

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



**ĐỒ ÁN CUỐI KÌ MÔN NODE JS**

# **WEBSITE QUẢN LÝ BÁN HÀNG CỦA HÀNG ĐIỆN TỬ**

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

*Người thực hiện:* **PHẠM TRẦN THẢO NGUYỄN–52100913**

**NGUYỄN NGỌC XUÂN NHƯ – 52100825**

**VÕ THY TRÚC–52100340**

**Lớp : 21050201, 21050301**

**Khoá : 25**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023**

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



**ĐỒ ÁN CUỐI KÌ MÔN NODE JS**

# **WEBSITE QUẢN LÝ BÁN HÀNG CỦA HÀNG ĐIỆN TỬ**

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

Người thực hiện: **PHẠM TRẦN THẢO NGUYỄN**

**NGUYỄN NGỌC XUÂN NHƯ**

**VÕ THY TRÚC**

Lớp : **21050201, 21050301**

Khoá : **25**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023**

## LỜI CẢM ƠN

Chúng em xin được bày tỏ lòng biết ơn giảng viên bộ môn và nhà trường đã tạo điều kiện hỗ trợ, giúp đỡ chúng em trong suốt quá trình học tập và hoàn thiện đồ án cuối kỳ này.

Chúng em xin gửi lời cảm ơn chân thành đến Thầy Vũ Đình Hồng đã truyền đạt vốn kiến thức quý báu cho chúng em trong suốt thời gian học môn Phát triển ứng dụng web NodeJs. Nhờ có những lời hướng dẫn, quan tâm và dạy bảo chúng em mới có thể hoàn thiện đồ án cuối kỳ của mình một cách hoàn thiện.

Một lần nữa chúng em xin gửi lời cảm ơn sâu sắc đến thầy. Đồng thời do kinh nghiệm của chúng em về một số vấn đề còn có những hạn chế. Do đó, bài báo cáo của chúng em không thể tránh khỏi những thiếu sót, chúng em rất mong nhận được những ý kiến đóng góp quý báu của Thầy để kiến thức của chúng em trong lĩnh vực này được hoàn thiện hơn ở những báo cáo sau.

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

Tôi xin cam đoan đây là sản phẩm đồ án của riêng tôi / chúng tôi 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 tôi 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 tôi gây ra trong quá trình thực hiện (nếu có).

*TP. Hồ Chí Minh, ngày 10 tháng 12 năm 2023*

*Tác giả*

*(ký tên và ghi rõ họ tên)*

*Phạm Trần Thảo Nguyên*

*Nguyễn Ngọc Xuân Như*

*Võ Thy Trúc*

## **PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN**

### **Phần xác nhận của GV hướng dẫn**

---

---

---

---

---

---

---

---

Tp. Hồ Chí Minh, ngày 10 tháng 12 năm 2023  
(kí và ghi họ tên)

### **Phần đánh giá của GV chấm bài**

---

---

---

---

---

---

---

---

Tp. Hồ Chí Minh, ngày 10 tháng 12 năm 2023  
(kí và ghi họ tên)

## **TÓM TẮT**

Bài báo cáo trình bày quá trình bày những hiểu biết về Node Js và các vấn đề liên quan. Giới thiệu đề tài sẽ tìm hiểu và giải thích cách hiện thực hóa đề tài POS – Hệ thống quản lý của cửa hàng bán điện thoại.

## MỤC LỤC

CẢM ƠN .....	i
PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN .....	iii
TÓM TẮT .....	iv
MỤC LỤC .....	1
DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ .....	4
CHƯƠNG 1 – GIỚI THIỆU ĐỀ TÀI .....	5
1.1 Trình bày về đề tài .....	5
1.2. Lý do chọn đề tài.....	5
1.3. Yêu cầu đề tài cần được thực hiện .....	5
1.3.1 Ràng buộc về ứng dụng .....	6
1.3.2 Người dùng hệ thống .....	6
1.3.3 Các chức năng của hệ thống .....	6
1.3.4 Các trường dữ liệu .....	12
CHƯƠNG 2 – CƠ SỞ LÝ THUYẾT .....	14
1.1 ExpressJS .....	14
1.2 MongoDB.....	14
1.3 Javascript.....	15
1.4 Handlebars .....	17
CHƯƠNG 3 – PHÂN TÍCH THIẾT KẾ .....	18
3.1 Đặt vấn đề .....	18
3.2 Thiết kế sơ đồ hệ thống.....	18
3.3 Hoạt động của trang web .....	21
3.4 Cấu trúc cơ sở dữ liệu .....	26
.....	27
CHƯƠNG 4 – HIỆN THỰC HỆ THỐNG .....	28
4.1 Phân tích thiết kế yêu cầu .....	28

4.2 Thiết kế hệ thống.....	28
4.3 Phát triển các tính năng.....	30
4.4 Kiểm tra hệ thống và sửa lỗi .....	31
4.5 Triển khai và ghi bản nghiệm thu .....	32
CHƯƠNG 5 – KẾT QUẢ ĐẠT ĐƯỢC .....	33
Giao diện của trang chủ quản lý nhân viên.....	36
CHƯƠNG 6 – KẾT LUẬN.....	37
6.1. Ưu điểm của đề tài .....	37
6.2. Nhược điểm và hướng phát triển của đề tài .....	37
6.3. Hướng phát triển cho tương lai .....	38
PHỤ LỤC.....	40



## **DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT**

### **CÁC CHỮ VIẾT TẮT**

POS Point Of Sale

## **DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ**

### **DANH MỤC HÌNH**

Hình 3.1: Usecase hệ thống.....	19
Hình 3.2. Usecase của admin đối với hệ thống.....	19
Hình 3.3. Usecase của nhân viên đối với hệ thống .....	20
Hình 3.4. Hình ảnh miêu tả hoạt động đăng ký của hệ thống.....	21
Hình 3.5. Hình ảnh miêu tả hoạt động đăng nhập của hệ thống .....	22
Hình 3.6. Hình ảnh miêu tả hoạt động thêm sản phẩm, nhân viên của hệ thống.....	23
Hình 3.6. Hình ảnh miêu tả hoạt động thêm đơn hàng mới của hệ thống .....	24
Hình 3.7. Hình ảnh miêu tả hoạt động xóa sản phẩm, khách hàng của hệ thống .....	25
Hình 3.8. ERD cho cơ sở dữ liệu hệ thống .....	26
Hình 3.9. Hình ảnh sơ đồ quan hệ cho cơ sở dữ liệu hệ thống .....	27
Hình 5.1. Hình ảnh giao diện trang thống kê sản phẩm.....	35
Hình 5.2. Hình ảnh giao diện trang quản lý khách hàng.....	36
Hình 5.3. Hình ảnh giao diện trang thêm sản phẩm.....	36

### **DANH MỤC BẢNG**

Bảng 1.1: Bảng mô tả yêu cầu của admin đối với hệ thống .....	6
Bảng 1.2: Bảng mô tả yêu cầu của nhân viên đối với hệ thống.....	9
Bảng 1.3: Bảng mô tả các trường dữ liệu theo yêu cầu cho hệ thống .....	12

# CHƯƠNG 1 – GIỚI THIỆU ĐỀ TÀI

## 1.1 Trình bày về đề tài

Đồ án cuối kỳ môn “Phát triển ứng dụng web Node Js” sẽ thực hiện nghiên cứu và hiện thực hóa ứng dụng website toàn diện cung cấp chức năng Point of Sale (POS) cho cho một cửa hàng bán lẻ chuyên về điện thoại và phụ kiện.

Mục tiêu chính là tăng cường hiệu suất của các giao dịch bán hàng, quản lý sản phẩm một cách hiệu quả, quản lý nhân viên và cung cấp thông tin chi tiết thông qua báo cáo và thống kê.

Đồ án cuối kỳ sẽ tập trung hiện thực bằng công nghệ Node Js với sự hỗ trợ của framework Express NodeJs. Cơ sở dữ liệu được sử dụng MongoDB, một hệ quản lý cơ sở dữ liệu phi quan hệ (NoSQL) phổ biến. .

## 1.2. Lý do chọn đề tài

Phần mềm quản lý bán hàng (POS) cho cửa hàng di động là một ứng dụng quan trọng giúp cải thiện hiệu suất kinh doanh và quản lý tốt hơn. Trong lĩnh vực bán lẻ, việc sử dụng một hệ thống POS hiện đại là không thể thiếu để tối ưu hóa quy trình kinh doanh.

Phần mềm POS cho phép cửa hàng di động thực hiện các chức năng quan trọng như ghi nhận doanh số bán hàng, quản lý kho, theo dõi doanh thu, và tạo ra các báo cáo chi tiết. Việc này giúp đơn giản hóa quy trình bán hàng, tăng cường khả năng đáp ứng nhanh chóng đối với nhu cầu của khách hàng và cải thiện trải nghiệm mua sắm. Đồ án cuối kỳ sẽ xây dựng một ứng dụng POS hiện đại và linh hoạt, nhằm đáp ứng đầy đủ nhu cầu quản lý và kinh doanh của cửa hàng di động. Đồng thời, chúng tôi nhấn mạnh vào việc cung cấp giao diện thân thiện với người dùng để họ có thể sử dụng ứng dụng một cách dễ dàng. Phát triển một phần mềm POS hiệu quả sẽ giúp nâng cao khả năng cạnh tranh và quản lý kinh doanh một cách hiệu quả cho cửa hàng di động.

## 1.3. Yêu cầu đề tài cần được thực hiện

### 1.3.1 Ràng buộc về ứng dụng

Xây dựng ứng dụng web Point Of Sale cửa hàng bán điện thoại cùng các phụ kiện - hệ thống phần mềm dùng để quản lý quy trình bán hàng và các giao dịch tại điểm bán lẻ. Mục tiêu chính của hệ thống này dùng để tối ưu hóa và tự động hóa quá trình bán hàng, từ việc xử lý các sản phẩm, quản lý khách hàng, nhân viên đến việc thống kê và quản lý các dịch vụ thanh toán.

Các công nghệ chính sử dụng:

- Công nghệ: Node Js
- Framework: Express Js
- Cơ sở dữ liệu: MongoDB
- Mô hình: MVC

### 1.3.2 Người dùng hệ thống

Người dùng của ứng dụng web này chỉ có Admin/Quản lý cửa hàng và Nhân viên tại cửa hàng điện thoại đó. Khách hàng sẽ không tương tác với hệ thống.

### 1.3.3 Các chức năng của hệ thống

Đối với Administration/Quản lý cửa hàng

Bảng 1.1: Bảng mô tả yêu cầu của admin đối với hệ thống

Loại chức năng	Chức năng	Quy trình/Mô tả	Ràng buộc	Trường dữ liệu
Tài khoản	Đăng nhập			- Mặc định: * Username Admin * Password: Admin
	Đăng ký tài khoản cho nhân viên	+ Nhập thông tin		- Cần nhập: * Họ tên * Địa chỉ Email
		+ Đăng ký thành công. Email được gửi tới nhân viên	- Link chỉ có hiệu lực trong 1 phút	

	Xem thông tin hồ sơ của tài khoản			- Hình ảnh hồ sơ cá nhân. - Họ tên tài khoản
	Cập nhật thông tin hồ sơ			- Hình ảnh hồ sơ cá nhân. - Họ tên tài khoản
Nhân viên	Xem danh sách tất cả nhân viên			Avatar, fullname, inactivate status nếu có, locked status nếu có
	Xem thông tin chi tiết nhân viên			
	Thê hiện các hành động đối với từng nhân viên	- Cho phép gửi lại đường link đăng nhập về gmail người dùng. - Khóa, mở khóa tài khoản		
		- Xem thông tin bán hàng/ doanh số bán hàng của nhân viên		
Quản lý danh mục hàng hóa (Product Catalog Management)	Xem danh sách sản phẩm (viewing product listings)			
	Thêm sản phẩm mới			
	Cập nhật thông tin sản phẩm		Sản phẩm đã bán thì không thể cập nhật thông tin trừ số lượng sản phẩm, mô tả và thể loại	

	Xóa sản phẩm		Chỉ được xóa sản phẩm khi nó không có mặt trong đơn hàng (Khi chưa ai mua)	
Báo cáo và phân tích	Xem báo cáo và phân tích theo các mốc thời gian	Hiện thị ra các thông tin cần thiết + Tổng số tiền thu được + Số lượng đơn hàng + Số lượng sản phẩm với danh sách đơn hàng theo thời gian + Có thể chọn đơn hàng cụ thể xem chi tiết bên trong	Các mốc thời gian quy định: + Hôm nay + Hôm qua + Trong 7 ngày + Tháng hiện tại + Lựa chọn khoảng thời gian nhất định	
	Xem lợi nhuận			

Đối với nhân viên

Bảng 1.2: Bảng mô tả yêu cầu của nhân viên đối với hệ thống

Loại chức năng	Chức năng	Quy trình/Mô tả		Ràng buộc	Trường dữ liệu
Tài khoản	Đăng nhập	Lần đầu:	<ul style="list-style-type: none"> <li>- Nhập thông tin đăng nhập</li> <li>* Không thể truy cập tới form đăng nhập 1 cách trực tiếp như những người khác</li> </ul>	<ul style="list-style-type: none"> <li>- Nếu cố gắng đăng nhập trực tiếp, hiển thị Error Message. E.g: “Please login by clicking on the link in your email”</li> <li>- Yêu cầu dùng link được gửi trong email để đăng nhập</li> </ul>	<ul style="list-style-type: none"> <li>* Username</li> <li>* Password: Là tên Username (Lần đầu đăng nhập)</li> </ul>
			Đăng nhập thành công - Tạo mật khẩu mới (Không cần nhập lại mật khẩu cũ)	- Nếu không tạo mật khẩu mới. Khi truy cập vào các chức năng sẽ bị buộc log out	
		Lần 2:	Đăng nhập bình		<ul style="list-style-type: none"> <li>* Username</li> <li>* Password:</li> </ul>

			thường		
	Xem thông tin hồ sơ				- Hình ảnh hồ sơ cá nhân. - Họ tên tài khoản
	Cập nhật thông tin hồ sơ				- Hình ảnh hồ sơ cá nhân. - Họ tên tài khoản
Quản lý danh mục hàng hóa	Xem danh sách sản phẩm	Không thể chỉnh sửa bất kỳ thông tin nào		- Không thể xem được giá gốc của sản phẩm	
Quản lý khách hàng)	Tạo tài khoản khách hàng	SDT khách hàng mua lần một	Hệ thống sẽ tự động tạo tài khoản mới mà không cần tạo tay	Cần nhập họ tên và địa chỉ khách hàng	
		Sdt khách hàng đã từng mua	Hiện thị tên và địa chỉ khách hàng		
	Xem thông tin các khách hàng				-Tên - SDT - Địa chỉ
	Xem các lịch sử mua hàng của khách			Có thể xem thêm chi tiết đơn hàng để xem thông tin cụ thể	-Tổng số lượng -Tiền khách hàng đã đưa, -Tiền thừa còn lại - Ngày mua - Số lượng sản phẩm
Quá trình thanh	Nhập sản phẩm cần thanh	- Người mua có giỏ hàng, cần thanh toán. Tiến			



toán	toán	<p>hành tính tiền các sản phẩm:</p> <ul style="list-style-type: none"> <li>+ Nhập tên, tìm kiếm sản phẩm</li> <li>+ Nhập/quét mã barcode</li> <li>- Các sản phẩm được thêm vào lần lượt sẽ hiện thị lên listview mà không cần xác nhận</li> <li>-Hiện thị cùng các thông tin:               <ul style="list-style-type: none"> <li>+ Số lượng sản phẩm</li> <li>+ Đơn giá sản phẩm</li> <li>+ Tổng tiền sản phẩm</li> </ul> </li> <li>- Khi sản phẩm được thêm vào thì lần lượt các trường dữ liệu sẽ được cập nhật tự động</li> </ul>		
		Cập nhật sản phẩm, Xóa sản phẩm Khi update số lượng hoặc xóa sản phẩm ra khỏi list thì cũng sẽ cập nhật tự động sản phẩm		
	Xem hóa đơn trước khi xuất	<p>Hiện thị các thông tin:</p> <ul style="list-style-type: none"> <li>+ Tổng toàn bộ tiền của đơn hàng đó</li> </ul>	- Nếu là khách hàng mới thì nhập thêm tên và địa chỉ khách hàng.	

		+ Chỗ nhập thông tin khách hàng (số điện thoại) + Tiền khách hàng đưa + Tiền thừa trả lại khách hàng		
Báo cáo và phân tích (Reporting and Analytics)	Xem báo cáo và phân tích theo các mốc thời gian	Hiện thị ra các thông tin cần thiết + Tổng số tiền thu được (total amount received) ~ doanh thu + Số lượng đơn hàng (number of orders) + Số lượng sản phẩm với danh sách đơn hàng theo thời gian + Có thể chọn đơn hàng cụ thể xem chi tiết bên trong	Các mốc thời gian quy định: + Hôm nay + Hôm qua + Trong 7 ngày + Tháng hiện tại + Lựa chọn khoảng thời gian nhất định	

#### 1.3.4 Các trường dữ liệu

Bảng 1.3: Bảng mô tả các trường dữ liệu theo yêu cầu cho hệ thống

Dữ liệu	Yêu cầu	Kiểu dữ liệu
Tên người dùng	Phần đầu của gmail	String
Mật khẩu		String
Sản phẩm	Barcode	String
	Tên sản phẩm	String
	Giá nhập	Number
	Giá bán lẻ	Number
	Phân loại	String
	Ngày sản xuất sản phẩm	Date

Profile	/	/
Khách hàng	Tên, số điện thoại, địa chỉ	

## CHƯƠNG 2 – CƠ SỞ LÝ THUYẾT

*Tổng quan:* trình bày các cơ sở lý thuyết được áp dụng trong quá trình hiện thực đồ án cuối kỳ.

### 1.1 ExpressJS

ExpressJS là một framework linh hoạt và mạnh mẽ được sử dụng để xây dựng ứng dụng web bằng NodeJS. . Việc sử dụng ExpressJS trong project web POS (web quản lý bán hàng) có nhiều lợi ích như sau:

Xử lý yêu cầu và phản hồi: ExpressJS giúp dễ dàng xử lý yêu cầu HTTP từ phía client và phản hồi tương ứng. Điều này cho phép xây dựng các tính năng như đăng ký, đăng nhập, thêm sản phẩm vào giỏ hàng, thanh toán và xử lý đơn đặt hàng.

Định tuyến linh hoạt: ExpressJS cung cấp hệ thống định tuyến mạnh mẽ, giúp quản lý các tuyến đường (routes) trong ứng dụng web bán hàng. Bằng cách xác định các tuyến đường, có thể xử lý các yêu cầu từ client và hiển thị các trang tương ứng như trang chủ, danh sách sản phẩm, chi tiết sản phẩm, giỏ hàng, thanh toán, ...

Middleware và xác thực: ExpressJS hỗ trợ middleware, cho phép thực hiện các xử lý trung gian trước khi yêu cầu đến các xử lý cuối cùng. Có thể sử dụng middleware để xác thực người dùng, kiểm tra quyền truy cập, ghi nhật ký hoạt động, xử lý lỗi, và nhiều tác vụ khác liên quan đến việc bảo mật và xử lý dữ liệu trong ứng dụng web bán hàng.

Thao tác với cơ sở dữ liệu: ExpressJS không đi kèm với một cơ sở dữ liệu cụ thể, nhưng nó linh hoạt và có thể kết hợp với các cơ sở dữ liệu phổ biến như MongoDB hoặc MySQL. Có thể sử dụng các thư viện như Mongoose hoặc Sequelize để tương tác với cơ sở dữ liệu và thực hiện các thao tác như lấy dữ liệu sản phẩm, thêm/sửa/xóa sản phẩm, quản lý đơn đặt hàng, ...

### 1.2 MongoDB

MongoDB là một hệ quản trị cơ sở dữ liệu phi cấu trúc, NoSQL, được sử dụng

phổ biến trong các ứng dụng web hiện đại. Việc sử dụng MongoDB có nhiều lợi ích như sau:

**Phi cấu trúc và linh hoạt:** MongoDB là một hệ quản trị cơ sở dữ liệu phi cấu trúc, cho phép lưu trữ dữ liệu dưới dạng tài liệu (document) JSON. Điều này giúp linh hoạt trong việc thay đổi cấu trúc dữ liệu, thêm mới hoặc xóa các trường dữ liệu một cách dễ dàng mà không cần thay đổi cấu trúc tổng thể của cơ sở dữ liệu.

**Hiệu suất và mở rộng:** MongoDB được thiết kế để có hiệu suất cao và mở rộng tốt, cho phép xử lý lượng dữ liệu lớn và tải trọng công việc cao.

**Tốc độ truy xuất dữ liệu:** MongoDB sử dụng cơ chế lưu trữ trên bộ nhớ đệm để tăng tốc độ truy xuất dữ liệu, giúp cho ứng dụng hoạt động nhanh hơn.

**Khả năng sao lưu dữ liệu:** MongoDB cung cấp các khả năng và công cụ để sao lưu và phục hồi dữ liệu đảm bảo tính an toàn và khả dụng của dữ liệu.

**Tính bảo mật:** MongoDB cung cấp các tính năng bảo mật như xác thực người dùng, mã hóa dữ liệu và kiểm soát quyền truy cập. Điều này giúp bảo vệ dữ liệu khách hàng, thông tin giao dịch và các thông tin quan trọng khác trong quá trình quản lý bán hàng.

**Dễ dàng mở rộng:** MongoDB cho phép mở rộng cơ sở dữ liệu một cách dễ dàng. Tích hợp tốt với JavaScript: MongoDB được tích hợp tốt với JavaScript, cho phép sử dụng ngôn ngữ này cả trong ứng dụng back-end và front-end. Điều này giúp đơn giản hóa việc phát triển và tương tác với cơ sở dữ liệu từ các ứng dụng web sử dụng JavaScript hoặc các framework như Node.js hoặc Angular.

### **1.3 Javascript**

JavaScript là một ngôn ngữ lập trình phổ biến trong phát triển web, và nó được sử dụng rộng rãi. Việc sử dụng JavaScript trong phát triển web có nhiều lợi ích quan trọng:

**Tương tác người dùng:** JavaScript cho phép tạo ra các trang web tương tác và

đáp ứng, giúp cải thiện trải nghiệm người dùng. Bằng cách sử dụng JavaScript, có thể thực hiện các hành động như xử lý sự kiện, kiểm tra và xác nhận dữ liệu người dùng trước khi gửi lên máy chủ, thay đổi nội dung trang mà không cần tải lại toàn bộ trang và tạo ra các hiệu ứng động hấp dẫn.

Tích hợp dễ dàng: JavaScript là ngôn ngữ chủ đạo trong phát triển web, được hỗ trợ trên mọi trình duyệt hiện đại. Do đó, việc tích hợp mã JavaScript vào trang web là dễ dàng và không gặp nhiều rào cản.

Đa nền tảng: JavaScript có thể chạy trên nhiều nền tảng khác nhau, bao gồm trình duyệt web, máy chủ (với Node.js) và thiết bị di động. Điều này cho phép phát triển ứng dụng web và ứng dụng di động sử dụng cùng một ngôn ngữ, giúp tiết kiệm thời gian và công sức trong quá trình phát triển và bảo trì.

Cộng đồng mạnh mẽ: JavaScript có một cộng đồng phát triển đông đảo và sôi nổi. Có hàng ngàn thư viện, framework và công cụ JavaScript có sẵn để hỗ trợ phát triển web. Có thể tìm thấy tài liệu, ví dụ và hỗ trợ từ cộng đồng này, giúp nhanh chóng giải quyết các vấn đề và tận dụng tối đa tiềm năng của JavaScript.

Giao tiếp với server: JavaScript cho phép giao tiếp với server thông qua các công nghệ như AJAX (Asynchronous JavaScript and XML) hoặc Fetch API. AJAX cho phép gửi và nhận dữ liệu từ server mà không cần tải lại trang. Bằng cách sử dụng AJAX, có thể gửi yêu cầu HTTP bất đồng bộ đến server và xử lý kết quả trả về mà không làm gián đoạn trải nghiệm người dùng. Fetch API là một cách tiếp cận hiện đại hơn và cung cấp một giao diện dễ sử dụng để gửi và nhận dữ liệu từ server.

Xây dựng giao diện người dùng động: JavaScript cho phép tạo giao diện người dùng động và tương tác. Có thể sử dụng JavaScript để thêm, sửa, xóa các phần tử trên trang, tạo các hiệu ứng động, tương tác với người dùng qua sự kiện (như nhấp chuột, nhập liệu) và thay đổi nội dung trang mà không cần tải lại hoàn toàn trang. Điều này tạo ra trải nghiệm người dùng mượt mà và tương tác.

Xử lý dữ liệu phía client: JavaScript cho phép xử lý dữ liệu phía client trước khi gửi dữ liệu lên server. Có thể thực hiện kiểm tra dữ liệu người dùng, xác thực dữ liệu, thay đổi định dạng dữ liệu, tính toán và xử lý các tác vụ phức tạp khác trên dữ liệu trên trình duyệt. Điều này giúp giảm tải băng thông và tăng tốc độ phản hồi của ứng dụng, vì một phần xử lý dữ liệu được thực hiện trên trình duyệt thay vì gửi yêu cầu tới server.

#### **1.4 Handlebars**

Handlebars là một thư viện template engine phổ biến trong phát triển web. Nó cho phép lập trình viên tạo ra các template HTML linh hoạt và dễ dàng quản lý bằng cách sử dụng cú pháp tương tự như các ngôn ngữ lập trình khác.

Handlebars giúp tách biệt logic và giao diện trong ứng dụng web bằng cách cho phép bạn định nghĩa các template HTML chứa các thẻ và cú pháp đặc biệt. Bằng cách sử dụng Handlebars, bạn có thể thêm các biểu thức, điều kiện, vòng lặp vào template HTML để tạo ra các trang web động và tùy chỉnh.

## CHƯƠNG 3 – PHÂN TÍCH THIẾT KẾ

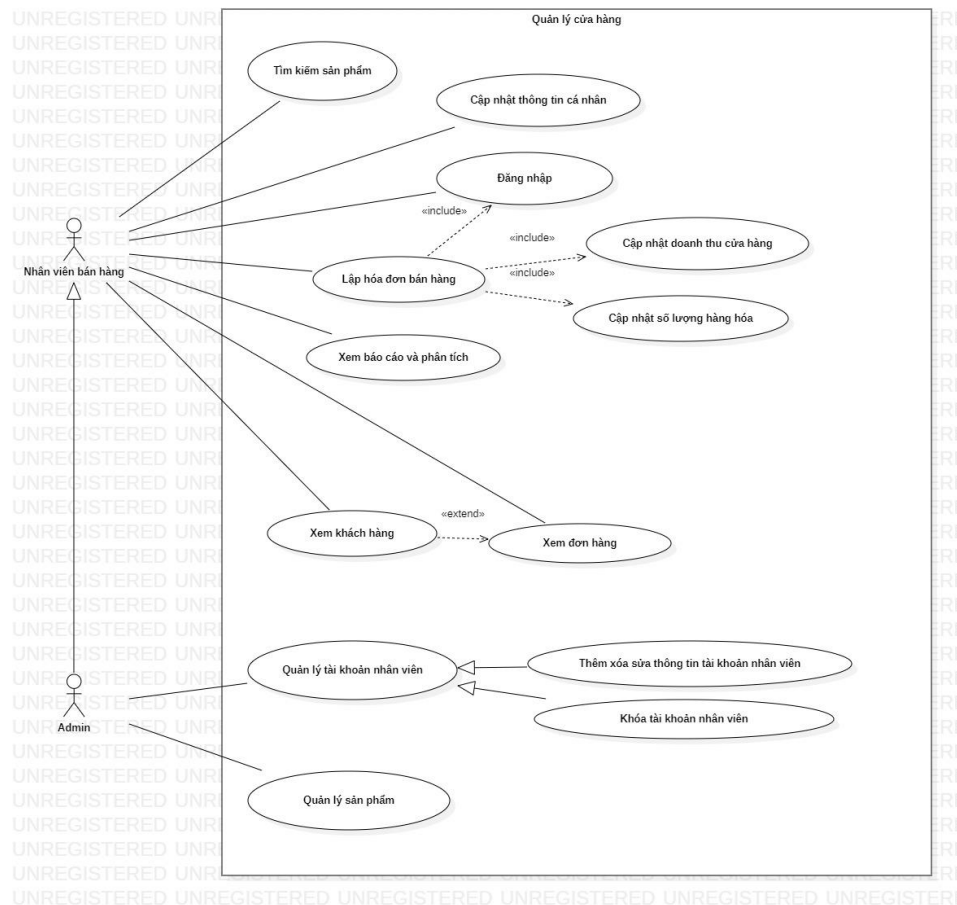
Chương 3 trình bày việc phân tích thiết kế hệ thống trực quan thông qua các sơ đồ hướng đối tượng.

### 3.1 Đặt vấn đề

Ứng dụng web POS sẽ giúp quản lý quá trình quản lý quá trình mua bán diễn ra tại cửa hàng. Việc giao tiếp với hệ thống chỉ diễn ra trên hai đối tượng Quản lý/Admin và nhân viên bán hàng. Hệ thống phải đáp ứng các chức năng chủ yếu liên quan đến mua bán.

### 3.2 Thiết kế sơ đồ hệ thống

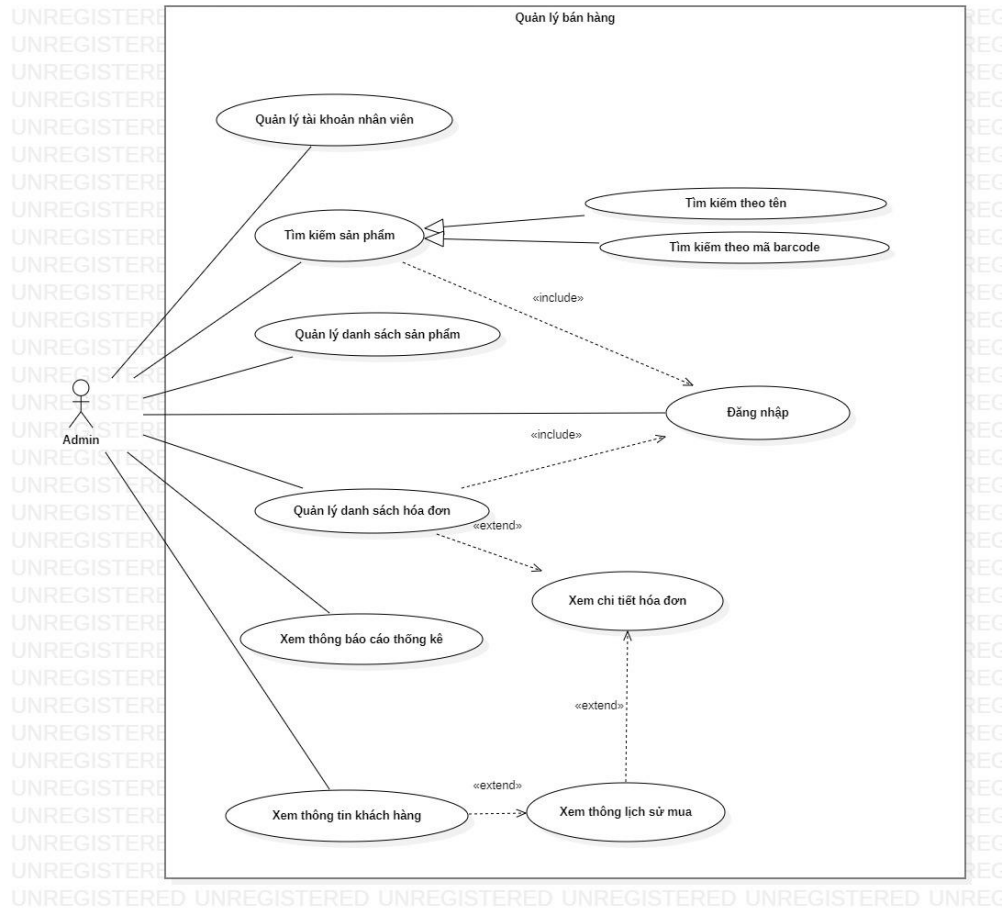
Mô hình use case tổng quát mô tả các chức năng của web và sự tương tác của admin với người dùng đối với trang web.





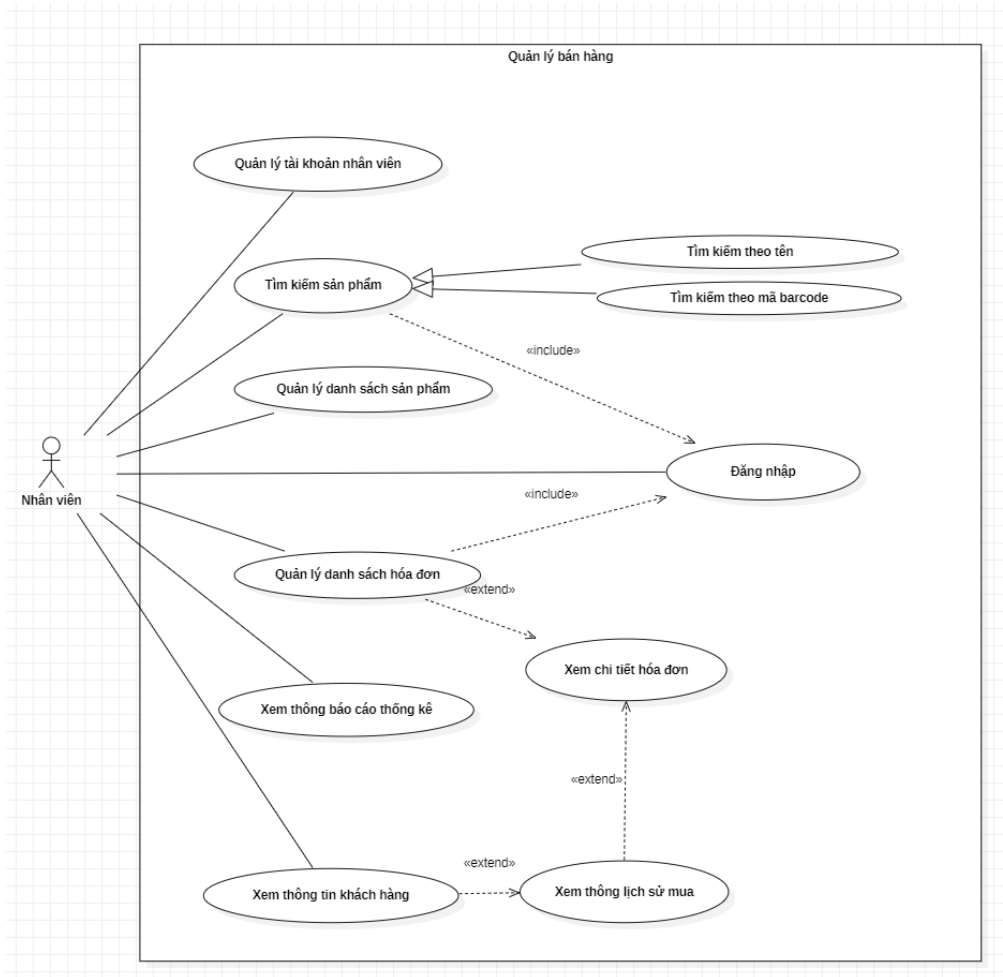
Hình 3.1: Usecase hệ thống

Hành vi của admin tương tác trên hệ thống



Hình 3.2. Usecase của admin đối với hệ thống

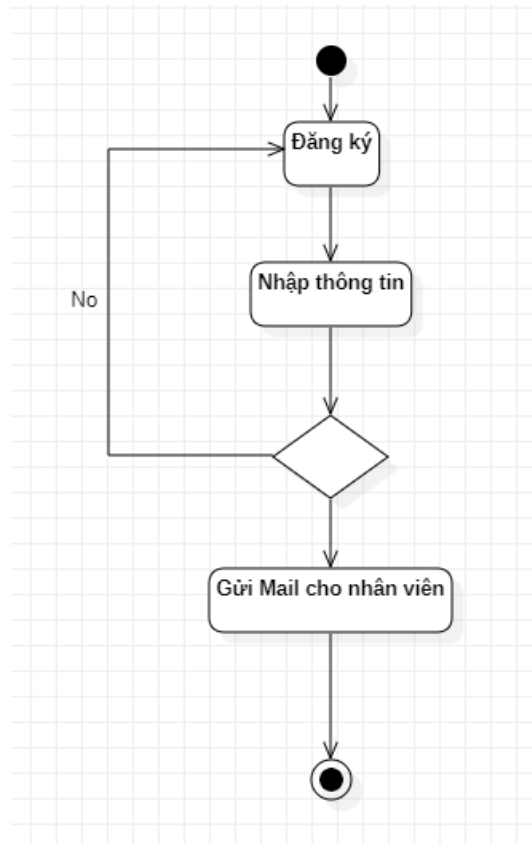
### Hành vi của người dùng tương tác trên hệ thống



Hình 3.3. Usecase của nhân viên đối với hệ thống

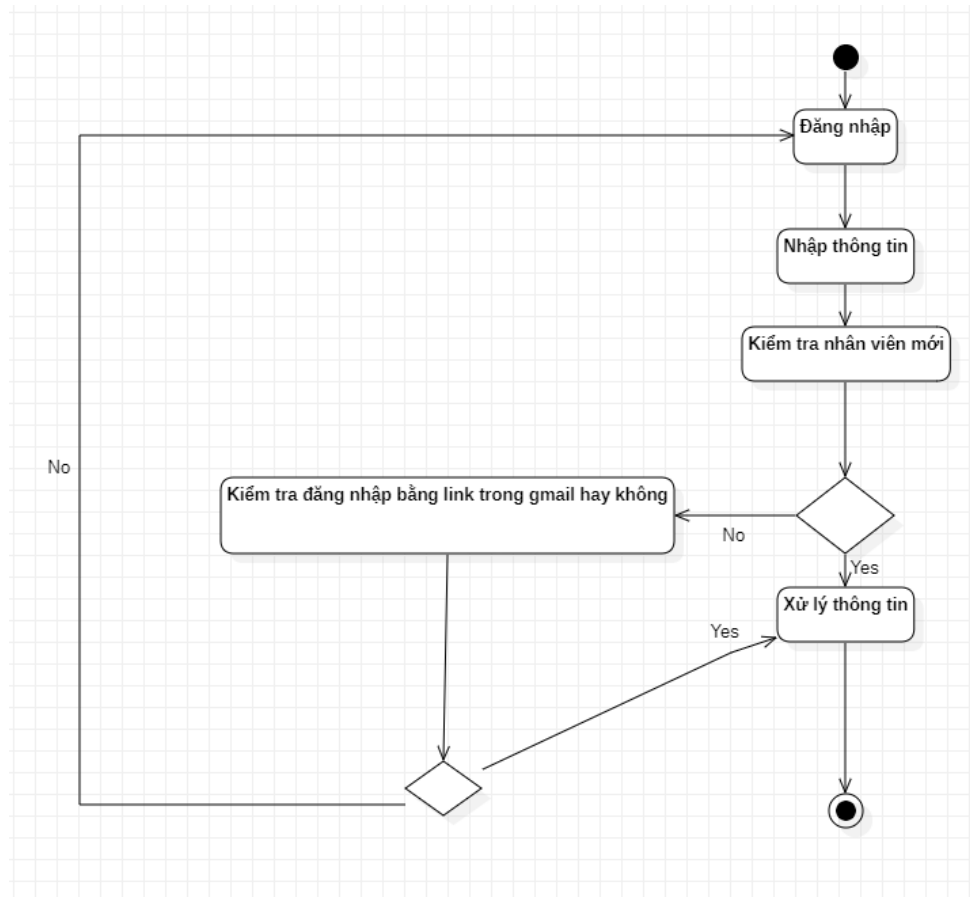
### 3.3 Hoạt động của trang web

Hoạt động đăng ký tài khoản mới của admin



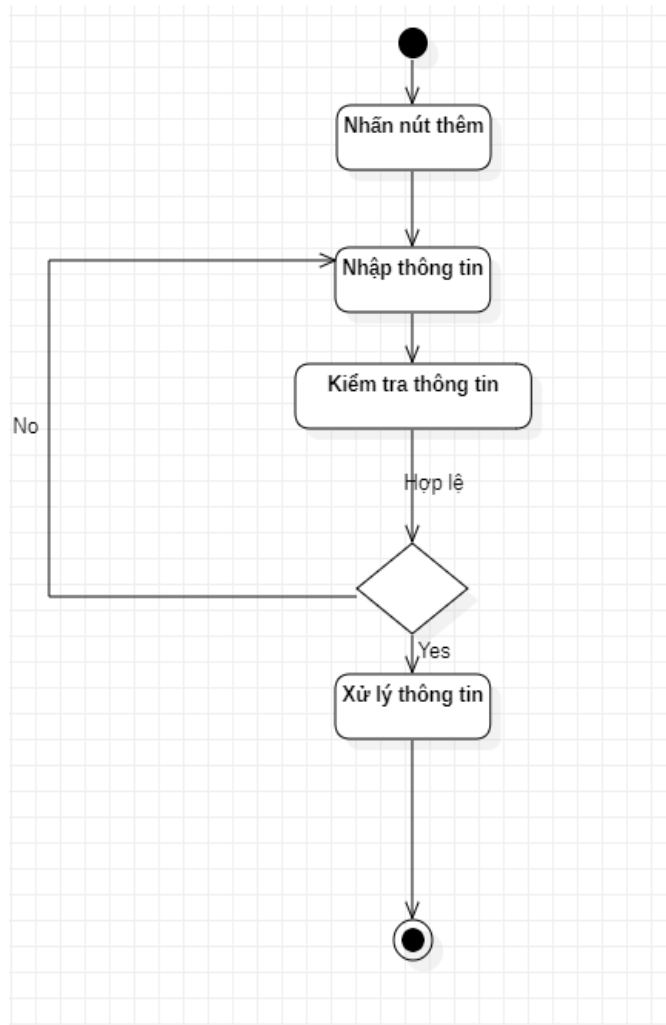
Hình 3.4. Hình ảnh miêu tả hoạt động đăng ký của hệ thống

- Luồng sự kiện cho chức năng đăng nhập



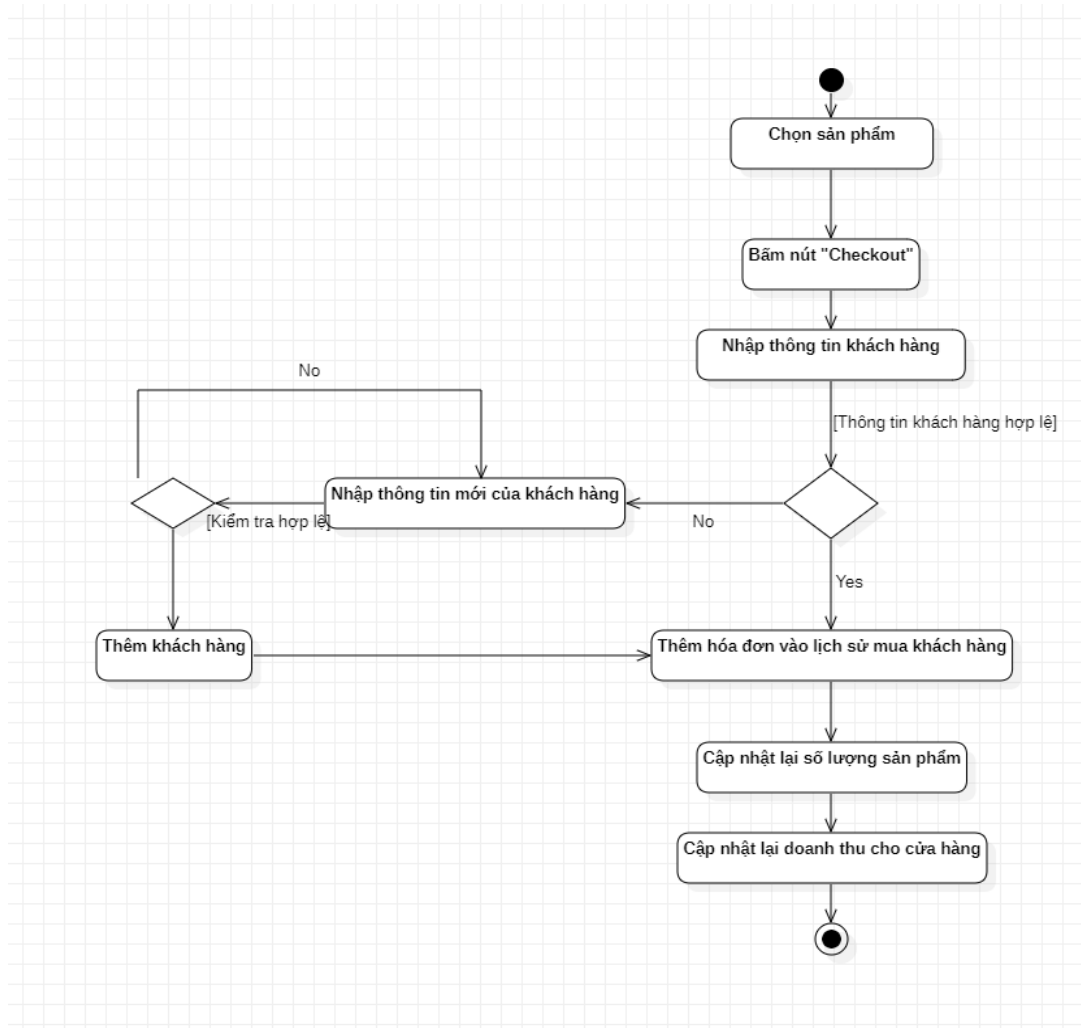
Hình 3.5. Hình ảnh miêu tả hoạt động đăng nhập của hệ thống

- Luồng sự kiện cho chức năng thêm khách hàng, thêm sản phẩm



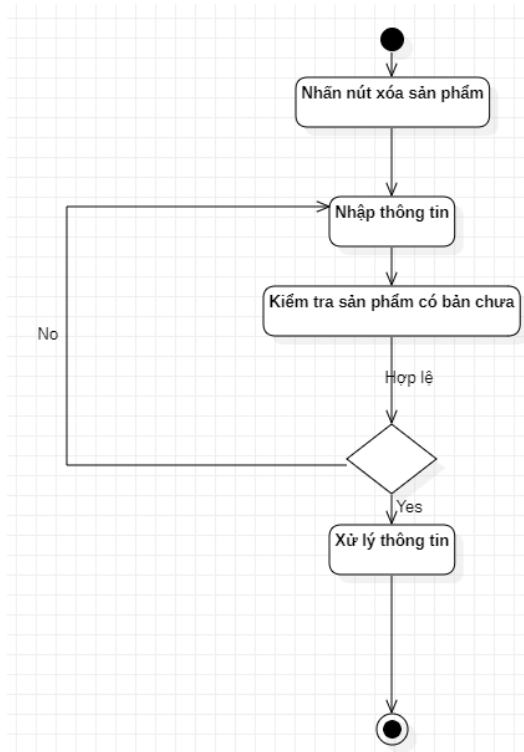
Hình 3.6. Hình ảnh miêu tả hoạt động thêm sản phẩm, nhân viên của hệ thống

- Luồng sự kiện cho chức năng thêm đơn hàng mới



Hình 3.6. Hình ảnh miêu tả hoạt động thêm đơn hàng mới của hệ thống

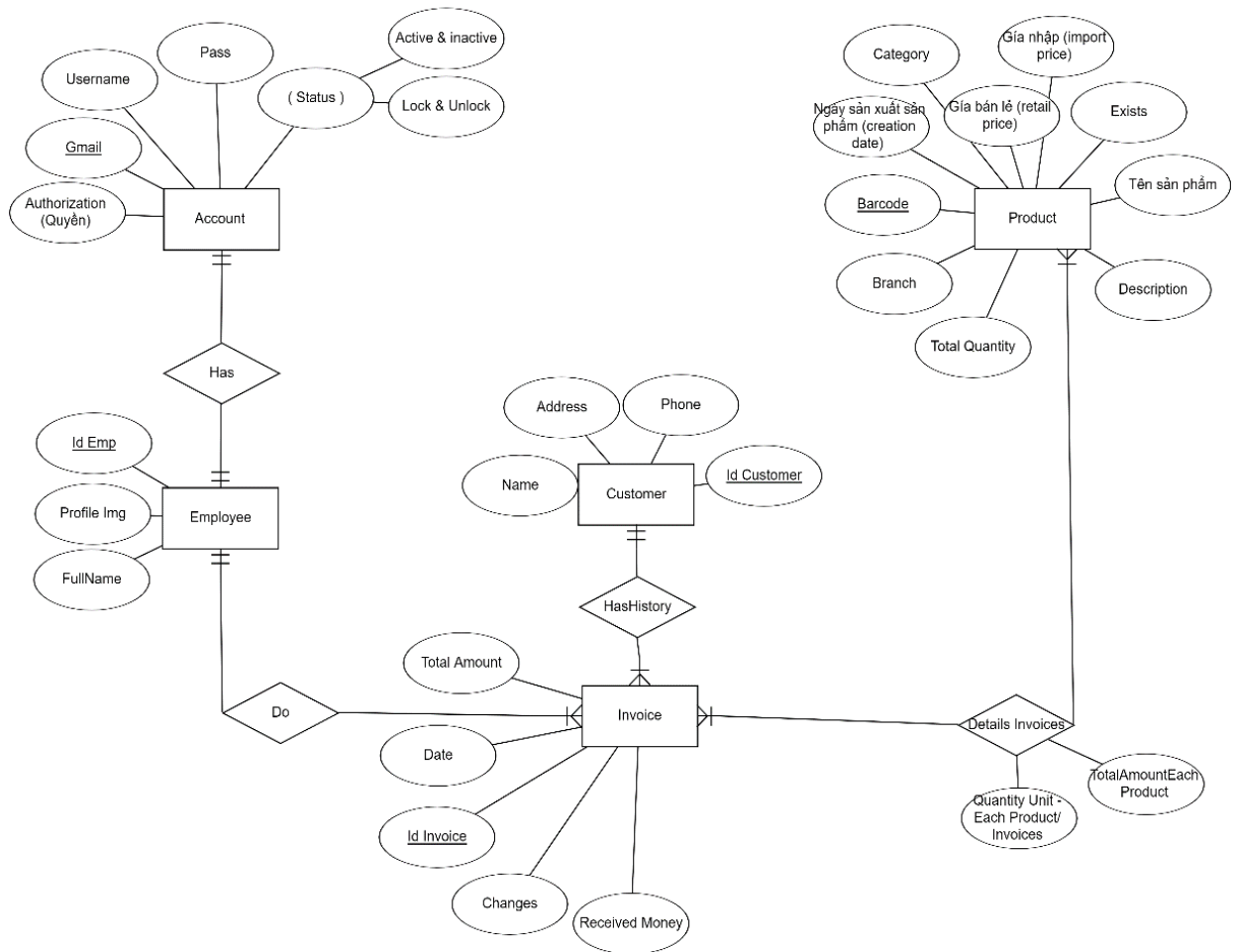
- Luồng sự kiện cho hoạt động xóa sản phẩm



Hình 3.7. Hình ảnh miêu tả hoạt động xóa sản phẩm, khách hàng của hệ thống

### 3.4 Cấu trúc cơ sở dữ liệu

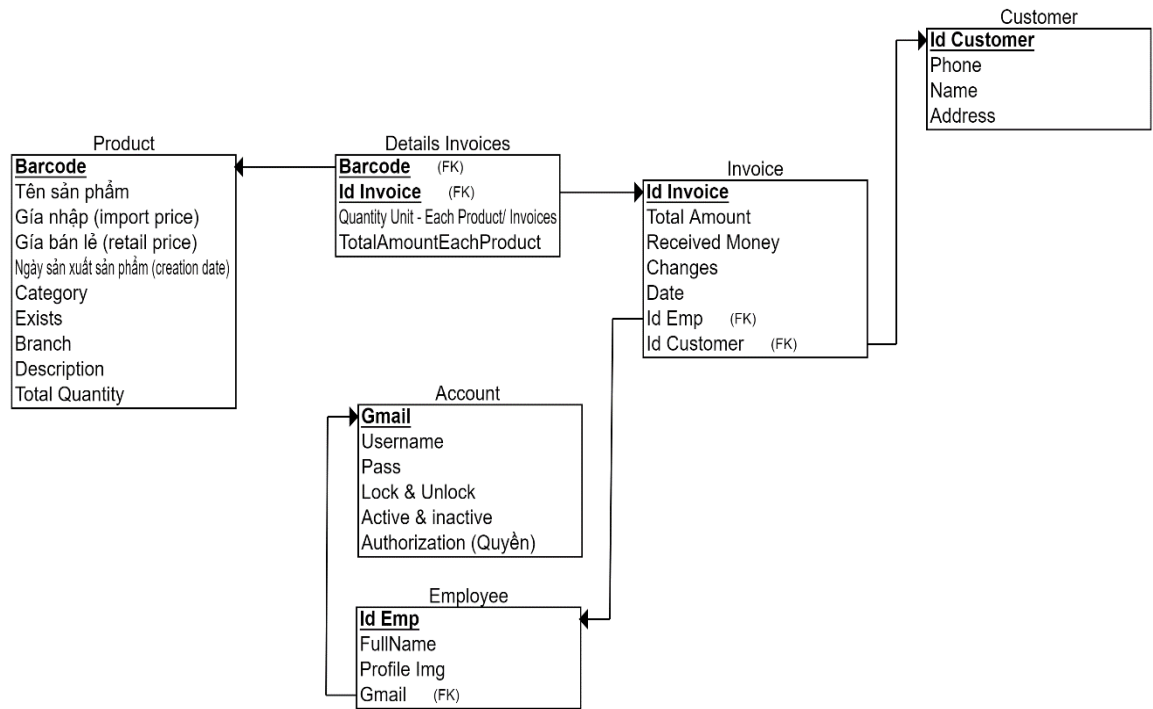
- Sơ đồ ERD cho cơ sở dữ liệu của ứng dụng web



Hình 3.8. ERD cho cơ sở dữ liệu hệ thống



- Sơ đồ quan hệ cho cơ sở dữ liệu của ứng dụng web



Hình 3.9. Hình ảnh sơ đồ quan hệ cho cơ sở dữ liệu hệ thống

## CHƯƠNG 4 – HIỆN THỰC HỆ THỐNG

*Chương 4 trình bày quá trình xây dựng của hệ thống từ kết quả đã phân tích ở trên.*

### 4.1 Phân tích thiết kế yêu cầu

Tại đây, chúng em sẽ thực hiện nghiên cứu các yêu cầu của hệ thống, các công nghệ mới sẽ được áp dụng vào hệ thống và xây dựng:

Mô hình cấu trúc hệ thống bao gồm các thành phần chính, mối quan hệ, và cách chúng tương tác với nhau.

Mô tả các chức năng cụ thể mà hệ thống sẽ thực hiện bao gồm quản lý người dùng, quản lý sản phẩm, tìm kiếm, xử lý thanh toán, quản lý bảo mật... Mỗi chức năng nên được mô tả chi tiết về các input, output, và hướng xử lý.

Thiết kế giao diện trang web Figma. Mô tả cách người dùng sẽ tương tác với hệ thống ứng với các chức năng cụ thể mà người dùng có thể sử dụng.

Dữ Liệu và Cơ Sở Dữ Liệu: Xây dựng cách tổ chức dữ liệu và lưu trữ trong hệ thống

### 4.2 Thiết kế hệ thống

Dựa trên các kết quả phân tích và nguồn lực, tài nguyên và các môi trường sẵn có, hệ thống quyết định sử dụng:

Mô hình MVC: Model (M): Đại diện cho dữ liệu và logic xử lý dữ liệu - sử dụng MongoDB để lưu trữ dữ liệu. View (V) Hiển thị giao diện và tương tác phía người dùng. Controller (C) – điều khiển luồng thông tin giữa Model và View.

Cơ sở dữ liệu MongoDB, Chọn MongoDB làm hệ quản trị cơ sở dữ liệu vì tính linh động, khả năng mở rộng, và khả năng lưu trữ dữ liệu theo mô hình JSON.

```

require('dotenv').config();
//Database
const mongoose = require('mongoose');
let URI = process.env.URI;
var opts = {
  server: {
    socketOptions: { keepAlive: 1 }
  }
}

//onnect
mongoose.connect(URI, {
  useNewUrlParser: true,
  useUnifiedTopology: true,
  //useCreateIndex: true,
});
mongoose.connect(URI, opts);

const db = mongoose.connection
db.on('error', (error) => {
  console.error('Lỗi kết nối đến MongoDB:', error);
});

db.once('open', () => {
  console.log('Đã kết nối thành công đến MongoDB!');
});

module.exports = db;

```

Hình 4.1: Triển khai kết nối database tới MongoDB

Framework hỗ trợ ExpressJs Express.js là một framework Node.js mạnh mẽ, giúp xây dựng các ứng dụng web và API dễ dàng. Ngoài ra, ứng dụng Ajax để tạo kênh giao tiếp chủ yếu giữa giao diện và server.



```

1  require('dotenv').config();
2  const express = require('express');
3  const hdb = require("express-handlebars");
4  const session = require('express-session');
5  const cookieParser = require('cookie-parser');
6  const jwt = require('jsonwebtoken');
7  const flash = require('connect-flash');
8  const moment = require('moment'); // Import the moment
   library
9
10 const app = express();
11 app.set('view engine', 'handlebars');
12 app.engine("handlebars", hdb.engine({
13   defaultLayout: "main",
14 }));
15
16 app.set('views', __dirname + '/src/views');
17 app.use(express.static(__dirname+'public'));
18
19 app.use(express.json());
20 app.use(express.urlencoded());
21 app.use((session({
22   secret: process.env.KEY,
23   cookie: {
24     sameSite: 'strict',
25     resave: false,
26     saveUninitialized: true,
27     expires: 60000*60
28   }
29 })))
30 app.use(flash());
31 app.use(cookieParser());
32
33 //Add handlebars helpers
34 var hbs = hdb.create({
35   helpers: { ...
36 }
37 })
38
39 app.engine('handlebars', hbs.engine);
40 //Database connect
41 const db = require('./src/config/db.config');
42 //Router
43 const invoiceRouter = require('./src/routes/invoiceRouter.js');
44 const accountRouter = require('./src/routes/accountRouter.js');
45 const productRouter = require('./src/routes/productRouter.js');
46 const errorRouter = require('./src/routes/errorRouter.js');
47 const customerRouter = require('./src/routes/customerRouter.js');
48 const adminRouter = require('./src/routes/adminRouter.js');
49 const reportsRouter = require('./src/routes/reportsRouter.js');
50
51 invoiceRouter(app);
52 adminRouter(app);
53 accountRouter(app);
54 customerRouter(app);
55 invoiceRouter(app);
56 productRouter(app);
57 reportsRouter(app);
58 errorRouter(app);
59 //Listen
60 app.listen(process.env.PORT, () => console.log(
61   `Express started on http://localhost:${process.env.PORT};` +
62   `press Ctrl-C to terminate.`
63 ))

```

Hình 4.2: Ứng dụng Express Js trong hệ thống

### 4.3 Phát triển các tính năng

Phát triển các tính năng theo thứ tự phân tích yêu cầu:

+ Trang quản lý sản phẩm: Thêm, xóa, sửa sản phẩm – một trong những thành phần quan trọng của hệ thống, thành phần cốt lõi. Các file liên quan: product.js, category.js, productService.js, productRouter.js, productController.js, index.handlebars, productIncategory.handlebars.

+ Trang thanh toán: Tạo một đơn hàng mới cho một khách hàng bất kỳ từ ứng dụng hệ thống. Các file liên quan: invoice.js, invoiceRouter.js, productService.js, invoiceService.js, customerService.js, employeeService.js

+ Trang quản lý tài khoản cá nhân: Thêm, xóa, sửa, xem chi tiết và khóa tài khoản nhân viên, và xem lịch sử bán hàng của nhân viên đó. Các file liên quan: account.js, employee.js, accountRouter.js, employeeRouter.js, accountController.js, employeeController.js, account.handlebars, email.handlebars, index.handlebars

+ Trang quản lý hóa đơn: Xem lịch sử, chi tiết các hóa đơn đã tạo trong cửa hàng. Các file liên quan: invoice.js, invoiceRouter.js, invoiceController.js, invoiceDetails.handlebars

- + Trang quản lý khách hàng: Xem thông tin khách hàng, cùng các lịch sử thông tin khách hàng đã mua. Các file liên quan: `customer.js` , `customerService.js`, `customerRouter.js`, `customerController.js`, `index.handlebars`, `history.handlebars`
- + Trang lỗi: Hiện thị lỗi cho các trường hợp không tìm thấy các trang trên hệ thống. Các file liên quan: `errorRouter.js`, `errorController.js`, `error.handlebars`.
- + Trang báo cáo và thống kê: Xem thông tin của tất cả doanh thu và lợi nhuận các khách hàng theo thời gian. Các file liên quan: `productService.js`, `invoicesService.js`, `reportRouter.js`, `reportControllers.js`, `chart.handlebars`.
- + Trang đăng nhập: Kiểm tra phương thức đăng nhập của tài khoản. Các file liên quan: `employeeService.js`, `account.js`, `accountController.js`, `login.handlebars`
- + Trang chủ của hệ thống POS: Hiện thị các sản phẩm có trong hệ thống để nhân viên thực hiện việc tạo hóa đơn mới. Các file liên quan: `productService.js`, `productRouter.js`, `productController.js`, `index.handlebars`.

#### **4.4 Kiểm tra hệ thống và sửa lỗi**

Thực hiện kiểm tra và phân tích các thành phần/module trong hệ thống theo các quy chuẩn:

- + Trường dữ liệu chỉ nhập số: Tiền nhập, tiền bán, tiền nhận từ khách hàng, số điện thoại khách hàng
- +Trường dữ liệu chỉ nhập chữ, và không được nhập các kiểu dữ liệu đặc biệt.
- +Trường dữ liệu giới hạn
- + Tính đúng đắn, và an toàn của dữ liệu, và đảm bảo dữ liệu hiện thị chính xác yêu cầu được đề ra.
- + Kiểm tra cảnh báo và hiện thị lỗi của hệ thống.
- + Kiểm tra tính bảo mật của hệ thống
- + Kiểm tra hiệu suất của hệ thống
- + Kiểm tra từng chức năng trong hệ thống.
- + Kiểm tra hệ thống

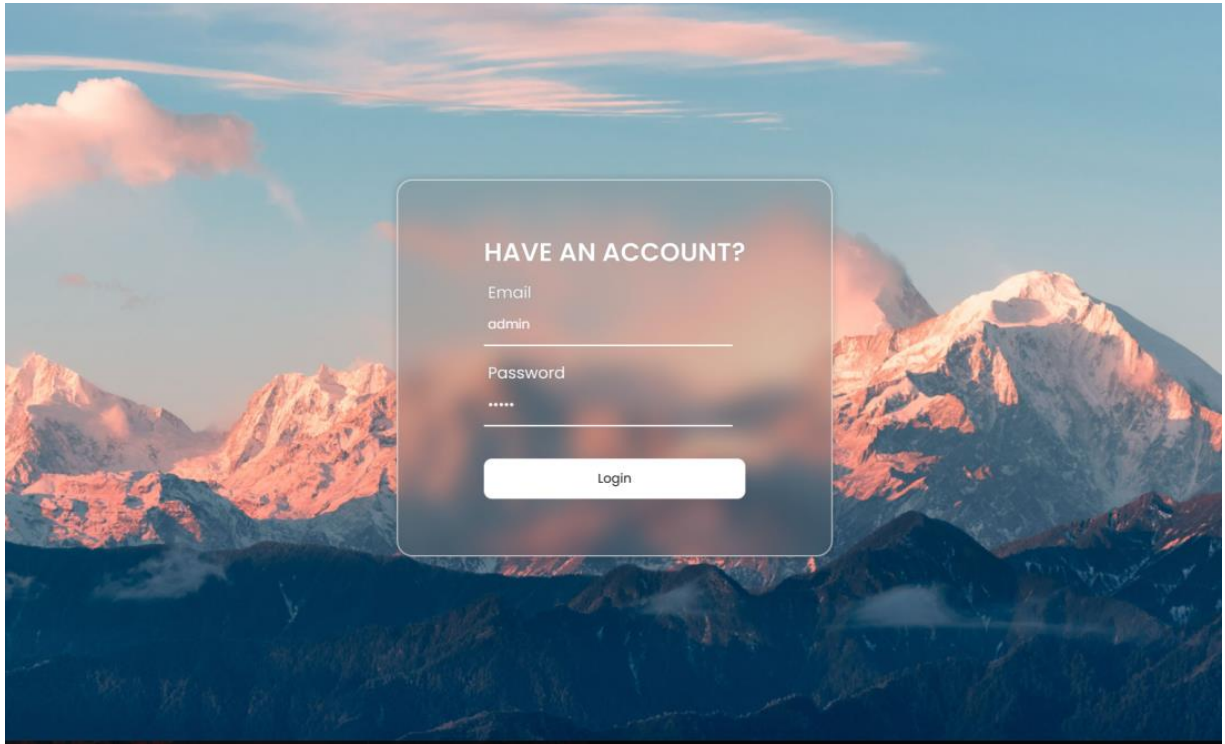
#### **4.5 Triển khai và ghi bản nghiệm thu**

Ghi nhận và trình bày kết quả đã triển khai thành công. Ghi nhận những ưu điểm và nhược điểm trong quá trình hoàn thiện đề tài

## CHƯƠNG 5 – KẾT QUẢ ĐẠT ĐƯỢC

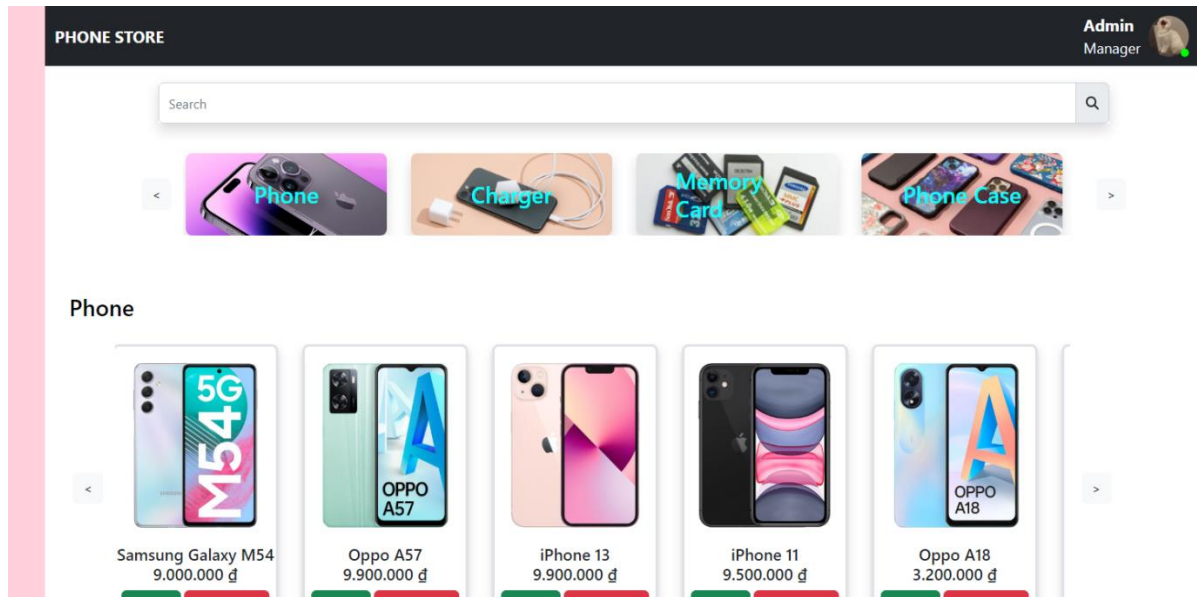
*Chương 5 trình bày quá trình xây dựng của hệ thống từ kết quả đã phân tích ở trên.*

### **Giao diện của trang đăng nhập**



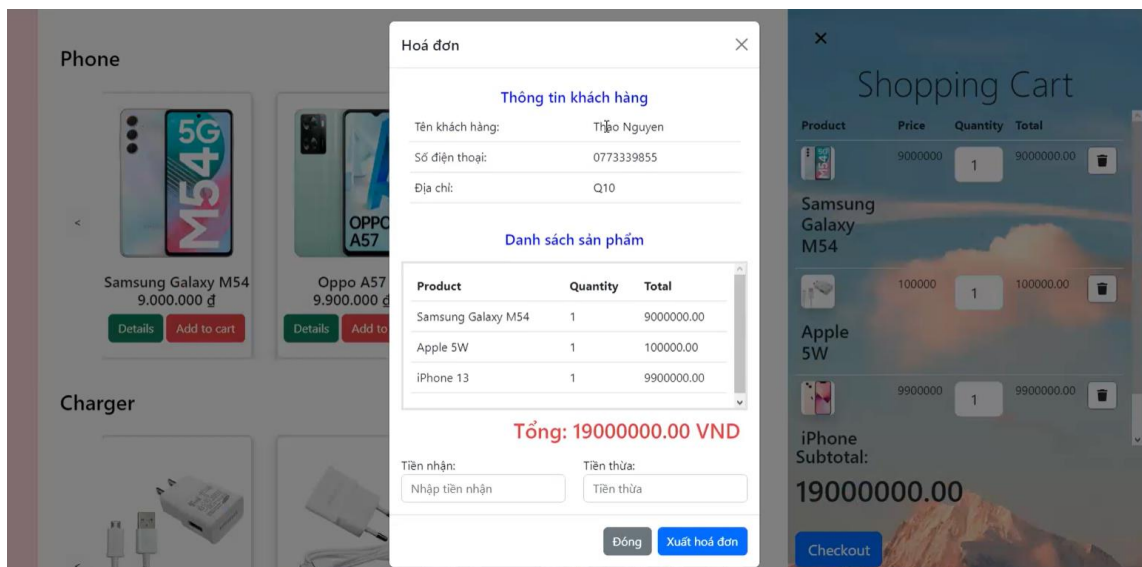
Hình 5.1. Hình ảnh giao diện đăng nhập của hệ thống

## Giao diện của trang chủ



Hình 5.2. Hình ảnh giao diện trang chủ của hệ thống

## Giao diện của trang tạo hóa đơn

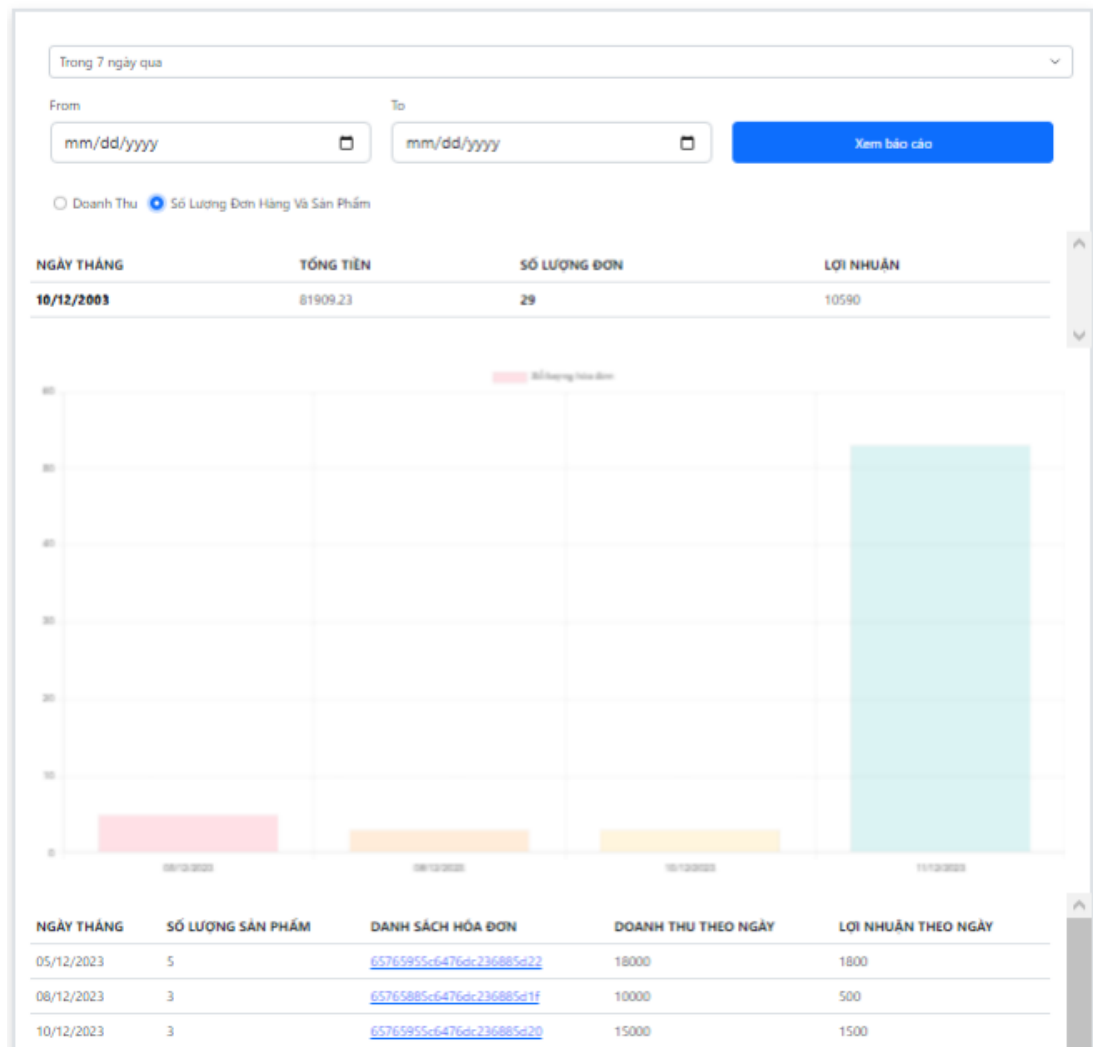


Hình 5.3. Hình ảnh giao diện trang tạo hóa đơn sản phẩm



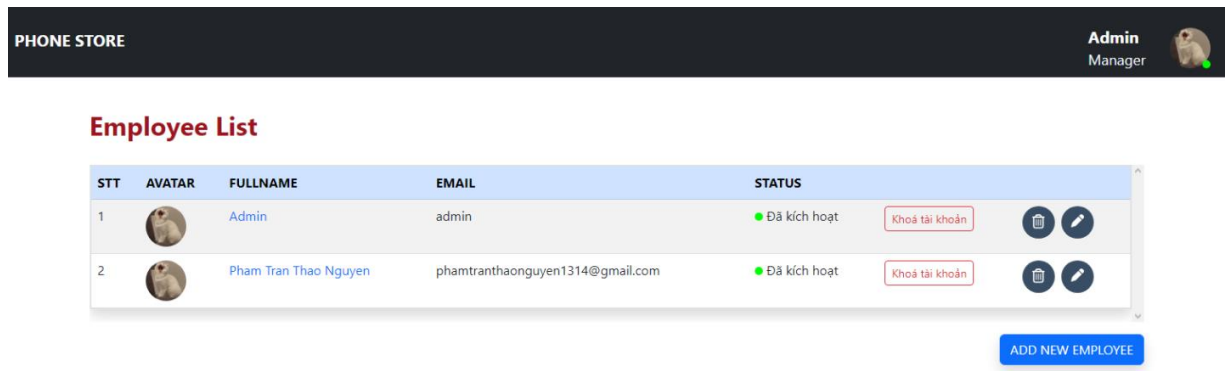
## Giao diện của trang thống kê và phân tích

### BÁO CÁO VÀ PHÂN TÍCH



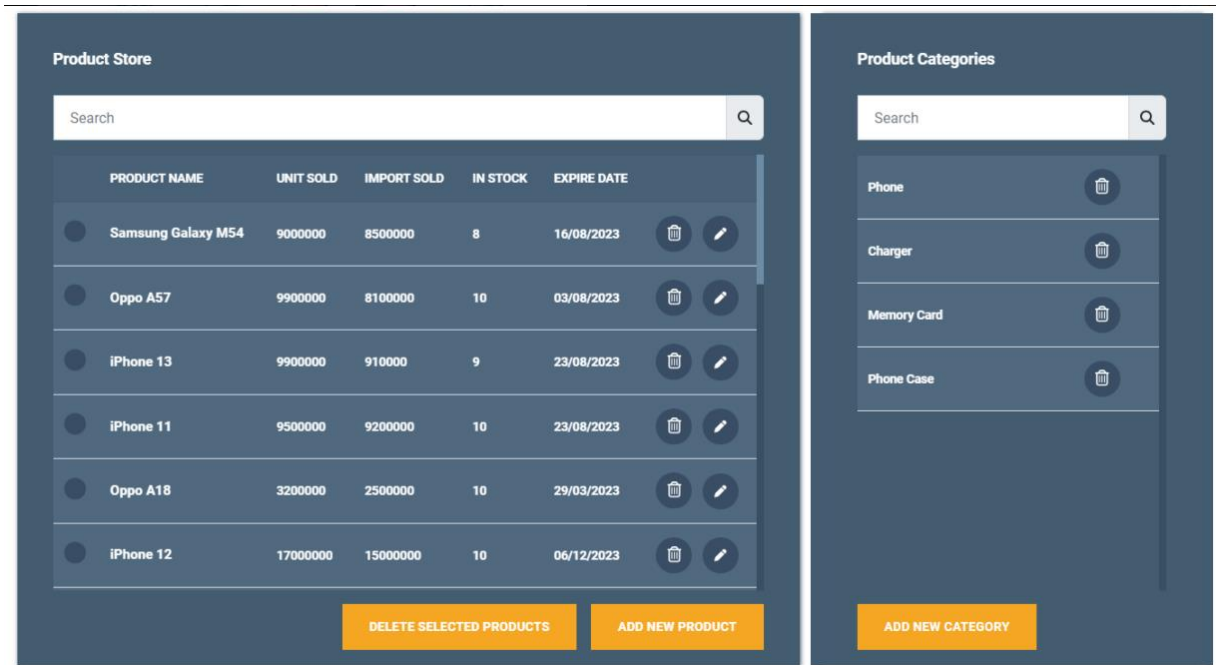
Hình 5.4. Hình ảnh giao diện trang thống kê sản phẩm

## Giao diện của trang chủ quản lý nhân viên



Hình 5.5. Hình ảnh giao diện trang quản lý khách hàng

## Giao diện của trang quản lý sản phẩm



Hình 5.6. Hình ảnh giao diện trang quản lý sản phẩm

## CHƯƠNG 6 – KẾT LUẬN

*Chương 6 trình bày những ưu, khuyết điểm trong quá trình xây dựng hệ thống từ đó đề ra hướng phát triển cho tương lai.*

### 6.1. Ưu điểm của đề tài

- Phát triển kiến thức: Tận dụng và phát huy được tốt các kiến thức đã học trong môn Phát triển ứng dụng web NodeJs.
- Giao diện người dùng: Thân thiện, dễ dàng tương tác và sử dụng.
- Hoàn thiện chức năng: Hệ thống đã đáp ứng tương đối đầy đủ và hoàn thiện các chức năng được yêu cầu.
- Phân chia module rõ ràng: Việc phân chia module trong hệ thống giúp các thành viên làm việc và tương tác dễ dàng.
- Tương thích: Hệ thống được thiết kế và phát triển với sự tương thích với các nền tảng và các công nghệ khác nhau. Điều này giúp hệ thống có khả năng tích hợp và hoạt động tốt với các hệ thống và ứng dụng khác, tạo điều kiện thuận lợi cho việc mở rộng và kết nối với các hệ thống khác.
- Tính mở rộng: Hệ thống được xây dựng với khả năng mở rộng linh hoạt, cho phép thêm mới các chức năng và tính năng trong tương lai một cách dễ dàng.

### 6.2. Nhược điểm và hướng phát triển của đề tài

- Trải nghiệm người dùng: Phần trải nghiệm giao diện người dùng có thể còn chưa đạt tối ưu, cần nâng cấp và cải thiện để cung cấp trải nghiệm tốt hơn cho người dùng.
- Bảo mật: Cần xem xét và tăng cường mật bảo mật để đảm bảo an toàn cho hệ thống và dữ liệu.
- Tối ưu hiệu suất: Cần nghiên cứu và thực hiện các biện pháp tối ưu hóa để cải thiện hiệu suất của hệ thống, đặc biệt là khi có nhiều người sử dụng.

### **6.3. Hướng phát triển cho tương lai**

- Nâng cấp giao diện người dùng: để mang lại trải nghiệm tốt hơn cho người dùng.
- Tăng cường bảo mật: để đảm bảo an toàn hơn cho hệ thống và dữ liệu.
- Tối ưu hiệu suất: Tiếp tục tìm hiểu và thực hiện các biện pháp tối ưu hóa để cải thiện hiệu suất của hệ thống, đặc biệt là trong tình huống có nhiều người sử dụng.

## TÀI LIỆU THAM KHẢO

### Tiếng Việt

- [1] TemplateMo, "Free Template 524 Product Admin," [Online]. Available: <https://templatemo.com/tm-524-product-admin>. [Accessed: 12-Nov-2023].
- [2] Colorlib, "Login Form V20 - Free Login In Form w/ Image Background 2023," [Online]. Available: <https://colorlib.com/wp/template/login-form-20/> . [Accessed: 15 Nov-2023].
- [3] TrungQuanDev, "NodeJS viết API gửi Email với OAuth2 và Nodemailer," [Online]. Available: <https://trungquandev.com/nodejs-viet-api-gui-email-voi-oauth2-va-nodemailer/>. [Accessed: 25-Nov-2023].
- [4] "Chart.js | Chart.js," [Online]. Available: <https://www.chartjs.org/docs/latest/>. [Accessed: 4-Dec-2023].

## PHỤ LỤC

1. Trang web mô tả giao diện hệ thống người dùng:

<https://www.figma.com/file/HqmJPpIdCMl8dMlpD52S87/webStore?type=design&node-id=0-1&mode=design&t=xckw4JmISN3JGvug-0>

2. Link video demo sản phẩm:

[https://drive.google.com/drive/folders/1kzDb6ozL-Dd8BO193EDA7V\\_-9JHKbD-J?usp=sharing](https://drive.google.com/drive/folders/1kzDb6ozL-Dd8BO193EDA7V_-9JHKbD-J?usp=sharing)