổng quan

Bán tại quầy

Đơn hàng

Sản phẩm

Khách hàng

Kho

Thống kê

Quản lý

Thông tin chung

Kế hoạch bán hàng

Đăng xuất

Thông tin nhân viên

QUAY LẠI

XÓA

LƯU

Quản lý 22

Đang làm việc

Thông tin nhân viên

Đổi mật khẩu

Họ và tên nhân viên \*

Quản lý 22

Số điện thoại \*

0321546987

Email

Nhập địa chỉ email

Vai trò nhân viên \*

Quản lý

Cửa hàng phụ trách \*

Điểm bán 2

Thông tin bổ sung

Ngày sinh

29/12/2024

Giới tính

Khác

Địa chỉ

Nhập địa chỉ

Hình 4.24: Giao diện quản lý nhân viên

Đây là giao diện quản lý nhân viên cung cấp đầy đủ thông tin cá nhân và công việc của một nhân viên. Giao diện hiển thị chi tiết các trường thông tin bao gồm họ tên, số điện thoại, vai trò nhân viên, cửa hàng phụ trách, và email (nếu có). Ngoài ra, chủ của hàng có thể thay đổi mật khẩu và bổ sung các thông tin như ngày sinh, giới tính và địa chỉ. Giao diện cũng hỗ trợ xóa nhân viên để chấm dứt hoạt động các tài khoản nhân viên không còn làm việc.

g, Giao diện kế hoạch bán hàng

Fashion Shop

Tổng quan

Bán tại quầy

Đơn hàng

Sản phẩm

Khách hàng

Kho

Thống kê

Quản lý

Thông tin chung

Kế hoạch bán hàng

Đăng xuất

Kế hoạch bán hàng

Chủ của hàng

Lịch ca làm việc từ 23/12/2024 đến 29/12/2024

Tạo mới kế hoạch

Tháng: 12 Năm: 2024

Tuần trước

Tuần sau

Chi tiết ca làm việc

Nhân viên: Bán hàng 11

Cửa hàng: Điểm bán 1

Vị trí: SALE

Ngày: 24/12/2024

Ca làm việc: KIP01

Chi tiết: 7:00-14:30

Đóng

TT	Mã NV	Họ và tên	24/12	25/12	26/12	27/12	28/12	29/12
1	A001	Chủ cửa hàng	KIP02	KIP01	KIP02	KIP01	KIP02	N
2	A002	Quản lý 01	KIP01	KIP01	KIP01	KIP01	KIP01	N
3	A003	Bán hàng 01	KIP01	KIP01	KIP01	KIP01	KIP01	N
4	A004	Bán hàng 02	KIP02	KIP02	KIP02	KIP02	KIP02	N
5	A005	Quản lý 11	KIP01	KIP01	KIP01	KIP01	KIP01	N
6	A006	Bán hàng 11	KIP01	KIP01	KIP01	KIP01	KIP01	N
7	A007	Bán hàng 12	KIP02	KIP02	KIP02	KIP02	KIP02	N
8	A008	Quản lý 21	KIP01	KIP01	KIP01	KIP01	KIP01	N
9	A009	Bán hàng 21	KIP01	KIP01	KIP01	KIP01	KIP01	N
10	A0010	Bán hàng 22	KIP02	KIP02	KIP02	KIP02	KIP02	N
11	A0011	Quản lý 02	KIP02	KIP02	KIP02	KIP02	KIP02	N

Hình 4.25: Giao diện kế hoạch bán hàng

Đây là giao diện kế hoạch bán hàng, chủ cửa hàng có thể theo dõi kế hoạch bán hàng bao gồm các ca làm của từng nhân viên theo từng tuần tại các thời điểm theo

cần theo dõi. Ngoài ra chủ cửa hàng có thể tạo kế hoạch bán hàng cho tuần tiếp theo khi nhấn nút tạo mới kế hoạch. Hệ thống cho phép tải lên file excel bản kế hoạch theo mẫu có sẵn.

#### 4.4 Kiểm thử

##### 4.4.1 Kiểm tra chức năng bán hàng

**Bảng 4.16:** Kết quả kiểm thử chức năng bán hàng

STT	Các trường hợp	Đầu vào	Kết quả mong muốn	Kết quả thực tế	Kết luận
1	Không chọn sản phẩm khi bán hàng	1. Chọn bán hàng 2. Không chọn sản phẩm nào ở giỏ hàng 3. Chọn Thanh toán	Hệ thống hiển thị thông báo Hãy thêm sản phẩm vào đơn hàng	Hệ thống hiển thị thông báo Hãy thêm sản phẩm vào đơn hàng	Đạt
2	Giảm số lượng sản phẩm trong đơn hàng	1. Chọn sản phẩm 2. Chọn giảm số lượng sản phẩm	Hệ thống hiển thị số lượng sản phẩm giảm	Hệ thống hiển thị số lượng sản phẩm giảm	Đạt
3	Xóa sản phẩm trong đơn hàng	1. Chọn sản phẩm 2. Chọn xóa sản phẩm trong giỏ hàng	Hệ thống hiển thị thông báo đã xóa sản phẩm khỏi đơn	Hệ thống hiển thị thông báo đã xóa sản phẩm khỏi đơn	Đạt
4	Tăng số lượng sản phẩm trong đơn hàng lên nhiều hơn số lượng sản phẩm hiện có	1. Chọn sản phẩm 2. Chọn tăng số lượng sản phẩm đến khi số lượng sản phẩm nhiều hơn số lượng sản phẩm hiện có	Hệ thống hiển thị thông báo Không thể tăng số lượng sản phẩm	Hệ thống hiển thị thông báo Không thể tăng số lượng sản phẩm	Đạt
5	Tạo đơn bán hàng thành công	1. Chọn sản phẩm 2. Nhập tiền khách đưa 3. Chọn In hóa đơn 4. Chọn Thanh toán	Hệ thống hiển thị thông báo Thanh toán thành công	Hệ thống hiển thị thông báo Thanh toán thành công	Đạt

#### 4.4.2 Kiểm tra chức năng luân chuyển hàng hóa

**Bảng 4.17:** Kết quả kiểm thử chức năng luân chuyển hàng hóa

STT	Các trường hợp	Đầu vào	Kết quả mong muốn	Kết quả thực tế	Kết luận
1	Không chọn địa chỉ xuất/nhập hàng	1. Chọn Tạo đơn chuyển hàng 2. Không chọn địa chỉ xuất/nhập hàng 3. Chọn Tạo đơn	Hệ thống hiển thị thông báo Chưa chọn địa chỉ xuất/nhập hàng	Hệ thống hiển thị thông báo Chưa chọn địa chỉ xuất/nhập hàng	Đạt
2	Không chọn sản phẩm	1. Chọn Tạo đơn chuyển hàng 2. Chọn địa chỉ xuất/nhập hàng 3. Không chọn sản phẩm chuyển hàng 4. Chọn Tạo đơn	Hệ thống hiển thị thông báo Chưa chọn sản phẩm	Hệ thống hiển thị thông báo Chưa chọn sản phẩm	Đạt
3	Xóa sản phẩm trong đơn hàng	1. Chọn sản phẩm 2. Chọn xóa sản phẩm trong giỏ hàng	Hệ thống hiển thị sản phẩm trong đơn đã bị xóa	Hệ thống hiển thị sản phẩm trong đơn đã bị xóa	Đạt
4	Tăng số lượng sản phẩm trong đơn hàng lên nhiều hơn số lượng sản phẩm hiện có	1. Chọn sản phẩm 2. Chọn tăng số lượng sản phẩm đến khi số lượng sản phẩm nhiều hơn số lượng sản phẩm hiện có	Hệ thống hiển thị thông báo Không thể tăng số lượng sản phẩm	Hệ thống hiển thị thông báo Không thể tăng số lượng sản phẩm	Đạt
5	Tạo đơn chuyển hàng thành công	1. Chọn Tạo đơn chuyển hàng 2. Chọn địa chỉ xuất/nhập hàng 3. Chọn Sản phẩm 4. Chọn Tạo đơn	Hệ thống hiển thị thông báo Tạo đơn thành công	Hệ thống hiển thị thông báo Tạo đơn thành công	Đạt

#### 4.5 Triển khai

Dưới đây là bảng thống kê thông tin của hệ thống cung cấp các thông tin chi tiết về hệ thống được sử dụng trong quá trình phát triển, bao gồm URL của backend và

frontend, cùng với các thông số phần cứng như hệ điều hành, CPU và RAM.

Thông số	Chi tiết
URL API	http://localhost:8000
URL UI	http://localhost:3000
Hệ điều hành	Windows 11
CPU	Intel Core i5
RAM	20GB

**Bảng 4.18:** Thông số cấu hình

## **CHƯƠNG 5. CÁC GIẢI PHÁP VÀ ĐÓNG GÓP NỔI BẬT**

### **5.1 Quản lý luân chuyển hàng hóa giữa các điểm bán**

#### **5.1.1 Đặt vấn đề**

Trong mô hình bán lẻ lưu động, việc quản lý luân chuyển hàng hóa giữa các điểm bán đóng vai trò quan trọng trong việc duy trì hoạt động liên tục và hiệu quả của hệ thống bán hàng. Một trong những thách thức lớn nhất trong việc quản lý luân chuyển hàng hóa chính là đảm bảo rằng các điểm bán lưu động luôn có đủ hàng hóa để phục vụ khách hàng mà không gây ra tình trạng thiếu hàng. Trong quá trình đó, mỗi điểm bán cần phải có đủ thông tin về số lượng, giá trị hàng hóa, người phụ trách và thông tin các điểm bán xuất nhập hàng hỗ trợ luân chuyển hàng hóa đúng và đủ.

Hệ thống cũng cần xây dựng quy trình luân chuyển hàng giữa các điểm bán, nhằm đảm bảo tính linh hoạt và khả năng đáp ứng kịp thời khi có sự thay đổi trong nhu cầu bán hàng tại các điểm bán. Quy trình luân chuyển hàng được xác định rõ ràng không chỉ đảm bảo hàng hóa được chuyển đến đúng địa điểm và đảm bảo số lượng mà giúp việc cung cấp sản phẩm tới khách hàng trở nên hiệu quả hơn, đồng thời cung cấp một hệ thống theo dõi giúp các nhân viên quản lý dễ dàng nắm bắt tình trạng của đơn hàng.

Một vấn đề nữa là việc theo dõi, cập nhật tồn kho sau các giao dịch luân chuyển hàng giữa kho và các điểm bán lưu động. Các giao dịch này cần được ghi nhận đầy đủ và chính xác trong cơ sở dữ liệu, từ khi hàng hóa được xuất kho cho đến khi được nhập kho tại điểm bán. Điều này giúp giảm thiểu các sai sót trong quá trình quản lý kho.

#### **5.1.2 Giải pháp**

##### **a, Truy xuất dữ Liệu khi tạo đơn luân chuyển hàng**

Khi tạo một đơn luân chuyển hàng, hệ thống cần phải thu thập và xác minh nhiều loại dữ liệu từ các nguồn khác nhau để đảm bảo tính hợp lệ và sự chính xác trong quá trình xử lý. Quá trình này bắt đầu bằng việc thu thập thông tin từ các kho và sản phẩm có liên quan.

Đầu tiên, thông tin về người quản lý tạo đơn cũng cần được xác minh. Hệ thống cần biết ai là người tạo đơn để xác định quyền hạn và trách nhiệm trong quá trình xử lý đơn luân chuyển hàng. Dữ liệu này được lấy từ bảng người dùng trong hệ thống, nơi chứa thông tin về các vai trò và quyền hạn của từng nhân viên.

Tiếp theo, hệ thống cần lấy thông tin về các kho xuất và kho nhận. Việc xác định



kho xuất và kho nhận là rất quan trọng, vì chúng sẽ quyết định hướng di chuyển của hàng hóa. Các thông tin này được lấy từ bảng dữ liệu chứa thông tin về các điểm bán hàng hoặc kho trong hệ thống. Các kho có thể là kho cố định hoặc các điểm bán hàng lưu động, và mỗi điểm bán hay kho sẽ có các dữ liệu liên quan như tên kho, địa chỉ, và mã kho.

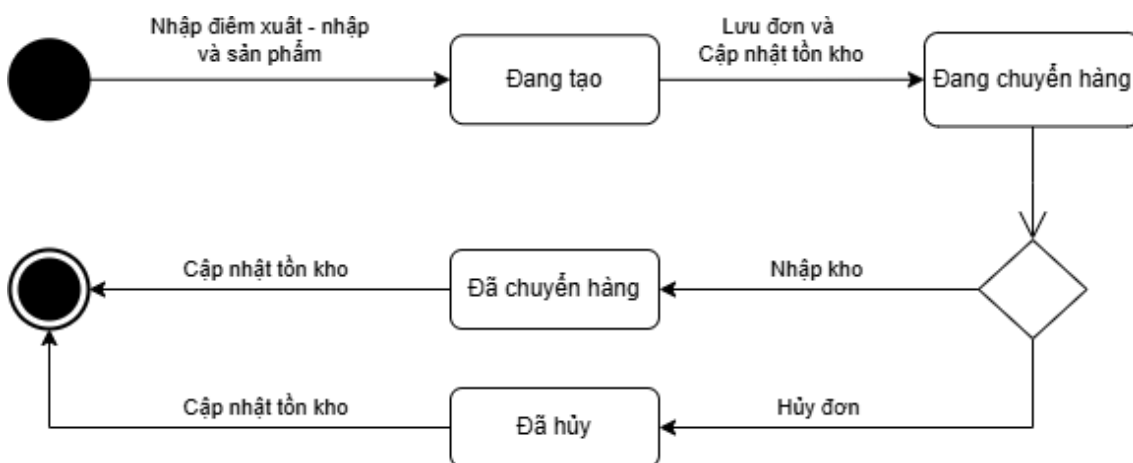
STT	Ảnh	Mã SKU	Tên sản phẩm	Số lượng nhập	Đơn giá	Chiết khấu	Thành tiền
1		SKU3100008	Áo hoodie nữ - PB2	10	200,000	0 %	2,000,000
2		SKU3100007	Áo hoodie nữ - PB1	10	200,000	0 %	2,000,000
3		SKU1100003	Áo thun nam - PB2	20	200,000	0 %	4,000,000
4		SKU1100002	Áo thun nam - PB1	20	200,000	0 %	4,000,000

**Hình 5.1:** Giao diện tạo đơn chuyển hàng

Cuối cùng, hệ thống cần lấy thông tin về sản phẩm sẽ được chuyển. Điều này bao gồm các chi tiết như tên sản phẩm, mã sản phẩm, và số lượng tồn kho. Thông tin về các sản phẩm này được lấy từ bảng dữ liệu sản phẩm hoặc bảng tồn kho. Dữ liệu này giúp xác định số lượng sản phẩm có sẵn để chuyển đi, từ đó đảm bảo rằng không có sự thiếu hụt trong kho xuất khi đơn chuyển hàng được tạo ra.

Tất cả các dữ liệu trên sẽ được thu thập và xác minh trước khi đơn luân chuyển hàng được tạo, đảm bảo rằng các thông tin này là chính xác và hợp lệ.

### b, Quá trình thay đổi trạng thái của đơn luân chuyển hàng



**Hình 5.2:** Sơ đồ chuyển đổi trạng thái

## CHƯƠNG 5. CÁC GIẢI PHÁP VÀ ĐÓNG GÓP NỔI BẬT

Trong suốt quá trình xử lý đơn luân chuyển hàng, đơn này sẽ trải qua nhiều trạng thái khác nhau. Mỗi trạng thái phản ánh một giai đoạn cụ thể trong quy trình chuyển hàng, từ khi đơn được tạo cho đến khi giao dịch hoàn tất.

STT	Tên TT	Mô tả	Trạng thái có thể chuyển sang
1	Đang tạo	Đơn hàng đang được tạo	Trạng thái 2 (Đang chuyển hàng)
2	Đang chuyển hàng	Đơn hàng đã được tạo và đang chuyển hàng từ kho của điểm xuất.	Trạng thái 3 (Đã chuyển hàng) hoặc Trạng thái 4 (Đã hủy)
3	Đã chuyển hàng	Đơn hàng đã được chuyển thành công tới điểm đến.	Không thể chuyển đến trạng thái khác
4	Đã hủy	Đơn hàng bị hủy do sự cố phát sinh.	Không thể chuyển đến trạng thái khác

**Bảng 5.1:** Mô tả các trạng thái chuyển hàng và quy tắc chuyển đổi

Fashion Shop

Tổng quan

Bán tại quầy

Đơn hàng

Sản phẩm

Khách hàng

Kho

Thống kê

Quản lý

Đăng xuất

Tạo chuyển kho

QUAY LẠI

TAO ĐƠN

Thông tin bổ sung

Cửa hàng xuất: Cửa hàng chính

Cửa hàng nhập: Điểm bán 1

Người phụ trách: Nguyễn Duy Thành

Địa chỉ: số 1

Địa chỉ: số 29

Bắt đầu tạo: 28/12/2024 22:31

Số điện thoại: 0333139967

Số điện thoại: 0325617744

Bắt đầu tạo: 28/12/2024 22:31

Email: cuahangchinh@example.com

Email: diemban1@gmail.com

Trạng thái: Đang tạo

Thông tin sản phẩm

Thêm mới sản phẩm

Đồng hồ nam - PB1  
SKU4100010  
1000000  
Tồn kho: 2

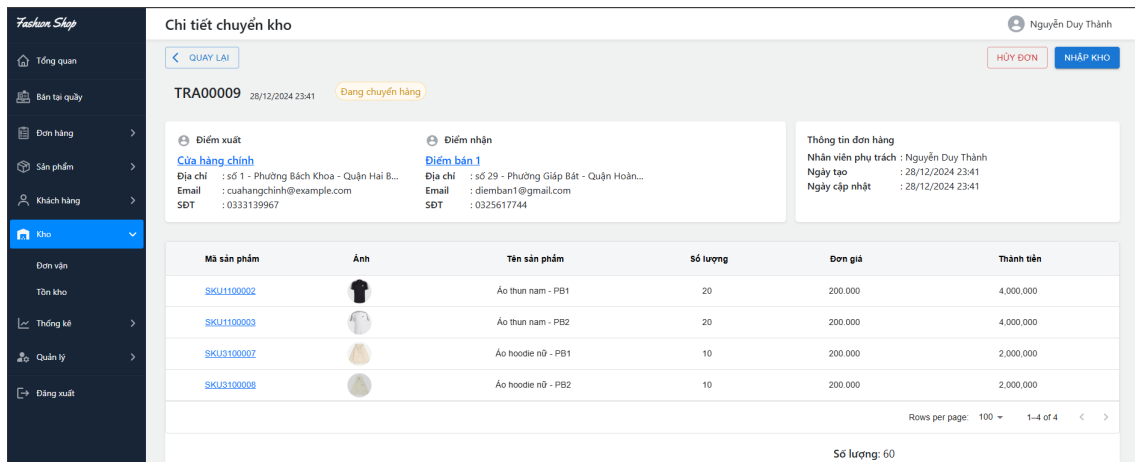
Áo hoodie nữ - PB2  
SKU3100008  
200000  
Tồn kho: 15

hiết khấu

Thành tiền

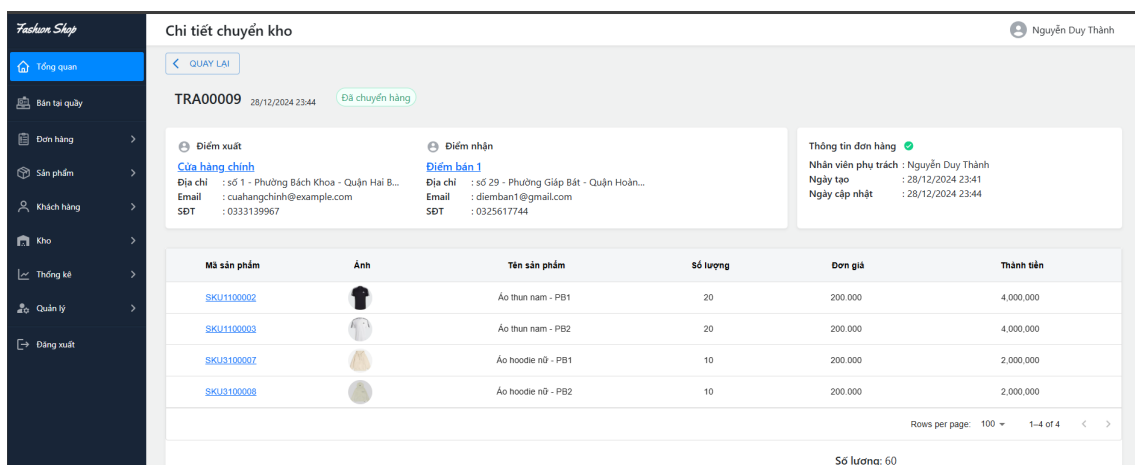
**Hình 5.3:** Giao diện đơn khi đang tạo

**Đang tạo:** Ban đầu, khi đơn luân chuyển hàng đang được tạo, trạng thái của nó sẽ là "Đang tạo". Đây là trạng thái ban đầu, khi đơn đang được tạo. Trạng thái này thể hiện rằng hệ thống đang truy xuất thông tin và quản lý đang nhập các thông tin cần thiết cho đơn hàng.



**Hình 5.4:** Giao diện đơn khi đang chuyển hàng

**Đang chuyển hàng:** Khi đơn đã được tạo và hàng hóa được xuất kho, dữ liệu trong bảng **Transaction** sẽ ghi lại trạng thái của đơn "Đang chuyển hàng". Trạng thái này cho thấy rằng đơn hàng đã được ghi nhận và các bước cần thiết để chuyển hàng đã được bắt đầu, hàng hóa đang trên đường đi. Trong quá trình này, dữ liệu về số lượng hàng hóa trong bảng **Inventory** sẽ được trừ đi từ kho xuất. Sự thay đổi này giúp đảm bảo rằng thông tin về tồn kho tại kho xuất luôn chính xác.



**Hình 5.5:** Giao diện đơn khi đã chuyển hàng

**Đã chuyển hàng:** Khi hàng hóa đã được giao và nhập kho tại kho nhận, hệ thống sẽ cập nhật lại trạng thái của đơn thành "Đã chuyển hàng" trong bảng **Transaction**. Trạng thái này xác nhận rằng hàng hóa đã đến nơi và được nhập kho thành công. Đây là trạng thái kết thúc quá trình chuyển hàng, và đơn sẽ được đóng lại. Lúc này, dữ liệu trong bảng **Inventory** cũng sẽ được cập nhật, tăng số lượng hàng tại kho nhận theo số lượng hàng hóa của đơn vận.

Fashion Shop

Tổng quan

Bán tại quầy

Đơn hàng

Sản phẩm

Khách hàng

Kho

Đơn vận

Tồn kho

Thống kê

Quản lý

Đăng xuất

Chi tiết chuyển kho

Nguyễn Duy Thành

QUAY LẠI

TRA00004

23/12/2024 23:48

Đã hủy

Điểm xuất

Điểm bán 1

Địa chỉ : số 29 - Phường Giáp Bát - Quận Hoàn...

Email : diemban1@gmail.com

SĐT : 0325617744

Điểm nhận

Điểm bán 2

Địa chỉ : số 53 - Phường Khương Mai - Quận T...

Email : diemban2@gmail.com

SĐT : 0987235661

Thông tin đơn hàng

Nhân viên phụ trách : Nguyễn Duy Thành

Ngày tạo : 23/12/2024 23:48

Ngày cập nhật : 23/12/2024 23:48

Mã sản phẩm	Ảnh	Tên sản phẩm	Số lượng	Đơn giá	Thành tiền
SKU1100002		Áo thun nam - PB1	5	200.000	1.000.000
SKU1100003		Áo thun nam - PB2	5	200.000	1.000.000
SKU2100007		Áo hoodie nữ - PB1	5	200.000	1.000.000
SKU2100008		Áo hoodie nữ - PB2	5	200.000	1.000.000

Rows per page: 100 1-4 of 4

Số lượng: 20

Hình 5.6: Giao diện đơn khi đã hủy

**Đã hủy:** Tuy nhiên, trong một số trường hợp, nếu có bất kỳ vấn đề nào phát sinh trong quá trình giao hàng, quản lý có thể điều chỉnh trên hệ thống để cập nhật trạng thái của đơn thành "Đã hủy" trong bảng **Transaction** và tất cả các giao dịch liên quan sẽ được hủy bỏ. Đây là trạng thái cuối cùng trong quá trình theo dõi đơn, phản ánh rằng đơn không còn hiệu lực nữa. Khi đó, hàng sẽ quay về kho xuất, hệ thống cập nhật tồn kho của điểm xuất hàng trong bảng **Inventory** và kho xuất sẽ cộng thêm số lượng hàng hóa từ đơn vận.

5.1.3 Kết quả đạt được

Sau khi triển khai các giải pháp trên, hệ thống đã xây dựng được một giao diện quản lý chuyển hàng giữa các điểm bán lưu động, giúp các nhân viên quản lý có thể theo dõi tình trạng của các giao dịch chuyển hàng một cách dễ dàng. Hệ thống cũng cung cấp các thông tin chi tiết về tiến trình chuyển hàng, bao gồm trạng thái của từng giao dịch, số lượng hàng hóa chuyển đi và thời gian thực hiện.

Fashion Shop

Tổng quan

Bán tại quầy

Đơn hàng

Sản phẩm

Khách hàng

Kho

Thống kê

Quản lý

Đăng xuất

Danh sách chuyển kho

Nguyễn Duy Thành

+ TẠO ĐƠN CHUYỂN KHO

Chuyển kho

Nhập hàng

🔍

Tìm mã đơn chuyển kho, tên, SĐT cửa hàng

Mã đơn hàng	Cửa hàng xuất	Cửa hàng nhập	Trạng thái	Ngày tạo đơn
TRA00004	Điểm bán 1	Điểm bán 2	Đã hủy	23/12/2024 23:48
TRA00003	Cửa hàng chính	Điểm bán 1	Đã Chuyển hàng	23/12/2024 23:13
TRA00002	Cửa hàng chính	Điểm bán 2	Đang chuyển hàng	23/12/2024 23:12
TRA00001	Cửa hàng chính	Điểm bán 1	Đã Chuyển hàng	18/12/2024 23:25

Rows per page: 101-4 of 4

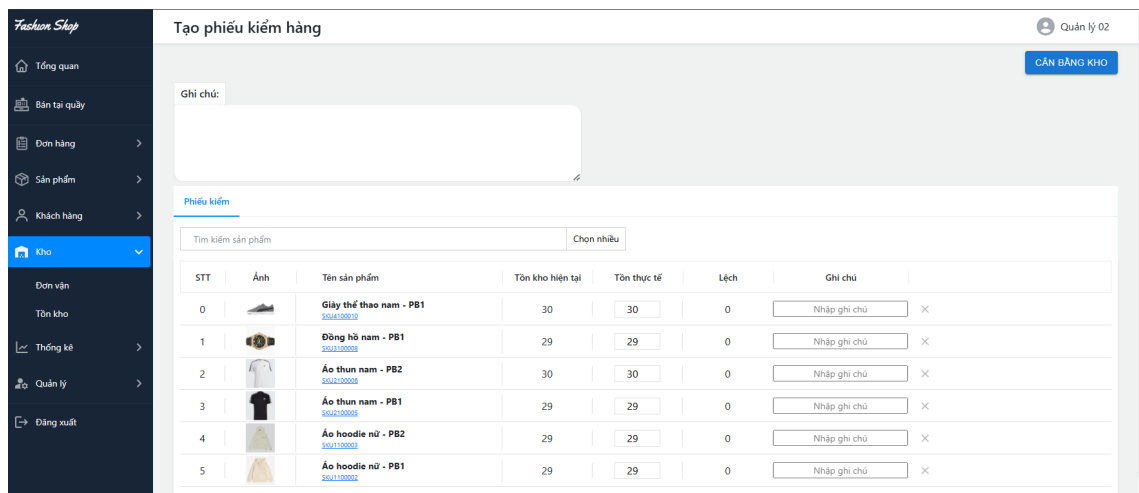
Hình 5.7: Giao diện quản lý chuyển hàng

Giao diện hệ thống sẽ phân loại các giao dịch chuyển hàng theo trạng thái, giúp người quản lý có cái nhìn tổng quan về các giao dịch đang diễn ra và các giao dịch đã hoàn thành. Bên cạnh đó, người quản lý cũng có thể xem chi tiết các giao dịch chuyển hàng, bao gồm thông tin về điểm xuất kho, điểm đến và số lượng hàng hóa. Hệ thống này không chỉ giúp quản lý tốt hơn việc chuyển hàng mà còn giúp tối ưu hóa quy trình phân phối hàng hóa, đảm bảo rằng hàng hóa luôn có mặt đúng thời gian tại các điểm bán cần thiết.

Mã sản phẩm	Tên sản phẩm	Tồn kho	Ngày cập nhật
SKU1100002	Áo thun nam - PB1	20	2024-12-28T16:41:24.008+00:00
SKU1100003	Áo thun nam - PB2	29	2024-12-28T16:41:23.986+00:00
SKU2100005	Giày thể thao nam - PB1	0	2024-12-28T15:52:29.320+00:00
SKU3100007	Áo hoodie nữ - PB1	10	2024-12-28T16:41:23.963+00:00
SKU3100008	Áo hoodie nữ - PB2	5	2024-12-28T16:41:23.940+00:00
SKU4100010	Đồng hồ nam - PB1	2	2024-12-28T15:50:16.452+00:00

**Hình 5.8:** Giao diện chi tiết tồn kho điểm bán

Bên cạnh đó, quản lý cũng có thể theo dõi tồn kho của từng điểm bán để đảm bảo rằng dữ liệu về tồn kho luôn chính xác trong suốt quá trình vận hành. Hệ thống cung cấp giao diện trực quan, giúp quản lý có thể nắm bắt được tình trạng tồn kho điểm bán lưu động. Quản lý cũng có thể kiểm tra chi tiết tồn kho cho từng sản phẩm, bao gồm cả các thông tin liên quan như số lượng hàng hóa còn lại, các mặt hàng sắp hết, và các sản phẩm không còn trong kho đều được hiển thị rõ ràng, giúp quản lý nhanh chóng đưa ra các quyết định bổ sung hoặc điều chỉnh kế hoạch phân phối.



**Hình 5.9:** Giao diện tạo phiếu kiểm kho

Cùng với đó, quản lý cũng có thể tạo phiếu kiểm kho cho điểm bán mình phụ trách để đảm bảo rằng dữ liệu về tồn kho trên hệ thống luôn được cập nhật đúng với tồn kho thực tế. Nếu có chênh lệch so với thực tế có thể do nhiều lý do như hàng bị hỏng, hàng bị thất thoát trong khi vận chuyển, quản lý điểm bán có thể cập nhật số liệu tồn kho thông qua phiếu kiểm, đồng thời bổ sung các ghi chú cần thiết về lý do cân bằng kho (nếu có). Sau khi quản lý thực hiện cân bằng kho, hệ thống sẽ cập nhật lại tồn kho theo số liệu thực tế được ghi nhận.

## 5.2 Tạo kế hoạch bán hàng

### 5.2.1 Đặt vấn đề

Trong mô hình bán lẻ lưu động, việc tạo kế hoạch bán hàng là một yếu tố quan trọng giúp các điểm bán lưu động có thể hoạt động hiệu quả và đáp ứng kịp thời nhu cầu của khách hàng. Một kế hoạch bán hàng tốt không chỉ giúp các nhân viên bán hàng có định hướng rõ ràng về địa điểm, thời gian làm việc, mà còn giúp đảm bảo điều phối hợp lý các ca làm của từng nhân viên cho từng điểm bán, tránh tình trạng thiếu hụt nhân sự tại các điểm bán hàng.

Tuy nhiên, việc xây dựng kế hoạch bán hàng cho các điểm bán lưu động không phải là một công việc đơn giản. Việc thiếu kế hoạch bán hàng hợp lý có thể dẫn đến các vấn đề như thiếu nhân sự, không đáp ứng được nhu cầu khách hàng đặc biệt trong các khung giờ cao điểm hoặc trong thời gian diễn ra các chương trình sự kiện đặc biệt.

### 5.2.2 Giải pháp

Để giải quyết các vấn đề trên, hệ thống POS cho mô hình bán lẻ lưu động sẽ tích hợp chức năng tạo kế hoạch bán hàng, giúp chủ cửa hàng có thể dễ dàng tạo ra một kế hoạch bán hàng phù hợp với tình hình thực tế của từng điểm bán. Chức

năng này sẽ bao gồm các bước sau:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	Mã NV	Họ và tên	Cửa hàng	Vị trí	2025-01-06	2025-01-07	2025-01-08	2025-01-09	2025-01-10	2025-01-11	2025-01-12					
2	1	Chủ cửa hàng	Cửa hàng chính, Điểm bán 1, Điểm bán 2	ADMIN	N	N	N	N	N	N	N					
3	2	Quản lý 01	Cửa hàng chính	MANAGER	N	N	N	N	N	N	N					
4	11	Quản lý 02	Cửa hàng chính	MANAGER	N	N	N	N	N	N	N					
5	3	Bán hàng 01	Cửa hàng chính	SALE	N	N	N	N	N	N	N					
6	4	Bán hàng 02	Cửa hàng chính	SALE	N	N	N	N	N	N	N					
7	5	Quản lý 11	Điểm bán 1	MANAGER	N	N	N	N	N	N	N					
8	12	Quản lý 12	Điểm bán 1	MANAGER	N	N	N	N	N	N	N					
9	6	Bán hàng 11	Điểm bán 1	SALE	N	N	N	N	N	N	N					
10	7	Bán hàng 12	Điểm bán 1	SALE	N	N	N	N	N	N	N					
11	8	Quản lý 21	Điểm bán 2	MANAGER	N	N	N	N	N	N	N					
12	13	Quản lý 22	Điểm bán 2	MANAGER	N	N	N	N	N	N	N					
13	9	Bán hàng 21	Điểm bán 2	SALE	N	N	N	N	N	N	N					
14	10	Bán hàng 22	Điểm bán 2	SALE	N	N	N	N	N	N	N					
15	N	Nghỉ														
16	KIP01	Kip 1 (7:00-14h:30)														
17	KIP02	Kip 2 (14h30:22h)														
18																
19																
20																
21																
22																
23																
24																
25																
26																
27																
28																

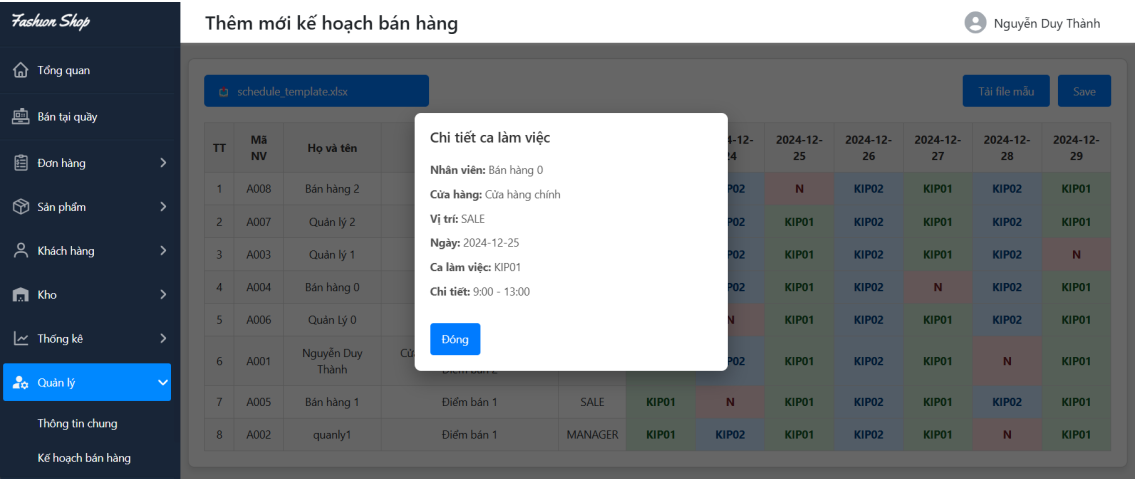
**Hình 5.10:** Bản mẫu kế hoạch bán hàng

**Tải xuống bản mẫu kế hoạch bán hàng theo tuần:** Hệ thống sẽ sử dụng dữ liệu nhân viên đang làm việc, thời gian khởi tạo là 1 tuần làm việc (tính từ ngày thứ 2 tiếp theo) và xây dựng thành bản mẫu excel hoàn chỉnh để tải xuống.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	Mã NV	Họ và tên	Cửa hàng	Vị trí	23/12/2024	24/12/2024	25/12/2024	26/12/2024	27/12/2024	28/12/2024	29/12/2024					
2	1	Chủ cửa hàng	Cửa hàng chính, Điểm bán 1, Điểm bán 2	ADMIN	KIP01	KIP02	KIP01	KIP02	KIP01	KIP02	N					
3	2	Quản lý 01	Cửa hàng chính	MANAGER	KIP01	KIP01	KIP01	KIP01	KIP01	KIP01	N					
4	11	Quản lý 02	Cửa hàng chính	MANAGER	KIP02	KIP02	KIP02	KIP02	KIP02	KIP02	N					
5	3	Bán hàng 01	Cửa hàng chính	SALE	KIP01	KIP01	KIP01	KIP01	KIP01	KIP01	N					
6	4	Bán hàng 02	Cửa hàng chính	SALE	KIP02	KIP02	KIP02	KIP02	KIP02	KIP02	N					
7	5	Quản lý 11	Điểm bán 1	MANAGER	KIP01	KIP01	KIP01	KIP01	KIP01	KIP01	N					
8	12	Quản lý 12	Điểm bán 1	MANAGER	KIP02	KIP02	KIP02	KIP02	KIP02	KIP02	N					
9	6	Bán hàng 11	Điểm bán 1	SALE	KIP01	KIP01	KIP01	KIP01	KIP01	KIP01	N					
10	7	Bán hàng 12	Điểm bán 1	SALE	KIP02	KIP02	KIP02	KIP02	KIP02	KIP02	N					
11	8	Quản lý 21	Điểm bán 2	MANAGER	KIP01	KIP01	KIP01	KIP01	KIP01	KIP01	N					
12	13	Quản lý 22	Điểm bán 2	MANAGER	KIP02	KIP02	KIP02	KIP02	KIP02	KIP02	N					
13	9	Bán hàng 21	Điểm bán 2	SALE	KIP01	KIP01	KIP01	KIP01	KIP01	KIP01	N					
14	10	Bán hàng 22	Điểm bán 2	SALE	KIP02	KIP02	KIP02	KIP02	KIP02	KIP02	N					
15	N	Nghỉ														
16	KIP01	Kip 1 (7:00-14h:30)														
17	KIP02	Kip 2 (14h30:22h)														
18																
19																
20																
21																
22																
23																
24																
25																
26																
27																

**Hình 5.11:** Giao diện điều chỉnh kế hoạch bán hàng

**Lập kế hoạch bán hàng:** Dựa trên tình hình thực tế, chủ cửa hàng sẽ điều chỉnh về thời gian tuần làm việc và phân bổ lại ca làm để phù hợp với mục tiêu và điều kiện thực tế của từng điểm bán.

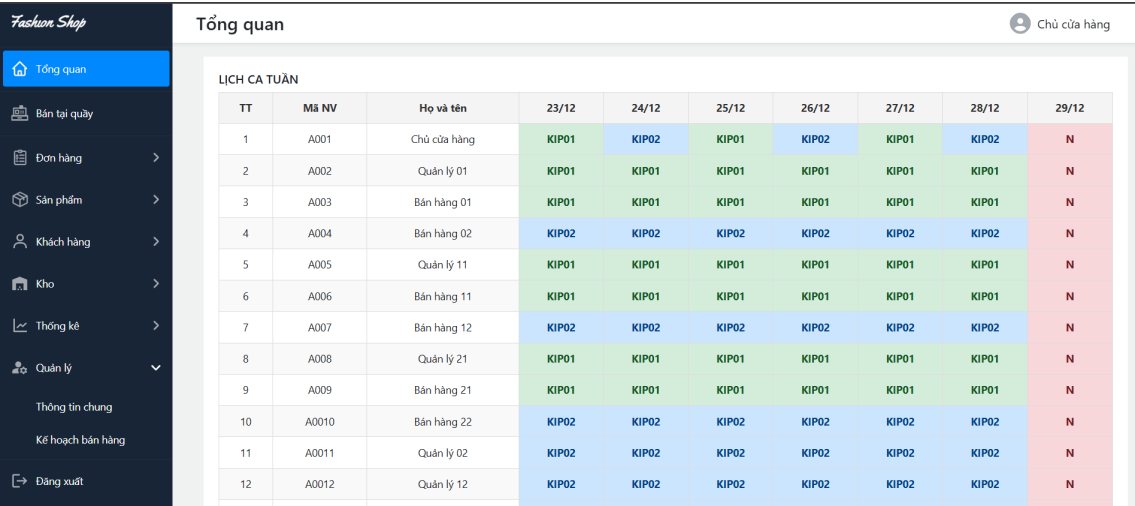


Hình 5.12: Giao diện tạo kế hoạch bán hàng

**Tải lên bản kế hoạch bán hàng và cập nhật hệ thống:** Chủ cửa hàng sau khi điều chỉnh bản kế hoạch làm việc sẽ tải lên hệ thống và cập nhật cho lịch làm việc. Hệ thống sẽ tạo mới hoặc ghi đè dữ liệu lịch làm việc và hiển thị lên trang tổng quan để toàn bộ nhân viên có thể nắm được lịch làm việc.

5.2.3 Kết quả đạt được

Sau khi triển khai chức năng tạo kế hoạch bán hàng, hệ thống đã giúp tối ưu hóa việc lập kế hoạch bán hàng, hỗ trợ chủ cửa hàng có thể thao tác nhanh chóng và giảm thiểu sai sót so với việc lập kế hoạch thủ công và các nhân viên có thể theo dõi lịch làm việc và ca làm mà mình phụ trách.



Hình 5.13: Giao diện theo dõi kế hoạch bán hàng

Giao diện hệ thống cũng cho phép chủ cửa hàng quản lý các ca làm việc theo các tuần trong năm. Bên cạnh đó, bằng việc sử dụng dạng bảng, chủ cửa hàng cũng dễ dàng theo dõi chi tiết từng ca làm việc của mỗi nhân viên làm việc tại từng điểm



bán.

**Kế hoạch bán hàng**

Lịch ca làm việc từ 23/12/2024 đến 29/12/2024

Tạo mới kế hoạch

Tháng: 12 Năm: 2024 Tuần trước Tuần sau

TT	Mã NV	Họ và tên	Chi tiết ca làm việc	24/12	25/12	26/12	27/12	28/12	29/12
1	A001	Chủ cửa hàng	Nhân viên: Bán hàng 11	KIP02	KIP01	KIP02	KIP01	KIP02	N
2	A002	Quản lý 01	Cửa hàng: Điểm bán 1	KIP01	KIP01	KIP01	KIP01	KIP01	N
3	A003	Bán hàng 01	Vị trí: SALE	KIP01	KIP01	KIP01	KIP01	KIP01	N
4	A004	Bán hàng 02	Ngày: 24/12/2024	KIP02	KIP02	KIP02	KIP02	KIP02	N
5	A005	Quản lý 11	Ca làm việc: KIP01	KIP01	KIP01	KIP01	KIP01	KIP01	N
6	A006	Bán hàng 11	Chi tiết: 7:00-14:30	KIP01	KIP01	KIP01	KIP01	KIP01	N
7	A007	Bán hàng 12		KIP02	KIP02	KIP02	KIP02	KIP02	N
8	A008	Quản lý 21	Điểm bán 2	KIP01	KIP01	KIP01	KIP01	KIP01	N
9	A009	Bán hàng 21	Điểm bán 2	KIP01	KIP01	KIP01	KIP01	KIP01	N
10	A0010	Bán hàng 22	Điểm bán 2	KIP02	KIP02	KIP02	KIP02	KIP02	N
11	A0011	Quản lý 02	Cửa hàng chính	KIP02	KIP02	KIP02	KIP02	KIP02	N

Hình 5.14: Giao diện chi tiết kế hoạch bán hàng

## CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

### 6.1 Kết luận

Dưới sự hướng dẫn tận tình của thầy ThS.Nguyễn Mạnh Tuấn, sau một quá trình dài nghiên cứu và thực hiện, em đã hoàn thành việc xây dựng hệ thống quản lý bán hàng POS cho mô hình bán lẻ lưu động. Hệ thống này đã được thiết kế và triển khai với các chức năng chính bao gồm: quản lý người dùng, quản lý sản phẩm, xử lý giao dịch bán hàng, quản lý tồn kho, và theo dõi thống kê hoạt động kinh doanh. Các chức năng này được tối ưu hóa nhằm đáp ứng yêu cầu thực tế của mô hình kinh doanh bán lẻ lưu động, trong đó nổi bật là khả năng lập kế hoạch nhanh chóng dựa trên các mẫu có sẵn, quản lý luân chuyển hàng hóa linh hoạt giữa các điểm bán lưu động, và hỗ trợ thanh toán với các hình thức đa dạng như tiền mặt và quét mã QR.

Quá trình thực hiện đồ án không chỉ giúp em hoàn thiện sản phẩm mà còn mang lại nhiều giá trị học tập đáng kể. Em đã rèn luyện và nâng cao các kỹ năng như phân tích và giải quyết vấn đề, đặc biệt là trong việc xử lý các tình huống phức tạp phát sinh trong quá trình triển khai thực tế. Đồng thời, em cũng đã có cơ hội áp dụng những kiến thức đã học về phân tích yêu cầu, thiết kế hệ thống và xây dựng cơ sở dữ liệu vào một dự án thực tiễn. Bên cạnh đó, em còn học hỏi được nhiều công nghệ mới, bao gồm các công cụ hỗ trợ phát triển hệ thống, các framework lập trình hiện đại, và các tiêu chuẩn thiết kế giao diện nhằm nâng cao chất lượng phần mềm.

Tuy nhiên, do hạn chế về thời gian và tài nguyên, hệ thống hiện tại vẫn còn tồn tại một số hạn chế cần khắc phục. Cụ thể, giao diện người dùng chưa đạt mức thân thiện và thẩm mỹ cao, hiệu năng xử lý dữ liệu lớn còn hạn chế, và hệ thống thiếu các tính năng phân tích báo cáo chuyên sâu. Việc quản lý thông tin sản phẩm cũng như các chức năng quản trị hệ thống hiện vẫn chưa được tối ưu hóa hoàn toàn để đáp ứng tốt hơn nhu cầu của người dùng.

### 6.2 Hướng phát triển

Nhằm hướng đến việc hoàn thiện và mở rộng hệ thống trong tương lai, em dự kiến sẽ triển khai các cải tiến và nâng cấp quan trọng như sau:

**Cải thiện tính năng thanh toán:** Tích hợp hệ thống thông báo thời gian thực khi khách hàng hoàn tất giao dịch chuyển khoản, giúp nhân viên bán hàng dễ dàng kiểm tra trạng thái thanh toán và xử lý nhanh chóng hơn. Đồng thời, mở rộng hỗ trợ các phương thức thanh toán điện tử khác để tăng tính tiện lợi và phù hợp với xu hướng hiện đại.

**Ứng dụng AI và Machine Learning:** Nghiên cứu và áp dụng các thuật toán trí tuệ nhân tạo để phân tích hành vi mua sắm của khách hàng, từ đó đưa ra các gợi ý nhập hàng thông minh, dự báo nhu cầu sản phẩm, và tối ưu hóa luân chuyển hàng hóa giữa các kho và điểm bán.

**Phát triển chức năng thống kê chi tiết:** Tạo ra các báo cáo bán hàng chi tiết và đa chiều hơn, bao gồm các biểu đồ trực quan và số liệu phân tích chuyên sâu. Điều này sẽ hỗ trợ chủ cửa hàng trong việc đưa ra các quyết định chiến lược, nâng cao hiệu quả quản lý và phát triển kinh doanh.

**Mở rộng mô hình kinh doanh:** Nâng cấp hệ thống để có thể áp dụng cho nhiều ngành hàng khác nhau ngoài lĩnh vực bán lẻ quần áo. Hệ thống cần được tối ưu hóa để đáp ứng các yêu cầu đặc thù của từng ngành hàng, tạo điều kiện cho các doanh nghiệp mở rộng quy mô và lĩnh vực kinh doanh.

**Tăng cường hiệu năng và bảo mật:** Nâng cấp hệ thống cơ sở dữ liệu để đảm bảo khả năng xử lý khối lượng lớn dữ liệu một cách nhanh chóng và ổn định hơn. Đồng thời, cải thiện các biện pháp bảo mật nhằm bảo vệ thông tin người dùng và dữ liệu giao dịch một cách tối ưu.

Những hướng phát triển này không chỉ giúp hệ thống trở nên toàn diện hơn mà còn mở ra cơ hội ứng dụng thực tiễn rộng rãi trong các lĩnh vực kinh doanh khác nhau. Em rất mong nhận được sự đóng góp ý kiến từ thầy cô để tiếp tục hoàn thiện sản phẩm và nâng cao chất lượng đồ án.

## TÀI LIỆU THAM KHẢO

- [1] *Lightspeed*. [Online]. Available: <https://www.lightspeedhq.com/> (visited on 10/28/2024).
- [2] *Zettle*. [Online]. Available: <https://www.zettle.com/gb> (visited on 10/28/2024).
- [3] *Api*. [Online]. Available: <https://www.postman.com/what-is-an-api/> (visited on 10/28/2024).
- [4] *Spring boot*. [Online]. Available: <https://spring.io/projects/spring-boot> (visited on 10/28/2024).
- [5] *React*. [Online]. Available: <https://react.dev/> (visited on 10/28/2024).
- [6] *How virtual-dom and diffing works in react*. [Online]. Available: <https://medium.com/@gethylgeorge/how-virtual-dom-and-diffing-works-in-react-6fc805f9f84e> (visited on 10/28/2024).
- [7] *Mysql*. [Online]. Available: <https://www.mysql.com/> (visited on 10/28/2024).
- [8] *Three-tier architecture overview*. [Online]. Available: <https://docs.aws.amazon.com/whitepapers/latest/serverless-multi-tier-architectures-api-gateway-lambda/three-tier-architecture-overview.html> (visited on 10/28/2024).
- [9] *Mvc architecture*. [Online]. Available: <https://www.geeksforgeeks.org/spring-boot-starter-web/> (visited on 10/28/2024).

# PHỤ LỤC

## A. ĐẶC TẢ BẢNG TRONG CƠ SỞ DỮ LIỆU

### A.1 Bảng **balance\_variant**

STT	Tên cột	Kiểu dữ liệu	Mô tả
1	id	int	Khoá chính, tự động tăng (Auto Increment).
2	created_at	datetime(6)	Thời gian tạo bản ghi.
3	updated_at	datetime(6)	Thời gian cập nhật bản ghi.
4	note	varchar(255)	Ghi chú liên quan đến chênh lệch tồn kho.
5	real_quantity	int	Số lượng sản phẩm thực tế.
6	saved_quantity	int	Số lượng sản phẩm lưu trữ trong hệ thống.
7	inventory_id	int	Khoá ngoại tham chiếu đến bảng tồn kho (inventory).
8	balance_id	int	Khoá ngoại tham chiếu đến bảng phiếu cân đối.

**Bảng A.1:** Danh sách thuộc tính của bảng **balance\_variant**

Bảng **balance\_variant** ghi nhận thông tin về sự chênh lệch tồn kho thực tế so với hệ thống. Các trường như **real\_quantity** và **saved\_quantity** giúp so sánh số lượng sản phẩm thực tế và số liệu lưu trữ trên hệ thống, trong khi **inventory\_id** và **balance\_id** liên kết với tồn kho hiện tại và phiếu cân đối. Bảng này hỗ trợ kiểm soát chính xác tồn kho.

### A.2 Bảng **base\_product**

STT	Tên cột	Kiểu dữ liệu	Mô tả
1	id	int	Khoá chính, tự động tăng (Auto Increment).
2	created_at	datetime(6)	Thời gian tạo bản ghi.
3	updated_at	datetime(6)	Thời gian cập nhật bản ghi.
4	attribute1	varchar(255)	Thuộc tính sản phẩm 1.
5	attribute2	varchar(255)	Thuộc tính sản phẩm 2.
6	attribute3	varchar(255)	Thuộc tính sản phẩm 3.
7	is_deleted	tinyint(1)	Trạng thái xóa sản phẩm.
8	label	varchar(255)	Nhãn sản phẩm.
9	name	varchar(255)	Tên sản phẩm.

**Bảng A.2:** Danh sách thuộc tính của bảng **base\_product**

Bảng **base\_product** lưu trữ thông tin chung về sản phẩm, bao gồm các thuộc tính **attribute1**, **attribute2**, và **attribute3**. Trường **name** và **label** cung cấp thông

tin nhận diện. Bảng này giúp quản lý danh mục sản phẩm cơ bản của cửa hàng.

### A.3 Bảng customer

STT	Tên cột	Kiểu dữ liệu	Mô tả
1	id	int	Khoá chính, tự động tăng (Auto Increment).
2	created_at	datetime(6)	Thời gian tạo bản ghi.
3	updated_at	datetime(6)	Thời gian cập nhật bản ghi.
4	address	varchar(255)	Địa chỉ của khách hàng.
5	gender	enum	Giới tính của khách hàng ( <b>FEMALE</b> , <b>MALE</b> , <b>OTHER</b> ).
6	name	varchar(255)	Tên của khách hàng.
7	phone	varchar(255)	Số điện thoại khách hàng.
8	group_id	int	Khoá ngoại tham chiếu đến bảng nhóm khách hàng.
9	customer_code	varchar(255)	Mã khách hàng duy nhất.
10	date_of_birth	datetime(6)	Ngày sinh của khách hàng.
11	email	varchar(255)	Email khách hàng.

**Bảng A.3:** Danh sách thuộc tính của bảng **customer**

Bảng **customer** lưu thông tin chi tiết về khách hàng, bao gồm tên, địa chỉ, số điện thoại và email. Các trường như **group\_id** giúp phân loại khách hàng theo nhóm, trong khi **customer\_code** quản lý mã khách hàng duy nhất. Bảng này hỗ trợ theo dõi và chăm sóc khách hàng hiệu quả.

### A.4 Bảng customer\_group

STT	Tên cột	Kiểu dữ liệu	Mô tả
1	id	int	Khoá chính, tự động tăng (Auto Increment).
2	created_at	datetime(6)	Thời gian tạo bản ghi.
3	updated_at	datetime(6)	Thời gian cập nhật bản ghi.
4	discount	double	Mức chiết khấu áp dụng cho nhóm khách hàng.
5	name	varchar(255)	Tên nhóm khách hàng.

**Bảng A.4:** Danh sách thuộc tính của bảng **customer\_group**

Bảng **customer\_group** quản lý thông tin nhóm khách hàng, bao gồm tên nhóm và mức chiết khấu. Bảng này hỗ trợ việc phân loại và áp dụng chính sách khuyến mãi linh hoạt cho từng nhóm.

**A.5 Bảng feedback**

STT	Tên cột	Kiểu dữ liệu	Mô tả
1	id	int	Khoá chính, tự động tăng (Auto Increment).
2	created_at	datetime(6)	Thời gian tạo bản ghi.
3	updated_at	datetime(6)	Thời gian cập nhật bản ghi.
4	content	varchar(255)	Nội dung phản hồi của khách hàng.
5	title	varchar(255)	Tiêu đề phản hồi.
6	customer_id	int	Khoá ngoại tham chiếu đến bảng khách hàng.
7	evaluate	int	Điểm đánh giá của khách hàng.

**Bảng A.5:** Danh sách thuộc tính của bảng **feedback**

Bảng **feedback** ghi nhận phản hồi từ khách hàng, bao gồm nội dung, tiêu đề, và điểm đánh giá. Bảng giúp theo dõi và xử lý phản hồi, đảm bảo cải thiện chất lượng dịch vụ.

**A.6 Bảng import\_item**

STT	Tên cột	Kiểu dữ liệu	Mô tả
1	id	int	Khoá chính, tự động tăng (Auto Increment).
2	created_at	datetime(6)	Thời gian tạo bản ghi.
3	updated_at	datetime(6)	Thời gian cập nhật bản ghi.
4	import_price	int	Giá nhập của sản phẩm.
5	quantity	int	Số lượng sản phẩm nhập.
6	import_id	int	Khoá ngoại tham chiếu đến bảng phiếu nhập.
7	inventory_id	int	Khoá ngoại tham chiếu đến bảng sản phẩm.
8	discount	int	Mức chiết khấu áp dụng.

**Bảng A.6:** Danh sách thuộc tính của bảng **import\_item**

Bảng **import\_item** lưu trữ thông tin chi tiết về sản phẩm trong từng phiếu nhập hàng. Các trường như **import\_price**, **quantity**, và **discount** giúp quản lý giá cả và số lượng hàng nhập, hỗ trợ tối ưu hoá quá trình nhập hàng.



**A.7 Bảng order\_line**

STT	Tên cột	Kiểu dữ liệu	Mô tả
1	id	int	Khoá chính, tự động tăng (Auto Increment).
2	created_at	datetime(6)	Thời gian tạo bản ghi.
3	updated_at	datetime(6)	Thời gian cập nhật bản ghi.
4	price	int	Giá bán của sản phẩm trong đơn hàng.
5	quantity	int	Số lượng sản phẩm bán ra.
6	return_quantity	int	Số lượng sản phẩm được trả lại.
7	order_id	int	Khoá ngoại tham chiếu đến bảng đơn hàng.
8	inventory_id	int	Khoá ngoại tham chiếu đến bảng sản phẩm.

**Bảng A.7:** Danh sách thuộc tính của bảng **order\_line**

Bảng **order\_line** quản lý chi tiết sản phẩm trong đơn hàng, bao gồm giá bán, số lượng, và số lượng trả lại. Trường **order\_id** liên kết với bảng đơn hàng để xác định đơn cụ thể.

**A.8 Bảng return\_order**

STT	Tên cột	Kiểu dữ liệu	Mô tả
1	id	int	Khoá chính, tự động tăng (Auto Increment).
2	created_at	datetime(6)	Thời gian tạo bản ghi.
3	updated_at	datetime(6)	Thời gian cập nhật bản ghi.
4	return_reason	varchar(255)	Lý do trả hàng.
5	base_order	int	Đơn hàng gốc liên quan đến trả hàng.
6	swap_order	int	Đơn hàng thay thế nếu có.

**Bảng A.8:** Danh sách thuộc tính của bảng **return\_order**

Bảng **return\_order** lưu trữ thông tin về các phiếu trả hàng, bao gồm lý do trả hàng và thời gian trả hàng. Trường **base\_order** và **swap\_order** giúp theo dõi các đơn hàng liên quan.

**A.9 Bảng return\_order\_line**

STT	Tên cột	Kiểu dữ liệu	Mô tả
1	id	int	Khoá chính, tự động tăng (Auto Increment).
2	created_at	datetime(6)	Thời gian tạo bản ghi.
3	updated_at	datetime(6)	Thời gian cập nhật bản ghi.
4	original_price	int	Giá gốc của sản phẩm trả lại.
5	return_price	int	Giá trị hoàn trả của sản phẩm.
6	return_quantity	int	Số lượng sản phẩm trả lại.
7	return_id	int	Khoá ngoại tham chiếu đến bảng phiếu trả hàng.
8	inventory_id	int	Khoá ngoại tham chiếu đến bảng sản phẩm.

**Bảng A.9:** Danh sách thuộc tính của bảng **return\_order\_line**

Bảng **return\_order\_line** ghi nhận chi tiết từng sản phẩm trong phiếu trả hàng, bao gồm giá gốc, giá hoàn trả, và số lượng trả lại. Các thông tin này giúp quản lý chính xác các giao dịch trả hàng.

**A.10 Bảng payment**

STT	Tên cột	Kiểu dữ liệu	Mô tả
1	id	int	Khoá chính, tự động tăng (Auto Increment).
2	created_at	datetime(6)	Thời gian tạo bản ghi.
3	updated_at	datetime(6)	Thời gian cập nhật bản ghi.
4	amount	int	Số tiền thanh toán.
5	order_id	int	Khoá ngoại tham chiếu đến bảng đơn hàng liên quan.
6	order_type	enum	Loại đơn hàng: <b>IMPORT</b> , <b>ORDER</b> , hoặc <b>RETURN</b> .
7	pay_date	date	Ngày thanh toán.
8	payment_method	enum	Phương thức thanh toán: <b>CASH</b> , <b>CREDIT</b> , hoặc <b>TRANSFER</b> .
9	payment_status	enum	Trạng thái thanh toán: <b>CANCELLED</b> , <b>COMPLETE</b> , hoặc <b>INIT</b> .

**Bảng A.10:** Danh sách thuộc tính của bảng **payment**

Bảng **payment** lưu trữ thông tin các giao dịch thanh toán, bao gồm số tiền thanh toán, phương thức thanh toán, trạng thái thanh toán và ngày thực hiện. Mỗi bản ghi liên kết với một đơn hàng thông qua trường **order\_id**, đồng thời xác định loại đơn

hàng như nhập hàng, bán hàng hoặc trả hàng. Bảng này giúp theo dõi chi tiết mọi khoản thanh toán trong hệ thống.

#### A.11 Bảng **role**

STT	Tên cột	Kiểu dữ liệu	Mô tả
1	id	int	Khoá chính, tự động tăng (Auto Increment).
2	created_at	datetime(6)	Thời gian tạo bản ghi.
3	updated_at	datetime(6)	Thời gian cập nhật bản ghi.
4	name	enum	Tên vai trò: <b>ADMIN</b> , <b>CARE</b> , <b>MANAGER</b> , hoặc <b>SALE</b> .

**Bảng A.11:** Danh sách thuộc tính của bảng **role**

Bảng **role** quản lý các vai trò người dùng trong hệ thống như quản trị viên, quản lý và nhân viên bán hàng. Mỗi vai trò được định danh bằng một mã duy nhất, giúp phân quyền rõ ràng cho từng nhóm người dùng.

#### A.12 Bảng **user\_role**

STT	Tên cột	Kiểu dữ liệu	Mô tả
1	user_id	int	Khoá ngoại tham chiếu đến bảng người dùng.
2	role_id	int	Khoá ngoại tham chiếu đến bảng vai trò.

**Bảng A.12:** Danh sách thuộc tính của bảng **user\_role**

Bảng **user\_role** dùng để liên kết người dùng với vai trò của họ. Mỗi bản ghi gồm **user\_id** và **role\_id**, giúp xác định quyền hạn cụ thể của từng người trong hệ thống.

#### A.13 Bảng **user\_shop\_access**

STT	Tên cột	Kiểu dữ liệu	Mô tả
1	user_id	int	Khoá chính, tham chiếu đến bảng người dùng.
2	shop_id	int	Khoá chính, tham chiếu đến bảng cửa hàng.

**Bảng A.13:** Danh sách thuộc tính của bảng **user\_shop\_access**

Bảng **user\_shop\_access** lưu trữ thông tin về quyền truy cập của người dùng vào các cửa hàng. Mỗi bản ghi bao gồm **user\_id** và **shop\_id**, đảm bảo rằng chỉ những người dùng được phép mới có thể quản lý các cửa hàng tương ứng.

**A.14 Bảng shift**

STT	Tên cột	Kiểu dữ liệu	Mô tả
1	id	int	Khoá chính, tự động tăng (Auto Increment).
2	created_at	datetime(6)	Thời gian tạo bản ghi.
3	updated_at	datetime(6)	Thời gian cập nhật bản ghi.
4	code	varchar(10)	Mã ca làm việc.
5	detail	varchar(100)	Mô tả chi tiết về ca làm việc.

**Bảng A.14:** Danh sách thuộc tính của bảng **shift**

Bảng **shift** lưu thông tin về các ca làm việc, bao gồm mã ca làm việc và mô tả chi tiết. Dữ liệu này hỗ trợ việc quản lý và sắp xếp lịch làm việc cho nhân viên một cách hiệu quả.

**A.15 Bảng vendor**

STT	Tên cột	Kiểu dữ liệu	Mô tả
1	id	int	Khoá chính, tự động tăng (Auto Increment).
2	created_at	datetime(6)	Thời gian tạo bản ghi.
3	updated_at	datetime(6)	Thời gian cập nhật bản ghi.
4	address	varchar(255)	Địa chỉ của nhà cung cấp.
5	email	varchar(255)	Email liên hệ của nhà cung cấp.
6	name	varchar(255)	Tên nhà cung cấp.
7	phone	varchar(255)	Số điện thoại liên hệ.
8	shop_id	int	Khoá ngoại tham chiếu đến bảng cửa hàng.
9	description	varchar(255)	Mô tả về nhà cung cấp.
10	fax	varchar(255)	Số fax của nhà cung cấp.
11	tax	varchar(255)	Mã số thuế của nhà cung cấp.
12	website	varchar(255)	Website của nhà cung cấp.

**Bảng A.15:** Danh sách thuộc tính của bảng **vendor**

Bảng **vendor** lưu trữ thông tin về nhà cung cấp như tên, địa chỉ, số điện thoại, email, và các chi tiết liên quan. Dữ liệu từ bảng này giúp hệ thống dễ dàng quản lý và duy trì mối quan hệ với các nhà cung cấp.

**A.16 Bảng transfer\_item**

STT	Tên cột	Kiểu dữ liệu	Mô tả
1	id	int	Khoá chính, tự động tăng (Auto Increment).
2	transfer_id	int	Khoá ngoại tham chiếu đến bảng phiếu chuyển hàng.
3	inventory_id	int	Khoá ngoại tham chiếu đến bảng sản phẩm.
4	quantity	int	Số lượng sản phẩm được chuyển.

**Bảng A.16:** Danh sách thuộc tính của bảng **transfer\_item**

Bảng **transfer\_item** theo dõi chi tiết các sản phẩm trong từng phiếu chuyển hàng, bao gồm mã sản phẩm, số lượng và phiếu chuyển liên quan. Bảng này giúp quản lý chặt chẽ việc luân chuyển hàng hóa giữa các cửa hàng.