

**TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN CUỐI KỲ MÔN
PHÁT TRIỂN ỨNG DỤNG WEB VỚI NODE.JS**

HỆ THỐNG POS BÁN LẺ ĐIỆN THOẠI VÀ PHỤ KIỆN

Người hướng dẫn: **ThS. VŨ ĐÌNH HỒNG**

Người thực hiện: **LÊ THỊ XUÂN NGÂN – 52100065**

TRẦN LÊ DUY – 52100788

NGUYỄN ANH QUÂN – 52100095

Lớp : 21050201

Khoá : 25

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2024

**TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN CUỐI KỲ MÔN
PHÁT TRIỂN ỨNG DỤNG WEB VỚI NODE.JS**

HỆ THỐNG POS BÁN LẺ ĐIỆN THOẠI VÀ PHỤ KIỆN

Người hướng dẫn: **ThS. VŨ ĐÌNH HỒNG**

Người thực hiện: **LÊ THỊ XUÂN NGÂN**

TRẦN LÊ DUY

NGUYỄN ANH QUÂN

Lớp : **21050201**

Khoá : **25**

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2024

LỜI CẢM ƠN

Chúng em xin chân thành cảm ơn thầy Vũ Đình Hồng – Giảng viên khoa Công nghệ thông tin – Trường đại học Tôn Đức Thắng đã tận tình giảng dạy và hướng dẫn, giải đáp những thắc mắc trong quá trình thực hiện dự án.

Mặc dù rất nỗ lực nhưng do hiểu biết còn hạn chế nên bài báo cáo cuối kỳ không thể tránh khỏi những thiếu sót. Chúng em rất mong nhận được sự góp ý của các thầy cô để bài báo cáo cuối kỳ được hoàn thiện hơn.

Chúng em xin chân thành cảm ơn.

ĐỒ ÁN ĐƯỢC HOÀN THÀNH TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG

Chúng em xin cam đoan đây là sản phẩm đồ án của riêng chúng em và được sự hướng dẫn của ThS Vũ Đình Hồng. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong đồ án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

Nếu phát hiện có bất kỳ sự gian lận nào chúng em xin hoàn toàn chịu trách nhiệm về nội dung đồ án của mình. Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do chúng em gây ra trong quá trình thực hiện (nếu có).

TP. Hồ Chí Minh, ngày 15 tháng 11 năm 2024

Tác giả

(ký tên và ghi rõ họ tên)

Ngân

Duy

Lê Thị Xuân Ngân

Trần Lê Duy

Quân

Nguyễn Anh Quân

PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN

Phần xác nhận của GV hướng dẫn

Tp. Hồ Chí Minh, ngày tháng năm
(kí và ghi họ tên)

Phần đánh giá của GV chấm bài

Tp. Hồ Chí Minh, ngày tháng năm
(kí và ghi họ tên)

MỤC LỤC

LỜI CẢM ƠN	1
PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN	3
MỤC LỤC.....	4
DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ	7
CHƯƠNG 1 – TỔNG QUAN HỆ THỐNG	10
1.1 Giới thiệu đề tài.....	10
1.2 Cơ sở lý thuyết	11
CHƯƠNG 2 – PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG	13
2.1 Yêu cầu chức năng của hệ thống	13
2.2 Yêu cầu phi chức năng của hệ thống	15
2.3 Sơ đồ Usecase	16
2.3.1 Use case tổng quát	16
2.3.2 Đặc tả Use case	17
2.4 Sơ đồ ERD	39
2.5 Sơ đồ Class.....	41
2.6 Sơ đồ Activity	42
CHƯƠNG 3 – HIỆN THỰC HỆ THỐNG	49
3.1 Tổng quan hiện thực hệ thống.....	49
3.1.1 Công cụ phát triển.....	49
3.1.2 Framework	49
3.1.3 Ngôn ngữ lập trình	49
3.1.4 Cơ sở dữ liệu	49
3.1.5 Môi trường phát triển.....	49
3.2 Kiến trúc hệ thống.....	49
3.2.1. Mô tả kiến trúc tổng thể	50
3.2.2 Các thành phần chính.....	50

3.3 Các đoạn mã nguồn quan trọng	51
3.3.1 Chức năng tạo tài khoản nhân viên mới	51
3.3.2 Chức năng đăng nhập.....	54
3.3.3 Hàm thêm sản phẩm mới	56
3.3.4 Hàm xóa sản phẩm theo ID.....	59
3.3.5 Hàm thanh toán hóa đơn	61
CHƯƠNG 4 - KẾT QUẢ ĐẠT ĐƯỢC	64
4.1 Giao diện đăng nhập	64
4.2 Giao diện Dashboard.....	65
4.3 Giao diện Products	68
4.4 Giao diện Product Detail.....	71
4.5 Giao diện Cart	72
4.6 Giao diện Order.....	73
4.7 Giao diện Checkout.....	75
4.8 Giao diện Employees	77
4.9 Giao diện Employee Profile.....	78
4.10 Giao diện Customers	79
4.11 Giao diện Sign up.....	81
4.12 Giao diện Resend Email.....	81
4.13 Giao diện My Profile.....	82
CHƯƠNG 5 - KẾT LUẬN	83
5.1 Ưu điểm.....	83
5.2 Khuyết điểm.....	83
5.3 Hướng phát triển tương lai.....	83
TÀI LIỆU THAM KHẢO.....	84

DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT

CÁC CHỮ VIẾT TẮT

API Application Programming Interface

ID Identifier

DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ

DANH MỤC HÌNH

Hình 2.1 Sơ đồ Use case tổng quát	16
Hình 2.2 Lược đồ ERD	39
Hình 2.3 Sơ đồ Class	41
Hình 2.4 Sơ đồ Activity cho use case Đăng nhập	42
Hình 2.5 Sơ đồ Activity cho use case Tạo tài khoản nhân viên mới	43
Hình 2.6 Sơ đồ Activity cho use case Thêm sản phẩm mới	44
Hình 2.7 Sơ đồ Activity cho use case Xóa sản phẩm	45
Hình 2.8 Sơ đồ Activity cho use case Tạo đơn hàng mới	46
Hình 2.9 Sơ đồ Activity cho use case Cập nhật đơn hàng	47
Hình 2.10: Sơ đồ Activity cho use case Thanh toán đơn hàng	48
Hình 3.1 Sơ đồ kiến trúc tổng thể của hệ thống	50
Hình 3.2 Hàm tạo tài khoản nhân viên mới	52
Hình 3.3 Hàm đăng nhập	54
Hình 3.4 Hàm thêm sản phẩm	57
Hình 3.5 Hàm xóa sản phẩm theo ID	59
Hình 3.6 Hàm thanh toán hóa đơn	62
Hình 4.1 Giao diện đăng nhập	64
Hình 4.2 Giao diện Dashboard 1	65
Hình 4.3 Giao diện Dashboard 2	65
Hình 4.4 Giao diện Dashboard 3	66
Hình 4.4 Giao diện Products	68
Hình 4.5 Giao diện Search Product	68
Hình 4.6 Giao diện Add Product	69
Hình 4.7 Giao diện Filter By Category	70
Hình 4.8 Giao diện Product Detail	71

Hình 4.9 Giao diện Cart	72
Hình 4.10 Giao diện Order	73
Hình 4.11 Giao diện tạo khách hàng mới	74
Hình 4.12 Giao diện Create Order	74
Hình 4.13 Giao diện Checkout thất bại	75
Hình 4.14 Giao diện Checkout thành công 1	75
Hình 4.14 Giao diện Checkout thành công 2	76
Hình 4.15 Giao diện Hóa đơn định dạng pdf	76
Hình 4.16 Giao diện Employees	77
Hình 4.17 Giao diện Employee Profile	78
Hình 4.18 Giao diện Customers	79
Hình 4.19 Giao diện Customer Profile	80
Hình 4.20 Giao diện Chi tiết đơn hàng	80
Hình 4.21 Giao diện Sign up	81
Hình 4.22 Giao diện Resend Email	81
Hình 4.23 Giao diện My Profile	82

DANH MỤC BẢNG

Bảng 2.1: Đặc tả use case đăng nhập	17
Bảng 2.2: Đặc tả use case xem thông tin cá nhân	18
Bảng 2.3: Đặc tả use case cập nhật thông tin cá nhân	19
Bảng 2.4: Đặc tả use case đổi mật khẩu	20
Bảng 2.5: Đặc tả use case xem danh sách sản phẩm	21
Bảng 2.6: Đặc tả use case xem báo cáo	22
Bảng 2.7: Đặc tả use case tạo tài khoản nhân viên mới	23
Bảng 2.8: Đặc tả use case xem danh sách nhân viên	24
Bảng 2.9: Đặc tả use case xem chi tiết nhân viên	25

Bảng 2.10: Đặc tả use case khóa/mở khóa tài khoản nhân viên	26
Bảng 2.11: Đặc tả use case thêm sản phẩm mới	27
Bảng 2.12: Đặc tả use case cập nhật thông tin sản phẩm.....	28
Bảng 2.13: Đặc tả use case xóa sản phẩm.....	29
Bảng 2.14: Đặc tả use case thêm khách hàng mới.....	30
Bảng 2.15: Đặc tả use case xem thông tin khách hàng	31
Bảng 2.16: Đặc tả use case xem lịch sử mua hàng của khách hàng	32
Bảng 2.17: Đặc tả use case tạo đơn hàng mới	33
Bảng 2.18: Đặc tả use case xem đơn hàng theo số điện thoại	34
Bảng 2.19: Đặc tả use case xem chi tiết đơn hàng.....	35
Bảng 2.20: Đặc tả use case cập nhật đơn hàng	36
Bảng 2.21: Đặc tả use case thanh toán đơn hàng.....	37
Bảng 2.22: Đặc tả use case báo cáo và phân tích.....	38

CHƯƠNG 1 – TỔNG QUAN HỆ THỐNG

1.1 Giới thiệu đề tài

Đề tài này hướng đến xây dựng một hệ thống web hỗ trợ quản lý bán hàng (POS) cho cửa hàng điện thoại và phụ kiện, giúp tối ưu hóa quá trình bán hàng, quản lý sản phẩm, khách hàng và báo cáo doanh thu. Trong bối cảnh phát triển nhanh chóng của các cửa hàng bán lẻ điện thoại tại Việt Nam, việc sở hữu một hệ thống POS hiệu quả trở thành một nhu cầu thiết yếu, giúp cửa hàng tối ưu hóa hiệu suất bán hàng, cải thiện trải nghiệm của nhân viên và khách hàng.

Hệ thống POS này cung cấp các chức năng quan trọng bao gồm: quản lý tài khoản (với sự phân quyền giữa nhân viên bán hàng và quản trị viên), quản lý danh mục sản phẩm, quản lý thông tin khách hàng, xử lý giao dịch bán hàng và thống kê doanh thu. Các yêu cầu cụ thể của đề tài bao gồm:

- **Giao diện thân thiện cho nhân viên bán hàng:** Giúp nhân viên dễ dàng thao tác trong quá trình thanh toán và tìm kiếm sản phẩm nhanh chóng qua mã vạch. Tính năng này bao gồm khả năng nhập thông tin khách hàng, lưu thông tin của khách hàng mới và in hóa đơn dưới dạng PDF khi giao dịch hoàn tất.
- **Quản lý tài khoản và phân quyền truy cập:** Hệ thống thiết lập một tài khoản quản trị viên mặc định, giúp người quản trị tạo và quản lý tài khoản cho nhân viên bán hàng. Chỉ tài khoản quản trị viên mới có quyền tạo mới, khóa/mở tài khoản nhân viên khi cần thiết, cũng như xem thông tin bán hàng của nhân viên.
- **Quản lý danh mục sản phẩm:** Quản trị viên có thể thêm mới, cập nhật, hoặc xóa sản phẩm khi cần, với thông tin bao gồm mã sản phẩm (barcode), tên sản phẩm, giá nhập, giá bán, danh mục, và ngày tạo. Sản phẩm chỉ được phép xóa khi không có đơn hàng nào liên quan để đảm bảo tính toàn vẹn dữ liệu. Nhân

viên bán hàng có thể xem danh sách sản phẩm nhưng bị hạn chế quyền sửa và không thể thấy giá nhập sản phẩm.

- **Quản lý thông tin khách hàng:** Nhân viên bán hàng có thể xem lịch sử mua hàng của khách hàng, bao gồm thông tin chi tiết về mỗi giao dịch như ngày mua, số lượng sản phẩm, và tổng tiền thanh toán. Nếu khách hàng mua hàng lần đầu, hệ thống sẽ cho phép tạo tài khoản cho khách hàng để lưu trữ thông tin cho các lần mua sau.
- **Báo cáo và thống kê doanh thu:** Hệ thống cung cấp khả năng báo cáo chi tiết doanh thu và số lượng đơn hàng dựa trên các mốc thời gian khác nhau. Báo cáo bao gồm tổng doanh thu, số lượng đơn hàng, số lượng sản phẩm bán ra, và danh sách đơn hàng theo thứ tự thời gian. Quản trị viên có thể xem thêm thông tin lợi nhuận tổng hợp để có cái nhìn tổng quát về tình hình kinh doanh.

1.2 Cơ sở lý thuyết

Để xây dựng hệ thống quản lý bán hàng cho cửa hàng điện thoại và phụ kiện, đề tài đã sử dụng các nền tảng và công nghệ chính như Node.js, Express.js, MongoDB và Redis. Mỗi công nghệ có vai trò quan trọng trong việc đảm bảo hệ thống đáp ứng đúng các yêu cầu của đề tài và hỗ trợ khả năng mở rộng:

- **Node.js** là một nền tảng server-side phổ biến cho phép xây dựng các ứng dụng web với ngôn ngữ JavaScript hiệu suất cao và dễ dàng mở rộng. **Express.js** là một back end framework của **Node.js**, cung cấp các công cụ hữu ích để quản lý routing, middleware, và xử lý yêu cầu HTTP, từ đó giúp đơn giản hóa việc phát triển ứng dụng RESTful API. **Node.js** và **Express.js** tạo thành nền tảng cho các tính năng quan trọng của hệ thống.
- **MongoDB** là một hệ quản trị cơ sở dữ liệu NoSQL. Khả năng mở rộng và hỗ trợ JSON của MongoDB giúp dễ dàng quản lý dữ liệu phức tạp, phù hợp với hệ thống POS yêu cầu lưu trữ dữ liệu người dùng, khách hàng, sản phẩm và giao dịch.

- **Redis** là một hệ thống lưu trữ dữ liệu dạng key-value có hiệu suất cao, được sử dụng để caching. Trong ứng dụng, Redis hỗ trợ tăng tốc độ truy xuất dữ liệu, đặc biệt là cho các tính năng yêu cầu tốc độ phản hồi nhanh và dữ liệu thường không bị thay đổi như tra cứu thông tin khách hàng hoặc báo cáo, phân tích tình hình kinh doanh. **Redis** giúp giảm tải cho cơ sở dữ liệu chính và cải thiện trải nghiệm người dùng bằng cách giảm thiểu thời gian chờ đợi.
- **Brevo API** được dùng để gửi email thay thế cho Nodemailer, hỗ trợ gửi các email một cách hiệu quả và linh hoạt. **Brevo** cung cấp tính năng gửi email với API RESTful, dễ tích hợp và kiểm soát nội dung, tệp đính kèm email được gửi đi.
- **Bcrypt** là thư viện mã hóa dùng để băm và bảo vệ mật khẩu người dùng. Trong hệ thống, Bcrypt đảm bảo các mật khẩu của tài khoản không bị lộ thông qua việc lưu trữ trực tiếp, mà thay vào đó là các chuỗi băm được mã hóa an toàn.
- **JWT** là một phương thức để xác thực người dùng qua các tokens. Khi người dùng đăng nhập thành công, hệ thống sẽ tạo ra một token **JWT** để xác nhận danh tính của họ trong suốt phiên làm việc. Phương pháp này giúp đảm bảo tính bảo mật và dễ dàng mở rộng hệ thống xác thực mà không cần quản lý phiên phức tạp.
- **Swagger** là công cụ giúp mô tả và tài liệu hóa API. Bằng cách sử dụng **Swagger**, hệ thống cung cấp một giao diện trực quan cho nhóm phát triển thử nghiệm các endpoints của API, đảm bảo quá trình kiểm thử và phát triển diễn ra thuận tiện và dễ dàng hơn.

CHƯƠNG 2 – PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

Hệ thống quản lý bán hàng (POS) được xây dựng để đáp ứng các yêu cầu quản lý nhân viên, khách hàng, sản phẩm và giao dịch. Hệ thống cần đảm bảo cung cấp các chức năng đầy đủ và giao diện thân thiện, nhằm tạo ra trải nghiệm sử dụng tốt nhất cho người dùng là nhân viên bán hàng và quản trị viên

2.1 Yêu cầu chức năng của hệ thống

Hệ thống POS cần cung cấp các tính năng chính như đăng nhập, quản lý tài khoản, quản lý sản phẩm, giao dịch bán hàng, và báo cáo doanh thu. Cụ thể:

Quản lý tài khoản: Cho phép quản trị viên và nhân viên đăng ký tài khoản, đăng nhập, và đăng xuất khỏi hệ thống.

- Đăng ký tài khoản: Người dùng mới (nhân viên) có thể được quản trị viên tạo tài khoản với các thông tin cơ bản như email, mật khẩu, tên người dùng, số điện thoại, và vai trò.
- Đăng nhập.
- Phân quyền người dùng: Tài khoản được chia thành hai loại chính: tài khoản quản trị viên (admin) và tài khoản nhân viên bán hàng.
 - Quản trị viên có quyền cao nhất, có thể quản lý tài khoản nhân viên, tạo báo cáo, xem lịch sử giao dịch, và quản lý toàn bộ sản phẩm.
 - Nhân viên bán hàng có quyền hạn giới hạn hơn, chỉ có thể thực hiện các thao tác liên quan đến bán hàng và xem danh sách sản phẩm mà không được chỉnh sửa hoặc xóa.
- Quản lý tài khoản: Cho phép thay đổi mật khẩu, cập nhật thông tin cá nhân và khóa/mở tài khoản nhân viên nếu cần thiết.

Quản lý khách hàng:

- Truy xuất thông tin qua số điện thoại: Khi thanh toán, nhân viên bán hàng sẽ yêu cầu số điện thoại của khách hàng. Nếu khách hàng đã từng mua hàng trước đây, hệ thống sẽ tự động hiển thị tên và địa chỉ của khách.
- Tạo tài khoản khách hàng: Đối với khách hàng mới, nhân viên sẽ thông tin của khách hàng. Hệ thống sẽ tự động tạo một tài khoản mới cho khách hàng dựa trên thông tin này.
- Hiển thị thông tin khách hàng:
 - Thông tin cá nhân: Nhân viên có thể dễ xem thông tin cơ bản của khách hàng như tên, số điện thoại, email và địa chỉ thông qua giao diện bán hàng.
 - Lịch sử mua hàng: Nhân viên bán hàng có thể truy cập vào lịch sử mua hàng của khách

Quản lý sản phẩm: Hỗ trợ thêm, sửa, xóa và xem thông tin sản phẩm.

- Tìm kiếm sản phẩm: Cung cấp công cụ tìm kiếm sản phẩm hiệu quả dựa trên tên sản phẩm, mã vạch, nhà sản xuất cũng như tầm giá.
- Thêm sản phẩm: Mỗi sản phẩm cần có các thuộc tính như mã vạch, tên sản phẩm, giá nhập, giá bán lẻ, danh mục và ngày tạo.
- Chỉnh sửa và xóa sản phẩm: Cho phép quản trị viên cập nhật hoặc xóa thông tin sản phẩm khi cần. Chỉ những sản phẩm chưa có trong bất kỳ giao dịch nào mới có thể bị xóa.
- Phân loại sản phẩm: Các sản phẩm được phân loại theo danh mục (ví dụ: điện thoại, máy tính bảng, phụ kiện), giúp việc tìm kiếm và quản lý dễ dàng hơn.
- Cập nhật lượng tồn kho: Số lượng sản phẩm tồn kho cần được cập nhật tự động khi có thay đổi về mua bán.

Quản lý giao dịch bán hàng: Hỗ trợ nhân viên bán hàng thực hiện và ghi nhận giao dịch, thay đổi đơn hàng, thêm, xóa sản phẩm khỏi đơn hàng.

- Tạo giao dịch mới: Khi bán sản phẩm, hệ thống sẽ lưu lại thông tin khách hàng, sản phẩm, và số lượng. Mỗi giao dịch sẽ tạo mã đơn hàng duy nhất.
- Thêm, xóa sản phẩm khỏi đơn hàng: Khi số lượng sản phẩm thay đổi hoặc một sản phẩm bị loại bỏ, các thông tin như tổng tiền của từng sản phẩm và tổng tiền đơn hàng sẽ tự động cập nhật
- Xuất hóa đơn: Hệ thống sẽ tạo hóa đơn dạng PDF để lưu trữ và gửi qua email cho khách hàng.

Báo cáo và thống kê: Hệ thống cung cấp báo cáo doanh thu, phân tích sản phẩm đã bán theo thời gian.

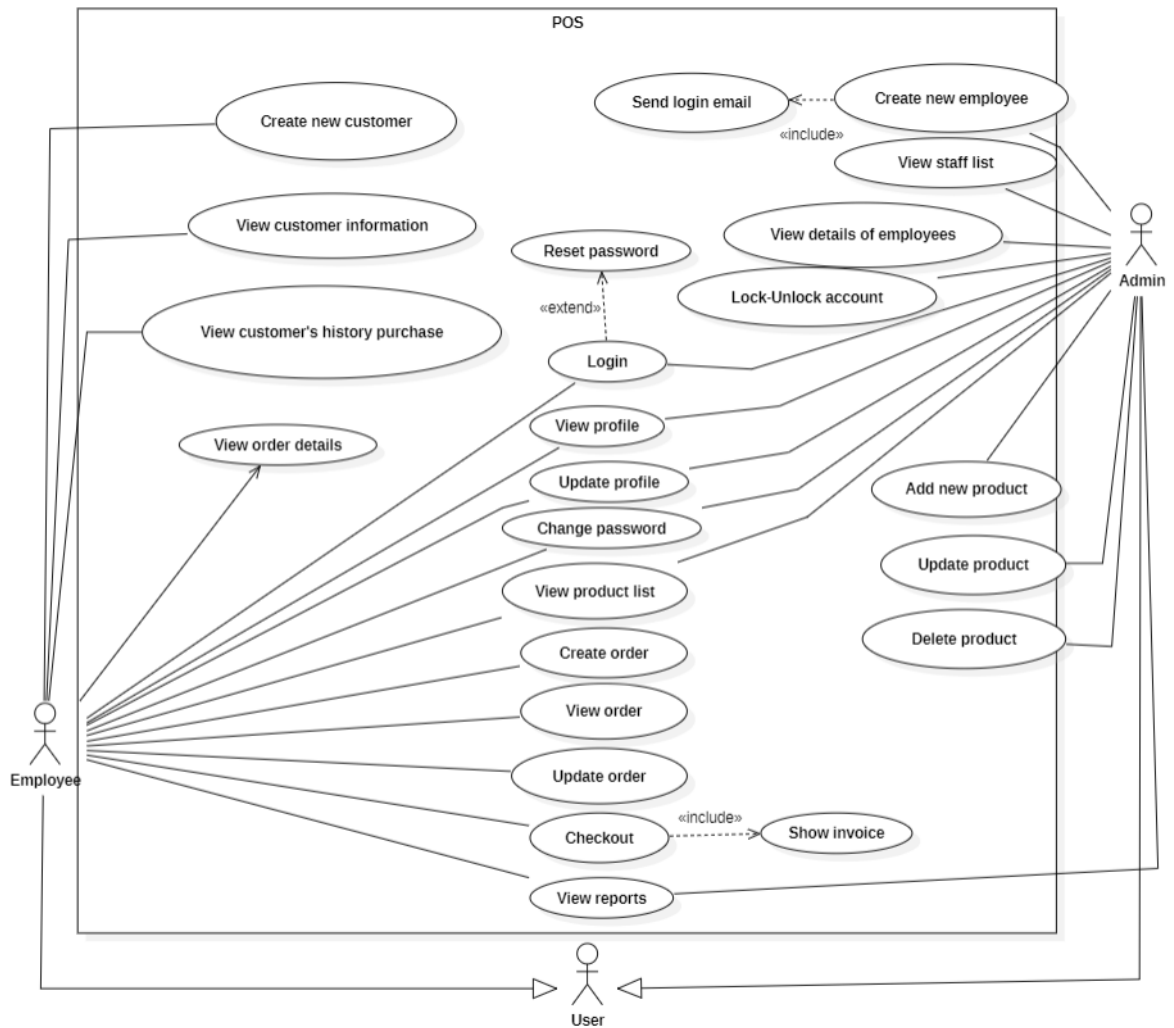
- Báo cáo doanh thu: Cung cấp dữ liệu doanh thu theo ngày, tuần, tháng để giúp quản lý đánh giá tình hình kinh doanh.
- Phân tích sản phẩm: Cho phép xem sản phẩm bán đã bán được theo ngày, tuần, tháng.

2.2 Yêu cầu phi chức năng của hệ thống

- Giao diện thân thiện với người dùng, dễ dàng sử dụng.
- Xác thực và phân quyền: Sử dụng JWT để bảo vệ các API và kiểm soát truy cập theo vai trò của người dùng. Mật khẩu người dùng được mã hóa bằng bcrypt trước khi lưu vào cơ sở dữ liệu.
- Caching: Redis được tích hợp để lưu trữ tạm thời dữ liệu, giảm tải cho cơ sở dữ liệu chính và tăng hiệu suất xử lý các truy vấn phức tạp.
- Xử lý email thông qua Brevo APIs.
- Dễ bảo trì: Với cấu trúc mã nguồn rõ ràng, tích hợp API document bằng Swagger, hệ thống dễ bảo trì và nâng cấp khi cần thiết.

2.3 Sơ đồ Usecase

2.3.1 Use case tổng quát



Hình 2.1 Sơ đồ Use case tổng quát

2.3.2 Đặc tả Use case

Bảng 2.1: Đặc tả use case đăng nhập

Mã use case	UC01	
Tên use case	Đăng nhập	
Ngữ cảnh	Người dùng sử dụng tài khoản đã tạo trước đó để đăng nhập	
Mô tả	Người dùng đăng nhập vào hệ thống để sử dụng các chức năng của hệ thống	
Tác nhân	Admin/ Nhân viên	
Sự kiện kích hoạt	Admin/ Nhân viên muốn truy cập vào hệ thống POS để thao tác	
Điều kiện tiên quyết	Admin/ Nhân viên đã có tài khoản trên hệ thống	
Kết quả	Admin/ Nhân viên đăng nhập thành công	
Luồng sự kiện	Tác nhân	Hệ thống
	<ul style="list-style-type: none"> Người dùng chọn mục “Đăng nhập” 	<ul style="list-style-type: none"> Hệ thống hiển thị giao diện đăng nhập
	<ul style="list-style-type: none"> Người dùng nhập tài khoản và mật khẩu 	
	<ul style="list-style-type: none"> Người dùng ấn nút đăng nhập 	<ul style="list-style-type: none"> Hệ thống kiểm tra xem người dùng đã nhập đúng thông tin tài khoản có trên hệ thống hay chưa, nếu đúng thì hiển thị giao diện của ứng dụng
Ngoại lệ	Người dùng nhập sai tên tài khoản hoặc mật khẩu, hệ thống báo lỗi và yêu cầu nhập lại	

Bảng 2.2: Đặc tả use case xem thông tin cá nhân

<i>Mã use case</i>	UC02	
<i>Tên use case</i>	Xem thông tin cá nhân	
<i>Ngữ cảnh</i>	Người dùng muốn xem thông tin cá nhân của mình trên hệ thống.	
<i>Mô tả</i>	Hệ thống cho phép người dùng kiểm tra thông tin tài khoản cá nhân.	
<i>Tác nhân</i>	Admin/ Nhân viên	
<i>Sự kiện kích hoạt</i>	Người dùng chọn chức năng xem thông tin cá nhân.	
<i>Điều kiện tiên quyết</i>	Admin/ Nhân viên đã đăng nhập thành công.	
<i>Kết quả</i>	Hiển thị thông tin chi tiết về tài khoản của người dùng	
<i>Luồng sự kiện</i>	Tác nhân	Hệ thống
	<ul style="list-style-type: none"> Người dùng chọn mục “Xem thông tin cá nhân”. 	<ul style="list-style-type: none"> Hệ thống hiển thị giao diện thông tin tài khoản, bao gồm tên, email, vai trò, v.v.
<i>Ngoại lệ</i>	Không có	

Bảng 2.3: Đặc tả use case cập nhật thông tin cá nhân

Mã use case	UC03	
Tên use case	Cập nhật thông tin cá nhân	
Ngữ cảnh	Người dùng muốn thay đổi thông tin cá nhân của mình trên hệ thống.	
Mô tả	Hệ thống cho phép người dùng thay đổi thông tin tài khoản cá nhân.	
Tác nhân	Admin/ Nhân viên	
Sự kiện kích hoạt	Người dùng chọn chức năng cập nhật thông tin cá nhân.	
Điều kiện tiên quyết	Admin/ Nhân viên đã đăng nhập thành công.	
Kết quả	Hiển thị thông tin chi tiết về tài khoản của người dùng	
Luồng sự kiện	Tác nhân	Hệ thống
	<ul style="list-style-type: none"> Người dùng chọn mục “Cập nhật thông tin cá nhân”. 	<ul style="list-style-type: none"> Hệ thống hiển thị giao diện chỉnh sửa thông tin tài khoản.
	<ul style="list-style-type: none"> Người dùng thay đổi các trường thông tin cần cập nhật. 	
	<ul style="list-style-type: none"> Người dùng nhấn nút “Lưu thay đổi” 	Hệ thống lưu thông tin mới và hiển thị thông báo thành công.
Ngoại lệ	Nếu thông tin nhập không hợp lệ, hệ thống hiển thị lỗi và yêu cầu sửa lại.	

Bảng 2.4: Đặc tả use case đổi mật khẩu

Mã use case	UC04	
Tên use case	Đổi mật khẩu	
Ngữ cảnh	Người dùng muốn thay đổi mật khẩu tài khoản	
Mô tả	Người dùng nhập mật khẩu cũ và mật khẩu mới để đổi mật khẩu.	
Tác nhân	Admin/ Nhân viên	
Sự kiện kích hoạt	Người dùng chọn chức năng đổi mật khẩu.	
Điều kiện tiên quyết	Admin/ Nhân viên đã đăng nhập vào hệ thống	
Kết quả	Mật khẩu được đổi thành công	
Luồng sự kiện	Tác nhân	Hệ thống
	<ul style="list-style-type: none"> Người dùng chọn mục “Đổi mật khẩu” 	<ul style="list-style-type: none"> Hệ thống hiển thị giao diện đổi mật khẩu
	<ul style="list-style-type: none"> Người dùng nhập mật khẩu cũ và mới 	
	<ul style="list-style-type: none"> Người dùng ấn nút “Xác nhận” 	<ul style="list-style-type: none"> Hệ thống kiểm tra mật khẩu cũ và mới. Nếu thành công thì thông báo “Thành công”
Ngoại lệ	Mật khẩu cũ không chính xác, hệ thống thông báo lỗi	

Mã use case	UC05	
Tên use case	Xem danh sách sản phẩm	
Ngữ cảnh	Người dùng muốn xem danh sách sản phẩm	
Mô tả	Người dùng truy cập vào danh mục sản phẩm để xem thông tin cơ bản như tên sản phẩm, giá cả, số lượng tồn kho, và trạng thái sản phẩm	
Tác nhân	Admin/ Nhân viên	
Sự kiện kích hoạt	Người dùng chọn chức năng xem danh sách sản phẩm	
Điều kiện tiên quyết	Admin/ Nhân viên đã đăng nhập vào hệ thống	
Kết quả	Hiển thị danh sách sản phẩm thành công	
Luồng sự kiện	Tác nhân	Hệ thống
	<ul style="list-style-type: none"> Người dùng chọn mục “Danh sách sản phẩm” 	<ul style="list-style-type: none"> Hệ thống hiển thị giao diện danh sách sản phẩm
	<ul style="list-style-type: none"> Người dùng có thể tìm kiếm hoặc lọc sản phẩm 	<ul style="list-style-type: none"> Hệ thống trả về kết quả danh sách theo tiêu chí tìm kiếm hoặc lọc
Ngoại lệ	Nếu không có sản phẩm nào trong hệ thống, hiển thị thông báo "Không có sản phẩm nào".	

Bảng 2.5: Đặc tả use case xem danh sách sản phẩm

Bảng 2.6: Đặc tả use case xem báo cáo

Mã use case	UC06	
Tên use case	Xem báo cáo	
Ngữ cảnh	Người dùng muốn xem báo cáo	
Mô tả	Hệ thống cung cấp các báo cáo về doanh số, sản phẩm bán chạy	
Tác nhân	Admin/ Nhân viên	
Sự kiện kích hoạt	Người dùng chọn chức năng xem báo cáo.	
Điều kiện tiên quyết	Admin/ Nhân viên đã đăng nhập vào hệ thống	
Kết quả	Báo cáo hiển thị thành công	
Luồng sự kiện	Tác nhân	Hệ thống
	<ul style="list-style-type: none"> Người dùng chọn mục “Báo cáo” 	<ul style="list-style-type: none"> Hệ thống hiển thị giao diện báo cáo tổng quát
	<ul style="list-style-type: none"> Người dùng chọn loại báo cáo (doanh số, sản phẩm) 	<ul style="list-style-type: none"> Hệ thống hiển thị báo cáo chi tiết tương ứng.
Ngoại lệ	Nếu không có dữ liệu báo cáo, hiển thị thông báo "Không có dữ liệu".	

Bảng 2.7: Đặc tả use case tạo tài khoản nhân viên mới

Mã use case	UC07	
Tên use case	Tạo tài khoản nhân viên mới	
Ngữ cảnh	Admin muốn thêm nhân viên mới vào hệ thống.	
Mô tả	Admin nhập thông tin tài khoản nhân viên và gửi email đăng nhập cho họ.	
Tác nhân	Admin	
Sự kiện kích hoạt	Admin chọn chức năng thêm nhân viên mới.	
Điều kiện tiên quyết	Admin đã đăng nhập vào hệ thống	
Kết quả	Nhân viên mới được tạo thành công và nhận email đăng nhập lần đầu.	
Luồng sự kiện	Tác nhân	Hệ thống
	<ul style="list-style-type: none"> Admin chọn mục “Thêm nhân viên mới”. 	<ul style="list-style-type: none"> Hệ thống hiển thị giao diện nhập thông tin.
	<ul style="list-style-type: none"> Admin nhập thông tin nhân viên (họ tên, email, vai trò,...). 	
	<ul style="list-style-type: none"> Admin nhấn nút “Lưu”. 	<ul style="list-style-type: none"> Hệ thống lưu thông tin và gửi email đăng nhập cho nhân viên.
Ngoại lệ	Nếu email không hợp lệ, hiển thị thông báo lỗi.	

Bảng 2.8: Đặc tả use case xem danh sách nhân viên

Mã use case	UC08	
Tên use case	Xem danh sách nhân viên	
Ngữ cảnh	Admin muốn xem danh sách các nhân viên hiện có.	
Mô tả	Admin truy cập danh mục nhân viên để quản lý thông tin.	
Tác nhân	Admin	
Sự kiện kích hoạt	Admin chọn chức năng xem danh sách nhân viên.	
Điều kiện tiên quyết	Admin đã đăng nhập vào hệ thống	
Kết quả	Hiển thị danh sách nhân viên.	
Luồng sự kiện	Tác nhân	Hệ thống
	<ul style="list-style-type: none"> Admin chọn mục “Danh sách nhân viên”. 	<ul style="list-style-type: none"> Hệ thống hiển thị danh sách nhân viên với thông tin cơ bản (tên, email, vai trò, trạng thái).
Ngoại lệ	Nếu không có nhân viên nào, hiển thị thông báo "Không có nhân viên nào".	

Bảng 2.9: Đặc tả use case xem chi tiết nhân viên

Mã use case	UC09	
Tên use case	Xem chi tiết nhân viên	
Ngữ cảnh	Admin muốn xem chi tiết thông tin một nhân viên.	
Mô tả	Admin chọn nhân viên từ danh sách để xem thông tin chi tiết.	
Tác nhân	Admin	
Sự kiện kích hoạt	Admin chọn chức năng xem chi tiết nhân viên.	
Điều kiện tiên quyết	Admin đã đăng nhập vào hệ thống	
Kết quả	Hiển thị chi tiết thông tin nhân viên.	
Luồng sự kiện	Tác nhân	Hệ thống
	<ul style="list-style-type: none"> Admin chọn một nhân viên trong danh sách. 	<ul style="list-style-type: none"> Hệ thống hiển thị thông tin chi tiết của nhân viên đó.
Ngoại lệ	Không có	

Bảng 2.10: Đặc tả use case khóa/mở khóa tài khoản nhân viên

Mã use case	UC10	
Tên use case	Khóa/Mở khóa tài khoản	
Ngữ cảnh	Admin muốn khóa hoặc mở khóa tài khoản của một nhân viên.	
Mô tả	Admin thay đổi trạng thái tài khoản nhân viên (khóa hoặc mở khóa).	
Tác nhân	Admin	
Sự kiện kích hoạt	Admin chọn chức năng khóa/mở khóa tài khoản.	
Điều kiện tiên quyết	Admin đã đăng nhập vào hệ thống	
Kết quả	Trạng thái tài khoản nhân viên được cập nhật.	
Luồng sự kiện	Tác nhân	Hệ thống
	<ul style="list-style-type: none"> Admin chọn nhân viên từ danh sách. 	
	<ul style="list-style-type: none"> Admin nhấn nút “Khóa/Mở khóa”. 	<ul style="list-style-type: none"> Hệ thống cập nhật trạng thái tài khoản nhân viên.
Ngoại lệ	Không có	

Bảng 2.11: Đặc tả use case thêm sản phẩm mới

Mã use case	UC11	
Tên use case	Thêm sản phẩm mới	
Ngữ cảnh	Admin muốn thêm sản phẩm mới vào hệ thống.	
Mô tả	Admin nhập thông tin sản phẩm và lưu vào hệ thống.	
Tác nhân	Admin	
Sự kiện kích hoạt	Admin chọn chức năng thêm sản phẩm.	
Điều kiện tiên quyết	Admin đã đăng nhập vào hệ thống	
Kết quả	Sản phẩm mới được thêm thành công	
Luồng sự kiện	Tác nhân	Hệ thống
	<ul style="list-style-type: none"> Admin chọn mục “Thêm sản phẩm mới”. 	<ul style="list-style-type: none"> Hệ thống hiển thị giao diện nhập thông tin sản phẩm.
	<ul style="list-style-type: none"> Admin nhập thông tin sản phẩm (tên, giá, số lượng, mã barcode,...). 	
	<ul style="list-style-type: none"> Admin nhấn nút “Lưu”. 	<ul style="list-style-type: none"> Hệ thống lưu sản phẩm vào cơ sở dữ liệu.
Ngoại lệ	Nếu thông tin không hợp lệ, hiển thị thông báo lỗi.	

Bảng 2.12: Đặc tả use case cập nhật thông tin sản phẩm

Mã use case	UC12	
Tên use case	Cập nhật thông tin sản phẩm	
Ngữ cảnh	Admin muốn thay đổi thông tin của một sản phẩm trong hệ thống.	
Mô tả	Admin chỉnh sửa thông tin sản phẩm như tên, giá, hoặc số lượng tồn kho.	
Tác nhân	Admin	
Sự kiện kích hoạt	Admin chọn chức năng cập nhật sản phẩm.	
Điều kiện tiên quyết	Admin đã đăng nhập vào hệ thống. Sản phẩm đã tồn tại trong hệ thống.	
Kết quả	Thông tin sản phẩm được cập nhật thành công.	
Luồng sự kiện	Tác nhân	Hệ thống
	<ul style="list-style-type: none"> Admin chọn sản phẩm cần cập nhật từ danh sách. 	<ul style="list-style-type: none"> Hệ thống hiển thị giao diện chỉnh sửa thông tin sản phẩm.
	<ul style="list-style-type: none"> Admin thay đổi các thông tin cần thiết. 	
	<ul style="list-style-type: none"> Admin nhấn nút “Lưu”. 	<ul style="list-style-type: none"> Hệ thống lưu thông tin cập nhật và hiển thị thông báo thành công.
Ngoại lệ	Nếu thông tin không hợp lệ, hiển thị thông báo lỗi.	

Bảng 2.13: Đặc tả use case xóa sản phẩm

Mã use case	UC13	
Tên use case	Xóa sản phẩm	
Ngữ cảnh	Admin muốn xóa một sản phẩm khỏi hệ thống.	
Mô tả	Admin xóa một sản phẩm không còn sử dụng hoặc đã hết hàng.	
Tác nhân	Admin	
Sự kiện kích hoạt	Admin chọn chức năng xóa sản phẩm.	
Điều kiện tiên quyết	Sản phẩm tồn tại trong hệ thống. Sản phẩm không có trong bất kỳ đơn đặt hàng nào. Admin đã đăng nhập vào hệ thống.	
Kết quả	Sản phẩm bị xóa thành công	
Luồng sự kiện	Tác nhân	Hệ thống
	<ul style="list-style-type: none"> Admin chọn sản phẩm cần xóa từ danh sách. 	<ul style="list-style-type: none"> Hệ thống hiển thị thông báo xác nhận xóa.
	<ul style="list-style-type: none"> Admin nhấn “Xác nhận”. 	<ul style="list-style-type: none"> Hệ thống xóa sản phẩm và hiển thị thông báo thành công.
Ngoại lệ	Nếu sản phẩm liên quan đến các hóa đơn hoặc giao dịch, hệ thống không cho phép xóa và hiển thị thông báo lỗi.	

Bảng 2.14: Đặc tả use case thêm khách hàng mới

Mã use case	UC14	
Tên use case	Thêm khách hàng mới	
Ngữ cảnh	Nhân viên muốn thêm thông tin khách hàng vào hệ thống.	
Mô tả	Nhân viên nhập thông tin khách hàng mới để lưu hồ sơ và tạo giao dịch.	
Tác nhân	Nhân viên	
Sự kiện kích hoạt	Nhân viên chọn chức năng thêm khách hàng.	
Điều kiện tiên quyết	Nhân viên đã đăng nhập vào hệ thống. Tài khoản nhân viên không bị khóa.	
Kết quả	Thông tin khách hàng được lưu thành công.	
Luồng sự kiện	Tác nhân	Hệ thống
	<ul style="list-style-type: none"> Nhân viên nhập thông tin (tên, số điện thoại, email,...) khi tạo giao dịch mới. 	
	<ul style="list-style-type: none"> Nhân viên nhấn “Lưu”. 	<ul style="list-style-type: none"> Hệ thống lưu thông tin khách hàng vào cơ sở dữ liệu.
Ngoại lệ	Nếu thông tin không hợp lệ, hiển thị thông báo lỗi.	

Bảng 2.15: Đặc tả use case xem thông tin khách hàng

Mã use case	UC15	
Tên use case	Xem thông tin khách hàng	
Ngữ cảnh	Nhân viên muốn tra cứu thông tin chi tiết của một khách hàng.	
Mô tả	Hệ thống cho phép nhân viên tìm kiếm và hiển thị thông tin khách hàng.	
Tác nhân	Nhân viên	
Sự kiện kích hoạt	Nhân viên chọn chức năng xem thông tin khách hàng	
Điều kiện tiên quyết	Nhân viên đã đăng nhập vào hệ thống. Tài khoản nhân viên không bị khóa.	
Kết quả	Hiển thị thông tin chi tiết của khách hàng.	
Luồng sự kiện	Tác nhân	Hệ thống
	<ul style="list-style-type: none"> Nhân viên nhập số điện thoại khách hàng vào ô tìm kiếm. 	<ul style="list-style-type: none"> Hệ thống hiển thị kết quả.
Ngoại lệ	Nếu không tìm thấy khách hàng, hệ thống hiển thị thông báo "Không tìm thấy khách hàng".	

Bảng 2.16: Đặc tả use case xem lịch sử mua hàng của khách hàng

Mã use case	UC16	
Tên use case	Xem lịch sử mua hàng của khách hàng	
Ngữ cảnh	Nhân viên muốn kiểm tra lịch sử giao dịch của một khách hàng.	
Mô tả	Hệ thống hiển thị danh sách các giao dịch trước đây của khách hàng.	
Tác nhân	Nhân viên	
Sự kiện kích hoạt	Nhân viên chọn chức năng xem lịch sử mua hàng.	
Điều kiện tiên quyết	Nhân viên đã đăng nhập vào hệ thống. Tài khoản nhân viên không bị khóa. Khách hàng tồn tại trong hệ thống.	
Kết quả	Hiển thị danh sách các giao dịch của khách hàng.	
Luồng sự kiện	Tác nhân	Hệ thống
	<ul style="list-style-type: none"> Nhân viên tìm kiếm khách hàng cần kiểm tra. 	<ul style="list-style-type: none"> Hệ thống hiển thị danh sách giao dịch của khách hàng đó.
	<ul style="list-style-type: none"> Nhân viên chọn giao dịch để xem chi tiết. 	
Ngoại lệ	Nếu khách hàng chưa có giao dịch nào, hệ thống hiển thị thông báo "Không có lịch sử giao dịch".	

Bảng 2.17: Đặc tả use case tạo đơn hàng mới

Mã use case	UC17	
Tên use case	Tạo đơn hàng mới	
Ngữ cảnh	Nhân viên muốn tạo đơn hàng mới	
Mô tả	Nhân viên nhập thông tin khách hàng để lưu hồ sơ và tạo giao dịch.	
Tác nhân	Nhân viên	
Sự kiện kích hoạt	Nhân viên chọn chức năng tạo đơn hàng mới	
Điều kiện tiên quyết	Nhân viên đã đăng nhập vào hệ thống. Tài khoản nhân viên không bị khóa. Khách hàng tồn tại	
Kết quả	Đơn hàng mới được tạo thành công.	
Luồng sự kiện	Tác nhân	Hệ thống
	<ul style="list-style-type: none"> Nhân viên nhập thông tin khách hàng (tên, số điện thoại, email,...) khi tạo đơn hàng mới. 	<ul style="list-style-type: none"> Hệ thống hiển thị đơn hàng.
	<ul style="list-style-type: none"> Nhân viên nhấn “Lưu”. 	<ul style="list-style-type: none"> Hệ thống thông báo tạo thành công.
Ngoại lệ	Nếu thông tin không hợp lệ, hiển thị thông báo lỗi.	

Bảng 2.18: Đặc tả use case xem đơn hàng theo số điện thoại

Mã use case	UC18	
Tên use case	Xem đơn hàng theo số điện thoại	
Ngữ cảnh	Nhân viên muốn tra cứu đơn hàng dựa trên số điện thoại của khách hàng.	
Mô tả	Hệ thống hiển thị danh sách các đơn hàng liên kết với số điện thoại được cung cấp.	
Tác nhân	Nhân viên	
Sự kiện kích hoạt	Nhân viên chọn chức năng xem đơn hàng theo số điện thoại.	
Điều kiện tiên quyết	Nhân viên đã đăng nhập vào hệ thống. Tài khoản nhân viên không bị khóa. Khách hàng tồn tại	
Kết quả	Hiển thị danh sách các đơn hàng của khách hàng.	
Luồng sự kiện	Tác nhân	Hệ thống
	<ul style="list-style-type: none"> Nhân viên nhập số điện thoại của khách hàng vào ô tìm kiếm. 	<ul style="list-style-type: none"> Hệ thống kiểm tra và hiển thị danh sách các đơn hàng liên quan đến số điện thoại.
	<ul style="list-style-type: none"> Nhân viên chọn một đơn hàng từ danh sách để xem chi tiết (nếu cần). 	
Ngoại lệ	Nếu không tìm thấy đơn hàng liên quan đến số điện thoại, hiển thị thông báo "Không tìm thấy đơn hàng".	

Bảng 2.19: Đặc tả use case xem chi tiết đơn hàng

Mã use case	UC19	
Tên use case	Xem chi tiết đơn hàng	
Ngữ cảnh	Nhân viên muốn kiểm tra thông tin chi tiết của một đơn hàng.	
Mô tả	Hệ thống hiển thị chi tiết đơn hàng bao gồm khách hàng, sản phẩm, và trạng thái.	
Tác nhân	Nhân viên	
Sự kiện kích hoạt	Nhân viên chọn chức năng xem chi tiết đơn hàng.	
Điều kiện tiên quyết	Nhân viên đã đăng nhập vào hệ thống. Tài khoản nhân viên không bị khóa. Khách hàng tồn tại Đơn hàng tồn tại	
Kết quả	Hiển thị thông tin chi tiết của đơn hàng.	
Luồng sự kiện	Tác nhân	Hệ thống
	<ul style="list-style-type: none"> Nhân viên tìm kiếm đơn hàng bằng số điện thoại khách hàng. 	<ul style="list-style-type: none"> Hệ thống hiển thị danh sách các đơn hàng phù hợp.
	<ul style="list-style-type: none"> Nhân viên chọn một đơn hàng để xem. 	<ul style="list-style-type: none"> Hệ thống hiển thị thông tin chi tiết của đơn hàng đó (khách hàng, sản phẩm, trạng thái, ngày tạo).
Ngoại lệ	Nếu không tìm thấy đơn hàng, hiển thị thông báo "Không tìm thấy đơn hàng".	

Bảng 2.20: Đặc tả use case cập nhật đơn hàng

Mã use case	UC20	
Tên use case	Cập nhật đơn hàng	
Ngữ cảnh	Nhân viên muốn chỉnh sửa thông tin của một đơn hàng hiện có.	
Mô tả	Hệ thống cho phép nhân viên cập nhật thông tin đơn hàng.	
Tác nhân	Nhân viên	
Sự kiện kích hoạt	Nhân viên chọn chức năng cập nhật đơn hàng.	
Điều kiện tiên quyết	Nhân viên đã đăng nhập và không bị khóa tài khoản Khách hàng và đơn hàng tồn tại	
Kết quả	Đơn hàng được cập nhật thành công	
Luồng sự kiện	Tác nhân	Hệ thống
	<ul style="list-style-type: none"> Nhân viên tìm kiếm và chọn đơn hàng cần chỉnh sửa. 	<ul style="list-style-type: none"> Hệ thống hiển thị thông tin chi tiết của đơn hàng.
	<ul style="list-style-type: none"> Nhân viên thực hiện các thay đổi cần thiết như trạng thái đơn hàng, thêm/xóa/sửa sản phẩm trong đơn, cập nhật thông tin khách hàng. 	<ul style="list-style-type: none"> Hệ thống hiển thị thông tin chi tiết của đơn hàng đó (khách hàng, sản phẩm, trạng thái, ngày tạo).
	<ul style="list-style-type: none"> Nhân viên nhấn “Lưu”. 	<ul style="list-style-type: none"> Hệ thống lưu các thay đổi và hiển thị thông báo thành công.
Ngoại lệ	Nếu thông tin không hợp lệ, hệ thống sẽ báo lỗi	

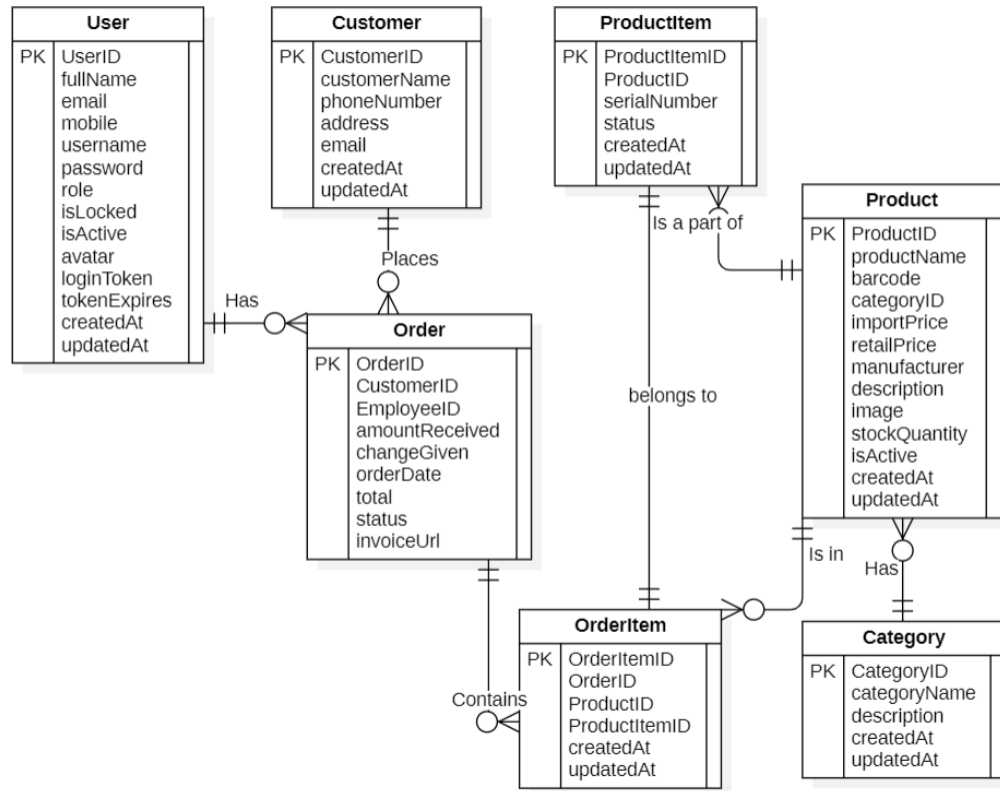
Bảng 2.21: Đặc tả use case thanh toán đơn hàng

Mã use case	UC21	
Tên use case	Thanh toán đơn hàng	
Ngữ cảnh	Nhân viên muốn thực hiện thanh toán cho khách hàng.	
Mô tả	Hệ thống xử lý đơn hàng và in hóa đơn sau khi thanh toán thành công.	
Tác nhân	Nhân viên	
Sự kiện kích hoạt	Nhân viên chọn chức năng thanh toán.	
Điều kiện tiên quyết	Nhân viên đã đăng nhập và không bị khóa tài khoản Đơn hàng đã được tạo và sẵn sàng thanh toán	
Kết quả	Thanh toán thành công, hóa đơn được in và gửi về email khách hàng	
Luồng sự kiện	Tác nhân	Hệ thống
	<ul style="list-style-type: none"> Nhân viên chọn đơn hàng cần thanh toán. 	<ul style="list-style-type: none"> Hệ thống hiển thị thông tin chi tiết của đơn hàng.
	<ul style="list-style-type: none"> Nhân viên nhấn “Xác nhận”. 	<ul style="list-style-type: none"> Hệ thống xử lý giao dịch và cập nhật trạng thái đơn hàng.
		<ul style="list-style-type: none"> Hệ thống in hóa đơn, gửi về email khách hàng và thông báo thành công.
Ngoại lệ	Nếu thông tin không hợp lệ, hệ thống sẽ báo lỗi	

Bảng 2.22: Đặc tả use case báo cáo và phân tích

Mã use case	UC22	
Tên use case	Báo cáo và phân tích	
Ngữ cảnh	Admin và nhân viên muốn xem báo cáo và phân tích	
Mô tả	Hệ thống thực hiện truy vấn và hiển thị báo cáo, phân tích tình hình kinh doanh của hệ thống.	
Tác nhân	Admin, Nhân viên	
Sự kiện kích hoạt	Người dùng chọn chức năng “Báo cáo, phân tích”	
Điều kiện tiên quyết	Người dùng đã đăng nhập	
Kết quả	Thanh toán thành công, hóa đơn được in và gửi về email khách hàng	
Luồng sự kiện	Tác nhân	Hệ thống
	<ul style="list-style-type: none"> Người dùng chọn chức năng “Báo cáo, phân tích” 	<ul style="list-style-type: none"> Hiển thị báo cáo doanh thu, danh sách đơn hàng
	<ul style="list-style-type: none"> Người dùng chọn mốc thời gian và nhấn “Xác nhận” 	<ul style="list-style-type: none"> Hệ thống xử lý và trả về giao diện báo cáo tương ứng với mốc thời gian đã chọn.
Ngoại lệ	Nếu thông tin không hợp lệ, hệ thống sẽ báo lỗi	

2.4 Sơ đồ ERD



Hình 2.2 Lược đồ ERD

Hệ thống bao gồm các thực thể:

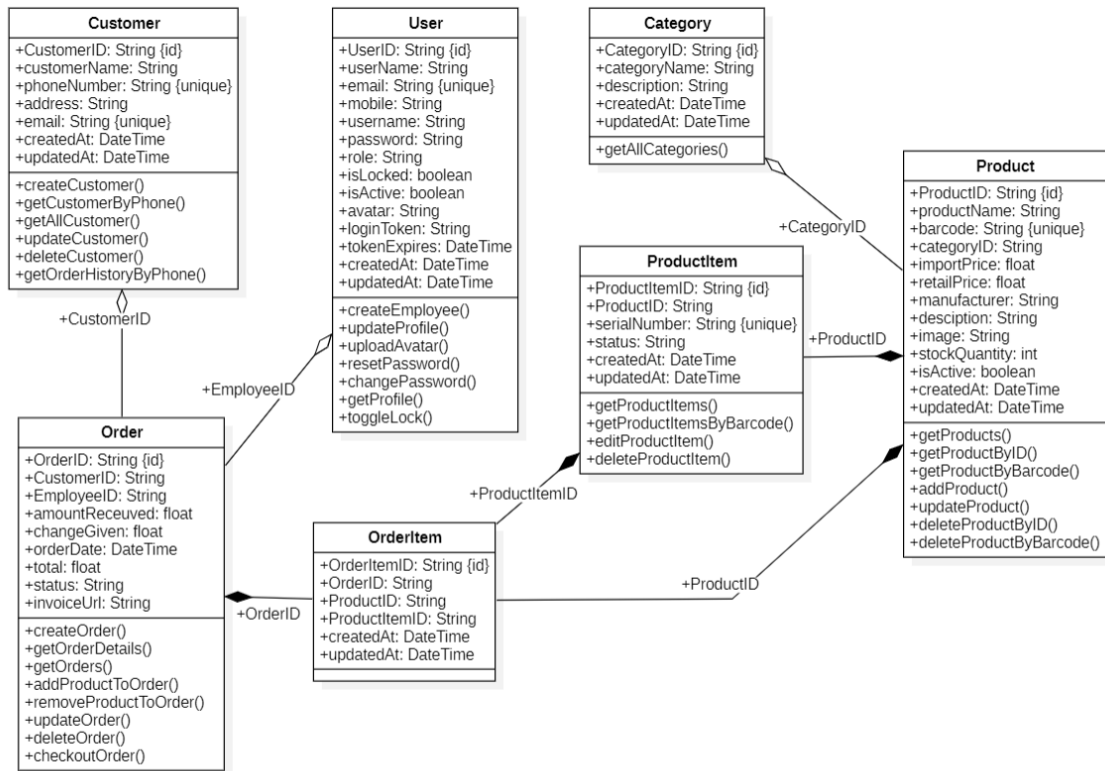
- **User**: Thực thể lưu trữ thông tin người dùng như tên, ngày sinh cũng như thông tin tài khoản như username, password,...
- **Customer**: Thực thể chứa thông tin cơ bản của khách hàng như họ tên, số điện thoại, địa chỉ nhà và địa chỉ email.
- **Category**: Thực thể dùng để phân loại mặt hàng được bán trên hệ thống.
- **Product**: Thực thể lưu trữ thông tin dòng sản phẩm đang có trên hệ thống.
- **ProductItem**: Thực thể đại diện cho từng sản phẩm có trong hệ thống.

- Order: Thực thể lưu trữ các thông tin của đơn mua hàng.
- OrderItem: Thực thể chứa thông tin chi tiết về những sản phẩm được mua trong đơn hàng.

Mối quan hệ giữa các thực thể:

- Quan hệ “One To Many” (1 - n):
 - Customer (1) có thể có nhiều đơn hàng Order (n), mỗi khách hàng có thể thực hiện nhiều đơn đặt hàng khác nhau.
 - User (hoặc Employee) (1) quản lý nhiều Order (n), mỗi đơn hàng sẽ liên kết với một người dùng hoặc nhân viên cụ thể.
 - Category (1) có nhiều Product (n), một danh mục có thể chứa nhiều sản phẩm khác nhau.
 - Order (1) có nhiều mục đơn hàng OrderItem (n), mỗi đơn hàng bao gồm nhiều mặt hàng.
 - Product (1) có thể có nhiều OrderItem (n) và nhiều ProductItem (n), mỗi sản phẩm có thể xuất hiện trong nhiều mục đơn hàng.

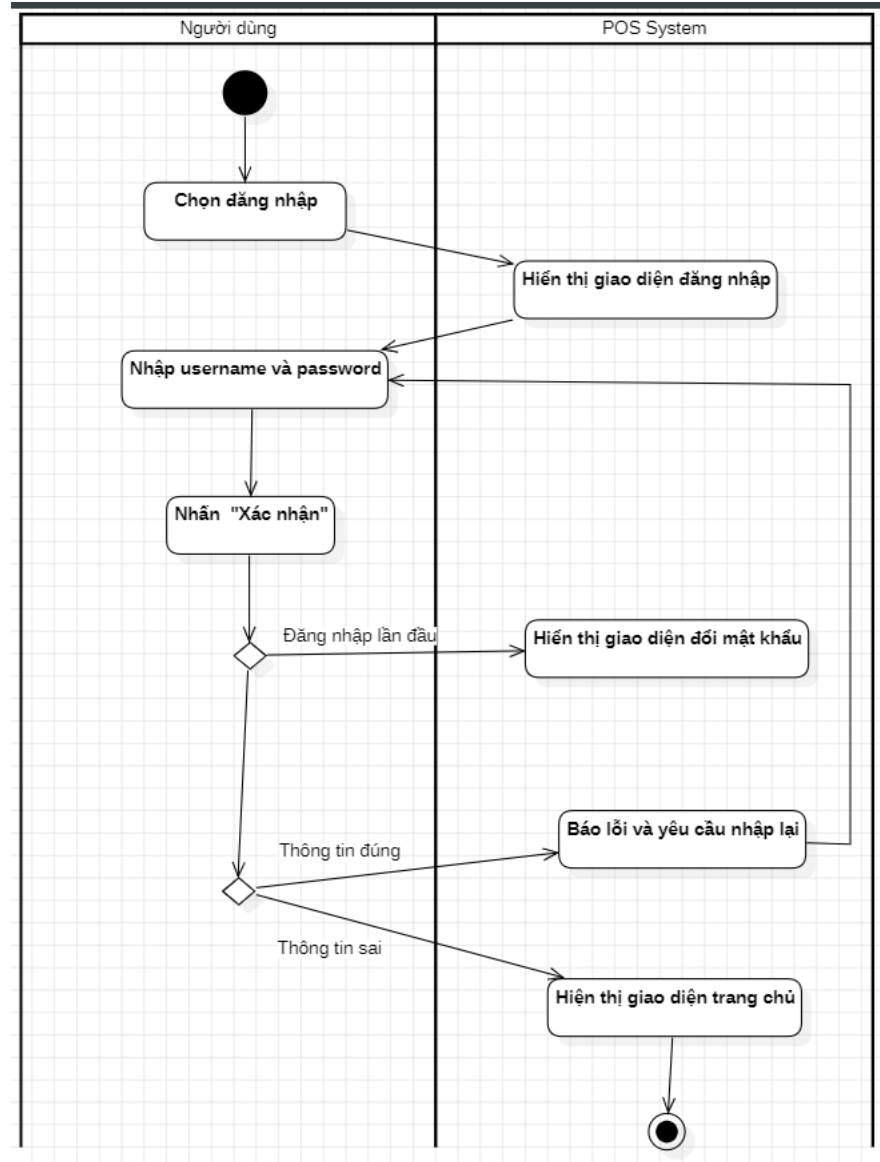
2.5 Sơ đồ Class



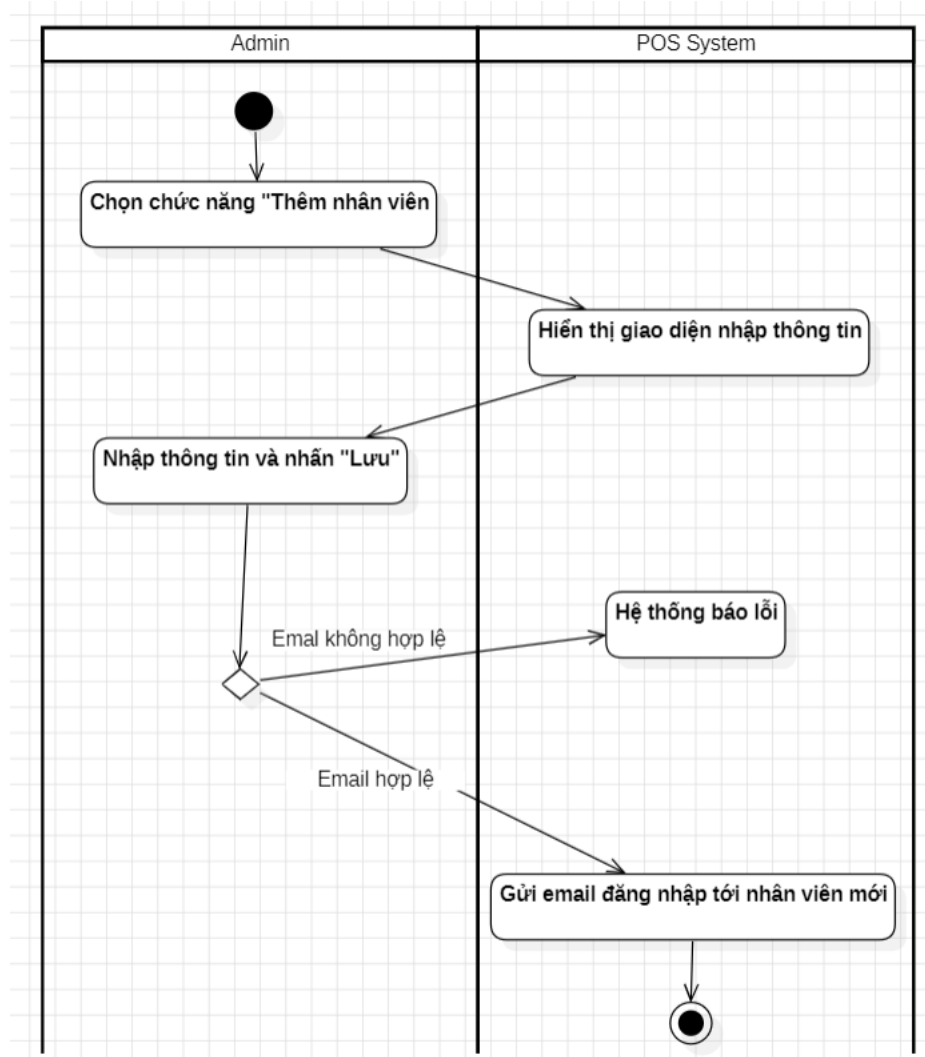
Hình 2.3 Sơ đồ Class

2.6 Sơ đồ Activity

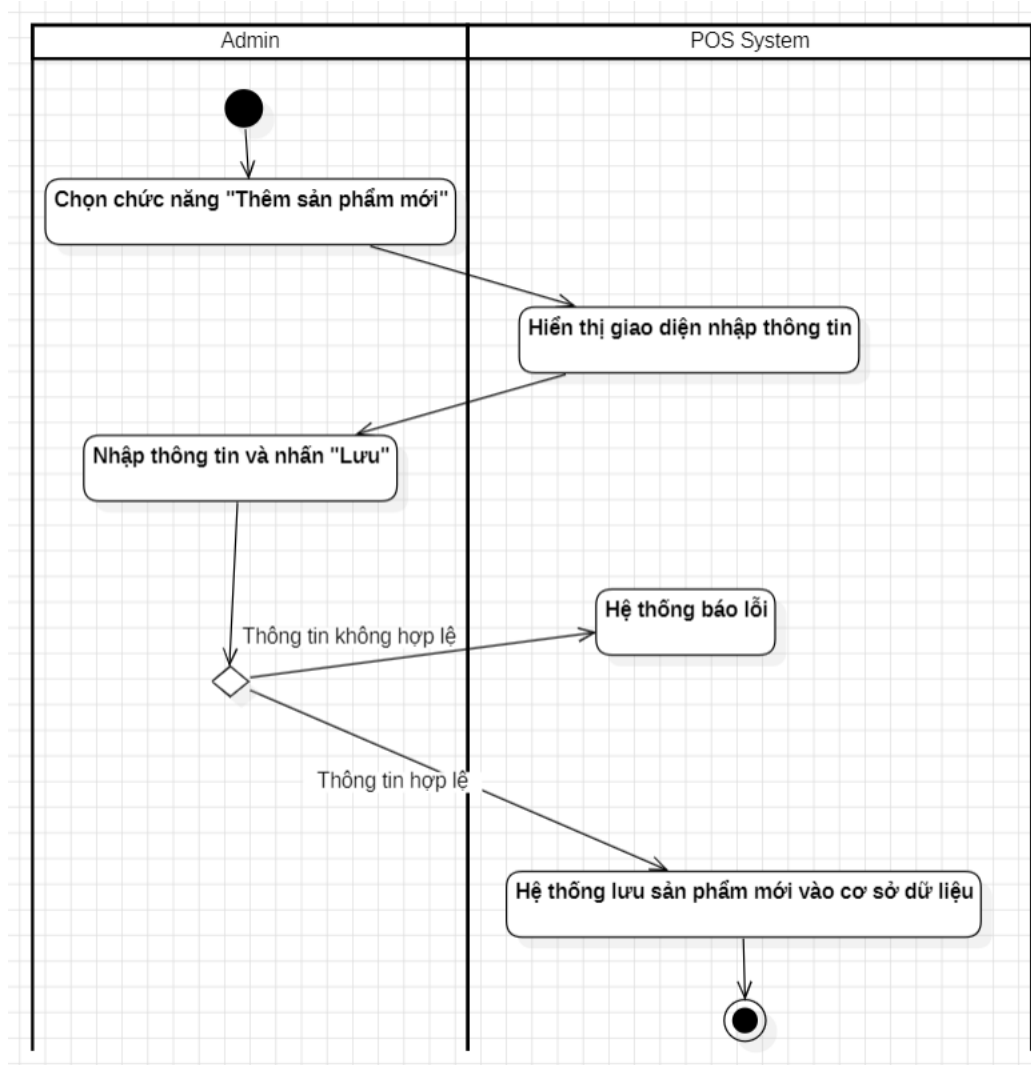
Sau đây là sơ đồ Activity cho một số use case quan trọng của hệ thống:



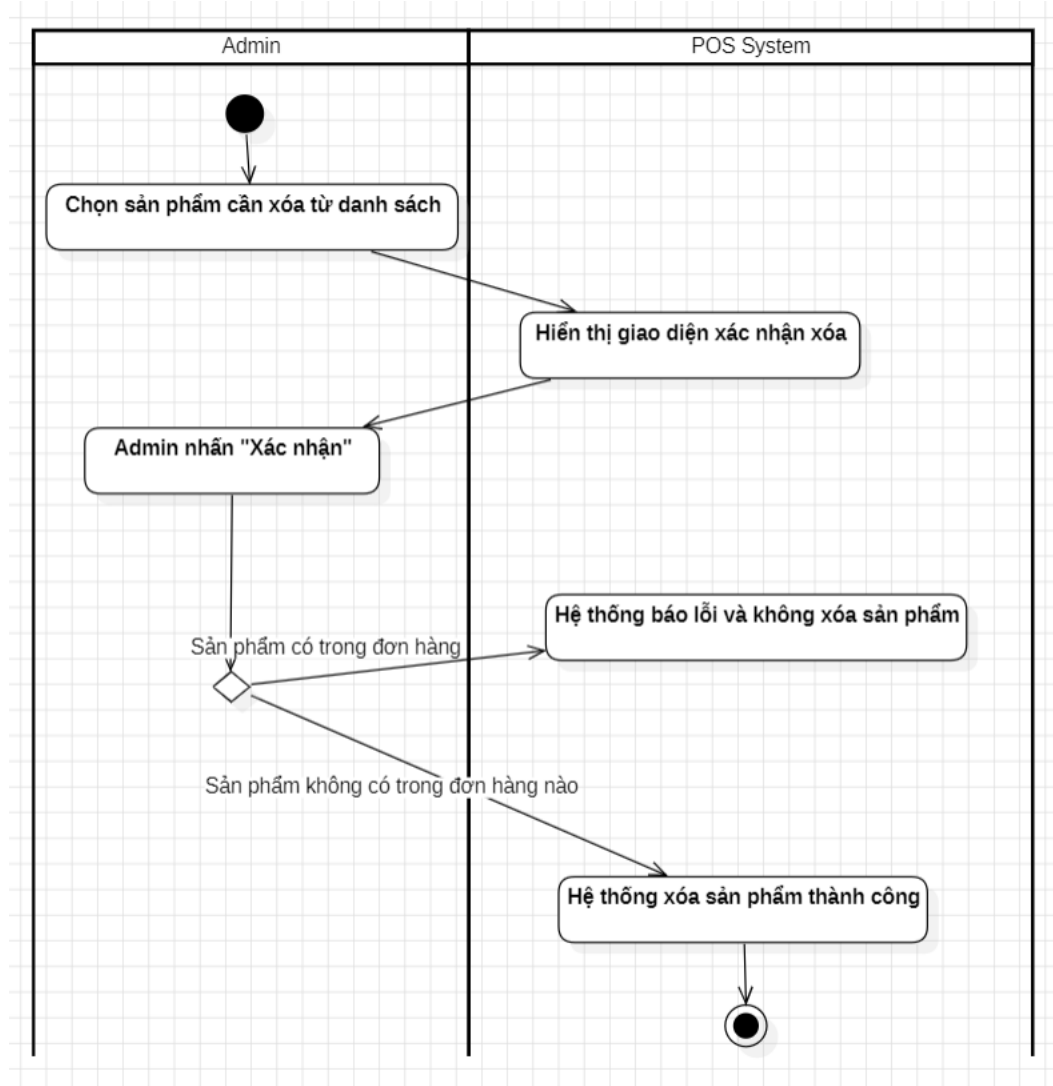
Hình 2.4 Sơ đồ Activity cho use case Đăng nhập



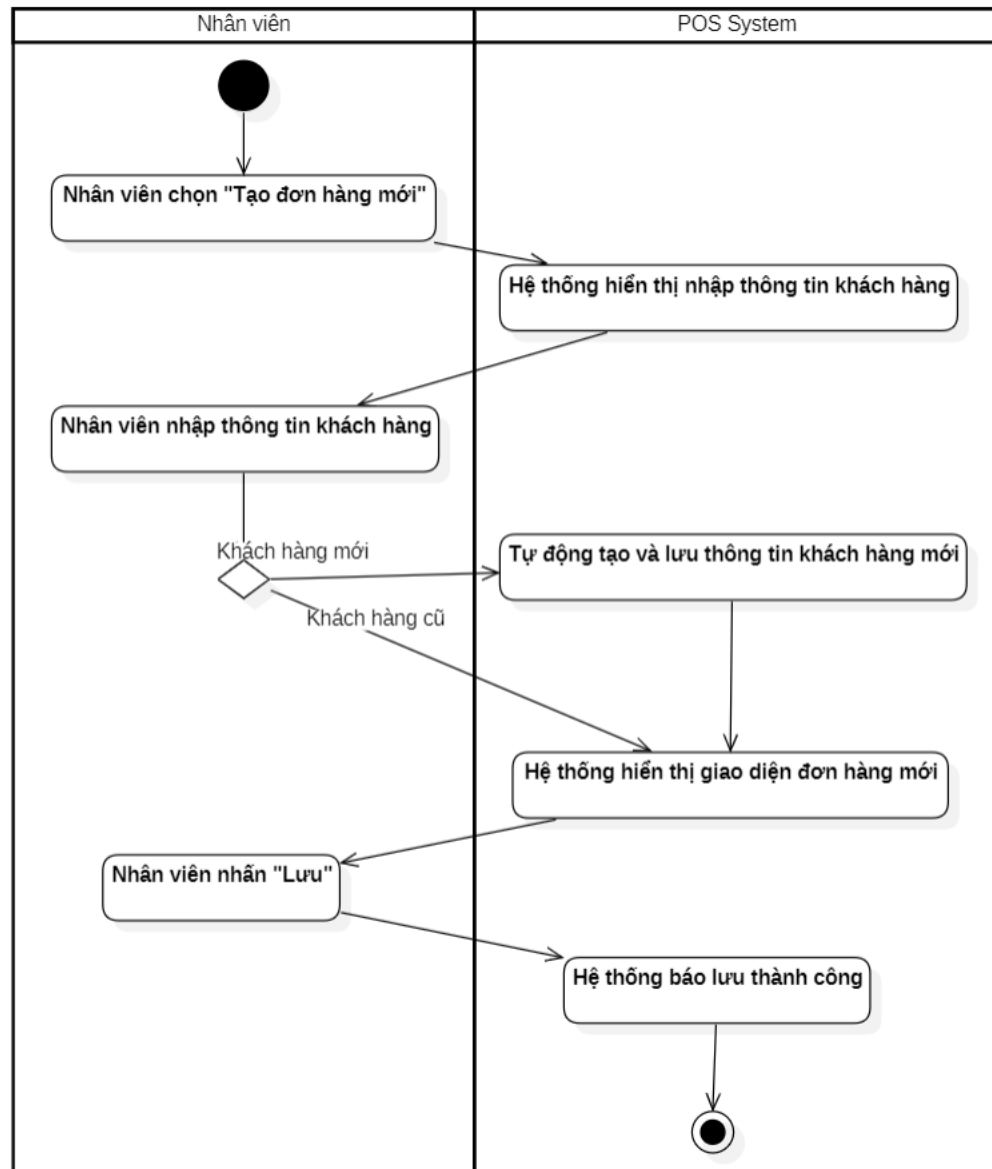
Hình 2.5 Sơ đồ Activity cho use case Tạo tài khoản nhân viên mới



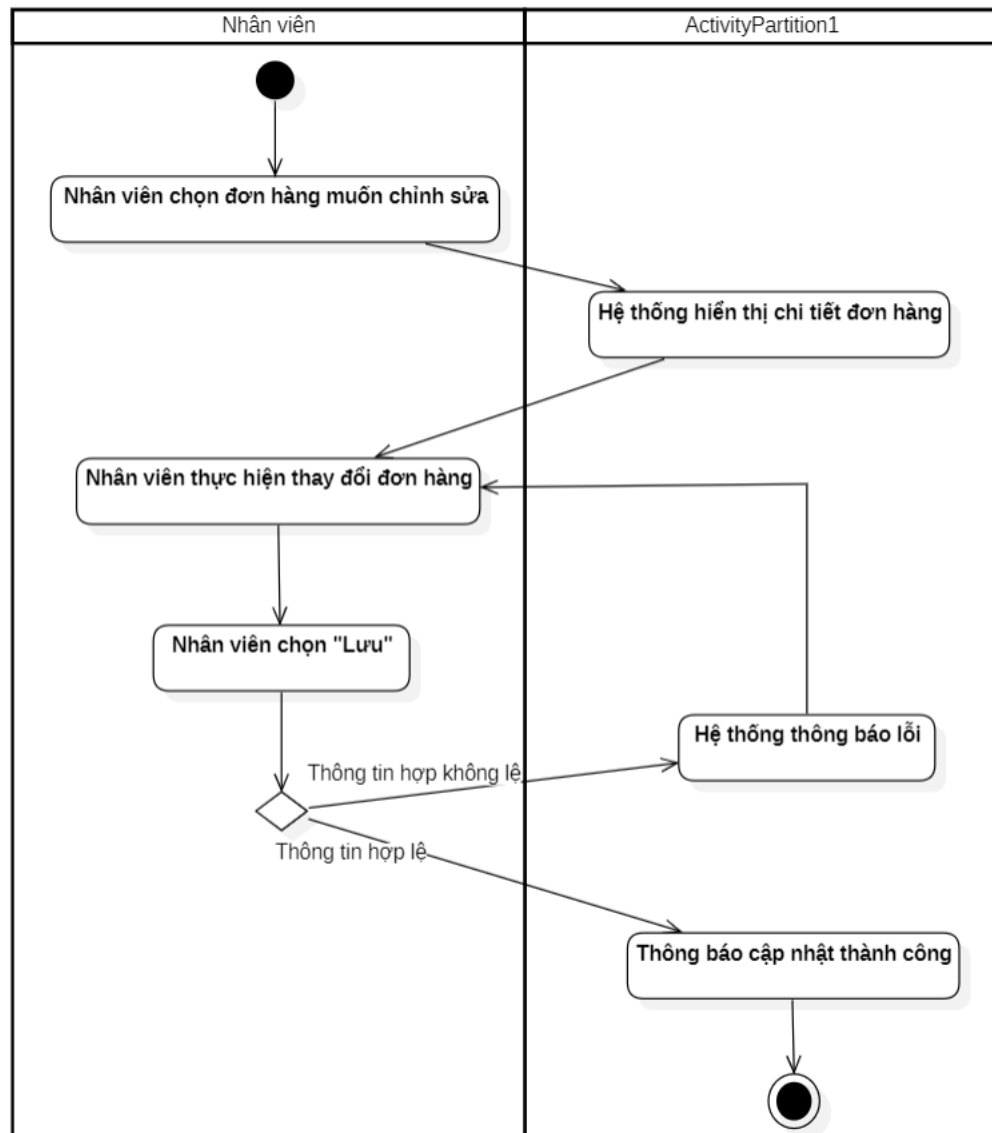
Hình 2.6 Sơ đồ Activity cho use case Thêm sản phẩm mới



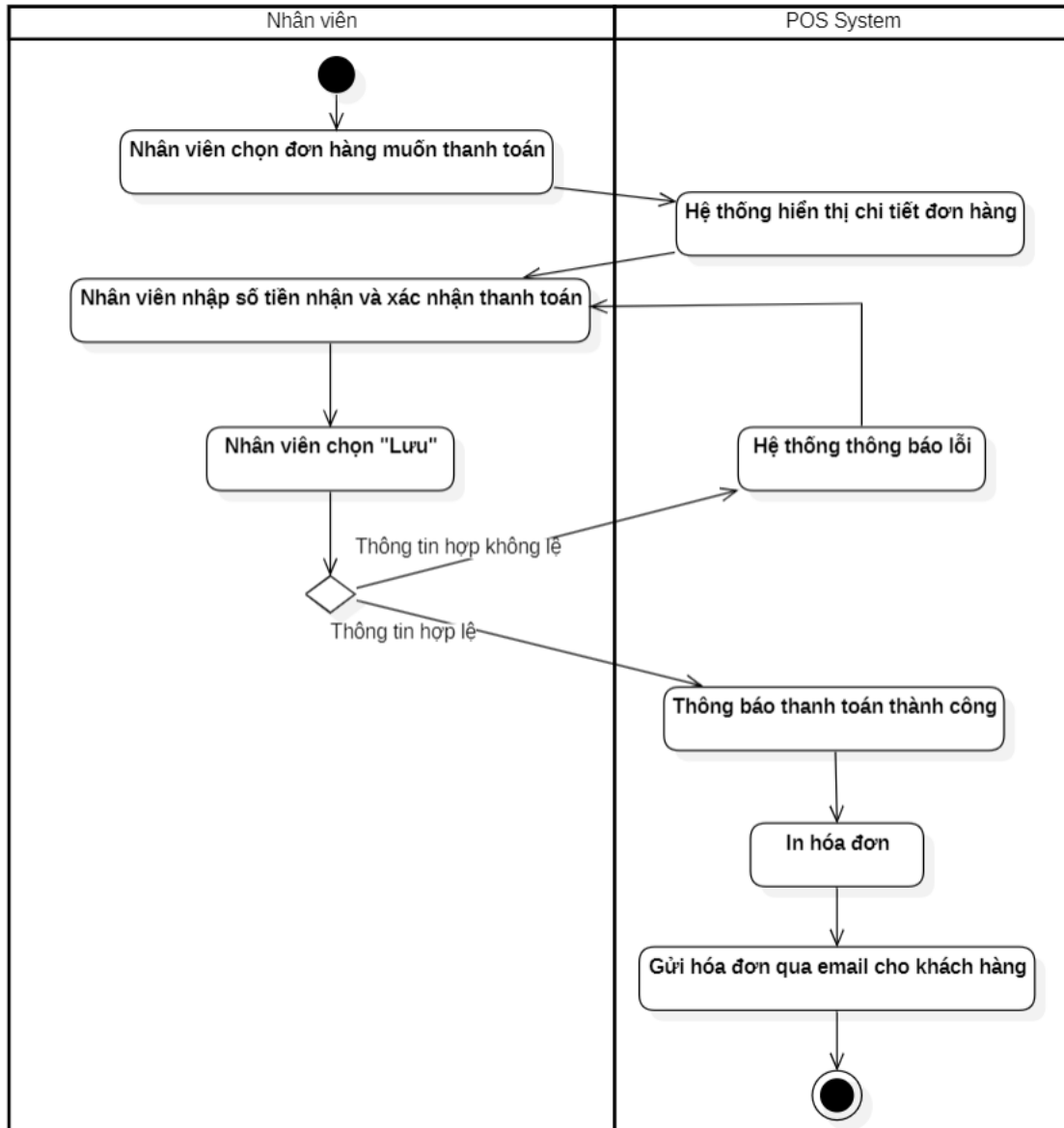
Hình 2.7 Sơ đồ Activity cho use case Xóa sản phẩm



Hình 2.8 Sơ đồ Activity cho use case Tạo đơn hàng mới



Hình 2.9 Sơ đồ Activity cho use case Cập nhật đơn hàng



Hình 2.10: Sơ đồ Activity cho use case Thanh toán đơn hàng

CHƯƠNG 3 – HIỆN THỰC HỆ THỐNG

3.1 Tổng quan hiện thực hệ thống

Mục tiêu chính của việc hiện thực hệ thống là xây dựng một website đáp ứng các yêu cầu chức năng và phi chức năng đã được xác định trong giai đoạn phân tích và thiết kế. Hệ thống sẽ cung cấp các tính năng như quản lý tài khoản, khách hàng, xử lý đơn hàng, báo cáo doanh thu.

3.1.1 Công cụ phát triển

IDE: Visual Code, Cursor

Hệ thống phiên bản quản lý: Git

3.1.2 Framework

Frontend: React.js

Backend: Node.js và Express.js

3.1.3 Ngôn ngữ lập trình

Javascript

3.1.4 Cơ sở dữ liệu

MongoDB Atlas cho lưu trữ dữ liệu trên đám mây

Redis cho caching

3.1.5 Môi trường phát triển

Sử dụng Docker để tạo môi trường phát triển đồng nhất

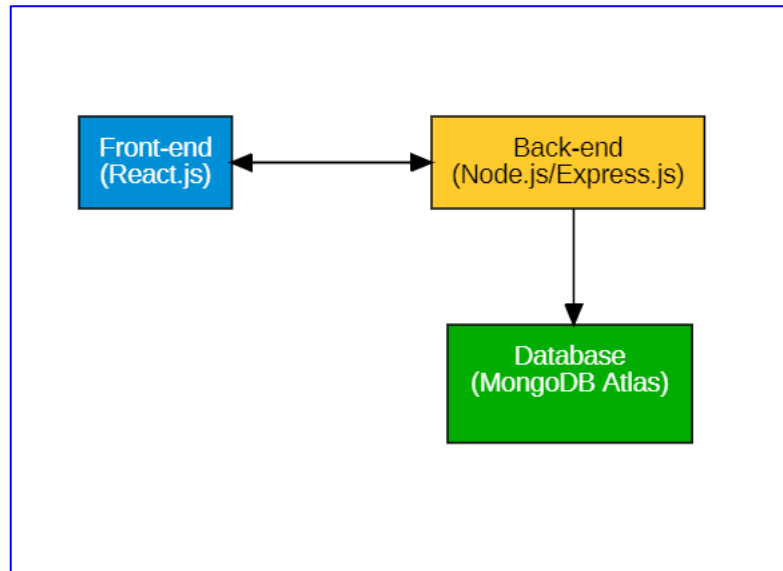
Sử dụng Postman và Swagger-ui để kiểm thử API

3.2 Kiến trúc hệ thống

Hệ thống được xây dựng theo kiến trúc Client-Server, trong đó frontend (client) sẽ giao tiếp với backend (server) thông qua các API RESTful. Backend sẽ xử lý logic nghiệp vụ và tương tác với cơ sở dữ liệu để lưu trữ và truy xuất dữ liệu.

3.2.1. Mô tả kiến trúc tổng thể

Sơ đồ kiến trúc tổng thể có thể được mô tả như sau:



Hình 3.1 Sơ đồ kiến trúc tổng thể của hệ thống

3.2.2 Các thành phần chính

- Front-end (Client):

- Vai trò: Giao diện người dùng, nơi người dùng tương tác với hệ thống. Front-end sẽ gửi yêu cầu đến back-end và hiển thị dữ liệu nhận được.
- Công nghệ: Sử dụng React.js để xây dựng giao diện người dùng động.

- Back-end (Server):

- Vai trò: Xử lý logic nghiệp vụ, quản lý dữ liệu và cung cấp API cho front-end. Back-end sẽ nhận yêu cầu từ front-end, xử lý và trả về kết quả.
- Công nghệ: Sử dụng Node.js với Express.js để xây dựng API RESTful.

- Cơ sở dữ liệu (Database):

- Vai trò: Lưu trữ dữ liệu của hệ thống, bao gồm thông tin người dùng, đơn hàng, sản phẩm, v.v.
- Công nghệ: Sử dụng MongoDB Atlas để lưu trữ dữ liệu trên cloud và Redis để caching.

- Môi liên hệ giữa các thành phần:

- Front-end gửi yêu cầu đến back-end thông qua các API.
- Back-end xử lý yêu cầu, tương tác với cơ sở dữ liệu để lưu trữ hoặc truy xuất dữ liệu.
- Kết quả từ back-end sẽ được trả về cho front-end để hiển thị cho người dùng.

3.3 Các đoạn mã nguồn quan trọng

3.3.1 Chức năng tạo tài khoản nhân viên mới

```
// Hàm tạo tài khoản nhân viên với error handling
const createEmployee = async (req, res, next) => {
  try {
    const { fullName, email, mobile, avatar } = req.body;
    if (!fullName || !email) {
      return next(new AppError('Fullname and Email is required', 400));
    }
    // Check if employee already exists
    const employeeExists = await User.findOne({ email });
    if (employeeExists) {
      return next(new AppError('Employee already exists', 400));
    }
    // Username và Password mặc định
    const username = email.split('@')[0];
    const password = username;

    // Tạo token đăng nhập và set thời gian hết hạn login token
    const loginToken = crypto.randomBytes(32).toString('hex');
    const tokenExpires = Date.now() + 60 * 1000;
  }
}
```

```

// Create the user in the database
const user = await User.create({
  fullName,
  email,
  mobile: mobile || '',
  password,
  username,
  role: 'employee',
  avatar: avatar || '',
  loginToken,
  tokenExpires,
});

// Generate login link and send email
const loginLink = `${req.protocol}://${req.get('host')}/api/auth/login/${loginToken}`;
const emailData = {
  to: user.email,
  subject: 'Login to POS',
  text: `Use this link to login: ${loginLink}. It expires in 1 minute.`,
  htmlContent: `<p>Use this link to login:
  |         <a href="${loginLink}">${loginLink}</a>. It expires in 1 minute.</p>`;
};
await sendEmail(emailData.to, emailData.subject, emailData.text, emailData.htmlContent);
await client.del(`employees_list`);

res.status(201).json({
  code: 201,
  success: true,
  message: 'Employee created successfully.',
  result: 'Login email sent.'
});
} catch (error) {
  return next(new AppError(`Error while creating new employee: ${error.message}`, 500));
}
;

```

Hình 3.2 Hàm tạo tài khoản nhân viên mới

Giải thích hàm:

1. Nhận dữ liệu từ yêu cầu:
 - Hàm nhận dữ liệu từ req.body, bao gồm fullName, email, mobile, và avatar.
2. Kiểm tra dữ liệu đầu vào:
 - Nếu fullName hoặc email không được cung cấp, hàm sẽ trả về lỗi 400 với thông báo 'Fullname and Email is required'.
3. Kiểm tra xem nhân viên đã tồn tại:

- Sử dụng `User.findOne({ email })` để kiểm tra xem có nhân viên nào đã tồn tại với địa chỉ email đó. Nếu có, hàm sẽ trả về lỗi 400 với thông báo 'Employee already exists'.

4. Tạo tên người dùng và mật khẩu:

- Tên người dùng (username) được tạo từ phần trước dấu '@' trong địa chỉ email.
- Mật khẩu mặc định được đặt bằng tên người dùng.

5. Tạo token đăng nhập:

- Sử dụng `crypto.randomBytes(32).toString('hex')` để tạo một token ngẫu nhiên cho việc đăng nhập.

- Thời gian hết hạn của token được đặt là 1 phút (60 giây).

6. Tạo người dùng trong cơ sở dữ liệu:

- Sử dụng `User.create({...})` để tạo một bản ghi người dùng mới trong cơ sở dữ liệu với các thông tin đã nhận và các thông tin đã tạo (username, password, loginToken, tokenExpires).

7. Tạo liên kết đăng nhập và gửi email:

- Tạo một liên kết đăng nhập (loginLink) sử dụng token đã tạo.
- Tạo một đối tượng emailData chứa thông tin email để gửi cho người dùng, bao gồm địa chỉ email, tiêu đề, nội dung văn bản và nội dung HTML.
- Gọi hàm `sendEmail` để gửi email đến người dùng.

8. Xóa cache:

- Gọi `await client.del('employees_list')` để xóa danh sách nhân viên trong cache, đảm bảo rằng danh sách được cập nhật.

9. Trả về phản hồi thành công:

- Nếu mọi thứ diễn ra suôn sẻ, hàm sẽ trả về phản hồi với mã trạng thái 201 và thông báo 'Employee created successfully.' cùng với thông tin rằng email đăng nhập đã được gửi.

10. Xử lý lỗi:

- Nếu có bất kỳ lỗi nào xảy ra trong quá trình thực hiện, hàm sẽ bắt lỗi và trả về một thông báo lỗi với mã trạng thái 500.

3.3.2 Chức năng đăng nhập

```
const login = async (req, res, next) => {
  try {
    const { username, password } = req.body;
    const user = await User.findOne({ username });
    if (user && user.role === 'employee' && user.loginToken) {
      return next(new AppError('Please login by clicking on the link in your email', 400));
    }
    if (user && (await bcrypt.compare(password, user.password))) {
      // Tạo token
      const token = generateToken(user._id);

      return res.status(200).json({
        code: 200,
        success: true,
        message: 'Login successful',
        result: {
          _id: user._id,
          fullName: user.fullName,
          email: user.email,
          token: token
        }
      });
    } else {
      return next(new AppError('Invalid email or password', 401));
    }
  } catch (error) {
    return next(new AppError(`Error while login: ${error.message}`, 500));
  }
};
```

Hình 3.3 Hàm đăng nhập

Giải thích hàm:

1. Nhận dữ liệu từ yêu cầu:

Hàm nhận dữ liệu từ req.body, bao gồm username và password.

2. Tìm kiếm người dùng:

Sử dụng User.findOne({ username }) để tìm kiếm người dùng trong cơ sở dữ liệu dựa trên tên người dùng đã cung cấp.

3. Kiểm tra điều kiện đăng nhập:

Nếu người dùng tồn tại và có vai trò là 'employee' và có loginToken, hàm sẽ trả về lỗi 400 với thông báo 'Please login by clicking on the link in your email'. Điều này có nghĩa là người dùng đã được tạo tài khoản nhưng cần phải đăng nhập qua liên kết trong email thay vì đăng nhập trực tiếp.

4. So sánh mật khẩu:

Nếu người dùng tồn tại, hàm sẽ sử dụng `bcrypt.compare(password, user.password)` để so sánh mật khẩu đã nhập với mật khẩu đã được mã hóa trong cơ sở dữ liệu.

Nếu mật khẩu khớp, hàm sẽ tiếp tục tạo token.

5. Tạo token:

Sử dụng hàm `generateToken(user._id)` để tạo một token JWT (JSON Web Token) cho người dùng. Token này sẽ được sử dụng để xác thực các yêu cầu trong tương lai.

6. Trả về phản hồi thành công:

Nếu đăng nhập thành công, hàm sẽ trả về phản hồi với mã trạng thái 200 và thông báo 'Login successful', cùng với thông tin người dùng (ID, tên đầy đủ, email) và token.

7. Xử lý lỗi:

Nếu không tìm thấy người dùng hoặc mật khẩu không khớp, hàm sẽ trả về lỗi 401 với thông báo 'Invalid email or password'.

Nếu có bất kỳ lỗi nào xảy ra trong quá trình thực hiện, hàm sẽ bắt lỗi và trả về một thông báo lỗi với mã trạng thái 500.

3.3.3 Hàm thêm sản phẩm mới

```
const addProduct = async (req, res, next) => {
  try {
    const {
      productName,
      importPrice,
      retailPrice,
      categoryId,
      manufacturer,
      description = '',
      stockQuantity = 1
    } = req.body;

    if (!productName || !importPrice || !retailPrice || !categoryId || !manufacturer) {
      return next(new AppError('Required fields must be filled', 400));
    }

    if (importPrice < 0 || retailPrice < 0 || stockQuantity < 0) {
      return next(new AppError('Price and stock quantity cannot be negative', 400));
    }

    // console.log(importPrice, retailPrice);
    if (Number(importPrice) > Number(retailPrice)) {
      return next(new AppError('Import price cannot be greater than retail price', 400));
    }

    if (!req.files || req.files.length === 0) {
      return next(new AppError('No images uploaded', 400));
    }
  }
}
```

```

const imageUrls = [];
for (const file of req.files) {
  const filename = `${productName}-${Date.now()}`;

  const result = await cloudinary.uploader.upload(file.path, {
    public_id: `product_images/${filename}`,
    folder: 'POS',
    use_filename: true,
    unique_filename: false,
    overwrite: true,
  });

  imageUrls.push(result.secure_url);
  fs.unlinkSync(file.path);
}

const newProduct = await Product.create({
  productName,
  barcode: generateBarcode(),
  categoryId,
  importPrice: Number(importPrice),
  retailPrice: Number(retailPrice),
  manufacturer,
  description,
  image: imageUrls,
  stockQuantity: Number(stockQuantity),
  isActive: true
});

const productItems = Array(Number(stockQuantity)).fill().map(() => ({
  productId: newProduct._id,
  serialNumber:
    newProduct.manufacturer.slice(0, 3).toUpperCase() + uuidv4().replace(/-/g, '').slice(-9),
  status: 'IN_STOCK'
})));

await ProductItem.insertMany(productItems);
res.status(201).json({
  code: 201,
  success: true,
  message: 'Product added successfully',
  result: newProduct
});
} catch (error) {
  return next(new AppError(`Error while adding product: ${error.message}`, 500));
}
};

```

Hình 3.4 Hàm thêm sản phẩm

Giải thích hàm:

1. Nhận dữ liệu từ yêu cầu:

- Hàm lấy thông tin sản phẩm từ req.body, bao gồm tên sản phẩm, giá nhập, giá bán, ID danh mục, nhà sản xuất, mô tả và số lượng tồn kho.

2. Kiểm tra dữ liệu đầu vào:

- Kiểm tra xem các trường bắt buộc có được cung cấp hay không.
- Đảm bảo rằng giá nhập, giá bán và số lượng tồn kho không âm.
- Kiểm tra rằng giá nhập không lớn hơn giá bán.

3. Kiểm tra hình ảnh: Nếu không có hình ảnh nào được tải lên, hàm sẽ trả về lỗi.

4. Tải hình ảnh lên Cloudinary: Hàm sử dụng Cloudinary để tải hình ảnh lên và lưu URL của hình ảnh vào mảng imageUrls.

5. Tạo sản phẩm mới: Sử dụng Product.create để thêm sản phẩm vào cơ sở dữ liệu với các thông tin đã nhận và URL hình ảnh.

6. Tạo các mục sản phẩm: Tạo một mảng các mục sản phẩm (ProductItem) dựa trên số lượng tồn kho và lưu chúng vào cơ sở dữ liệu.

7. Trả về phản hồi thành công: Nếu mọi thứ diễn ra suôn sẻ, hàm trả về mã trạng thái 201 và thông báo thành công cùng với thông tin sản phẩm mới.

8. Xử lý lỗi: Nếu có bất kỳ lỗi nào xảy ra trong quá trình thực hiện, hàm sẽ bắt lỗi và trả về thông báo lỗi với mã trạng thái 500.

3.3.4 Hàm xóa sản phẩm theo ID

```
const deleteProductById = async (req, res, next) => {
  try {
    const { productId } = req.params;

    if (!mongoose.Types.ObjectId.isValid(productId)) {
      return next(new AppError('Invalid ProductItemId', 400));
    }
    if (!productId) {
      return next(new AppError('ProductId is required', 400));
    }
    const product = await Product.findById(productId);
    if (!product) {
      return next(new AppError('Product not found', 404));
    }
    const orderItem = await OrderItem.findOne({ productId });
    if (orderItem) {
      return next(new AppError('Product cannot be deleted because it\'s in order', 400));
    }
    await ProductItem.deleteMany({ productId });
    await product.deleteOne();
    await client.del(`product_${product._id}`);
    await client.del(`product_barcode_${product.barcode}`);
    await client.del(`product_items_${product._id}`);
    await client.del(`product_items_barcode_${product.barcode}`);

    res.status(200).json({
      code: 200,
      success: true,
      message: 'Product item deleted successfully',
      result: product
    });
  } catch (error) {
    return next(new AppError(`Error deleting product: ${error.message}`, 500));
  }
}
```

Hình 3.5 Hàm xóa sản phẩm theo ID

Giải thích hàm:

1. Nhận ID sản phẩm: Hàm lấy productId từ tham số của yêu cầu.
2. Kiểm tra tính hợp lệ của ID:
 - Kiểm tra xem productId có phải là một ObjectId hợp lệ hay không.
 - Nếu không có productId, trả về lỗi 400.

3. Tìm sản phẩm: Sử dụng `Product.findById(productId)` để tìm sản phẩm trong cơ sở dữ liệu. Nếu không tìm thấy, trả về lỗi 404.

4. Kiểm tra xem sản phẩm có trong đơn hàng không:

- Sử dụng `OrderItem.findOne({ productId })` để kiểm tra xem sản phẩm có đang được sử dụng trong đơn hàng nào không. Nếu có, trả về lỗi 400.

5. Xóa các mục sản phẩm:

- Sử dụng `ProductItem.deleteMany({ productId })` để xóa tất cả các mục sản phẩm liên quan đến sản phẩm đó.

6. Xóa sản phẩm: Gọi `product.deleteOne()` để xóa sản phẩm khỏi cơ sở dữ liệu.

7. Xóa cache: Xóa các mục cache liên quan đến sản phẩm trong Redis.

8. Trả về phản hồi thành công: Nếu mọi thứ diễn ra suôn sẻ, trả về mã trạng thái 200 và thông báo thành công cùng với thông tin sản phẩm đã xóa.

8. Xử lý lỗi: Nếu có bất kỳ lỗi nào xảy ra trong quá trình thực hiện, hàm sẽ bắt lỗi và trả về thông báo lỗi với mã trạng thái 500.

3.3.5 Hàm thanh toán hóa đơn

```
const checkoutOrder = async (req, res, next) => {  
  const { orderId } = req.params;  
  const { amountReceived } = req.body;  
  
  try {  
    const order = await Order.findById(orderId);  
    if (!order) {  
      return next(new AppError('Order not found', 404));  
    }  
  
    if (!order.customerId || !order.employeeId || order.total <= 0) {  
      return next(new AppError('Order is not ready for checkout', 400));  
    }  
  
    const orderItems = await OrderItem.find({ orderId });  
    const products = await getProductDetailsFromOrderItems(orderItems);  
    const customer = await Customer.findById(order.customerId);  
  
    // Tạo file PDF và chờ cho đến khi hoàn tất  
    const invoiceUrl = await createInvoicePDF(order, products, customer);  
  
    order.status = OrderStatus.COMPLETED;  
    order.invoiceUrl = invoiceUrl;  
    if (!amountReceived && !order.amountReceived) {  
      return next(new AppError('Amount received is required', 400));  
    }  
  }  
}
```

```

order.amountReceived = amountReceived;
if (amountReceived < order.total) {
    return next(new AppError('Amount received is less than the total amount', 400));
}
order.changeGiven = amountReceived - order.total;
await order.save();

const pdfPath = path.resolve(invoiceUrl);

// Cấu trúc dữ liệu email với file đính kèm
const emailData = {
    to: customer.email,
    subject: 'Your Invoice from POS',
    text: 'Thank you for your shopping! This is your invoice.',
    htmlContent: `<p>Thank you for your shopping! This is your invoice.</p>`,
    attachmentPath: pdfPath
};

// Gọi hàm sendEmail để gửi email với file đính kèm
await sendEmail(emailData.to, emailData.subject, emailData.text, emailData.htmlContent,
    emailData.attachmentPath);

client.del(`orders_phoneNumber_${customer.phoneNumber}`);
client.del(`customer_orders_${customer.phoneNumber}`);
client.del(`order_${order._id}`);
res.status(200).json({
    code: 200,
    success: true,
    message: 'Order checked out successfully',
    result: order
});
} catch (error) {
    console.error(error);
    return next(new AppError('Error while checking out order: ${error.message}', 500));
}
}

```

Hình 3.6 Hàm thanh toán hóa đơn

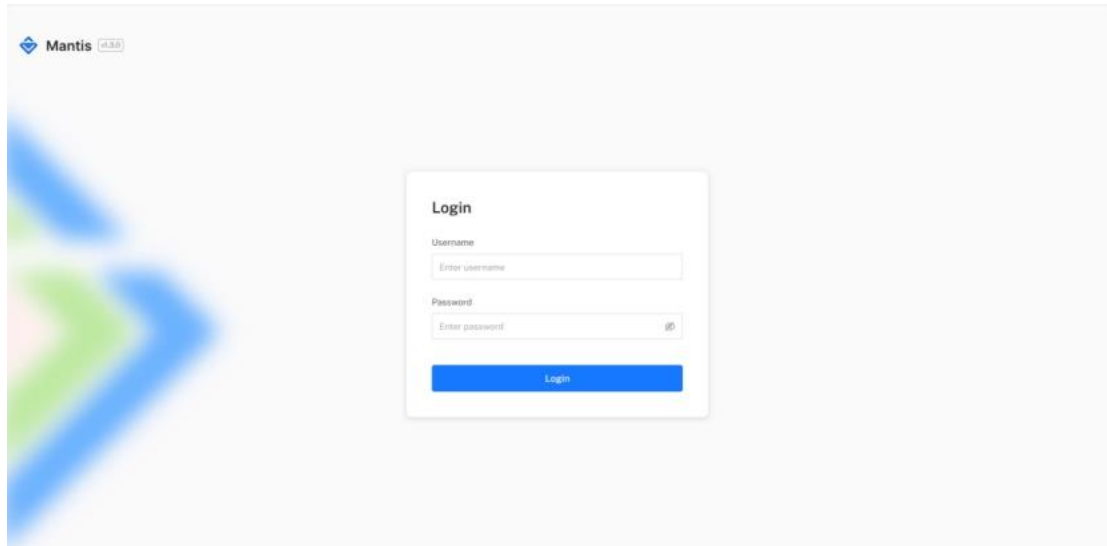
Giải thích hàm:

- Nhận dữ liệu: Lấy orderId từ tham số URL và amountReceived từ thân yêu cầu.
- Tìm kiếm đơn hàng: Sử dụng Order.findById(orderId) để tìm đơn hàng. Nếu không tìm thấy, trả về lỗi 404.
- Kiểm tra điều kiện thanh toán:
 - Đảm bảo rằng đơn hàng có customerId, employeeId, và tổng giá trị lớn hơn 0. Nếu không, trả về lỗi 400.
- Lấy thông tin sản phẩm và khách hàng:

- Tìm các mục đơn hàng (OrderItem) và lấy thông tin sản phẩm từ các mục đó.
- Tìm khách hàng dựa trên customerId.
 - Tạo hóa đơn PDF: Gọi createInvoicePDF để tạo hóa đơn và lưu URL của hóa đơn.
- Cập nhật trạng thái đơn hàng:
 - Đặt trạng thái đơn hàng là COMPLETED và lưu URL hóa đơn.
 - Kiểm tra xem amountReceived có được cung cấp và có lớn hơn tổng giá trị đơn hàng hay không. Nếu không, trả về lỗi 400.
- Tính tiền thối: Tính số tiền thối và lưu vào đơn hàng.
- Gửi email hóa đơn: Tạo dữ liệu email và gọi hàm sendEmail để gửi hóa đơn cho khách hàng.
- Xóa cache: Xóa các mục cache liên quan đến đơn hàng và khách hàng.
- Trả về phản hồi thành công: Nếu mọi thứ diễn ra suôn sẻ, trả về mã trạng thái 200 và thông báo thành công cùng với thông tin đơn hàng.
- Xử lý lỗi: Nếu có bất kỳ lỗi nào xảy ra, hàm sẽ bắt lỗi và trả về thông báo lỗi với mã trạng thái 500.

CHƯƠNG 4 - KẾT QUẢ ĐẠT ĐƯỢC

4.1 Giao diện đăng nhập



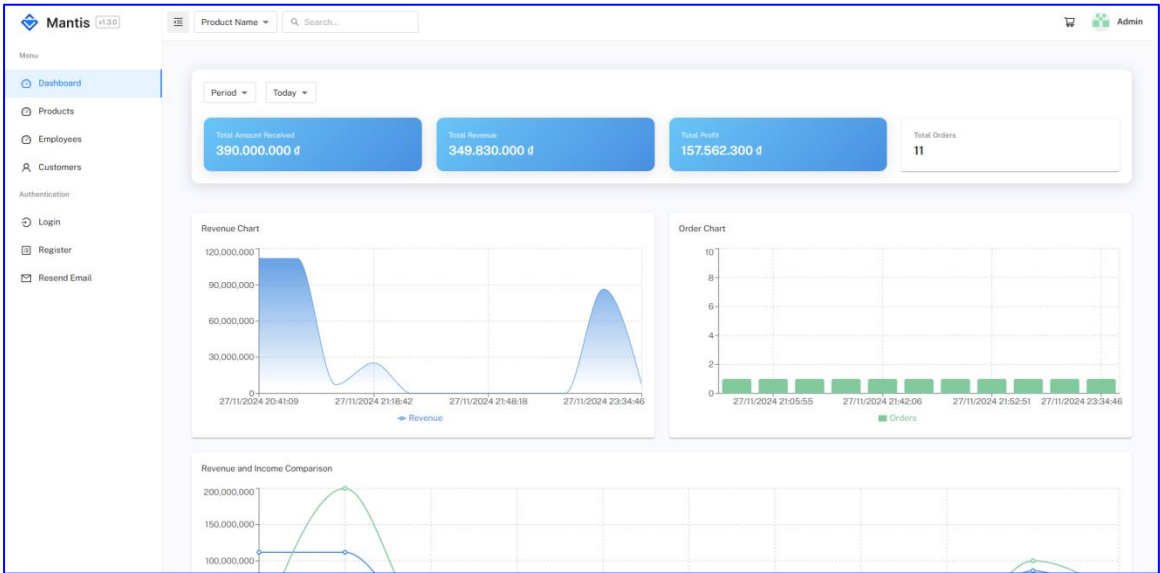
Hình 4.1 Giao diện đăng nhập

Tại đây để tiến hành truy cập vào hệ thống người dùng phải tiến hành đăng nhập vào bằng account của mình. Cả 2 trường username và password đều là required field. Ở field password có chức năng show/hide password để giúp người dùng tiện theo dõi trong quá trình đăng nhập.

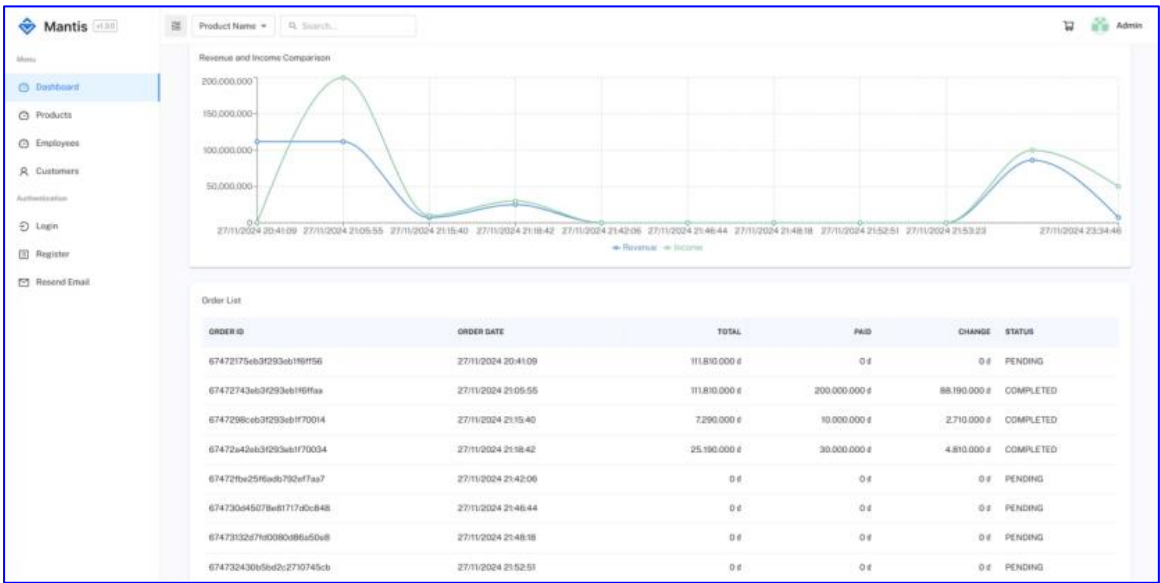
Admin: Sẽ đăng nhập bằng account đã được cung cấp trước đó

Salepeople: Account được tạo bởi Admin và sẽ tiến hành đăng nhập lần đầu bằng login link được gửi qua mailbox. Với thông tin đăng nhập lần đầu là Username và Password được lấy từ [Username@gmail.com](#) (ví dụ với [salepeople123@gmail.com](#) thì username và password để đăng nhập là salepeople123)

4.2 Giao diện Dashboard



Hình 4.2 Giao diện Dashboard 1



Hình 4.3 Giao diện Dashboard 2

The screenshot shows the Mantis v1.3.0 Dashboard. The left sidebar contains navigation links: Home, Dashboard (selected), Products, Employees, Customers, Authentication, Login, Register, and Resend Email. The main content area displays a table of transactions with columns for Product Name, Date, Amount, and Status. Below this is a 'Product Report (Yesterday)' table with columns for Product ID, Product Name, Barcode, Total Sold, Stock Quantity, and Status.

Product Name	Date	Amount	Status
6744b95ec54ac3b3c342145d	26/11/2024 00:52:30	7.290.000 ₺	PENDING
6744b97ec54ac3b3c3421472	26/11/2024 00:53:02	7.290.000 ₺	PENDING
6744b99ec54ac3b3c3421487	26/11/2024 00:54:36	7.290.000 ₺	PENDING
6744b9b2c54ac3b3c34214a6	26/11/2024 00:59:30	19.180.000 ₺	PENDING
6745c8b75ac3b3c34214f3	26/11/2024 20:10:15	0 ₺	PENDING
6745c8f15ac3b3c34214f7	26/11/2024 20:17:21	26.180.000 ₺	PENDING
6745c8f55ac3b3c34214f8	26/11/2024 20:17:33	26.180.000 ₺	PENDING
6745d8f99999c504bedf8449	26/11/2024 21:19:05	47.560.000 ₺	PENDING
6745d938999c504bedf84519	26/11/2024 21:20:40	47.560.000 ₺	COMPLETED

PRODUCT ID	PRODUCT NAME	BARCODE	TOTAL SOLD	STOCK QUANTITY	STATUS
6744b9c6d2eb34251361ecf37	Galaxy Z Fold Special	675710b49ebd94	2	11	Active
6744b9c6d2eb34251361ecf56	Galaxy A16	c304e9f5240083e	11	14	Active
6744b9c6d2eb34251361ecf84	Galaxy A16 5G	7804f5bad96a983	5	13	Active
6744b9c6d2eb34251361ecf0e	Galaxy M05	43a210c7f92286a	4	27	Active
6744b9c752eb34251361ec898	iPad Pro 12.9 (2022)	213194eb79500b7	1	6	Active

Hình 4.4 Giao diện Dashboard 3

Màn hình Dashboard cung cấp một cái nhìn tổng quan về các chỉ số quan trọng trong quản lý sản phẩm và doanh thu. Các biểu đồ và chỉ số chính giúp theo dõi hiệu suất bán hàng, doanh thu, số lượng sản phẩm, và các xu hướng hiện tại.

- **Timeline:** Có thể chọn xem report theo 2 dạng: Period và Costum
 - Period: Today, Yesterday, Last 7 Days, This Month
 - Costum: cho phép người dùng chọn bất kì Timeline nào muốn xem
- **Bảng Điều Khiển Tổng Quan:**
 - Total Amount Received: đây là số tiền nhận được từ khách hàng .
 - Total Revenue: đây là chỉ số quan trọng phản ánh tổng doanh thu từ tất cả các giao dịch trong một khoảng thời gian nhất định.
 - Total Profit: Số tiền lợi nhuận đạt được, tính bằng cách lấy tổng thu nhập trừ đi chi phí (Admin Only).
 - Total Orders: Xuất hiện với thông tin số đơn hàng, giúp quản lý số lượng đơn hàng đã được xử lý trong khoảng thời gian hiển thị.
- **Revenue Chart:**
 - Biểu đồ thể hiện diễn biến doanh thu theo thời gian, giúp người dùng phân tích các xu hướng và biến động doanh thu. Các đỉnh hoặc đáy có thể chỉ ra thời điểm cao điểm hoặc giảm sút trong doanh thu.
- **Order Chart:**
 - Biểu đồ này cung cấp thông tin về số lượng đơn hàng được đặt theo thời gian. Thông qua biểu đồ này, người dùng có thể theo dõi và đánh giá mức độ quan tâm của khách hàng.

- **Revenue and Income Comparison**

- Cung cấp cái nhìn tổng quan về sự so sánh giữa doanh thu và thu nhập. Biểu đồ này giúp xác định mối quan hệ giữa doanh thu và chi phí phát sinh, cho phép đánh giá tình hình tài chính của doanh nghiệp một cách rõ ràng.

- **Order List**

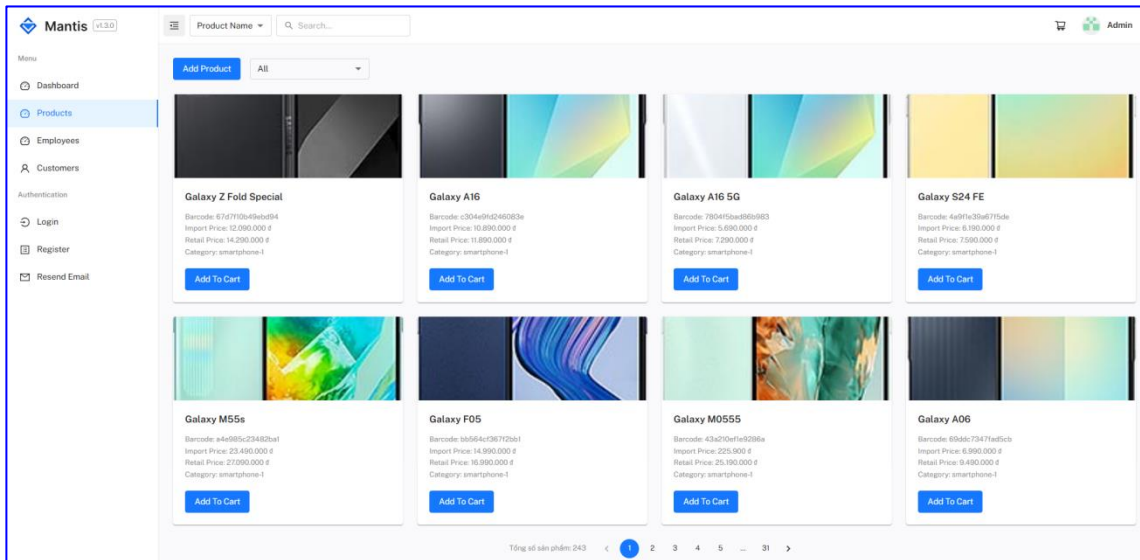
- **Order ID:** Mỗi đơn hàng được xác định bởi một mã ID duy nhất, cho phép theo dõi dễ dàng.
- **Order Date:** Ngày và giờ mà khách hàng thực hiện đơn hàng, thể hiện lịch sử giao dịch.
- **Total:** Tổng giá trị của đơn hàng, giúp đánh giá doanh thu từ từng giao dịch.
- **Paid:** Số tiền đã thanh toán cho đơn hàng, sẽ giúp kiểm tra tình trạng thanh toán của khách hàng.
- **Change:** Tiền thối lại cho khách, phản ánh mức độ chính xác trong quy trình thanh toán.
- **Status:** Trạng thái đơn hàng (COMPLETED, PENDING) giúp người quản lý theo dõi tiến độ xử lý đơn hàng.

- **Product Report**

- **Product ID:** Mã định danh duy nhất cho từng sản phẩm trong kho.
- **Product Name:** Tên sản phẩm, giúp nhân viên dễ dàng nhận biết và quản lý hàng hóa.
- **Barcode:** Mã vạch của sản phẩm, hỗ trợ trong việc kiểm tra hàng tồn kho và xử lý đơn hàng.
- **Total Sold:** Tổng số lượng sản phẩm đã bán, một chỉ số quan trọng phản ánh hiệu suất bán hàng.
- **Stock Quantity:** Số lượng hàng tồn kho hiện tại, giúp quản lý hàng hóa hiệu quả.
- **Status:** Trạng thái của sản phẩm trong kho, điều này hỗ trợ trong việc lập kế hoạch nhập hàng.

4.3 Giao diện Products

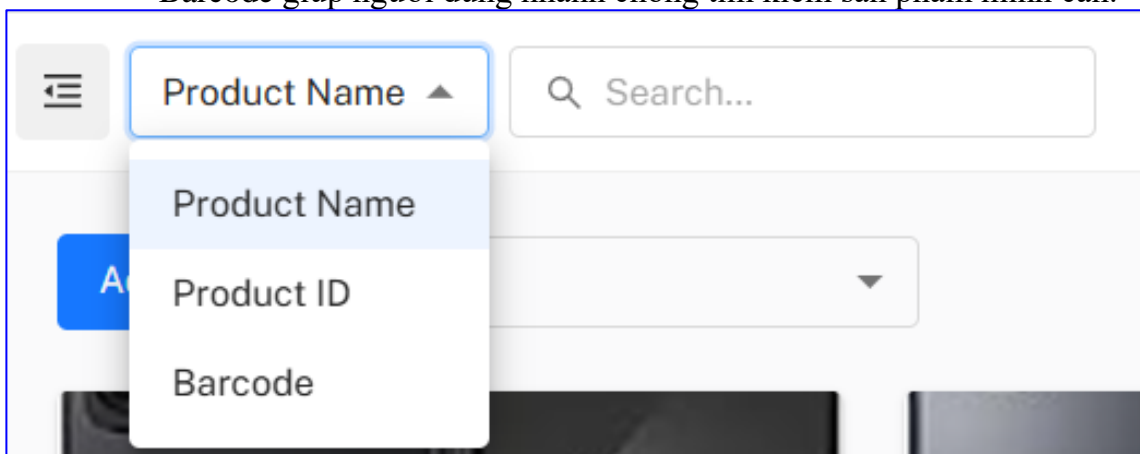
Giao diện "Products" của hệ thống bán thiết bị di động được thiết kế để quản lý và hiển thị thông tin về các sản phẩm, từ đó giúp người dùng dễ dàng theo dõi và xử lý các giao dịch liên quan đến sản phẩm.



Hình 4.4 Giao diện Products

Giao diện này cung cấp danh sách các sản phẩm hiện có trong kho cùng các thông tin quan trọng về từng sản phẩm.

- **Search:**
 - Có chức năng tìm kiếm sản phẩm theo Product Name, Product ID, Barcode giúp người dùng nhanh chóng tìm kiếm sản phẩm mình cần.



Hình 4.5 Giao diện Search Product

- **Add Product:**

- Giúp người dùng dễ dàng thêm mới sản phẩm vào hệ thống bằng cách click vào nút “Add Product”. Quá trình thêm sản phẩm thường yêu cầu thông tin chi tiết như:

Add new products

Product Name

Import Price

Retail Price

Manufacturer

Description

Stock Quantity

Category

Product Images

Choose Image

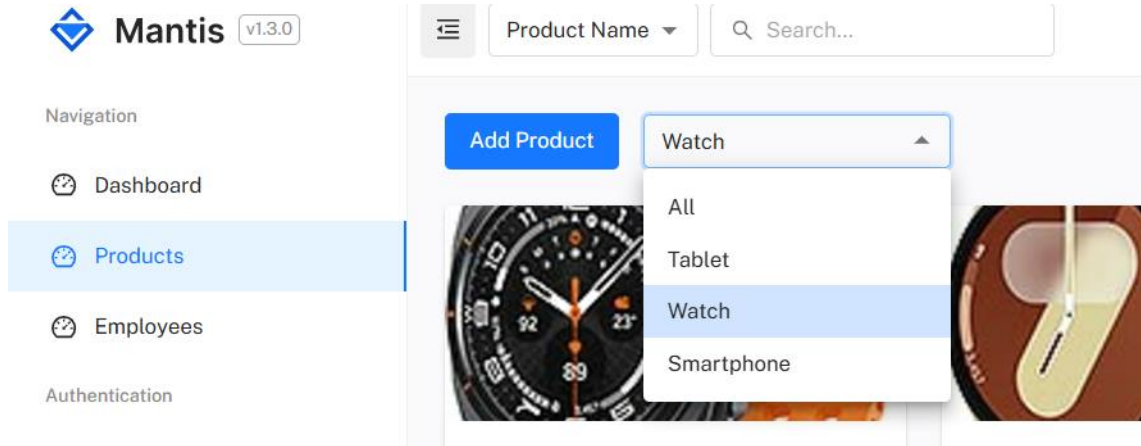
Cancel

Save

Hình 4.6 Giao diện Add Product

- **Filter by category:**

- Phần này giúp người dùng có thể tìm kiếm tất cả các sản phẩm cùng loại mà mình muốn xem.



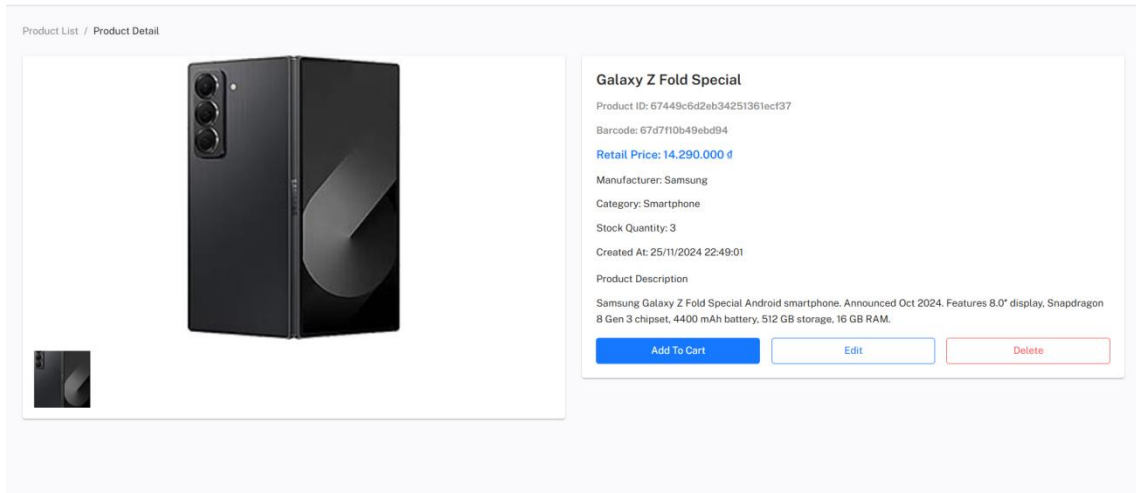
Hình 4.7 Giao diện Filter By Category

- **Danh Sách Sản Phẩm:**

- Mỗi sản phẩm trong danh sách hiển thị có các thông tin chính:
 - **Tên Sản Phẩm:** Tên gọi của sản phẩm (ví dụ: Galaxy Z Fold Special).
 - **Mã Vạch (Barcode):** Mã quản lý sản phẩm giúp dễ dàng theo dõi và kiểm kê.
 - **Giá Nhập (Import Price):** Giá mua vào từ nhà cung cấp (Admin only).
 - **Giá Bán Lẻ (Retail Price):** Giá bán ra cho khách hàng.
 - **Danh Mục (Category):** Phân loại sản phẩm (ví dụ: smartphone-1).
 - **Nút "Add To Cart":** Cho phép người dùng thêm sản phẩm vào giỏ hàng để thực hiện giao dịch.

- **Chỉ Số Tổng Sản Phẩm:** Ghi nhận tổng số sản phẩm hiện có trong hệ thống, giúp người quản lý nắm bắt được quy mô hàng hóa.

4.4 Giao diện Product Detail

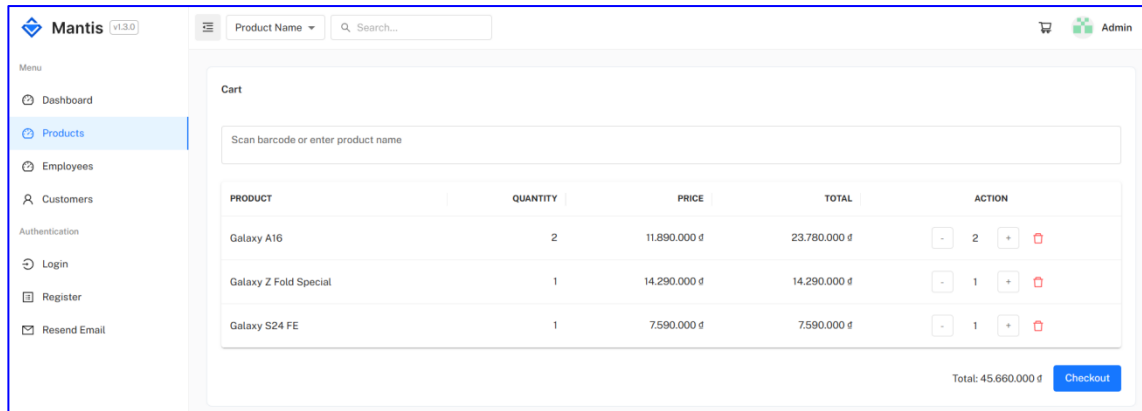


Hình 4.8 Giao diện Product Detail

Giao diện này cung cấp thông tin chi tiết về một sản phẩm cụ thể khi người dùng chọn vào đó.

- **Thông Tin Sản Phẩm:**
 - **Tên Sản Phẩm:** Hiển thị tên đầy đủ của sản phẩm.
 - **Mã Sản Phẩm (Product ID):** Mã định danh duy nhất cho sản phẩm.
 - **Mã Vạch:** Thông tin mã vạch liên quan đến sản phẩm.
 - **Giá Bán Lẻ (Retail Price):** Giá bán chính thức của sản phẩm.
 - **Nhà Sản Xuất (Manufacturer):** Thông tin về nhà sản xuất sản phẩm.
 - **Danh Mục:** Cung cấp thông tin về phân loại sản phẩm.
 - **Số Lượng Tồn Kho (Stock Quantity):** Hiển thị số lượng sản phẩm hiện có trong kho.
 - **Ngày Tạo (Created At):** Thông tin về thời gian sản phẩm được thêm vào hệ thống.
 - **Mô Tả Sản Phẩm:** Cung cấp thông tin chi tiết về tính năng và thông số kỹ thuật của sản phẩm.
- **Chức Năng Quản Lý:**
 - **Nút "Add To Cart":** Tương tự như trong giao diện sản phẩm, cho phép người dùng thêm sản phẩm vào giỏ hàng.
 - **Nút "Edit":** Dùng để chỉnh sửa thông tin sản phẩm hiện tại, bao gồm tên, giá, mô tả, v.v. (Admin only)
 - **Nút "Delete":** Cho phép người dùng xóa sản phẩm khỏi hệ thống với điều kiện sản phẩm đó đang không trong bất kì order nào (Admin only).

4.5 Giao diện Cart



Hình 4.9 Giao diện Cart

Giao diện "Cart" của hệ thống bán thiết bị di động là nơi người dùng có thể xem và quản lý các sản phẩm đã được thêm vào giỏ hàng trước khi tiến hành thanh toán. Giao diện này cung cấp thông tin rõ ràng về từng sản phẩm, số lượng, và tổng chi phí, hỗ trợ người dùng đưa ra quyết định cuối cùng trước khi hoàn tất giao dịch.

- **Scan barcode or enter product name:**
 - Cho phép salepeople có thể tiến hành thêm sản phẩm vào giỏ hàng nhanh bằng cách scan barcode và nhập tên sản phẩm mà không cần xác nhận lại.
- **Danh Sách Sản Phẩm:**
 - **PRODUCT:** Đây là tên các sản phẩm trong giỏ hàng.
- **Thông Tin Chi Tiết Sản Phẩm:**
 - **QUANTITY:** Hiện thị số lượng sản phẩm trong giỏ. Người dùng có thể điều chỉnh số lượng bằng cách sử dụng nút "-" để giảm và nút "+" để tăng.
 - **PRICE:** Giá của từng sản phẩm.
 - **TOTAL:** Tổng giá của từng sản phẩm, được tính bằng cách nhân giá với số lượng. Ví dụ, tổng của Galaxy A16 là **23.780.000 đ** (11.890.000 đ x 2).
- **Hành Động (ACTION):**
 - **Nút Xóa (Delete):** Cho phép người dùng xóa sản phẩm khỏi giỏ hàng nếu không còn muốn mua.
- **Tổng Chi Phí:**
 - **Total:** Hiện thị tổng số tiền của tất cả các sản phẩm có trong giỏ hàng. Đây là số tiền tổng mà người dùng sẽ phải thanh toán khi tiến hành thanh toán.
- **Nút Thanh Toán (Checkout):**

- Nút "Checkout" cho phép người dùng hoàn tất quá trình mua sắm, chuyển đến bước thanh toán. Nút này rất quan trọng trong việc thúc đẩy giao dịch.

4.6 Giao diện Order

The screenshot shows the Mantis v1.3.0 Order management interface. The sidebar menu includes: Dashboard, Products (selected), Employees, Customers, Login, Register, and Resend Email. The main content area displays the 'Order' form with the following sections:

- Customer phone number:** A text input field containing '7378478621'.
- Current order information:** A table with columns: PRODUCT, QUANTITY, PRICE, and TOTAL.

PRODUCT	QUANTITY	PRICE	TOTAL
Galaxy A16	2	11.890.000 đ	23.780.000 đ
- Tổng tiền:** 23.780.000 đ. Buttons: Back To Cart, Confirm Order.
- Customer information:** A form with fields: Customer ID: 67449c882eb34251361f19d, Customer name: Jackie Reichert, Phone number: 7378478621, Address: 6628 Grange Close, Email: Francesca98@hotmail.com, Created at: 25/11/2024 22:49:28.
- Customer order history:** A table with columns: ORDER ID, ORDER DATE, TOTAL, PAID, CHANGE, STATUS.

ORDER ID	ORDER DATE	TOTAL	PAID	CHANGE	STATUS
67449e172eb34251361f1971	29/07/2024 13:18:30	114.510.000 đ	114.510.026 đ	26 đ	COMPLETED
67449d5a2eb34251361f0648	26/07/2024 07:41:46	168.590.000 đ	168.590.082 đ	82 đ	COMPLETED

Hình 4.10 Giao diện Order

Giao diện "Order" của hệ thống bán thiết bị di động cung cấp chi tiết về đơn hàng đang được xử lý. Đây là công cụ giúp người dùng theo dõi, xác nhận và quản lý đơn hàng của mình, bao gồm thông tin sản phẩm, thông tin khách hàng và lịch sử giao dịch

- Customer phone number
 - Cho phép salepeople nhập số điện thoại của khách hàng, có 2 loại khách hàng là khách hàng mới và khách hàng cũ:
 - Nếu là khách hàng cũ khi salepeople nhập số điện thoại thì hệ thống sẽ tự động hiển thị các thông tin như
 - **Thông Tin Khách Hàng**
 - **Lịch Sử Đơn Hàng của Khách Hàng:**
 - **ORDER ID:** Mã định danh của từng đơn hàng trong lịch sử giao dịch, giúp khách hàng tra cứu và theo dõi dễ dàng.
 - **ORDER DATE:** Ngày và giờ mà đơn hàng được thực hiện.
 - **TOTAL:** Tổng giá trị của đơn hàng trong quá khứ.
 - **PAID:** Số tiền đã thanh toán cho từng đơn hàng.

- **CHANGE:** Số tiền thối lại cho khách hàng, thể hiện độ chính xác trong quy trình thanh toán.
 - **STATUS:** Trạng thái của đơn hàng (COMPLETED), cho biết đơn hàng đã được xử lý thành công hay chưa.
- Nếu là khách hàng mới khi salepeople nhập số điện thoại thì hệ thống sẽ tự động hiển thị một popup cho phép điền thông tin của khách hàng mới

The screenshot shows a 'Create new customer' modal form. The background 'Order' page includes a 'Customer phone number' field with '0123456789', a 'Current order information' section for 'Galaxy A16' with a total of 23,780,000 đ, and a table with columns 'PRODUCT', 'PRICE', and 'TOTAL'.

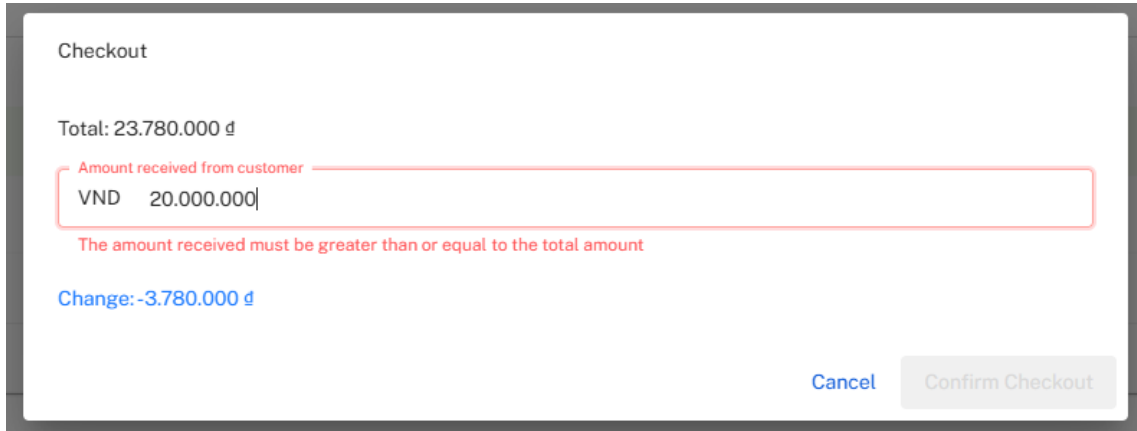
Hình 4.11 Giao diện tạo khách hàng mới

The screenshot shows the 'Create Order' page. A green banner at the top indicates 'Customer created successfully'. Below it, the 'Current order information' section shows 'Galaxy A16' with a quantity of 2 and a total of 23,780,000 đ. A table with columns 'PRODUCT', 'QUANTITY', 'PRICE', and 'TOTAL' is also present. At the bottom right, there are 'Back To Cart' and 'Confirm Order' buttons.

Hình 4.12 Giao diện Create Order

Sau khi tìm được khách hàng cũ hoặc đã tạo khách hàng mới thì hệ thống mới hiển thị buuton “Confirm Order” để đến bước Checkout

4.7 Giao diện Checkout



Checkout

Total: 23.780.000 đ

Amount received from customer

VND 20.000.000

The amount received must be greater than or equal to the total amount

Change: -3.780.000 đ

Cancel Confirm Checkout

Hình 4.13 Giao diện Checkout thất bại

Nếu số tiền nhập vào không hợp lệ sẽ hiển thị thông báo và button “Confirm Checkout” sẽ bị vô hiệu hóa



Checkout

Total: 23.780.000 đ

Amount received from customer

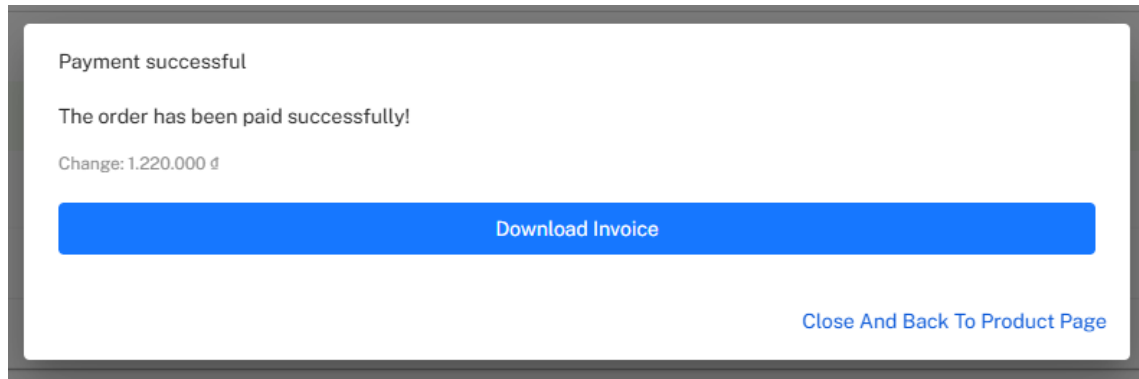
VND 25.000.000

Change: 1.220.000 đ

Cancel Confirm Checkout

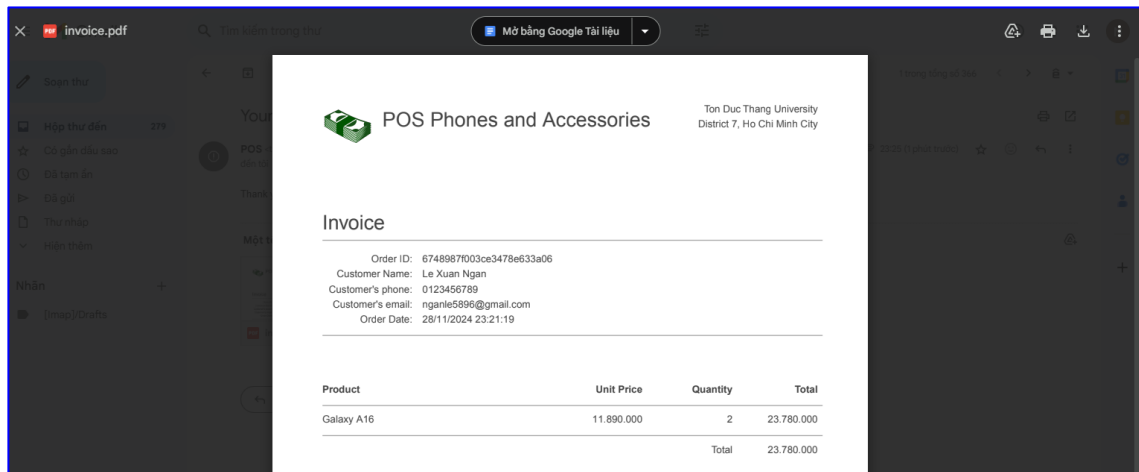
Hình 4.14 Giao diện Checkout thành công 1

Khi số tiền hợp lệ sẽ cho phép salepeople Checkout



Hình 4.14 Giao diện Checkout thành công 2

Sau khi Checkout thành công hệ thống sẽ tiến hành xuất hóa đơn và đồng thời cũng có 1 hóa đơn được gửi đến mailbox của khách hàng



Hình 4.15 Giao diện Hóa đơn định dạng pdf

4.8 Giao diện Employees

Avatar	Full Name	Email	Active Status	Locked Status
	Denise Goodwin	Will_Pfeffer93@gmail.com	Active	
	Dr. Adrian Purdy	Brittany84@gmail.com	Active	
	Freda Littell	Ruben80@yahoo.com	Active	
	Eula Larson	Hilton17@yahoo.com	Active	
	Ricky Mills	Mae.Stoltenberg45@hotmail.com	Active	
	Irvin Gorczany	Eddie_Rempel@yahoo.com	Active	
	Andre Wiegand	Elinore_Feil86@yahoo.com	Active	
	Dominic Kirlin PhD	Shanon81@hotmail.com	Active	
	Johanna Lockman DVM	Elmo.West@hotmail.com	Active	
	Miss Darlene Okuneva PhD	Cindy.Nitzsche65@hotmail.com	Active	Unlocked

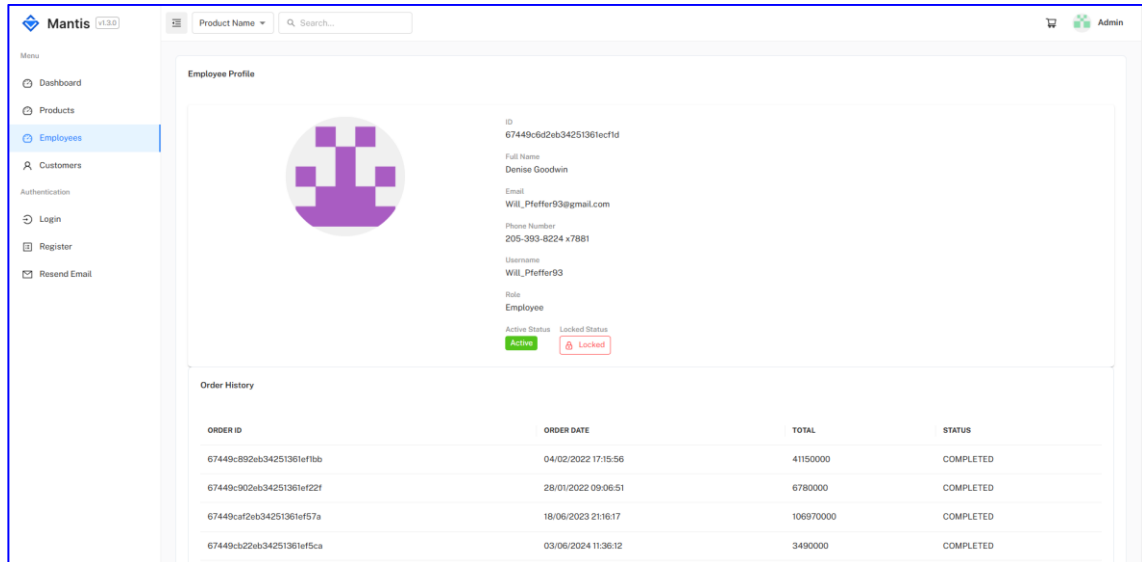
Hình 4.16 Giao diện Employees

Giao diện này hiển thị danh sách nhân viên đang làm việc trong hệ thống cùng với các thông tin cơ bản liên quan đến từng nhân viên (Admin only).

Các Thành Phần Chính:

- **Avatar:** Hình đại diện của nhân viên, giúp nhận diện dễ dàng.
- **Tên đầy đủ (Full Name):** Tên của nhân viên, chẳng hạn như **Denise Goodwin**, giúp phân biệt từng nhân viên trong danh sách.
- **Email:** Địa chỉ email cá nhân của nhân viên, sử dụng để liên lạc và gửi thông tin liên quan đến công việc (ví dụ: Will_Pfeffer93@gmail.com).
- **Trạng thái hoạt động (Active Status):** Chỉ ra tình trạng làm việc của nhân viên (Active hoặc Inactive)
 - **Active:** nhân viên đã đăng nhập vào hệ thống lần đầu tiên
 - **Inactive:** nhân viên chưa đăng nhập vào hệ thống bằng cách click vào login link ở mailbox
- **Trạng thái khóa (Locked Status):** Cho biết liệu tài khoản của nhân viên có đang bị khóa hay không, với các biểu tượng khác nhau để người quản lý dễ dàng theo dõi.
- **Tìm kiếm và phân trang:** Chức năng tìm kiếm và phân trang giúp người quản lý tìm kiếm nhanh chóng thông tin nhân viên cần thiết, nhất là khi có nhiều nhân sự trong danh sách.

4.9 Giao diện Employee Profile



Hình 4.17 Giao diện Employee Profile

Khi người quản lý chọn một nhân viên từ danh sách, giao diện "Employee Profile" sẽ hiển thị thông tin chi tiết về nhân viên đó.

Các Thành Phần Chính:

- **ID Nhân Viên:** Mã định danh duy nhất của nhân viên trong hệ thống, ví dụ **67449cGd2eb34251361ecf1d**.
- **Họ và tên đầy đủ (Full Name):** Tên đầy đủ của nhân viên.
- **Email:** Địa chỉ email cá nhân, cần thiết trong việc liên lạc và xác thực thông tin.
- **Số điện thoại (Phone Number):** Thông tin liên lạc bổ sung, giúp liên hệ nhanh chóng khi cần thiết.
- **Tên người dùng (Username):** Tên người dùng được sử dụng để đăng nhập vào hệ thống.
- **Vai trò (Role):** Chỉ rõ vai trò của nhân viên trong công ty (vd: Employee).
- **Trạng thái hoạt động (Active Status) và Trạng thái khóa (Locked Status):** Hiện trạng làm việc của nhân viên, cho biết họ có thể truy cập vào hệ thống hay không.
- **Lịch sử đơn hàng (Order History):**
 - **ORDER ID:** Mã định danh của từng đơn hàng mà nhân viên đã xử lý, giúp theo dõi lịch sử công việc.
 - **ORDER DATE:** Ngày giờ từng đơn hàng được thực hiện.
 - **TOTAL:** Tổng số tiền của từng đơn hàng, cho biết giá trị giao dịch.
 - **STATUS:** Trạng thái của đơn hàng (vd: COMPLETED), phản ánh kết quả xử lý đơn hàng.

4.10 Giao diện Customers

Giao diện "Customers" là nơi hiển thị danh sách toàn bộ khách hàng trong hệ thống, bao gồm các thông tin cơ bản để quản lý và theo dõi.

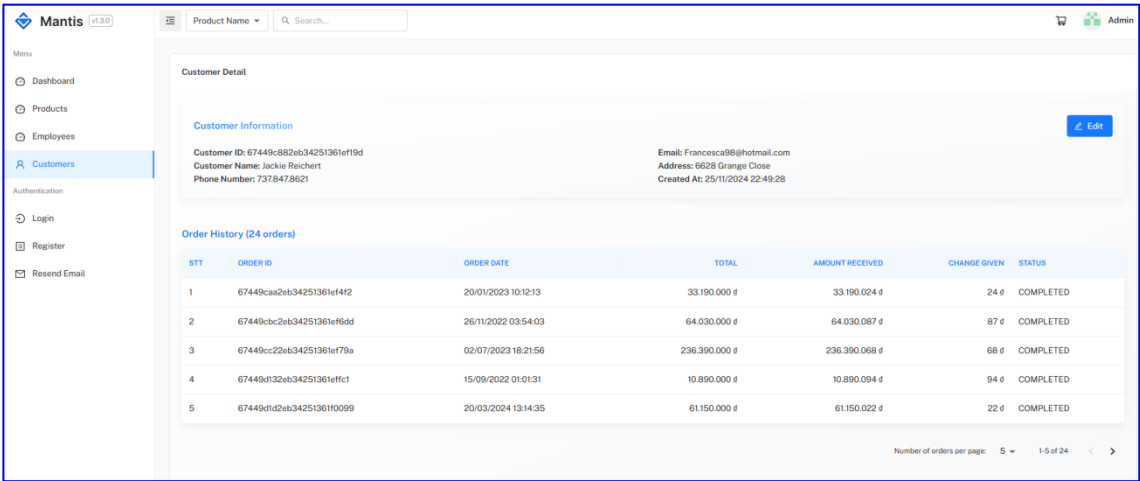
STT	CUSTOMER ID	CUSTOMER NAME	PHONE NUMBER	EMAIL	ADDRESS	CREATED AT	UPDATED AT
1	67449c882eb34251361ef19d	string	55555	string	5555	25/11/2024 22:49:28	27/11/2024 23:06:36
2	67449c882eb34251361ef19d	Jackie Reichert	7378478621	Francesca98@hotmail.com	6628 Grange Close	25/11/2024 22:49:28	25/11/2024 22:49:28
3	67449c882eb34251361ef19e	Ginger Volkman	(829) 316-3243 x42628	Trisha.Pouros@hotmail.com	487 N Front Street	25/11/2024 22:49:28	25/11/2024 22:49:28
4	67449c882eb34251361ef1a4	Jeffrey Terry	1-916-843-7502 x377	Oliver_Champin@hotmail.com	70031 Kuhn Squares	25/11/2024 22:49:28	25/11/2024 22:49:28
5	67449c882eb34251361ef1a9	Juan Glasasomnn 123	391-453-4931 x367	Stephanie60@yahoo.commm	391-453-4931 x367	25/11/2024 22:49:28	27/11/2024 23:56:38

Hình 4.18 Giao diện Customers

- **Thanh Tìm Kiếm:**
 - **Search by Phone Number:** Chức năng cho phép người dùng tìm kiếm khách hàng nhanh chóng bằng cách nhập số điện thoại, giúp tiết kiệm thời gian khi cần tra cứu thông tin cụ thể.
- **Thông Tin Khách Hàng:**
 - STT (Số Thứ Tự): Hiển thị số thứ tự của khách hàng trong danh sách.
 - CUSTOMER ID: Mã định danh duy nhất của khách hàng, ví dụ: 67449c882eb34251361ef19d, dùng để phân biệt các khách hàng trong hệ thống.
 - CUSTOMER NAME
 - PHONE NUMBER
 - EMAIL
 - ADDRESS
 - CREATED AT
 - UPDATED AT

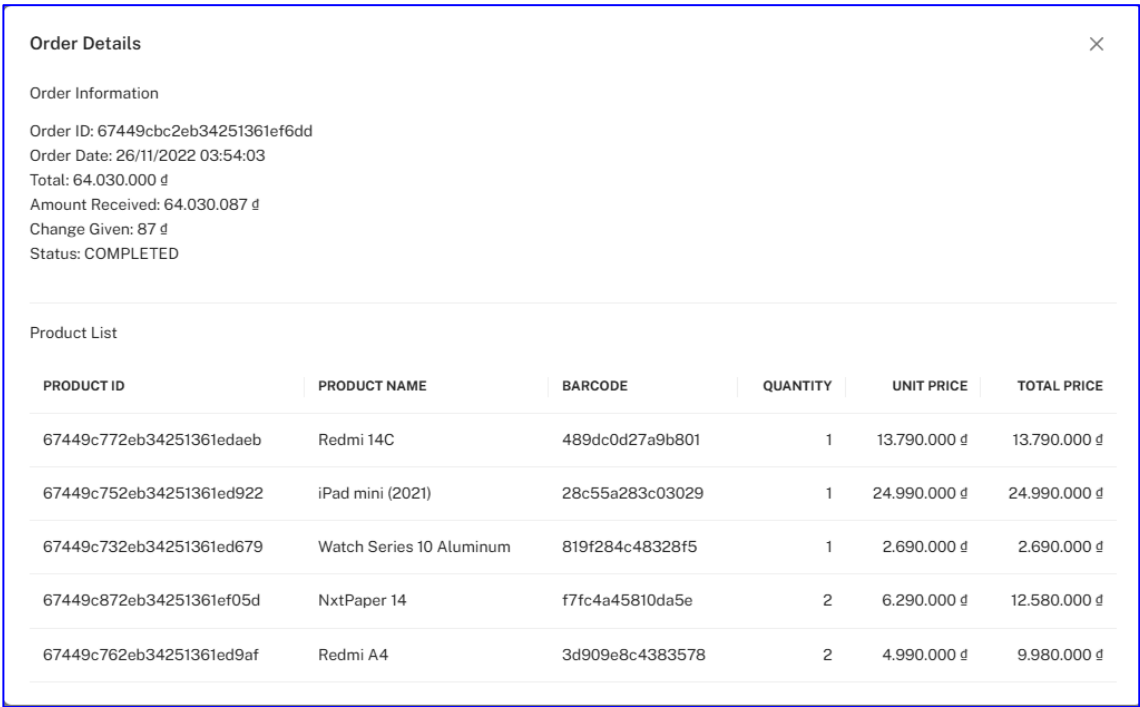
Giao diện "Customer Profile"

Giao diện này xuất hiện khi người quản lý chọn một khách hàng từ danh sách để xem thông tin chi tiết và lịch sử đơn hàng của họ.



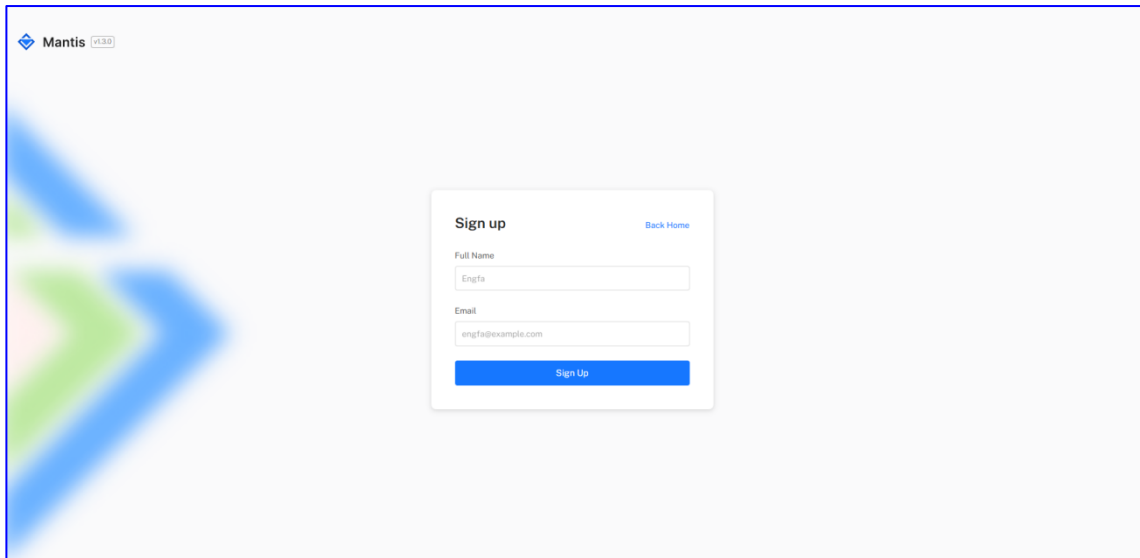
Hình 4.19 Giao diện Customer Profile

Chi Tiết Đơn Hàng:



Hình 4.20 Giao diện Chi tiết đơn hàng

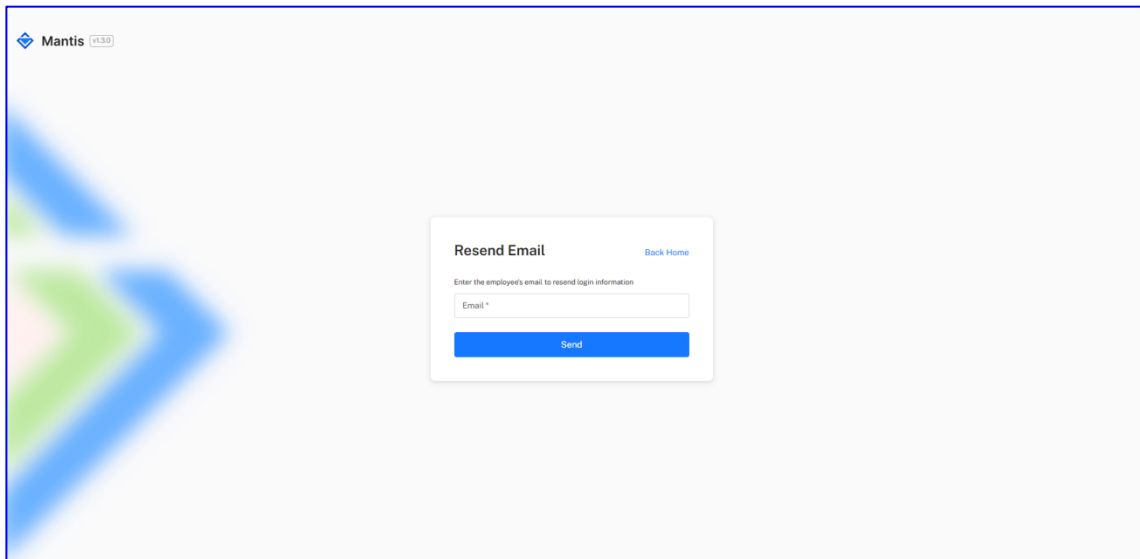
4.11 Giao diện Sign up



Hình 4.21 Giao diện Sign up

Màn hình để Admin có thể tạo account cho salepeople (Admin only)

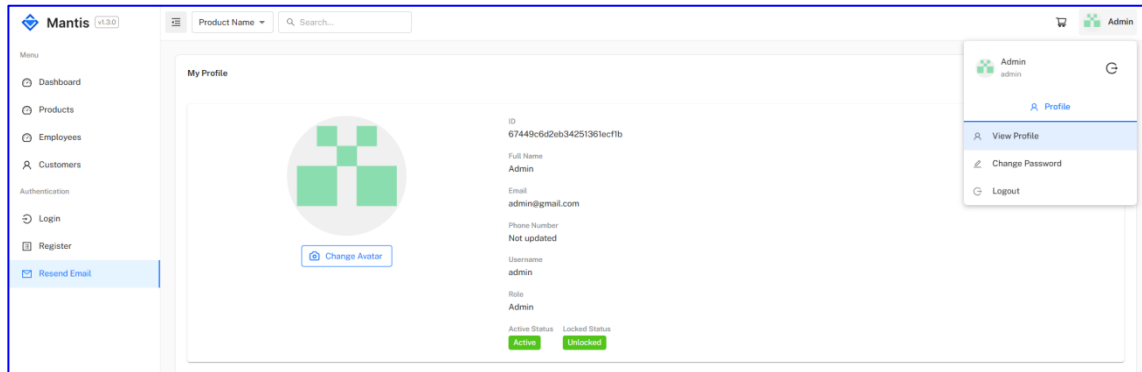
4.12 Giao diện Resend Email



Hình 4.22 Giao diện Resend Email

Khi đường link trong email đã quá 1 phút salepeople cần yêu cầu Admin gửi lại 1 link khác để tiến hành đăng nhập lần đầu

4.13 Giao diện My Profile



Hình 4.23 Giao diện My Profile

Tại đây hiển thị các thông tin của account và cho phép user có thể thay đổi change profile picture hoặc change password.

CHƯƠNG 5 - KẾT LUẬN

5.1 Ưu điểm

Đa dạng tính năng: Hệ thống đã được phát triển với nhiều tính năng hữu ích như quản lý sản phẩm, xử lý đơn hàng, báo cáo doanh thu và quản lý người dùng, giúp nâng cao hiệu quả hoạt động của doanh nghiệp.

Giao diện thân thiện: Giao diện người dùng được thiết kế trực quan, dễ sử dụng, giúp người dùng dễ dàng tương tác với hệ thống mà không gặp khó khăn.

Bảo mật thông tin: Hệ thống đã áp dụng các biện pháp bảo mật như mã hóa mật khẩu và xác thực người dùng, đảm bảo an toàn cho dữ liệu của khách hàng và doanh nghiệp.

5.2 Nhược điểm

Tối ưu hóa hiệu suất: Hiệu suất của hệ thống có thể chưa đạt tối ưu trong một số tình huống, đặc biệt khi xử lý một lượng lớn dữ liệu hoặc nhiều yêu cầu đồng thời.

5.3 Hướng phát triển tương lai

- Tích hợp thanh toán trực tuyến
- Tích hợp AI để cải thiện trải nghiệm người dùng
- Tối ưu tốc độ truy vấn dữ liệu

TÀI LIỆU THAM KHẢO

Tiếng Việt

1. <https://viblo.asia/p/su-dung-redis-lam-cache-de-tang-toc-do-truy-van-GrLZD0dwZk0>
2. <https://viblo.asia/p/cache-du-lieu-nodejs-voi-redis-bJzKmPNr59N>

Tiếng Anh

3. <https://developers.brevo.com/docs/send-a-transactional-email>
4. <https://blog.logrocket.com/multer-nodejs-express-upload-file>
5. <https://github.com/PSPDFKit-labs/pdfkit-invoice/blob/master/createInvoice.js>
6. <https://v9.fakerjs.dev/guide/>
7. <https://momentjs.com/docs/#/use-it>