**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ ĐÔNG Á**

****

ĐỒ ÁN TỐT NGHIỆP

**TÊN ĐỀ TÀI: XÂY DỰNG HỆ THỐNG QUẢN LÝ BÁN HÀNG SỬ DỤNG CÔNG NGHỆ JAVA SPRING BOOT, MYSQL**

**Giảng viên hướng dẫn: ThS. Lê Thị Huyền Trang Sinh viên thực hiện: Nguyễn Tiến Đạt**

**Ngày sinh: 04/11/2003 Lớp: DCCNTT12.10.1**

**Ngành: Công nghệ thông tin Khoa: Công nghệ thông tin Khóa: 12**

**Mã sinh viên: 20210270**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ ĐÔNG Á**

**NGUYỄN TIẾN ĐẠT**

**TÊN ĐỀ TÀI: XÂY DỰNG HỆ THỐNG QUẢN LÝ BÁN HÀNG SỬ DỤNG CÔNG NGHỆ JAVA SPRING BOOT, MYSQL**

**Giảng viên hướng dẫn: ThS. Lê Thị Huyền Trang**

# LỜI CAM ĐOAN

Tôi xin cam đoan rằng đồ án tốt nghiệp với đề tài “Xây dựng hệ thống quản lý bán hàng sử dụng công nghệ Java Spring Boot, MySQL” là nghiên cứu độc lập của tôi. Đồng thời những số liệu được cung cấp từ báo cáo đều là kết quả nghiên cứu hoàn toàn trung thực, không sao chép từ bất kì một công trình nghiên cứu khác nào.

Bắc Ninh, ngày …. tháng … năm 2025

**Sinh viên**

*(ký và ghi rõ họ tên)*

# LỜI CẢM ƠN

Em xin gửi lời cảm ơn trân thành nhất đến quý thầy cô TRƯỜNG ĐẠI HỌC CÔNG NGHỆ ĐÔNG Á, những người đã dìu dắt em tận tình, đã truyền đạt cho em những kiến thức và bài học quý báu trong suốt bốn năm học em theo học tại trường.

Em xin trân trọng gửi lời cảm ơn đến tất cả các thầy cô trong khoa Công nghệ thông tin, đặc biệt là cô giáo Lê Thị Huyền Trang, cô đã tận tình hướng dẫn, chỉ bảo và giúp đỡ em trong suốt quá trình làm đồ án tốt nghiệp. Với sự hướng dẫn của cô, em đã có những định hướng tốt trong việc triển khai và thực hiện các yêu cầu trong quá trình làm đồ án tốt nghiệp.

Em xin cảm ơn những người thân và gia đình đã quan tâm động viên và luôn tạo cho em những điều kiện tốt nhất trong quá trình học tập và làm đồ án tốt nghiệp.

Ngoài ra, em cũng xin gửi lời cảm ơn tới tất cả bạn bè, đặc biệt là các bạn trong lớp đã luôn gắn bó, cùng học tập và giúp đỡ em phát triển bản thân trong những năm qua và trong suốt quá trình thực hiện đồ án này.

Em xin trân thành cảm ơn!

**Sinh viên thực hiện**

# MỤC LỤC

[LỜI CAM ĐOAN 1](#_Toc196380568)

[LỜI CẢM ƠN 2](#_Toc196380569)

[MỤC LỤC 3](#_Toc196380570)

[DANH MỤC CÁC TỪ VIẾT TẮT 5](#_Toc196380571)

[DANH MỤC BẢNG BIỂU VÀ SƠ ĐỒ 6](#_Toc196380572)

[DANH MỤC HÌNH ẢNH 8](#_Toc196380573)

[CHƯƠNG I: GIỚI THIỆU TỔNG QUAN ĐỀ TÀI 10](#_Toc196380574)

[1.1. Lý do chọn đề tài 10](#_Toc196380575)

[1.2. Mục tiêu của đề tài 10](#_Toc196380576)

[1.3. Giới hạn và phạm vi của đề tài 11](#_Toc196380577)

[1.4. Phương pháp thực hiện 11](#_Toc196380578)

[1.5. Kết quả dự kiến đạt được 12](#_Toc196380579)

[CHƯƠNG II: KIẾN THỨC NỀN TẢNG 14](#_Toc196380580)

[2.1. Cơ sở lý thuyết 14](#_Toc196380581)

[2.1.1. Ngôn ngữ lập trình Java 14](#_Toc196380582)

[2.1.2. Thư viện Boostrap 16](#_Toc196380583)

[2.1.3. Framework Jquery 18](#_Toc196380584)

[2.1.4. MySQL 19](#_Toc196380585)

[2.1.5. IntelliJ IDEA 20](#_Toc196380586)

[CHƯƠNG III: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 23](#_Toc196380587)

[3.1. Khảo sát hệ thống và đặc tả nghiệp vụ 23](#_Toc196380588)

[3.1.1. Khảo sát thực tế 23](#_Toc196380589)

[3.1.2. Mô tả quy trình nghiệp vụ 25](#_Toc196380590)

[3.1.3. Kết luận về khảo sát 26](#_Toc196380591)

[3.1.4. Yêu cầu xây dựng hệ thống 27](#_Toc196380592)

[3.2. Phân tích hệ thống 29](#_Toc196380593)

[3.2.1. Phân tích và thiết kế hệ thống thông tin theo phương pháp hướng đối tượng với UML................................................................................................................................29](#_Toc196380594)

[3.2.2. Biểu đồ phân rã chức năng 35](#_Toc196380595)

[3.2.3. Các tác nhân 36](#_Toc196380596)

[3.2.4. Biều đồ use-case và đặc tả các use-case 36](#_Toc196380597)

[3.2.5. Biểu đồ tuần tự 46](#_Toc196380598)

[3.2.6. Biểu đồ trạng thái 53](#_Toc196380599)

[3.3. Thiết kế hệ thống 54](#_Toc196380600)

[3.3.1. Thiết kế lớp 54](#_Toc196380601)

[3.3.2. Thiết kế Database 59](#_Toc196380602)

[3.3.3. Thiết kế kiến trúc hệ thống 64](#_Toc196380603)

[CHƯƠNG IV: XÂY DỰNG HỆ THỐNG 65](#_Toc196380604)

[4.1. Lập trình xây dựng hệ thống 65](#_Toc196380605)

[4.1.1. Cấu trúc mã nguồn 65](#_Toc196380606)

[4.1.2. Hình ảnh giao diện phần mềm thực tế xây dựng 66](#_Toc196380607)

[4.2. Khởi tạo dữ liệu ban đầu cho hệ thống 74](#_Toc196380608)

[4.3. Kiểm thử và đánh giá chương trình 76](#_Toc196380609)

[KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 80](#_Toc196380610)

[ Kết quả đạt được 80](#_Toc196380611)

[ Hạn chế 80](#_Toc196380612)

[ Hướng phát triển 81](#_Toc196380613)

[TÀI LIỆU THAM KHẢO 82](#_Toc196380614)

# DANH MỤC CÁC TỪ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| **STT** | **Từ viết tắt** | **Giải thích** |
| 1 | UML | Unified Modelling Language |
| 2 | IoT | Internet of Thing |
| 3 | UI | User Interface |

# DANH MỤC BẢNG BIỂU VÀ SƠ ĐỒ

[Bảng 3.1: Mô tả chức năng nhiệm vụ của các thành viên trong cửa hàng 32](#_Toc196339682)

[Bảng 3.2: Mô tả các tác nhân trong hệ thống 37](#_Toc196339683)

[Bảng 3.3: Đặc tả use-case đăng nhập 42](#_Toc196339684)

[Bảng 3.4: Đặc tả use-case quản lý bán hàng 42](#_Toc196339685)

[Bảng 3.6: Đặc tả use-case danh mục 43](#_Toc196339686)

[Bảng 3.7: Đặc tả use-case quản lý nhà cung cấp 44](#_Toc196339687)

[Bảng 3.8: Đặc tả use-case quản lý hóa đơn 44](#_Toc196339688)

[Bảng 3.10: Đặc tả use-case thống kê 45](#_Toc196339689)

[Bảng 3.11: Danh sách thực thể trong hệ thống 55](#_Toc196339690)

[Bảng 3.12: Mô tả chi tiết lớp Category 56](#_Toc196339691)

[Bảng 3.13: Mô tả chi tiết lớp Customer 56](#_Toc196339692)

[Bảng 3.14: Mô tả chi tiết lớp Employee 56](#_Toc196339693)

[Bảng 3.15: Mô tả chi tiết lớp Order 57](#_Toc196339694)

[Bảng 3.16: Mô tả chi tiết lớp OrderDetail 58](#_Toc196339695)

[Bảng 3.17: Mô tả chi tiết lớp Product 58](#_Toc196339696)

[Bảng 3.18: Mô tả chi tiết lớp Promotion 59](#_Toc196339697)

[Bảng 3.19: Mô tả chi tiết lớp Supplier 59](#_Toc196339698)

[Bảng 3.20: Mô tả chi tiết lớp TempOrder 59](#_Toc196339699)

[Bảng 3.21: Mô tả chi tiết lớp TempOrderDetail 60](#_Toc196339700)

[Bảng 3.22: Mô tả chi tiết bảng Category 60](#_Toc196339701)

[Bảng 3.23: Mô tả chi tiết bảng Customer 60](#_Toc196339702)

[Bảng 3.24: Mô tả chi tiết bảng Employee 61](#_Toc196339703)

[Bảng 3.25: Mô tả chi tiết bảng Product 61](#_Toc196339704)

[Bảng 3.26: Mô tả chi tiết bảng Supplier 62](#_Toc196339705)

[Bảng 3.27: Mô tả chi tiết bảng Orders 62](#_Toc196339706)

[Bảng 3.28: Mô tả chi tiết bảng Order\_detail 63](#_Toc196339707)

[Bảng 3.29: Mô tả chi tiết bảng Temp\_orders 63](#_Toc196339708)

[Bảng 3.30: Mô tả chi tiết bảng Temp\_order\_details 64](#_Toc196339709)

[Bảng 4.1: Danh sách test case kiểm thử đánh giá hệ thống 79](#_Toc196339710)

# DANH MỤC HÌNH ẢNH

[Hình 2.1: Logo ngôn ngữ Java 14](#_Toc199080274)

[Hình 2.2: Logo thư viện Boostrap 16](#_Toc199080275)

[Hình 2.3: Logo framework Jquery 18](#_Toc199080276)

[Hình 2.4: Logo hệ quản trị cơ sở dữ liệu MySQL 19](#_Toc199080277)

[Hình 2.5: Logo phần mềm IntelliJ IDEA 20](#_Toc199080278)

[Hình 3.1: Quy trình phát triển phần mềm với UML 30](#_Toc199080279)

[Hình 3.2: Các góc nhìn của UML 30](#_Toc199080280)

[Hình 3.3: Biểu đồ phân rã chức năng 36](#_Toc199080281)

[Hình 3.4: Biểu đồ use-case tổng quát 37](#_Toc199080282)

[Hình 3.5: Biểu đồ use-case chức năng đăng nhập 38](#_Toc199080283)

[Hình 3.6: Biểu đồ use-case chức năng bán hàng 38](#_Toc199080284)

[Hình 3.7: Biểu đồ use-case chức năng quản lý sản phẩm 39](#_Toc199080285)

[Hình 3.8: Biểu đồ use-case chức năng quản lý danh mục 39](#_Toc199080286)

[Hình 3.9: Biểu đồ use-case chức năng quản lý nhà cung cấp 40](#_Toc199080287)

[Hình 3.10: Biểu đồ use-case chức năng quản lý hóa đơn 40](#_Toc199080288)

[Hình 3.11: Biểu đồ use-case chức năng quản lý khách hàng 40](#_Toc199080289)

[Hình 3.12: Biểu đồ use-case chức năng quản lý nhân viên 41](#_Toc199080290)

[Hình 3.13: Biểu đồ use-case chức năng thống kê 41](#_Toc199080291)

[Hình 3.14: Biểu đồ tuần tự chức năng đăng nhập 47](#_Toc199080292)

[Hình 3.15: Biểu đồ tuần tự chức năng bán hàng 47](#_Toc199080293)

[Hình 3.16: Biểu đồ tuần tự chức năng quản lý sản phẩm 48](#_Toc199080294)

[Hình 3.17: Biểu đồ tuần tự chức năng quản lý danh mục 49](#_Toc199080295)

[Hình 3.18: Biểu đồ tuần tự chức năng quản lý nhà cung cấp 50](#_Toc199080296)

[Hình 3.19: Biểu đồ tuần tự chức năng quản lý khách hàng 51](#_Toc199080297)

[Hình 3.20: Biểu đồ tuần tự chức năng quản lý hóa đơn 52](#_Toc199080298)

[Hình 3.21: Biểu đồ tuần tự chức năng quản lý nhân viên 53](#_Toc199080299)

[Hình 3.22: Biểu đồ tuần tự chức năng thống kê 54](#_Toc199080300)

[Hình 3.23: Biểu đồ trạng thái của đơn hàng 54](#_Toc199080301)

[Hình 3.24: Biểu đồ trạng thái của khuyến mãi 54](#_Toc199080302)

[Hình 3.25: Biểu đồ trạng thái nhân viên 55](#_Toc199080303)

[Hình 3.26: Biểu đồ ERD 65](#_Toc199080304)

[Hình 3.27: Biểu đồ triển khai 65](#_Toc199080305)

[Hình 4.1: Folder tổng quát project 66](#_Toc199080306)

[Hình 4.2: Giao diện màn hình đăng nhập thực tế 67](#_Toc199080307)

[Hình 4.3: Giao diện màn hình bán hàng 67](#_Toc199080308)

[Hình 4.4: Giao diện màn hình quản lý sản phẩm 68](#_Toc199080309)

[Hình 4.5: Giao diện màn hình sửa sản phẩm 68](#_Toc199080310)

[Hình 4.6: Giao diện màn hình quản lý danh mục 69](#_Toc199080311)

[Hình 4.7: Giao diện màn hình sửa danh mục 69](#_Toc199080312)

[Hình 4.8: Giao diện màn hình quản lý nhà cung cấp 70](#_Toc199080313)

[Hình 4.9: Giao diện màn hình sửa nhà cung cấp 70](#_Toc199080314)

[Hình 4.10: Giao diện màn hình quản lý hóa đơn 71](#_Toc199080315)

[Hình 4.11: Giao diện màn hình thống kê 71](#_Toc199080316)

[Hình 4.12: Giao diện màn hình danh sách nhân viên 72](#_Toc199080317)

[Hình 4.13: Giao diện màn hình danh sách khách hàng 72](#_Toc199080318)

[Hình 4.14: Giao diện màn hình phương thức thanh toán 73](#_Toc199080319)

[Hình 4.15: Giao diện màn hình thanh toán chuyển khoản 73](#_Toc199080320)

[Hình 4.16: Giao diện màn hình hóa đơn 74](#_Toc199080321)

[Hình 4.17: Thêm sản phẩm mới trong hệ thống 75](#_Toc199080322)

[Hình 4.18: Thêm danh mục trong hệ thống 75](#_Toc199080323)

[Hình 4.19: Thêm nhà cung cấp mới trong hệ thống 76](#_Toc199080324)

[Hình 4.20: Thêm nhân viên mới 76](#_Toc199080325)

[Hình 4.21: Thêm số điện thoại khách hàng mới 77](#_Toc199080326)

# CHƯƠNG I: GIỚI THIỆU TỔNG QUAN ĐỀ TÀI

## Lý do chọn đề tài

* + - Trong thời đại công nghệ 4.0 hiện nay, dữ liệu đã và đang trở thành một trong những tài nguyên quan trọng nhất đối với các cá nhân, doanh nghiệp và tổ chức. Đặc biệt trong lĩnh vực kinh doanh, việc quản lý thông tin bán hàng, sản phẩm, khách hàng và hóa đơn một cách hiệu quả đóng vai trò then chốt trong việc nâng cao hiệu suất hoạt động và đưa ra quyết định kinh doanh chính xác. Tuy nhiên, việc quản lý thủ công không chỉ mất nhiều thời gian mà còn dễ xảy ra sai sót, thất thoát dữ liệu và gây khó khăn trong việc tổng hợp, thống kê.
    - Trước thực trạng đó, việc xây dựng một **hệ thống quản lý bán hàng** ứng dụng công nghệ hiện đại là vô cùng cần thiết. Đề tài **“Xây dựng hệ thống quản lý bán hàng sử dụng công nghệ Java Spring Boot và MySQL”** được thực hiện nhằm đưa ra một giải pháp phần mềm hỗ trợ doanh nghiệp, cửa hàng trong việc quản lý quy trình bán hàng một cách khoa học, chính xác và tự động hóa. Hệ thống giúp theo dõi đơn hàng, thanh toán, lưu trữ hóa đơn, quản lý sản phẩm và khách hàng một cách dễ dàng, đồng thời có thể mở rộng thêm các tính năng như thống kê, khuyến mãi, tích hợp thanh toán online,...
    - Ứng dụng web này không chỉ giúp nhân viên bán hàng và quản lý vận hành công việc thuận tiện hơn mà còn hỗ trợ chủ cửa hàng nắm bắt tình hình kinh doanh mọi lúc, mọi nơi. Với giao diện thân thiện, dễ sử dụng cùng với độ bảo mật cao, hệ thống hứa hẹn sẽ trở thành một công cụ hỗ trợ đắc lực trong quá trình chuyển đổi số của các doanh nghiệp vừa và nhỏ.
    - Trong các phần tiếp theo của báo cáo, em sẽ trình bày chi tiết về phân tích yêu cầu, thiết kế hệ thống, triển khai chức năng, cũng như đánh giá hiệu quả ứng dụng sau khi triển khai. Hy vọng giải pháp này sẽ mang lại giá trị thiết thực cho các đơn vị kinh doanh và góp phần thúc đẩy quá trình hiện đại hóa công tác quản lý bán hàng.

## Mục tiêu của đề tài

* + - Đề tài tập trung vào việc **xây dựng một hệ thống quản lý bán hàng hoàn chỉnh**, đáp ứng các nhu cầu thực tế như: quản lý sản phẩm, đơn hàng, khách hàng, nhân viên, hóa đơn, thống kê doanh thu theo ngày/tuần/tháng, áp dụng khuyến mãi linh hoạt,... Hệ thống mang tính chuyên nghiệp, giúp tiết kiệm thời gian, đảm bảo tính chính xác cao và đơn giản hóa thao tác cho người sử dụng trong môi trường doanh nghiệp hoặc cửa hàng.
    - Trong quá trình xây dựng hệ thống, đề tài đã **áp dụng phương pháp phân tích và thiết kế hệ thống thông tin** một cách khoa học, bài bản để đảm bảo đáp ứng yêu cầu nghiệp vụ và có thể dễ dàng bảo trì, nâng cấp trong tương lai.
    - Về mặt công nghệ, đề tài sử dụng **ngôn ngữ lập trình Java kết hợp với framework Spring Boot** để xây dựng backend, **MySQL** để lưu trữ dữ liệu, **Thymeleaf** để tạo giao diện web động cùng với HTML, CSS, Bootstrap và JavaScript giúp giao diện trực quan, thân thiện với người dùng. Toàn bộ hệ thống được triển khai theo kiến trúc MVC, đảm bảo tính tách biệt giữa các tầng xử lý, dễ mở rộng và bảo trì. Kết quả đạt được cho thấy hệ thống hoạt động ổn định, đáp ứng tốt yêu cầu thực tế và hỗ trợ hiệu quả cho quá trình quản lý bán hàng.

## Giới hạn và phạm vi của đề tài

* Trong phạm vi của đề tài này, em sẽ tập trung nghiên cứu và triển khai các nội dung chính sau:

+ Tìm hiểu, khảo sát thực tế nhu cầu quản lý bán hàng tại các cửa hàng nhỏ và vừa, bao gồm các nghiệp vụ như bán hàng, quản lý sản phẩm, hóa đơn, khách hàng, nhân viên, và nhu cầu thống kê doanh thu, áp dụng khuyến mãi.

+ Phân tích và thiết kế chi tiết hệ thống quản lý bán hàng, xây dựng các sơ đồ chức năng, sơ đồ luồng dữ liệu, sơ đồ ERD, từ đó thiết kế cơ sở dữ liệu và kiến trúc hệ thống phù hợp với yêu cầu thực tiễn.

+ Nghiên cứu và áp dụng các công nghệ hiện đại như ngôn ngữ lập trình Java, framework Spring Boot, kết hợp với HTML, CSS, Bootstrap, JavaScript (JQuery) ở phía giao diện, và MySQL để quản lý cơ sở dữ liệu. Toàn bộ hệ thống sẽ được xây dựng theo mô hình kiến trúc MVC nhằm đảm bảo khả năng mở rộng, dễ bảo trì và vận hành hiệu quả.

## Phương pháp thực hiện

* + - **Nghiên cứu yêu cầu:** Tìm hiểu và thu thập các yêu cầu chức năng và phi chức năng của hệ thống quản lý bán hàng, bao gồm quản lý sản phẩm, hóa đơn, khách hàng, nhân viên, thống kê doanh thu,... Việc xác định đúng yêu cầu giúp đảm bảo hệ thống được xây dựng đúng mục tiêu và đáp ứng nhu cầu thực tế.
    - **Thiết kế hệ thống:** Dựa trên các yêu cầu đã thu thập, tiến hành thiết kế kiến trúc hệ thống, thiết kế cơ sở dữ liệu (ERD), thiết kế giao diện người dùng (UI/UX) và các sơ đồ luồng xử lý (DFD, Use Case, Sequence diagram, v.v).
    - **Phát triển phần mềm:** Sử dụng các công nghệ như **Java, Spring Boot, Thymeleaf, HTML, CSS, Bootstrap, JavaScript (JQuery)** và **MySQL** để xây dựng hệ thống theo mô hình phân lớp MVC, đảm bảo tính linh hoạt, dễ mở rộng và bảo trì.
    - **Thử nghiệm và kiểm thử:** Tiến hành kiểm thử các chức năng của hệ thống như thêm, sửa, xóa, tìm kiếm sản phẩm, lập hóa đơn, thống kê doanh thu... Đồng thời sửa lỗi phát sinh để đảm bảo hệ thống hoạt động ổn định và đúng yêu cầu đã đặt ra.
    - **Triển khai và hỗ trợ:** Triển khai hệ thống vào môi trường thực tế (có thể là tại một cửa hàng giả lập để thử nghiệm). Đồng thời cung cấp tài liệu hướng dẫn sử dụng và hỗ trợ người dùng trong quá trình vận hành hệ thống.

## Kết quả dự kiến đạt được

Hệ thống quản lý bán hàng được xây dựng nhằm đáp ứng nhu cầu thực tế tại các cửa hàng, với các chức năng chính bao gồm:

* + - Quản lý sản phẩm:

+ Thêm, sửa, xóa, tìm kiếm sản phẩm

+ Quản lý danh mục sản phẩm, đơn vị tính, số lượng tồn kho

* + - Quản lý đơn hàng:

+ Tạo đơn hàng mới (bao gồm sản phẩm, số lượng, khách hàng)

+ Sửa, hủy đơn hàng khi cần thiết

+ Tìm kiếm, lọc đơn hàng theo ngày, trạng thái thanh toán

* + - Quản lý khách hàng:

+ Thêm, sửa, xóa, tìm kiếm khách hàng

+ Phân loại khách hàng theo tần suất mua, lịch sử giao dịch

+ Ghi nhận số điện thoại để áp dụng chương trình khuyến mãi

* + - Quản lý nhân viên:

+ Thêm, sửa, xóa thông tin nhân viên

+ Gán quyền truy cập hệ thống (quản lý, thu ngân,...)

- Quản lý hóa đơn và thanh toán:

+ Tạo hóa đơn khi hoàn tất đơn hàng

+ Hỗ trợ nhiều phương thức thanh toán (tiền mặt, chuyển khoản VietQR)

+ Lưu trữ lịch sử hóa đơn, xem và in lại hóa đơn bất kỳ lúc nào

* + - Thống kê, báo cáo:

+ Thống kê tổng số đơn hàng, tổng doanh thu

+ Thống kê theo ngày, tuần, tháng, năm

+ Doanh thu theo nhân viên, theo sản phẩm, theo phương thức thanh toán

+ Biểu đồ doanh thu trực quan

* + - Quản lý khuyến mãi:

+ Tạo và áp dụng các chương trình giảm giá theo phần trăm trên tổng đơn hàng

+ Áp dụng tự động nếu khách hàng có thông tin trong hệ thống

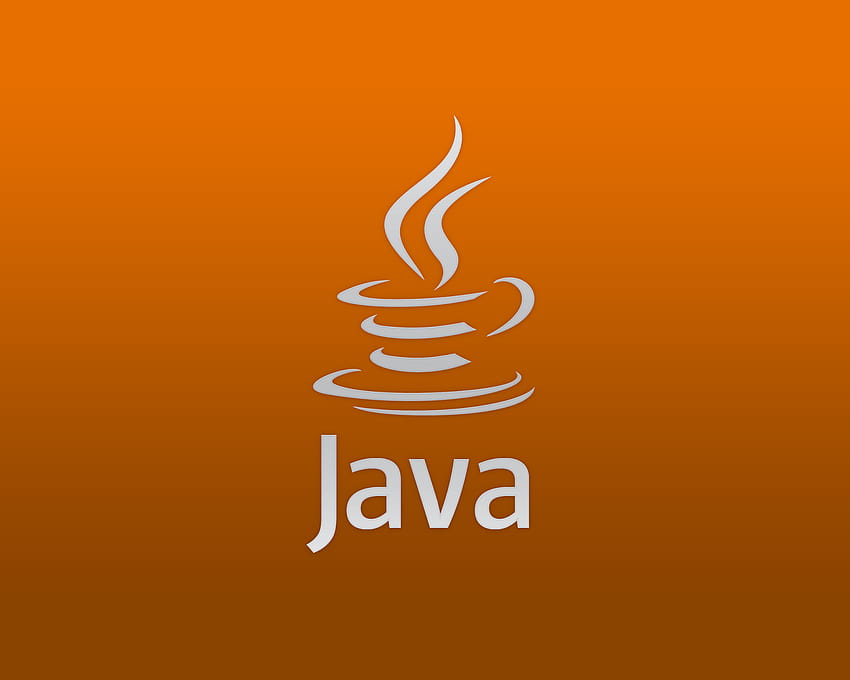
* + - Export dữ liệu:

+ Cho phép xuất danh sách sản phẩm, hóa đơn hoặc doanh thu ra file Excel để sao lưu hoặc phân tích.

# CHƯƠNG II: KIẾN THỨC NỀN TẢNG

## Cơ sở lý thuyết

### Ngôn ngữ lập trình Java



Hình 2.1: Logo ngôn ngữ Java

* **Java** là một ngôn ngữ lập trình hướng đối tượng, đa mục đích, được phát triển bởi Sun Microsystems và ra mắt lần đầu tiên vào năm 1995. Sau này, Java được Oracle tiếp quản và phát triển. Với khẩu hiệu “Write once, run anywhere” (Viết một lần, chạy mọi nơi), Java cho phép chạy trên nhiều nền tảng khác nhau nhờ vào máy ảo Java (JVM).
* Ưu điểm:

+ Tính đa nền tảng: Java có thể chạy trên bất kỳ hệ điều hành nào có cài đặt JVM, bao gồm Windows, macOS, Linux và các hệ điều hành di động. Điều này giúp Java trở thành lựa chọn hàng đầu cho các ứng dụng doanh nghiệp và hệ thống phân tán.

+ Lập trình hướng đối tượng: Java hỗ trợ đầy đủ các khái niệm của lập trình hướng đối tượng như kế thừa, đóng gói, đa hình,... giúp việc xây dựng và bảo trì phần mềm trở nên dễ dàng và hiệu quả hơn.

+ Thư viện phong phú: Java có một hệ sinh thái lớn với rất nhiều thư viện và framework mạnh mẽ như Spring, Hibernate, JavaFX, giúp lập trình viên dễ dàng phát triển từ ứng dụng web, desktop đến ứng dụng di động và IoT.

+ Bảo mật cao: Java được thiết kế với các cơ chế bảo mật tích hợp, giúp hạn chế lỗi và nguy cơ bị tấn công. Đây cũng là lý do Java thường được dùng trong các hệ thống tài chính, ngân hàng.

+ **Cộng đồng lớn và tài liệu phong phú:** Java là một trong những ngôn ngữ phổ biến nhất thế giới, có cộng đồng lập trình viên đông đảo, nhiều diễn đàn, khóa học và tài liệu miễn phí hỗ trợ học tập và giải quyết vấn đề.

- Nhược điểm:

+ **Hiệu năng thấp hơn so với ngôn ngữ biên dịch trực tiếp:** Java chạy trên JVM nên thường chậm hơn so với các ngôn ngữ như C/C++ trong các ứng dụng yêu cầu hiệu suất cao.

+ Giao diện người dùng không hiện đại: Các thư viện GUI như Swing hay AWT của Java có giao diện kém hấp dẫn so với các công nghệ hiện đại.

+ Yêu cầu bộ nhớ cao: Ứng dụng Java thường tiêu tốn nhiều bộ nhớ hơn do cơ chế quản lý bộ nhớ tự động (garbage collection), điều này có thể gây ảnh hưởng đến hiệu năng trong các hệ thống hạn chế tài nguyên.

+ Cú pháp có thể hơi dài dòng: Mặc dù rõ ràng và chuẩn hóa, nhưng cú pháp Java có thể dài và phức tạp hơn so với một số ngôn ngữ hiện đại khác như Python hoặc Kotlin.

\* **Giới thiệu framework Spring Boot**

- Hoàn cảnh ra đời của Spring Boot:

+ **Spring Boot** là một framework mã nguồn mở thuộc hệ sinh thái Spring, được phát triển bởi Pivotal (nay thuộc VMware). Spring Boot giúp đơn giản hóa quá trình xây dựng các ứng dụng Java độc lập, có thể chạy ngay (standalone), thường dùng cho ứng dụng web, hệ thống backend và dịch vụ vi mô (microservices).

+ Spring Framework ban đầu ra đời vào năm 2003 để giải quyết sự phức tạp của Java EE. Tuy nhiên, việc cấu hình Spring truyền thống khá phức tạp và đòi hỏi nhiều file XML.

+ Spring Boot được giới thiệu vào năm 2014 như một giải pháp đơn giản hóa việc khởi tạo, cấu hình và triển khai ứng dụng Spring. Mục tiêu chính của nó là giúp lập trình viên có thể "just run" một ứng dụng mà không cần lo quá nhiều đến cấu hình phức tạp.

+ Spring Boot loại bỏ nhu cầu cấu hình boilerplate, đồng thời tích hợp sẵn các thành phần phổ biến như web server, database connector, security, v.v.

- Những nền tảng cải tiến với Spring Boot:

+ **Tự động cấu hình (Auto Configuration):** Giảm đáng kể lượng cấu hình cần thiết, giúp khởi tạo ứng dụng nhanh chóng.

+ **Tích hợp sẵn Tomcat/Jetty/Undertow:** Ứng dụng có thể chạy như một file JAR độc lập mà không cần deploy vào web server.

+ Hỗ trợ phát triển Microservices: Kết hợp mạnh mẽ với Spring Cloud để xây dựng hệ thống phân tán, hỗ trợ load balancing, service discovery, API gateway,...

+ Tích hợp RESTful API dễ dàng: Cung cấp các annotation như @RestController, @RequestMapping,... giúp xây dựng API nhanh chóng.

+ Hệ thống cấu hình linh hoạt: Hỗ trợ cấu hình theo môi trường (application.yml/application.properties), dễ dàng triển khai cloud-native apps.

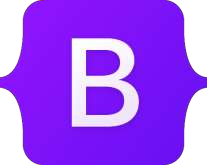
+ Hỗ trợ Dependency Injection mạnh mẽ: Kế thừa toàn bộ sức mạnh của Spring DI container.

+ Cộng đồng lớn và tài liệu đầy đủ: Là một trong những framework phổ biến nhất trong thế giới Java, có rất nhiều tài nguyên học tập và hỗ trợ.

+ Đa nền tảng: Ứng dụng Spring Boot có thể chạy trên Windows, macOS và Linux mà không cần thay đổi mã nguồn.

+ Mã nguồn mở và phát triển bởi cộng đồng: Spring Boot được duy trì bởi cộng đồng lớn và thường xuyên cập nhật tính năng mới.

### Thư viện Boostrap

******

Hình 2.2: Logo thư viện Boostrap

* Bootstrap là một front-end framework mã nguồn mở miễn phí giúp quá trình phát triển web được nhanh và dễ dàng hơn. Nó cung cấp một bộ công cụ và các lớp CSS có sẵn để tạo giao diện người dùng, bao gồm hệ thống lưới, nút, biểu mẫu và nhiều thành phần khác.
* Ưu điểm của Bootstrap:

+ Tiết kiệm thời gian và công sức: Cung cấp một bộ công cụ và các lớp CSS có sẵn, giúp bạn nhanh chóng tạo giao diện người dùng mà không cần viết CSS từ đầu. Bạn có thể sử dụng các thành phần và kiểu dáng đã được chuẩn hóa sẵn, giảm bớt công việc lặp lại và tăng tốc quá trình phát triển.

+ Giao diện đáp ứng (responsive design): Hỗ trợ xây dựng trang web có giao diện đáp ứng, tức là giao diện sẽ tự động thích ứng và hiển thị tốt trên các thiết bị và kích thước màn hình khác nhau. Điều này giúp đảm bảo trải nghiệm người dùng tốt trên di động, máy tính bảng và máy tính để bàn mà không cần phải viết mã CSS phức tạp.

+ Hệ thống lưới linh hoạt: Hệ thống lưới (grid system) mạnh mẽ, giúp bạn xây dựng cấu trúc trang linh hoạt và tương thích với các kích thước màn hình khác nhau. Bạn có thể dễ dàng điều chỉnh vị trí và kích thước của các phần tử trên trang web.

+ Tính nhất quán và tương thích trên nhiều trình duyệt: Bootstrap được kiểm thử và tối ưu để hoạt động tốt trên nhiều trình duyệt phổ biến như Chrome, Firefox, Safari

và Internet Explorer. Điều này giúp đảm bảo rằng trang web của bạn sẽ có trải nghiệm đồng nhất trên các nền tảng khác nhau.

+ Cộng đồng phát triển mạnh mẽ: Bootstrap là một framework phổ biến với cộng đồng lớn. Bạn có thể tìm thấy nhiều tài liệu, ví dụ và giải pháp sẵn có từ cộng đồng, giúp bạn giải quyết các vấn đề và tận dụng tối đa khả năng của Bootstrap.

### Framework Jquery

******

Hình 2.3: Logo framework Jquery

* Jquery là một thư viện JavaScript rất phổ biến được sử dụng để tạo ra các hiệu ứng động, thay đổi nội dung của trang web, tương tác với người dùng và thao tác với tài liệu HTML và CSS một cách dễ dàng và tiện lợi. jQuery cung cấp các phương thức để thao tác với HTML DOM, xử lý sự kiện, tạo ra hiệu ứng động, gọi Ajax để gửi và nhận dữ liệu từ máy chủ, thao tác với CSS và nhiều tính năng khác.
* Tính năng nổi bật:

+ Thao tác với HTML DOM: jQuery cho phép bạn thao tác với HTML DOM một cách dễ dàng và tiện lợi hơn bằng cách sử dụng các phương thức như

$(selector).method() để chọn các phần tử trên trang và thay đổi nội dung của chúng, thêm hoặc xóa các phần tử, thay đổi thuộc tính và lớp CSS.

+ Xử lý sự kiện: jQuery cung cấp các phương thức để xử lý sự kiện như click, hover, submit, change, và keypress. Bạn có thể sử dụng các phương thức này để xử lý các sự kiện của người dùng một cách dễ dàng.

+ Hiệu ứng động: jQuery cung cấp các phương thức để tạo ra các hiệu ứng động trên trang web như fadeIn, fadeOut, slideUp, slideDown, animate và toggle. Bạn có thể sử dụng các phương thức này để thêm tính năng động cho trang web của mình.

+ Ajax: jQuery cho phép bạn gửi và nhận dữ liệu từ máy chủ bằng cách sử dụng phương thức Ajax. Bạn có thể sử dụng các phương thức như $.ajax() và $.getJSON() để gửi và nhận dữ liệu từ máy chủ một cách nhanh chóng và tiện lợi.

+ Thao tác với CSS: jQuery cho phép bạn thao tác với CSS trên trang web của mình bằng cách sử dụng các phương thức như addClass, removeClass, và css. Bạn có thể sử dụng các phương thức này để thêm hoặc xóa lớp CSS hoặc thay đổi thuộc tính CSS của các phần tử trên trang.

+ Hỗ trợ đa trình duyệt: jQuery được thiết kế để hoạt động trên tất cả các trình duyệt phổ biến, bao gồm cả Internet Explorer, Firefox, Chrome, Safari và Opera.

### MySQL



Hình 2.4: Logo hệ quản trị cơ sở dữ liệu MySQL

* **MySQL** là một hệ quản trị cơ sở dữ liệu quan hệ (Relational Database Management System - RDBMS) mã nguồn mở, được phát triển ban đầu bởi công ty MySQL AB vào năm 1995 và hiện thuộc sở hữu của Oracle Corporation. MySQL được sử dụng phổ biến trong các hệ thống web, đặc biệt là kết hợp với PHP trong mô hình LAMP (Linux, Apache, MySQL, PHP).
* Các thành phần chính của MySQL:

+ **MySQL Server:** Là thành phần cốt lõi, xử lý truy vấn SQL, quản lý cơ sở dữ liệu, lưu trữ dữ liệu và xử lý các giao dịch.

+ **MySQL Workbench:** Là công cụ đồ họa dùng để thiết kế, quản lý và truy vấn cơ sở dữ liệu MySQL một cách trực quan. Nó tích hợp nhiều chức năng như thiết kế schema, migration, quản lý người dùng, và hiệu suất.

+ **MySQL Shell:** Là một công cụ dòng lệnh nâng cao, hỗ trợ truy vấn tương tác bằng SQL, JavaScript và Python, phù hợp cho cả quản trị viên và lập trình viên.

+ **MySQL Connectors:** Là các thư viện giúp các ứng dụng kết nối và làm việc với MySQL thông qua các ngôn ngữ lập trình như Java, Python, C++, .NET,...

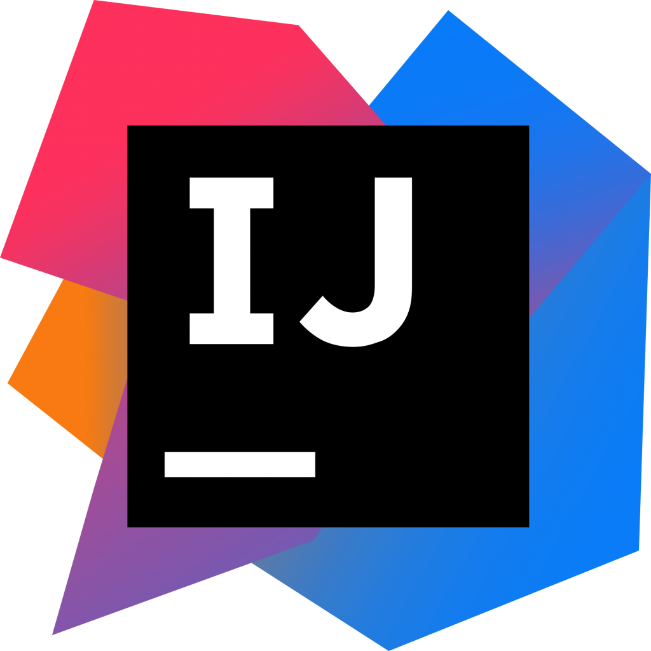
+ **InnoDB Storage Engine:** Là bộ máy lưu trữ mặc định trong MySQL, hỗ trợ đầy đủ các tính năng của RDBMS như khóa ngoại, giao dịch ACID và tính toàn vẹn dữ liệu.

+ **MySQL Router:** Là thành phần trung gian hỗ trợ định tuyến yêu cầu đến các máy chủ MySQL trong các hệ thống phân tán hoặc cluster.

+ **MySQL Enterprise Edition:** Là phiên bản thương mại của MySQL với các tính năng nâng cao như bảo mật, giám sát, sao lưu, và công cụ hiệu suất.Công cụ sử dụng.

+ **MySQL HeatWave (trên Oracle Cloud):** Là dịch vụ cloud tối ưu hóa cho phân tích dữ liệu theo thời gian thực, kết hợp xử lý OLTP và OLAP trên một nền tảng duy nhất.

### 2.1.5. IntelliJ IDEA



Hình 2.5: Logo phần mềm IntelliJ IDEA

* **IntelliJ IDEA** là một môi trường phát triển tích hợp (IDE) mạnh mẽ và chuyên nghiệp, được phát triển bởi JetBrains. Đây là một trong những IDE hàng đầu dành cho ngôn ngữ lập trình Java, đồng thời cũng hỗ trợ nhiều ngôn ngữ khác như Kotlin, Groovy, Scala, JavaScript, TypeScript, SQL,... IntelliJ IDEA nổi bật với giao diện trực quan, hiệu suất cao và hàng loạt tính năng thông minh giúp tăng năng suất lập trình.
* Các tính năng nổi bật của IntelliJ IDEA:

+ **Hỗ trợ ngôn ngữ lập trình đa dạng:** IntelliJ IDEA không chỉ hỗ trợ tuyệt vời cho Java mà còn tương thích với nhiều ngôn ngữ khác như Kotlin, Groovy, Scala, Python, JavaScript, TypeScript, SQL... Đây là lựa chọn lý tưởng cho các lập trình viên phát triển đa nền tảng, đa dự án với nhiều công nghệ khác nhau trong một môi trường làm việc duy nhất.

+ **Phân tích mã nguồn thông minh:** IntelliJ IDEA cung cấp khả năng phân tích mã theo ngữ cảnh cực kỳ mạnh mẽ, giúp phát hiện lỗi tiềm ẩn, gợi ý cải tiến mã (code suggestions), và tái cấu trúc mã (refactor) một cách thông minh và chính xác. Điều này giúp giảm thiểu lỗi, cải thiện chất lượng và khả năng bảo trì mã nguồn.

+ **Tự động hoàn thành mã nâng cao:** Với công nghệ IntelliSense tiên tiến, IntelliJ IDEA hỗ trợ tính năng tự động gợi ý, hoàn thành mã theo ngữ cảnh cực kỳ hiệu quả. Từ tên biến, tên class, phương thức đến cả các đoạn mã phức tạp, IDE này đều đưa ra gợi ý chính xác, giúp tăng tốc quá trình phát triển và giảm thiểu lỗi đánh máy..

+ **Tích hợp hệ thống kiểm soát phiên bản:** IntelliJ IDEA tích hợp sẵn các hệ thống quản lý mã nguồn như Git, GitHub, Mercurial và SVN. Người dùng có thể thực hiện các thao tác như commit, push, pull, merge và kiểm tra lịch sử thay đổi ngay trong giao diện IDE một cách dễ dàng, trực quan.

+ **Tích hợp công cụ xây dựng và quản lý phụ thuộc:** IntelliJ IDEA hỗ trợ các công cụ như Maven, Gradle để quản lý thư viện, phụ thuộc và quy trình build. Điều này giúp đơn giản hóa việc cấu hình dự án và tự động hóa quy trình phát triển phần mềm.

+ **Gỡ lỗi mạnh mẽ và trực quan:** IntelliJ IDEA đi kèm với hệ thống gỡ lỗi tiên tiến, cho phép đặt breakpoint, quan sát giá trị biến, theo dõi luồng thực thi và biểu diễn trực quan quá trình debug, rất hiệu quả trong việc tìm và sửa lỗi.

+ **Hỗ trợ phát triển Spring Boot tuyệt vời:** Là IDE được yêu thích trong cộng đồng Java, IntelliJ IDEA hỗ trợ sâu rộng cho framework Spring, đặc biệt là Spring Boot. Nó cung cấp auto-completion, cấu hình tự động, trình duyệt cấu trúc Spring beans và công cụ chạy ứng dụng một cách mượt mà.

+ **Hỗ trợ phát triển frontend:** IntelliJ IDEA Ultimate còn hỗ trợ phát triển giao diện người dùng với HTML, CSS, JavaScript và các framework hiện đại như React, Angular, Vue.j.

+ **Khả năng phát triển đa nền tảng:** Người dùng có thể sử dụng IntelliJ IDEA để phát triển ứng dụng trên nhiều nền tảng khác nhau như Windows, macOS, và Linux. Ngoài ra, nó cũng hỗ trợ phát triển ứng dụng backend, frontend, mobile (qua plugin) và các dịch vụ web hiện đại.

+ **Kho plugin phong phú:** IntelliJ IDEA có hệ sinh thái plugin rộng lớn, từ các công cụ hỗ trợ phát triển như Lombok, Database Tools, Docker, Kubernetes cho đến các công cụ hỗ trợ UI, quản lý task, quản lý tài liệu,... giúp tùy biến IDE phù hợp với từng loại dự án.

# CHƯƠNG III: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

## Khảo sát hệ thống và đặc tả nghiệp vụ

### Khảo sát thực tế

***a, Giới thiệu về bài toán***

* Trong thời đại công nghệ số phát triển mạnh mẽ, việc áp dụng chuyển đổi số vào các hoạt động kinh doanh không còn là một xu hướng mà đã trở thành nhu cầu cấp thiết đối với mọi doanh nghiệp, kể cả những cửa hàng bán lẻ nhỏ lẻ. Tuy nhiên, thực tế cho thấy vẫn còn rất nhiều cửa hàng truyền thống đang gặp không ít khó khăn trong việc quản lý hoạt động kinh doanh do vẫn sử dụng các phương pháp thủ công như ghi chép hóa đơn bằng tay, theo dõi hàng tồn kho trên sổ sách, hoặc sử dụng những bảng tính đơn giản mà thiếu tính hệ thống và dễ xảy ra sai sót. Việc không có một hệ thống phần mềm quản lý đồng bộ khiến cho các công việc như thống kê doanh thu, theo dõi lịch sử bán hàng, phân tích hiệu suất nhân viên hoặc kiểm tra tồn kho trở nên phức tạp, tốn thời gian, và dễ dẫn đến sai lệch dữ liệu.
* Một trong những vấn đề nổi cộm mà nhiều chủ cửa hàng gặp phải đó là việc **mất kiểm soát dữ liệu** do lưu trữ rải rác và không có hệ thống sao lưu hợp lý. Trong nhiều trường hợp, khi cần tra cứu lại thông tin đơn hàng, khách hàng hay lịch sử bán hàng để xử lý khiếu nại hoặc đối soát, thì lại mất rất nhiều thời gian hoặc thậm chí không thể tìm thấy dữ liệu. Điều này không chỉ ảnh hưởng đến uy tín mà còn làm giảm hiệu quả hoạt động kinh doanh. Ngoài ra, việc tính toán doanh thu thủ công thường dẫn đến sai lệch số liệu, ảnh hưởng trực tiếp đến quyết định nhập hàng, lập kế hoạch kinh doanh, cũng như phân tích lợi nhuận – lỗ.
* Chính vì vậy, **nhu cầu xây dựng một hệ thống phần mềm quản lý bán hàng thông minh, hiện đại, dễ sử dụng** là hoàn toàn thực tế và cần thiết. Chủ cửa hàng mong muốn có một hệ thống có thể **thay thế hoàn toàn phương pháp quản lý thủ công,** hỗ trợ nhập liệu nhanh chóng, giảm thiểu sai sót, đồng thời nâng cao trải nghiệm làm việc cho nhân viên bán hàng và quản lý. Hệ thống cần cho phép người dùng dễ dàng thao tác các chức năng như: quản lý sản phẩm, kiểm tra tồn kho, tạo và in hóa đơn, tính tiền, lưu trữ thông tin khách hàng, và đặc biệt là có khả năng tổng hợp báo cáo doanh thu theo ngày, tuần hoặc tháng để người quản lý có cái nhìn tổng quan và ra quyết định chính xác hơn.
* Một yêu cầu quan trọng khác là **hệ thống cần được xây dựng trên nền tảng web,** giúp người dùng có thể truy cập và sử dụng mọi lúc, mọi nơi – chỉ cần có kết nối Internet. Điều này sẽ tạo điều kiện cho các chủ cửa hàng dễ dàng giám sát hoạt động kinh doanh ngay cả khi không có mặt trực tiếp tại cửa hàng. Ngoài ra, hệ thống cũng nên có khả năng **tích hợp với các phương thức thanh toán điện tử hiện đại** như chuyển khoản qua mã VietQR, ví điện tử Momo, ZaloPay… nhằm giúp khách hàng thanh toán dễ dàng hơn, đồng thời giúp chủ cửa hàng theo dõi giao dịch một cách tự động và chính xác hơn.
* Tóm lại, trong bối cảnh chuyển đổi số đang diễn ra mạnh mẽ, việc xây dựng một **hệ thống quản lý bán hàng sử dụng công nghệ hiện đại như Java Spring Boot kết hợp với MySQL** không chỉ là giải pháp cấp thiết để giải quyết các khó khăn hiện tại, mà còn là nền tảng vững chắc để các cửa hàng phát triển bền vững trong tương lai.

**b,** **Bảng nội dung mô tả chức năng nhiệm vụ của các thành viên trong cửa hàng**

Bảng 3.1: Mô tả chức năng nhiệm vụ của các thành viên trong cửa hàng

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên công việc** | **Người thực hiện** | **Hồ sơ dữ liệu** |
| 1 | Cập nhật danh sách sản phẩm | Quản lý | Danh sách sản phẩm |
| 2 | Quản lý thông tin khách hàng | Quản lý | Hồ sơ khách hàng |
| 3 | Thêm / Cập nhật khuyến mãi | Quản lý | Thông tin khuyến mãi |
| 4 | Tạo và xử lý đơn hàng | Quản lý/Nhân viên | Hóa đơn / Đơn hàng |
| 5 | Tính tiền và in hóa đơn | Quản lý/Nhân viên | Hóa đơn bán hàng |
| 6 | Quản lý kho và kiểm tra tồn kho | Quản lý/Nhân viên | Phiếu nhập – xuất kho |
| 7 | Thống kê doanh thu theo ngày / tháng | Quản lý | Báo cáo doanh thu |
| 8 | Theo dõi hiệu suất bán hàng của nhân viên | Quản lý | Báo cáo hiệu suất |
| 9 | Xử lý thanh toán qua mã QR / tiền mặt | Quản lý/Nhân viên | Lịch sử giao dịch |
| 10 | Hỗ trợ, chăm sóc khách hàng trực tiếp | Quản lý/Nhân viên | Phản hồi / Thông tin khách hàng |

### Mô tả quy trình nghiệp vụ

* Khi bắt đầu ngày làm việc, quản lý đăng nhập vào hệ thống để kiểm tra tồn kho, cập nhật giá sản phẩm nếu cần thiết, thêm các chương trình khuyến mãi mới hoặc điều chỉnh chính sách giảm giá cho khách hàng thân thiết.
* Nhân viên bán hàng tiến hành nhập thông tin sản phẩm, kiểm tra mã vạch hoặc chọn từ danh sách sản phẩm có sẵn trên hệ thống khi khách hàng đến mua hàng. Nếu khách có thông tin lưu trong hệ thống, nhân viên sẽ nhập số điện thoại để áp dụng ưu đãi tự động (nếu có).
* Hệ thống tự động tính tổng tiền hóa đơn sau khi nhân viên chọn đầy đủ các mặt hàng và số lượng. Nếu có khuyến mãi, hệ thống sẽ tự động trừ phần trăm giảm giá tương ứng. Nhân viên có thể chọn phương thức thanh toán: tiền mặt, chuyển khoản QR, Momo, ZaloPay…
* Sau khi khách hàng thanh toán, hệ thống sẽ tạo hóa đơn bán hàng và cập nhật kho trừ số lượng tồn tương ứng. Hóa đơn được in và giao cho khách hàng, đồng thời lưu vào hệ thống để phục vụ cho thống kê, báo cáo và tra cứu sau này.

- Nếu khách hàng thanh toán bằng chuyển khoản VietQR, hệ thống sẽ tạo mã QR theo số tiền đơn hàng, sau đó lắng nghe webhook từ ngân hàng để xác nhận giao dịch thành công. Khi xác nhận thành công, hóa đơn sẽ được tự động đánh dấu là “Đã thanh toán” và in ra như thanh toán tiền mặt.

* Quản lý có thể thực hiện sao lưu dữ liệu hệ thống định kỳ, quản lý tài khoản nhân viên, phân quyền truy cập và xử lý các sự cố nếu có. Ngoài ra, hệ thống còn có thể mở rộng tích hợp với máy in hóa đơn, thiết bị quét mã vạch và các cổng thanh toán điện tử khác trong tương lai.
* **Công thức tính tiền giảm giá:**

**Tổng tiền giảm giá** = Tổng tiền hóa đơn × (Phần trăm giảm giá / 100)

Trong đó:   
+ Tổng tiền hóa đơn: Là tổng số tiền trước khi áp dụng giảm giá.

+ Phần trăm giảm giá: Là giá trị phần trăm khuyến mãi được áp dụng (ví dụ: 10%, 15%, 20%,...).

### Kết luận về khảo sát

* Qua quá trình khảo sát, để đáp ứng được nhu cầu thực tế trong việc quản lý bán hàng và các hoạt động liên quan, cần xây dựng một hệ thống quản lý bán hàng hiện đại giúp việc quản lý đơn hàng, tính toán doanh thu, kiểm soát tồn kho, và theo dõi hiệu suất nhân viên trở nên chính xác và nhanh chóng. Việc tin học hóa các công việc quản lý bán hàng giúp giảm thiểu sai sót, tối ưu hóa quy trình và tiết kiệm thời gian, đồng thời nâng cao hiệu quả công việc của nhân viên và quản lý. Hệ thống không chỉ giúp in hóa đơn và báo cáo doanh thu, mà còn cho phép dễ dàng sao lưu và khôi phục dữ liệu khi cần thiết.

-Hệ thống được xây dựng sẽ giúp các nhân viên bán hàng, quản lý cửa hàng có thể tham gia vào quá trình quản lý một cách dễ dàng và hiệu quả. Thông tin về khách hàng, sản phẩm, đơn hàng và doanh thu sẽ được lưu trữ trong hệ thống, giúp việc thống kê, tìm kiếm và cập nhật dữ liệu nhanh chóng. Những thông tin này sẽ giúp quản lý cửa hàng có cái nhìn tổng thể về hoạt động kinh doanh, từ đó đưa ra các quyết định chiến lược phù hợp để nâng cao chất lượng dịch vụ, phát triển cửa hàng và cải thiện hiệu quả kinh doanh.

### Yêu cầu xây dựng hệ thống

Xây dựng hệ thống phần quản lý bán hàng đáp ứng các yêu cầu sau:

* ***Yêu cầu chức năng***
* Quản lý bán hàng
* Quản lý sản phẩm
* Quản lý danh mục
* Quản lý nhà cung cấp
* Quản lý hóa đơn
* Quản lý nhân viên
* Thống kê
* ***Yêu cầu phi chức năng***
* Mĩ thuật:

+ **Giao diện đơn giản và trực quan**: Giao diện người dùng cần dễ hiểu, dễ sử dụng và trực quan, giúp người dùng dễ dàng tìm thấy các chức năng chính mà không phải mất thời gian tìm kiếm.

+ Màu sắc hài hòa: Sử dụng màu sắc phù hợp, hài hòa và không quá chói mắt. Các màu sắc cần được lựa chọn sao cho dễ nhìn và tạo cảm giác thoải mái cho người dùng.

+ Đồng nhất về thiết kế: Đảm bảo sự nhất quán trong thiết kế trên toàn bộ các trang của ứng dụng. Các yếu tố như màu sắc, phông chữ, nút bấm, và biểu tượng nên được sử dụng một cách đồng bộ.

+ Typography: Sử dụng các phông chữ dễ đọc, kích thước chữ phù hợp và bố cục văn bản rõ ràng. Tránh sử dụng quá nhiều loại phông chữ khác nhau.

+ Khoảng trắng (Whitespace): Sử dụng khoảng trắng hợp lý để giúp nội dung dễ đọc và tạo ra một giao diện gọn gàng, thông thoáng.

* Hiệu năng:

+ Tính dễ sử dụng: Hệ thống cần dễ dàng sử dụng cho cả người mới lẫn người có kinh nghiệm, không yêu cầu kiến thức kỹ thuật cao từ người dùng. Các chức năng phải được thiết kế sao cho dễ dàng truy cập và sử dụng

+ Phản hồi người dùng: Hệ thống cần có các phản hồi nhanh chóng và rõ ràng cho mọi hành động của người dùng, như thông báo khi thực hiện thao tác thành công hoặc khi có lỗi.

+ Định vị rõ ràng: Các phần tử giao diện cần được bố trí hợp lý, với các menu, nút bấm và liên kết được phân biệt rõ ràng và dễ dàng nhận diện.

+ Hướng dẫn và hỗ trợ: Cung cấp các tài liệu hướng dẫn chi tiết và hỗ trợ người dùng dễ dàng trong quá trình sử dụng hệ thống. Các biểu tượng trợ giúp và hướng dẫn trực quan cần phải có sẵn để người dùng dễ dàng tiếp cận.

+ Thiết kế đáp ứng (Responsive Design): Giao diện cần được tối ưu để hoạt động tốt trên nhiều thiết bị khác nhau như máy tính, điện thoại di động, và máy tính bảng, đảm bảo trải nghiệm người dùng mượt mà trên mọi kích thước màn hình.

+ Tối ưu hóa cho thiết bị di động: Hệ thống cần đảm bảo khả năng thao tác dễ dàng trên các thiết bị di động, bao gồm việc hiển thị các nút và menu một cách thuận tiện, dễ thao tác.

* Bảo mật:

+ Phân quyền người dùng (User Authorization): Cần phân quyền người dùng hợp lý để đảm bảo mỗi người chỉ có thể truy cập các chức năng và dữ liệu mà họ được phép. Các vai trò như quản lý, nhân viên, và khách hàng cần có quyền hạn khác nhau trong hệ thống.

+ Sử dụng thư viện và framework bảo mật: Đảm bảo sử dụng các thư viện và framework bảo mật đã được kiểm chứng, được cập nhật thường xuyên để giảm thiểu nguy cơ từ các lỗ hổng bảo mật. Điều này giúp bảo vệ dữ liệu của người dùng và hệ thống khỏi các mối đe dọa tiềm ẩn.

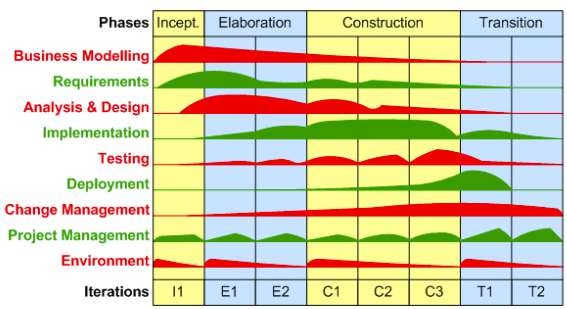
+ **Mã hóa dữ liệu**: Các thông tin nhạy cảm như mật khẩu, thông tin thanh toán cần được mã hóa trước khi lưu trữ, nhằm đảm bảo an toàn tuyệt đối cho người dùng. Hệ thống cần sử dụng các thuật toán mã hóa mạnh mẽ và an toàn.

+ **Đảm bảo kết nối an toàn:** Cần sử dụng giao thức HTTPS để đảm bảo rằng tất cả dữ liệu truyền tải giữa client và server đều được mã hóa và bảo mật.

## Phân tích hệ thống

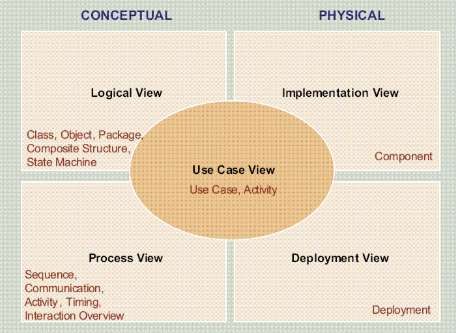
### Phân tích và thiết kế hệ thống thông tin theo phương pháp hướng đối tượng với UML

* **Tổng quan về UML**
  + - * + UML (Unified Modelling Language) là ngôn ngữ mô hình hoá tổng quát.
        + Được xây dựng để đặc tả, phát triển và viết tài liệu cho các khía cạnh trong phát triển phần mềm hướng đối tượng.
        + UML giúp người phát triển hiểu rõ và ra quyết định liên quan đến phần mềm cần xây dựng.
        + UML bao gồm một tập các khái niệm, các ký hiệu, các biểu đồ và hướng dẫn.
        + UML là ký pháp chứ không phải là phương pháp.
* **Quy trình phát triển phần mềm với UML**
  + - * + UML có thể áp dụng cho tất cả các pha của quy trình phát triển phần mềm
        + Quy trình phát triển cho UML – Rational Unified Process (RUP)



Hình 3.1: Quy trình phát triển phần mềm với UML

* **Các thành phần của ngôn ngữ UML**
* **Views (Các góc nhìn)**



Hình 3.2: Các góc nhìn của UML

* + - * + UML cung cấp 5 góc nhìn khác nhau nhằm đặc tả một hệ thống:

+ Use Case View:

Là góc nhìn phía bên ngoài hệ thống.

Mô tả các chức năng lớn và các quy trình của hệ thống.

Là góc nhìn của người dùng cuối, người phân tích, người kiểm định.

+ Logical View:

Là góc nhìn phía bên trong hệ thống.

Mô tả cấu trúc tĩnh (lớp, đối tượng, quan hệ) và cấu trúc động (chuyển trạng thái) của các thành phần trong hệ thống.

Là góc nhìn chủ yếu của người thiết kế, người phát triển.

+ Process View:

Là góc nhìn cho thấy các quá trình hoạt động, trình tự hoạt động (song hành hay đồng bộ) trong hệ thống.

Là góc nhìn chủ yếu của người thiết kế, người phát triển.

+ Implemetation View:

Là góc nhìn cho thấy các thành phần (packages, files, modules,..) được lắp ráp và tương tác với nhau để tạo ra hệ thống.

Là góc nhìn chủ yếu của người thiết kế, người phát triển.

+ Deployment View:

Là góc nhìn cho thấy mô hình phần cứng mà trên đó hệ thống được thực hiện.

Chỉ rõ sự phân bố, sự sắp đặt các phần của hệ thống vật lý trên các đơn vị phần cứng (servers, network)

Là góc nhìn chủ yếu của người thiết kế, người triển khai.

* **Diagram (Biểu đồ)**
  + - * + Các biểu đồ cấu trúc:

+ Package diagram (biểu đồ gói)

+ Class diagram (biểu đồ lớp)

+ Component diagram (biểu đồ thành phần)

+ Deployment diagram (biểu đồ triển khai)

* + - * + Các biểu đồ hành vi:

+ Activity diagram (biểu đồ hoạt động)

+ Use Case diagram (biểu đồ trường hợp sử dụng)

+ Sequence diagram (biểu đồ tuần tự)

+ State machine diagram (biểu đồ trạng thái)

* **Biểu đồ Use case (Use case Diagram)**
  + - * + Biểu đồ Use case biểu diễn sơ đồ chức năng của hệ thống: Từ tập yêu cầu của hệ thống, biểu đồ Use case sẽ phải chỉ ra hệ thống cần thực hiện điều gì để thoả mãn các yêu cầu của người dùng hệ thống đó.
        + Biểu đồ Use case chỉ ra sự tương tác giữa các tác nhân và hệ thống thông qua các Use case.
        + Đi kèm với biểu đồ Use case là các kịch bản (scenario).
        + Các bước xây dựng mô hình Use case

+ Xác định các Actor (Tác nhân)

+ Xác định các Use case (Trường hợp sử dụng)

+ Xác định mối quan hệ và phân rã biểu đồ Use case

+ Đặc tả Use case (Biểu diễn các Use case thông qua các kịch bản)

+ Kiểm tra và hiệu chỉnh mô hình

* **Đặc tả Use-case**
  + - * + Mẫu đặc tả Use case gồm các thông tin chính sau:

+ Mã số Use case

+ Tên Use case

+ Mức: Mức của Use case trong biểu đồ phân rã

+ Tên tác nhân: Tác nhân chính, tác nhân phụ

+ Tiền điều kiện: Điều kiện để kích hoạt Use case

+ Kích hoạt: Sự kiện kích hoạt Use case

+ Hậu điều kiện: Kết quả trong trường hợp Use case hoàn thành

+ Kịch bản chính: Chuỗi các sự kiện chính của Use case

+ Kịch bản ngoại lệ: Chuỗi các sự kiện xử lý ngoại lệ

* + - * + Có thể kết hợp sử dụng biểu đồ hoạt động (Activity Diagram) để mô tả các kịch bản.
* **Activity Diagram (Biểu đồ hoạt động)**
  + - * + Biểu đồ hoạt động (Activity Diagram) được sử dụng để mô tả:

+ Các hoạt động trong một quy trình nghiệp vụ.

+ Biểu diễn các hành động được thực hiện trong một use case.

=> Có thể sử dụng biểu đồ hoạt động để phân tích quy trình nghiệp vụ hoặc để đặc tả (mô tả chi tiết) các use case.

* **Class Diagram (Biểu đồ lớp)**
  + - * + Biểu đồ lớp (Class diagram) chỉ ra sự tồn tại của các lớp và mối quan hệ giữa chúng trong bản thiết kế logic của một hệ thống.
        + Sử dụng hình chữ nhật gồm 3 thành phần

+ Tên lớp

+ Các thuộc tính

+ Các phương thức

* **State Diagram (Biểu đồ trạng thái)**
  + - * + Biểu đồ trạng thái được sử dụng để biểu diễn các trạng thái và sự chuyển tiếp giữa các trạng thái của các đối tượng trong một lớp xác định. Thông thường, mỗi lớp sẽ có một biểu đồ trạng thái (trừ lớp trừu tượng là lớp không có đối tượng).
        + Có hai dạng biểu đồ trạng thái:

+ Biểu đồ trạng thái cho một use case: Mô tả các trạng thái và chuyển tiếp trạng thái của một đối tượng thuộc một lớp nào đó trong hoạt động của một use case cụ thể.

+ Biểu đồ trạng thái hệ thống: Mô tả tất cả các trạng thái của một đối tượng trong toàn bộ hoạt động của cả hệ thống.

* **Sequence Diagram (Biểu đồ trình tự)**
  + - * + Biểu diễn mối quan hệ giữa các đối tượng theo thứ tự thời gian. Biểu đồ tuần tự nhấn mạnh thứ tự thực hiện của các tương tác.
        + Biểu đồ trình tự thông thường được sử dụng để biểu diễn thứ tự các tương tác giữa các đối tượng bên trong mỗi UC:

+ Mô tả có thứ tự các hành vi giữa tác nhân và hệ thống (coi hệ thống là 1 đối tượng đặc biệt) theo từng UC.

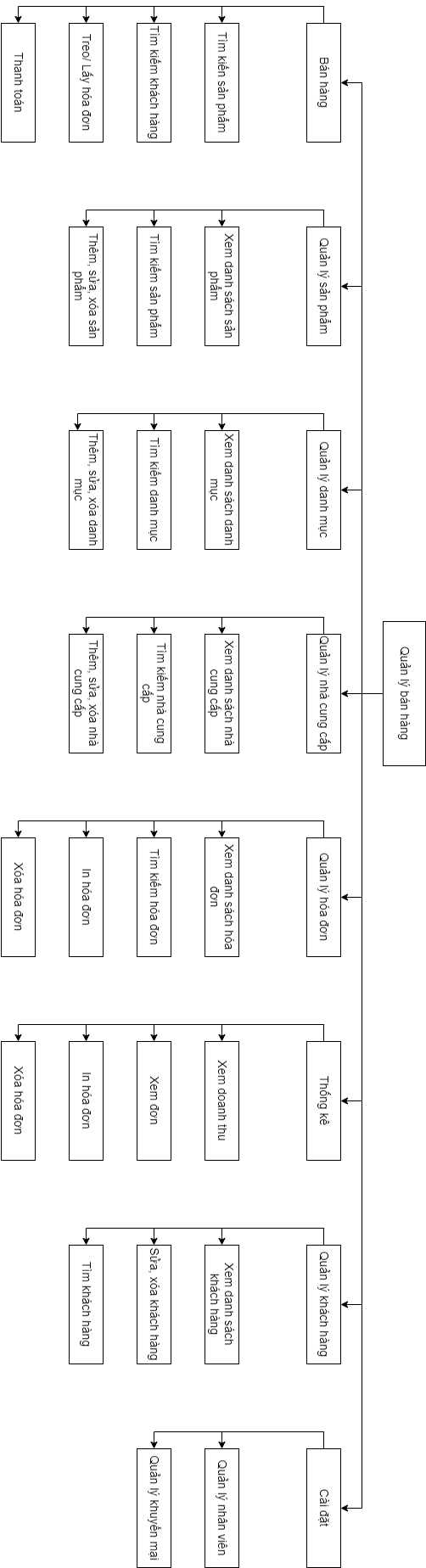
+ Mô tả có thứ tự các hành vi giữa các đối tượng trong biểu đồ lớp phân tích (Analysic Class Diagram) của từng UC.

* **Component Diagram (Biểu đồ thành phần)**
  + - * + Được sử dụng để biểu diễn các thành phần cấu thành nên hệ thống.
        + Mỗi thành phần có thể coi như một module nhỏ hơn được xây dựng cho quá trình triển khai hệ thống.
        + Trong phần mềm thì các thành phần có thể là các gói thư viện (package, .dll,

.jar) hoặc các phần mềm nhỏ được tạo ra từ các thành phần nhỏ hơn như các lớp và các thư viện chức năng.

* **Deployment Diagram (Biểu đồ triển khai)**
  + - * + Biểu diễn kiến trúc cài đặt và triển khai hệ thống dưới dạng các nodes và các mối quan hệ giữa các node đó. Thông thường, các nodes được kết nối với nhau thông qua các liên kết truyền thông như các kết nối mạng, liên kết TCP- IP, microwave…
        + Biểu đồ triển khai chỉ ra bố trí vật lý của mạng và các thành phần hệ thống sẽ đặt ở đâu.

### Biểu đồ phân rã chức năng



Hình 3.3: Biểu đồ phân rã chức năng

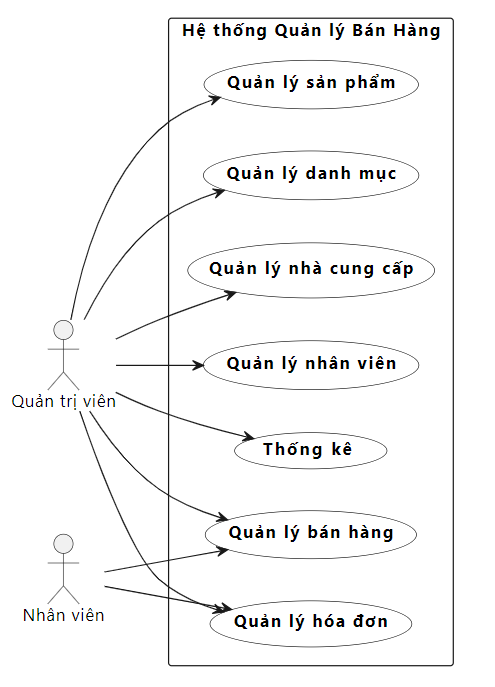
### Các tác nhân

Bảng 3.2: Mô tả các tác nhân trong hệ thống

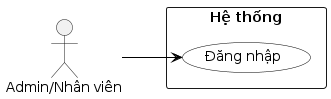
|  |  |  |
| --- | --- | --- |
| **STT** | **Tên tác nhân** | **Mô tả** |
| 1 | Admin | Là người quản trị hệ thống, có quyền cao nhất trong hệ thống, quản  lý và kiểm soát các hoạt động, tài nguyên và người dùng trong hệ thống. |
| 2 | Nhân viên | Là người sử dụng hệ thống để thực hiện bán hàng, xem tồn kho hàng. |

### Biều đồ use-case và đặc tả các use-case

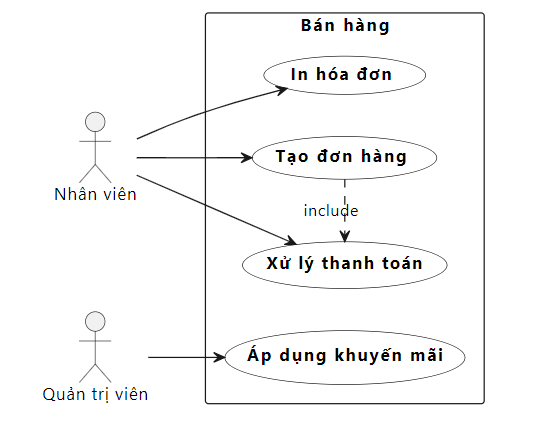
**a, Biểu đồ use-case**

****

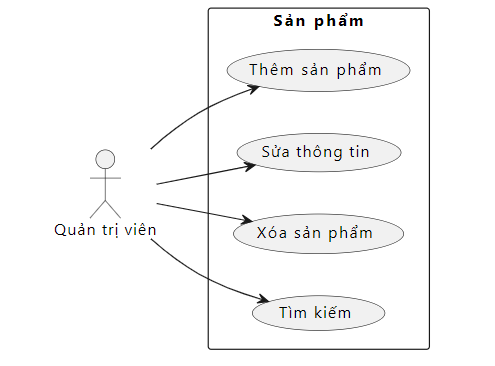
Hình 3.4: Biểu đồ use-case tổng quát



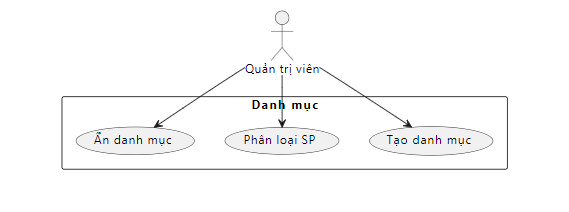
Hình 3.5: Biểu đồ use-case chức năng đăng nhập

**

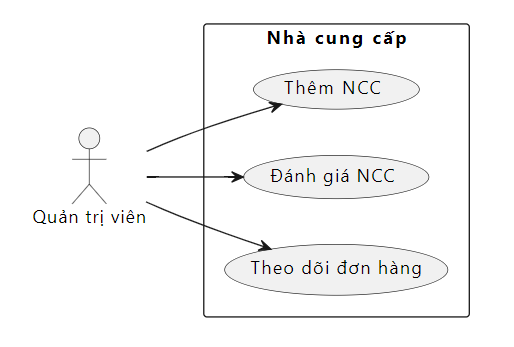
Hình 3.6: Biểu đồ use-case chức năng bán hàng

**

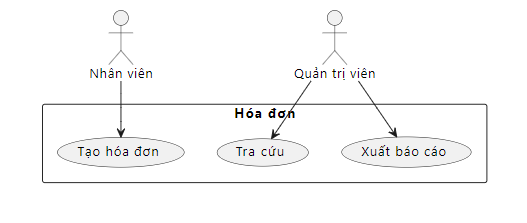
Hình 3.7: Biểu đồ use-case chức năng quản lý sản phẩm

**

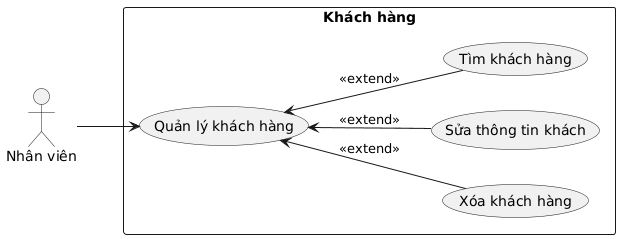
Hình 3.8: Biểu đồ use-case chức năng quản lý danh mục

**

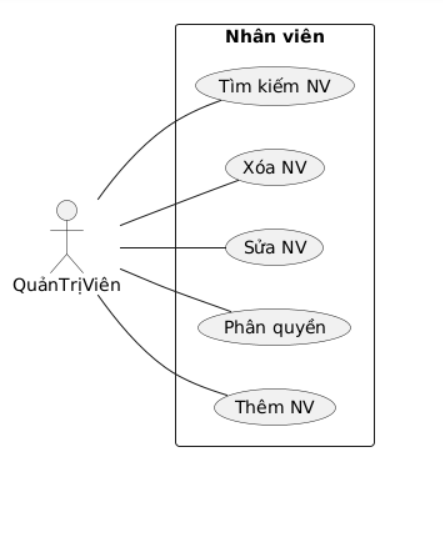
Hình 3.9: Biểu đồ use-case chức năng quản lý nhà cung cấp



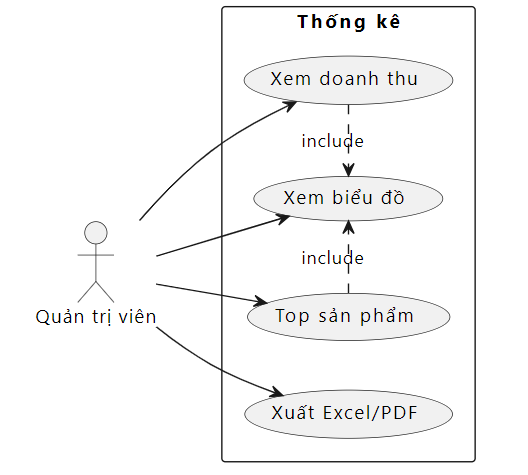
Hình 3.10: Biểu đồ use-case chức năng quản lý hóa đơn



Hình 3.11: Biểu đồ use-case chức năng quản lý khách hàng

**

Hình 3.12: Biểu đồ use-case chức năng quản lý nhân viên

**

Hình 3.13: Biểu đồ use-case chức năng thống kê

**b, Đặc tả các use-case**

Bảng 3.3: Đặc tả use-case đăng nhập

|  |  |
| --- | --- |
| **User case** | **Nội dung** |
| **Tên use case** | Đăng nhập |
| **Mô tả** | Chức năng này cho phép người dùng đăng nhập tài khoản vào hệ thống để truy cập vào ứng dụng và sử dụng các chức năng. |
| **Tác nhân** | Người dùng |
| **Tiền điều kiện** | 1. Người dùng phải truy cập vào trang web.  2. Người dùng đã có tài khoản trên hệ thống. |
| **Hậu điều kiện** | Người dùng đăng nhập thành công vào hệ thống. |
| **Luồng sự kiện chính** | 1. Hệ thống hiển thị màn hình đăng nhập.  2. Người dùng nhập tài khoản và mật khẩu.  3. Hệ thống kiểm tra thông tin đăng nhập.  4. Nếu đăng nhập thành công, hệ thống chuyển hướng người dùng đến trang chủ.  5. Nếu đăng nhập thất bại, hệ thống yêu cầu người dùng nhập lại thông tin đăng nhập. |

Bảng 3.4: Đặc tả use-case quản lý bán hàng

|  |  |
| --- | --- |
| **User case** | **Nội dung** |
| **Tên use case** | Quản lý bán hàng |
| **Mô tả** | Chức năng này cho phép nhân viên bán hàng quản lý các đơn hàng, xử lý thanh toán, in hóa đơn và áp dụng các chương trình khuyến mãi cho khách hàng. |
| **Tác nhân** | Nhân viên, Quản trị viên |
| **Tiền điều kiện** | 1. Nhân viên đã đăng nhập vào hệ thống. 2. Quản trị viên đã cấu hình các chương trình khuyến mãi nếu có. |
| **Hậu điều kiện** | Đơn hàng được tạo, thanh toán và in hóa đơn thành công, khuyến mãi được áp dụng nếu đủ điều kiện. |
| **Luồng sự kiện chính** | 1.Nhân viên tạo đơn hàng mới.  2. Nhân viên xử lý thanh toán cho đơn hàng.  3. Nhân viên in hóa đơn cho khách hàng.  4. Nếu có chương trình khuyến mãi, nhân viên áp dụng khuyến mãi khi thanh toán. |

Bảng 3.5: Đặc tả use-case quản lý sản phẩm

|  |  |
| --- | --- |
| **User case** | **Nội dung** |
| **Tên use case** | Quản lý sản phẩm |
| **Mô tả** | Chức năng này cho phép quản trị viên quản lý các sản phẩm trong hệ thống, bao gồm thêm mới, sửa thông tin, xóa sản phẩm và tìm kiếm sản phẩm. |
| **Tác nhân** | Quản trị viên |
| **Tiền điều kiện** | 1. Quản trị viên đã đăng nhập vào hệ thống.  2. Quản trị viên có quyền truy cập vào chức năng quản lý sản phẩm. |
| **Hậu điều kiện** | Các sản phẩm được quản lý, thông tin sản phẩm được cập nhật, và sản phẩm có thể bị xóa hoặc tìm kiếm thành công. |
| **Luồng sự kiện chính** | 1. Quản trị viên thêm sản phẩm mới vào hệ thống.  2. Quản trị viên sửa thông tin của sản phẩm hiện tại.  3. Quản trị viên xóa sản phẩm không còn bán.  4. Quản trị viên tìm kiếm sản phẩm trong hệ thống. |

Bảng 3.6: Đặc tả use-case danh mục

|  |  |
| --- | --- |
| **User case** | **Nội dung** |
| **Tên use case** | Quản lý danh mục |
| **Mô tả** | Chức năng này cho phép quản trị viên quản lý các danh mục sản phẩm, bao gồm tạo mới, phân loại sản phẩm vào danh mục, và ẩn danh mục khi không còn sử dụng. |
| **Tác nhân** | Quản trị viên |
| **Tiền điều kiện** | 1. Quản trị viên đã đăng nhập vào hệ thống. 2. Quản trị viên có quyền truy cập vào chức năng quản lý danh mục. |
| **Hậu điều kiện** | Các danh mục sản phẩm được tạo, phân loại và ẩn khi cần thiết. |
| **Luồng sự kiện chính** | 1. Quản trị viên tạo một danh mục sản phẩm mới.  2. Quản trị viên phân loại sản phẩm vào các danh mục đã có.  3. Quản trị viên ẩn các danh mục không còn sử dụng. |

Bảng 3.7: Đặc tả use-case quản lý nhà cung cấp

|  |  |
| --- | --- |
| **User case** | **Nội dung** |
| **Tên use case** | Quản lý nhà cung cấp |
| **Mô tả** | Chức năng này cho phép quản trị viên quản lý các nhà cung cấp, bao gồm thêm mới nhà cung cấp, đánh giá nhà cung cấp, và theo dõi các đơn hàng từ nhà cung cấp. |
| **Tác nhân** | Quản trị viên |
| **Tiền điều kiện** | 1. Quản trị viên đã đăng nhập vào hệ thống.  2. Quản trị viên có quyền truy cập vào chức năng quản lý nhà cung cấp. |
| **Hậu điều kiện** | Các nhà cung cấp được thêm mới, đánh giá và theo dõi đơn hàng đúng cách. |
| **Luồng sự kiện chính** | 1. Quản trị viên thêm nhà cung cấp mới vào hệ thống. 2. Quản trị viên đánh giá chất lượng nhà cung cấp. 3. Quản trị viên theo dõi tiến độ các đơn hàng của nhà cung cấp. |

Bảng 3.8: Đặc tả use-case quản lý hóa đơn

|  |  |
| --- | --- |
| **User case** | **Nội dung** |
| **Tên use case** | Quản lý hóa đơn |
| **Mô tả** | Chức năng này cho phép nhân viên tạo hóa đơn, quản trị viên tra cứu và xuất báo cáo hóa đơn. |
| **Tác nhân** | Nhân viên, Quản trị viên |
| **Tiền điều kiện** | 1. Nhân viên và quản trị viên đã đăng nhập vào hệ thống. 2. Quản trị viên có quyền truy cập vào chức năng tra cứu và xuất báo cáo. |
| **Hậu điều kiện** | Hóa đơn được tạo, tra cứu và báo cáo được xuất thành công. |
| **Luồng sự kiện chính** | 1. Nhân viên tạo hóa đơn cho đơn hàng.  2. Quản trị viên tra cứu thông tin hóa đơn.  3. Quản trị viên xuất báo cáo hóa đơn theo yêu cầu. |

*Bảng 3.9: Đặc tả use-case quản lý nhân viên*

|  |  |
| --- | --- |
| **User case** | **Nội dung** |
| **Tên use case** | Quản lý nhân viên |
| **Mô tả** | Chức năng này cho phép quản trị viên thực hiện các thao tác liên quan đến nhân viên như thêm mới, phân quyền và tính lương. |
| **Tác nhân** | Quản trị viên |
| **Tiền điều kiện** | 1. Quản trị viên đã đăng nhập vào hệ thống.  2. Quản trị viên có quyền truy cập vào chức năng quản lý nhân sự. |
| **Hậu điều kiện** | Danh sách nhân viên được cập nhật, quyền được phân bổ đúng, và lương được tính toán chính xác. |
| **Luồng sự kiện chính** | 1. Quản trị viên thêm nhân viên mới vào hệ thống.  2. Quản trị viên phân quyền cho nhân viên theo vai trò.  3. Quản trị viên sửa, xóa nhân viên. |

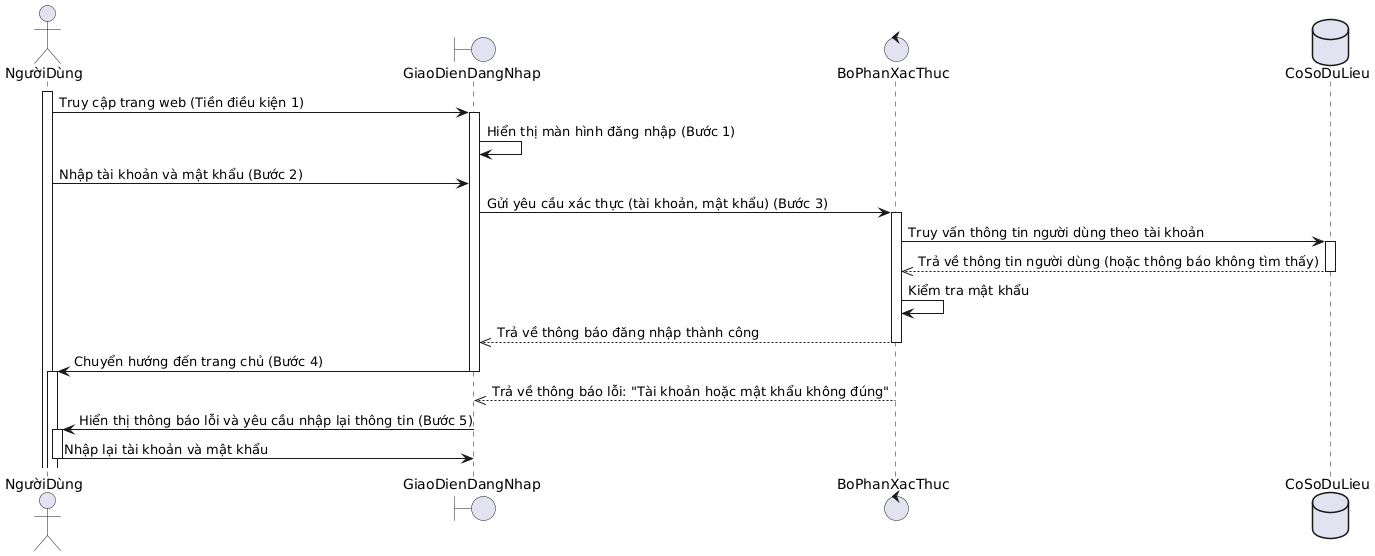
Bảng 3.10: Đặc tả use-case thống kê

|  |  |
| --- | --- |
| **User case** | **Nội dung** |
| **Tên use case** | Thống kê |
| **Mô tả** | Chức năng này cho phép quản trị viên theo dõi tình hình kinh doanh qua việc xem doanh thu, các sản phẩm bán chạy, biểu đồ trực quan và xuất dữ liệu dưới dạng Excel hoặc PDF. |
| **Tác nhân** | Quản trị viên |
| **Tiền điều kiện** | 1. Quản trị viên đã đăng nhập vào hệ thống. 2. Dữ liệu bán hàng đã được ghi nhận trong hệ thống. |
| **Hậu điều kiện** | Dữ liệu thống kê được hiển thị, báo cáo được xuất thành công. |
| **Luồng sự kiện chính** | 1. Quản trị viên chọn chức năng xem thống kê.  2. Hệ thống hiển thị doanh thu và top sản phẩm bán chạy.  3. Hệ thống trình bày dữ liệu dưới dạng biểu đồ trực quan.  4. Quản trị viên có thể xuất báo cáo thành file Excel hoặc PDF. |

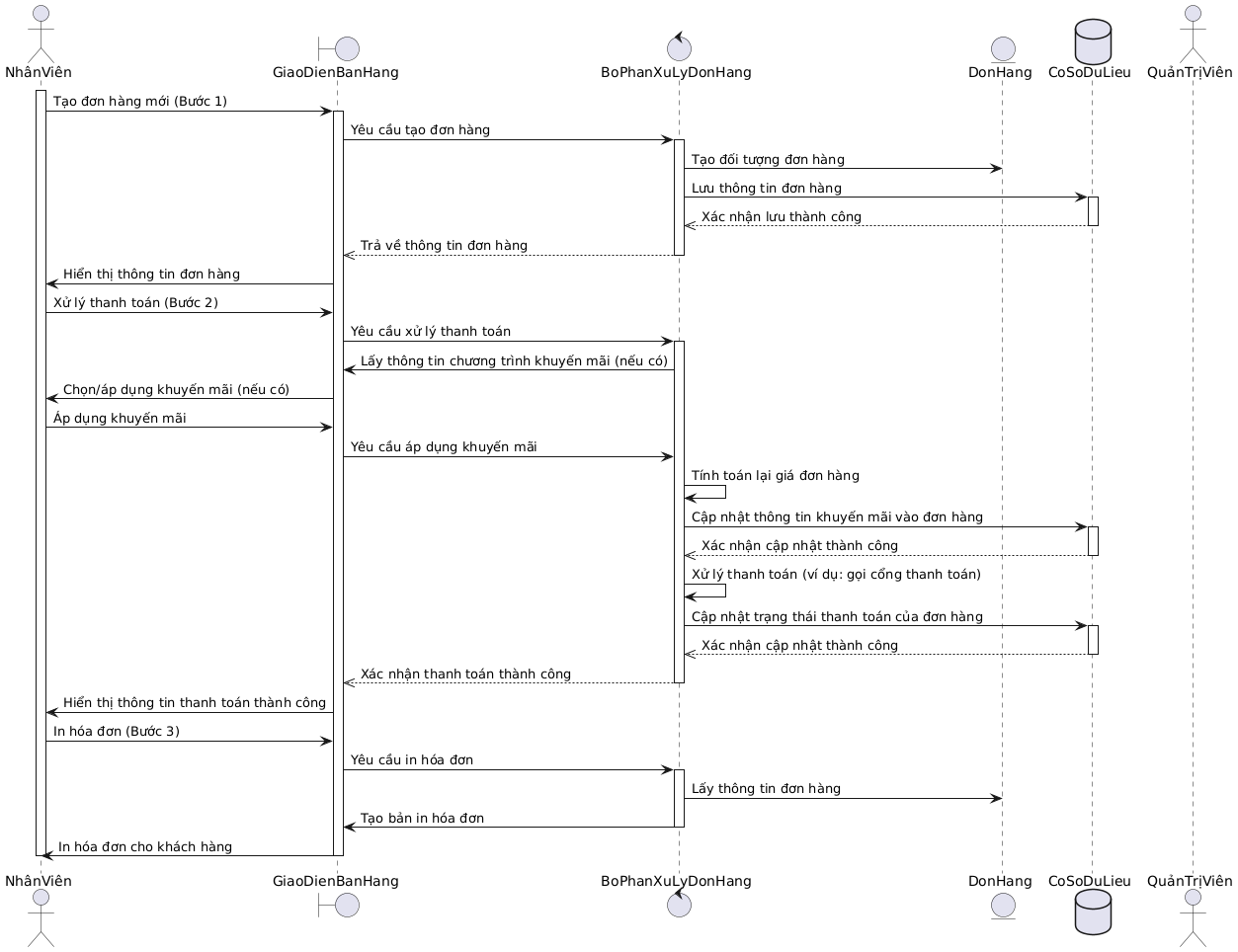
*Bảng 3.11: Đặc tả use-case quản lý khách hàng*

|  |  |
| --- | --- |
| **User case** | **Nội dung** |
| **Tên use case** | Quản lý khách hàng |
| **Mô tả** | Chức năng này cho phép quản trị viên thực hiện các thao tác liên quan đến khách hàng như sửa, xóa. |
| **Tác nhân** | Quản trị viên |
| **Tiền điều kiện** | 1. Quản trị viên đã đăng nhập vào hệ thống.  2. Quản trị viên có quyền truy cập vào chức năng quản lý khách hàng. |
| **Hậu điều kiện** | Danh sách khách hàng được cập nhật. |
| **Luồng sự kiện chính** | 1. Quản trị viên sửa, xóa khách hàng trong hệ thống  2. Quản trị viên có thể tìm khách hàng trong hệ thống |

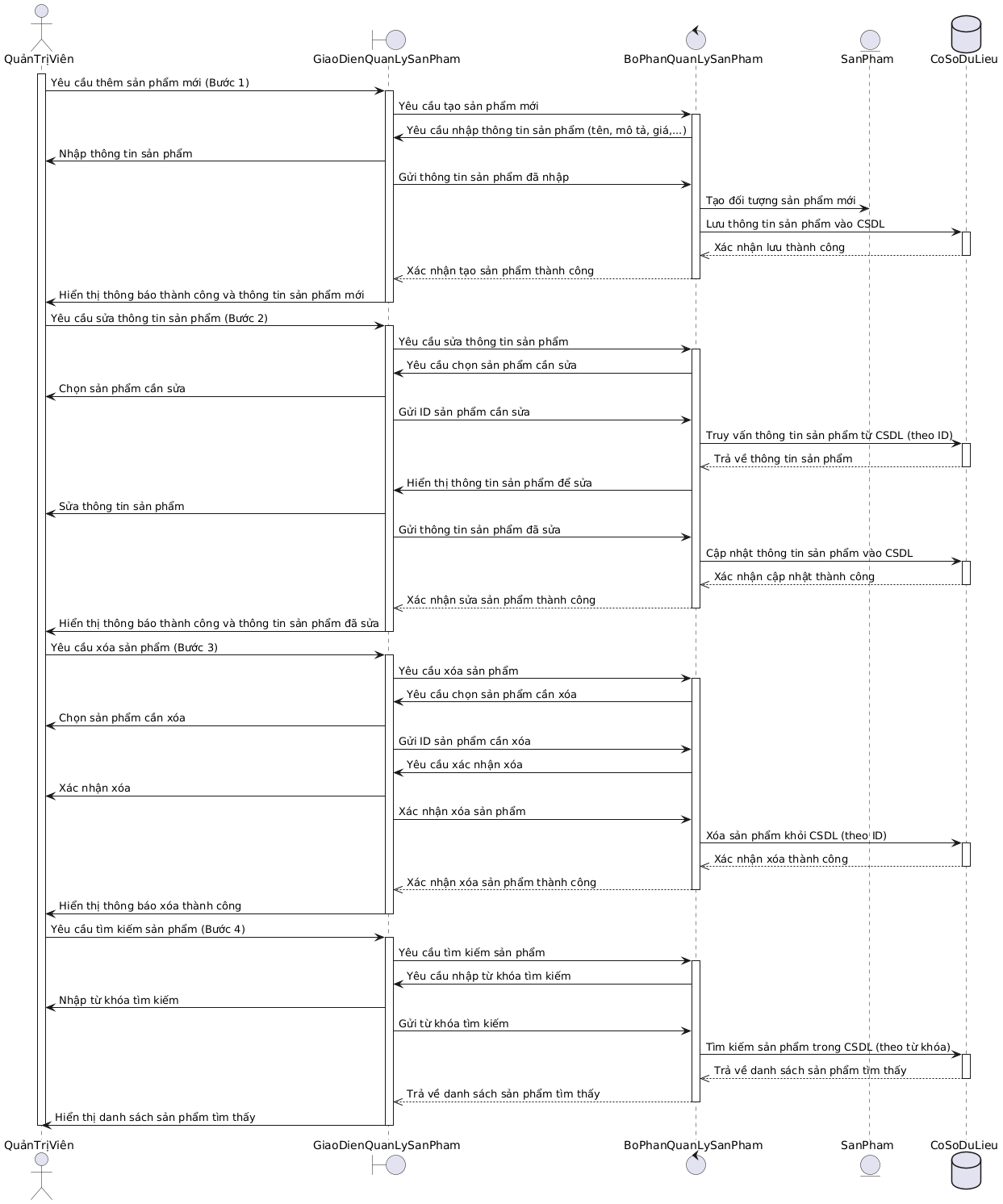
### Biểu đồ tuần tự



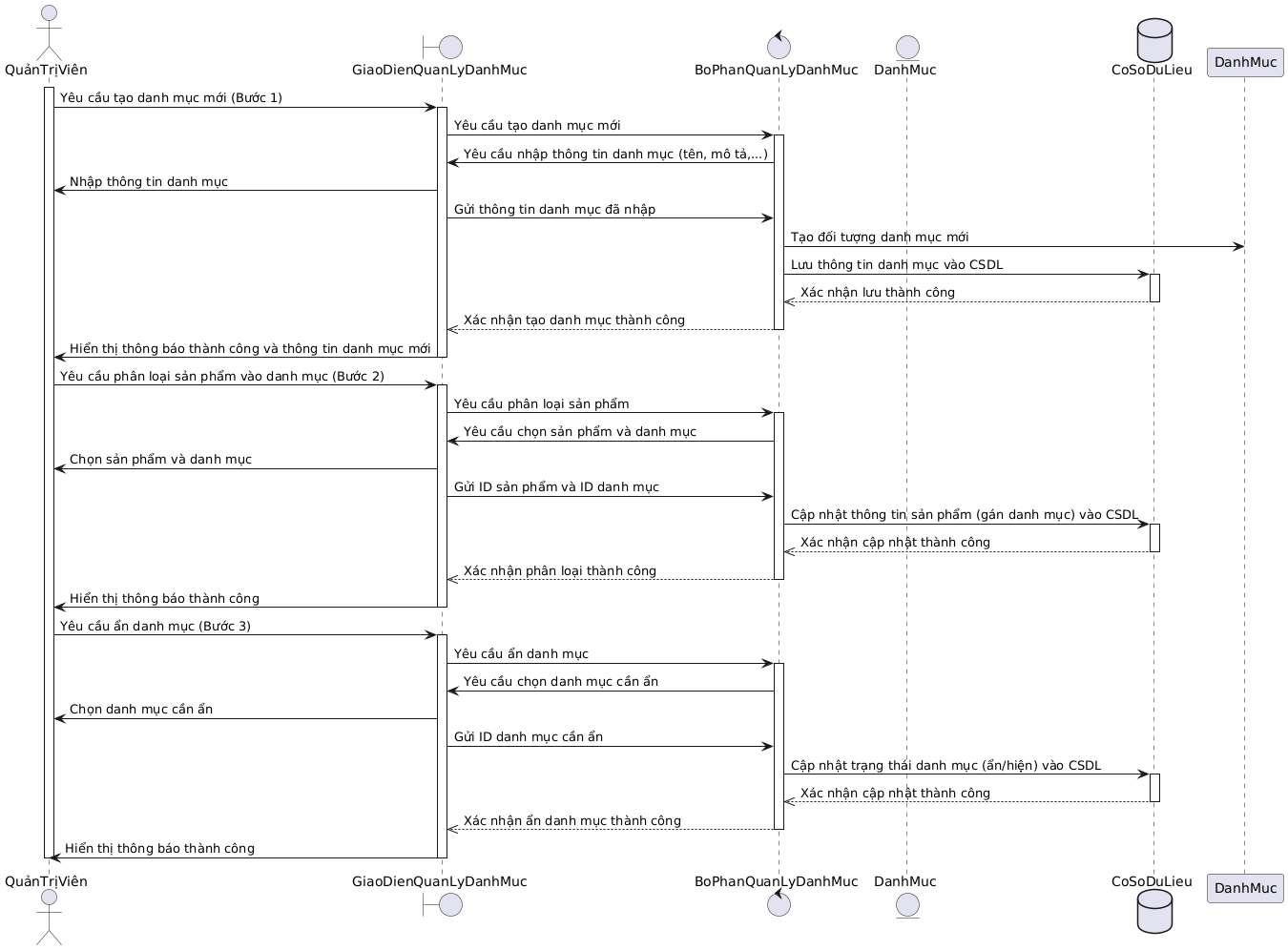
Hình 3.14: Biểu đồ tuần tự chức năng đăng nhập



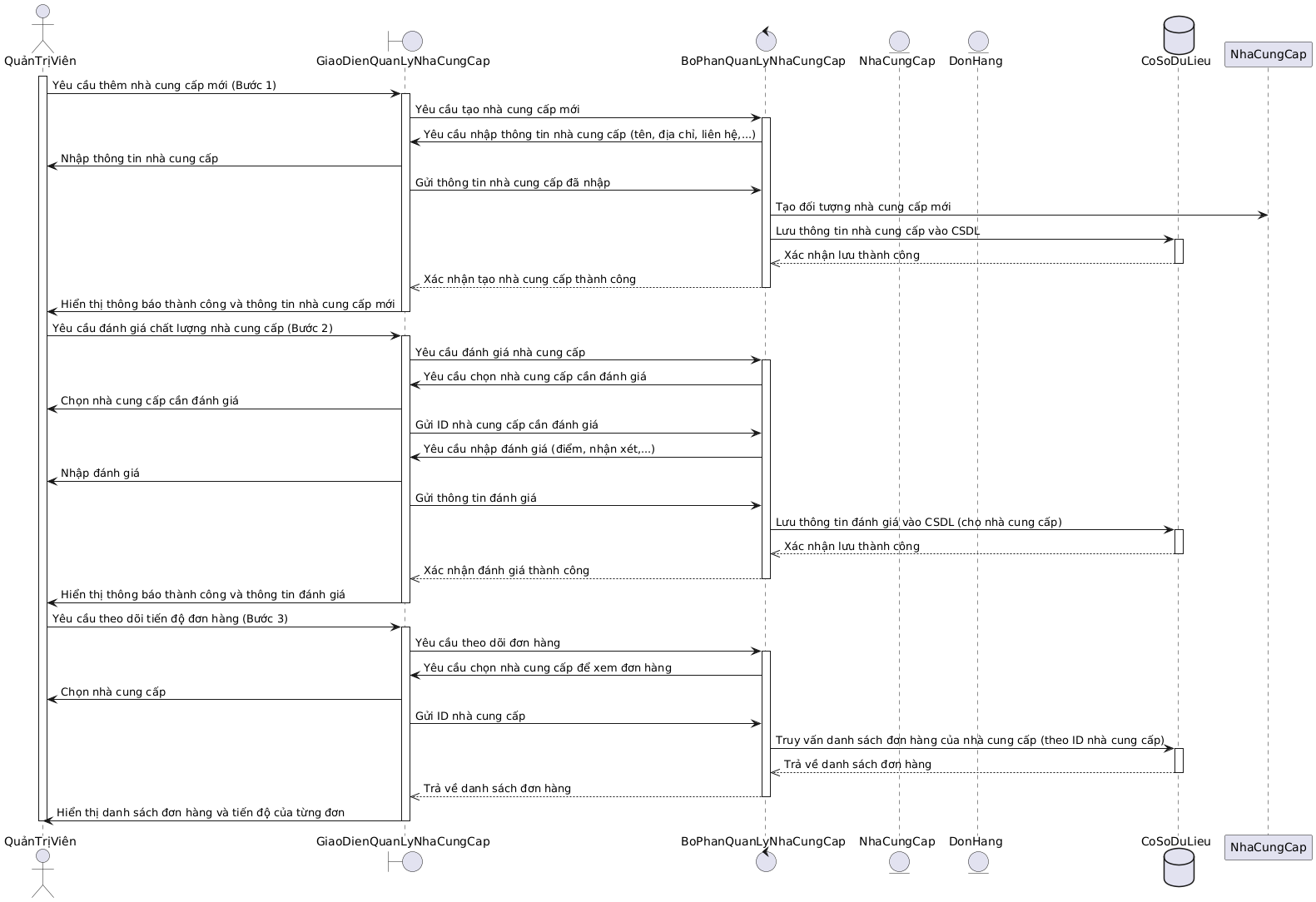
Hình 3.15: Biểu đồ tuần tự chức năng bán hàng

**

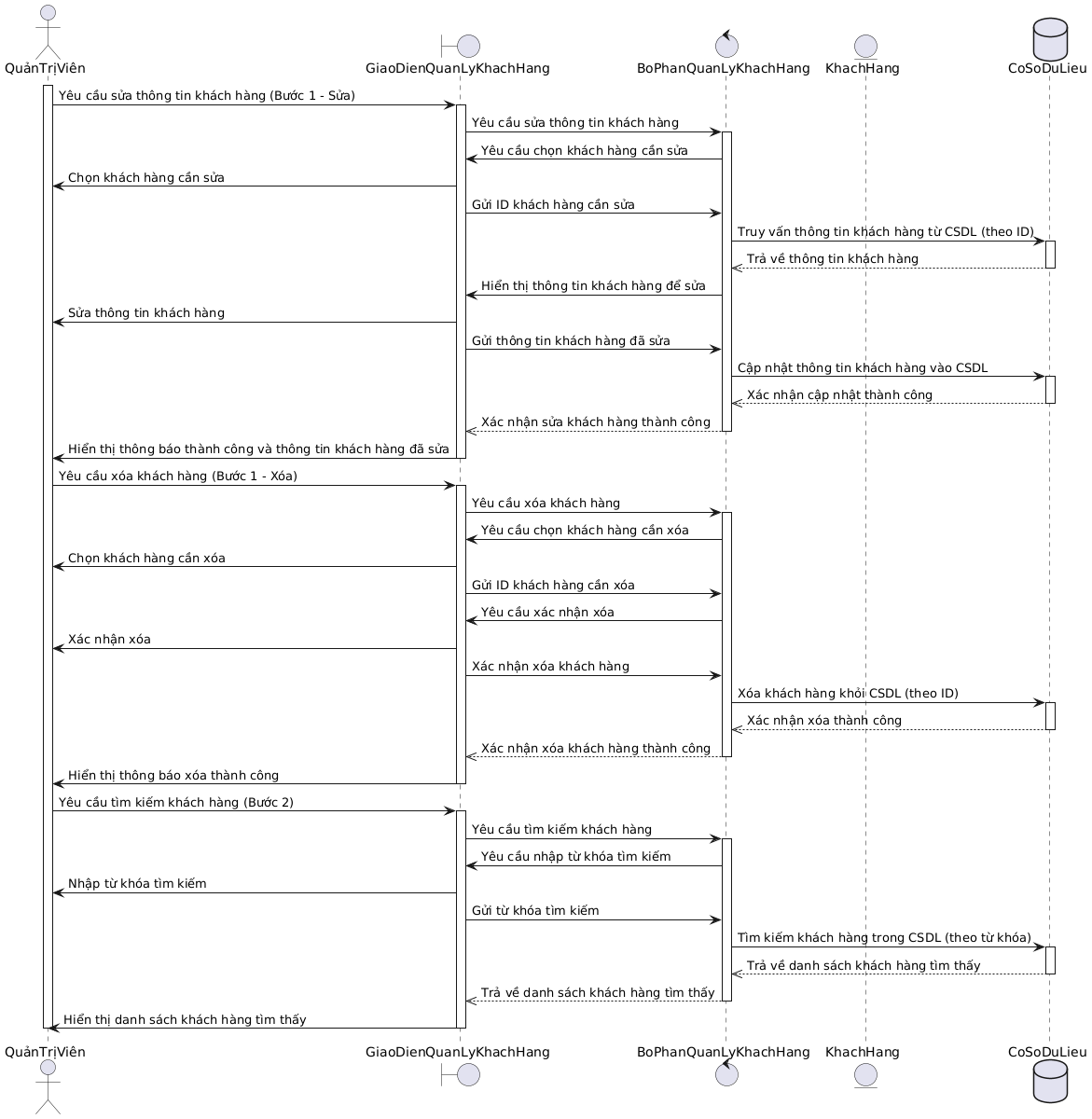
Hình 3.16: Biểu đồ tuần tự chức năng quản lý sản phẩm



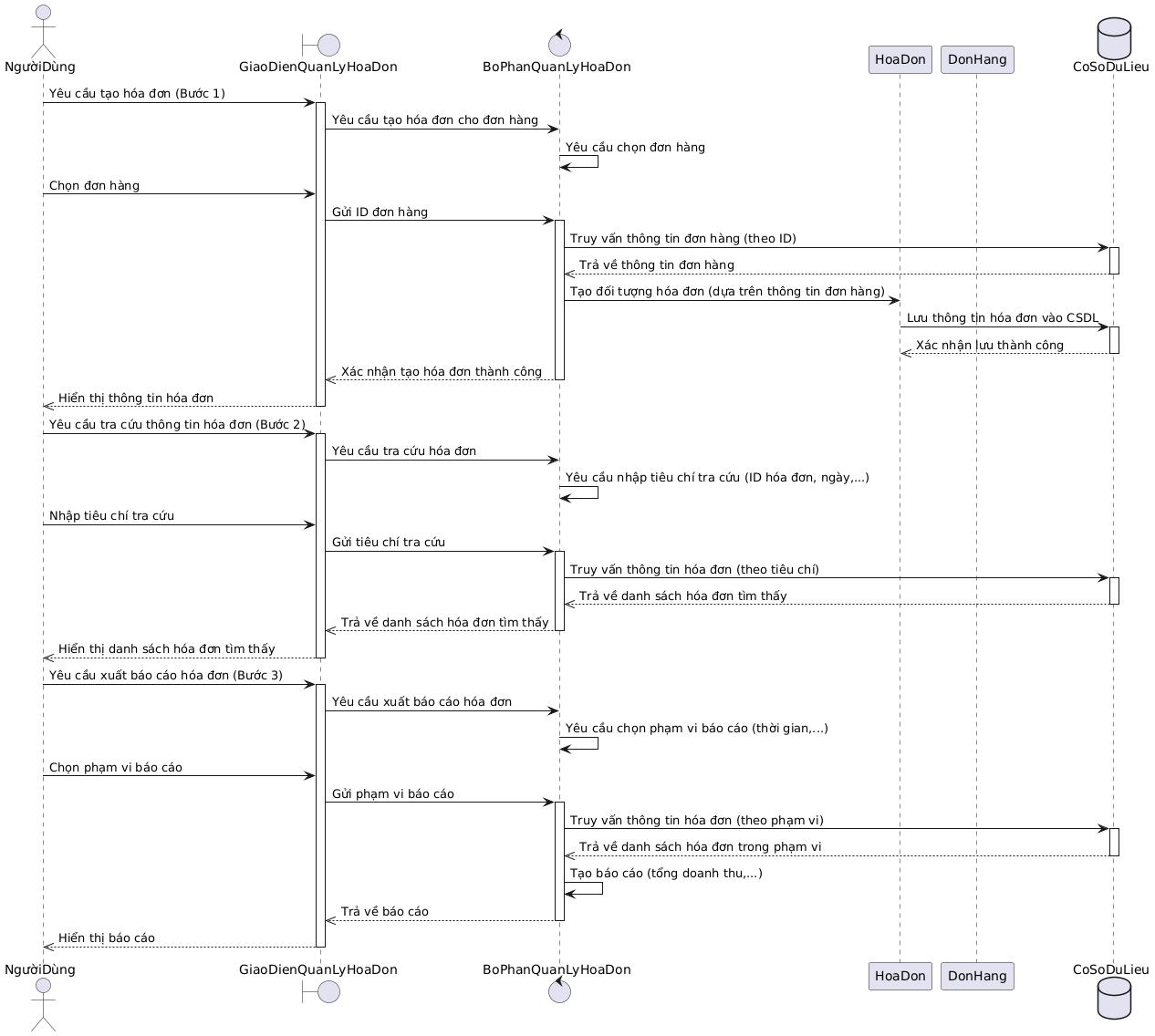
Hình 3.17: Biểu đồ tuần tự chức năng quản lý danh mục



Hình 3.18: Biểu đồ tuần tự chức năng quản lý nhà cung cấp



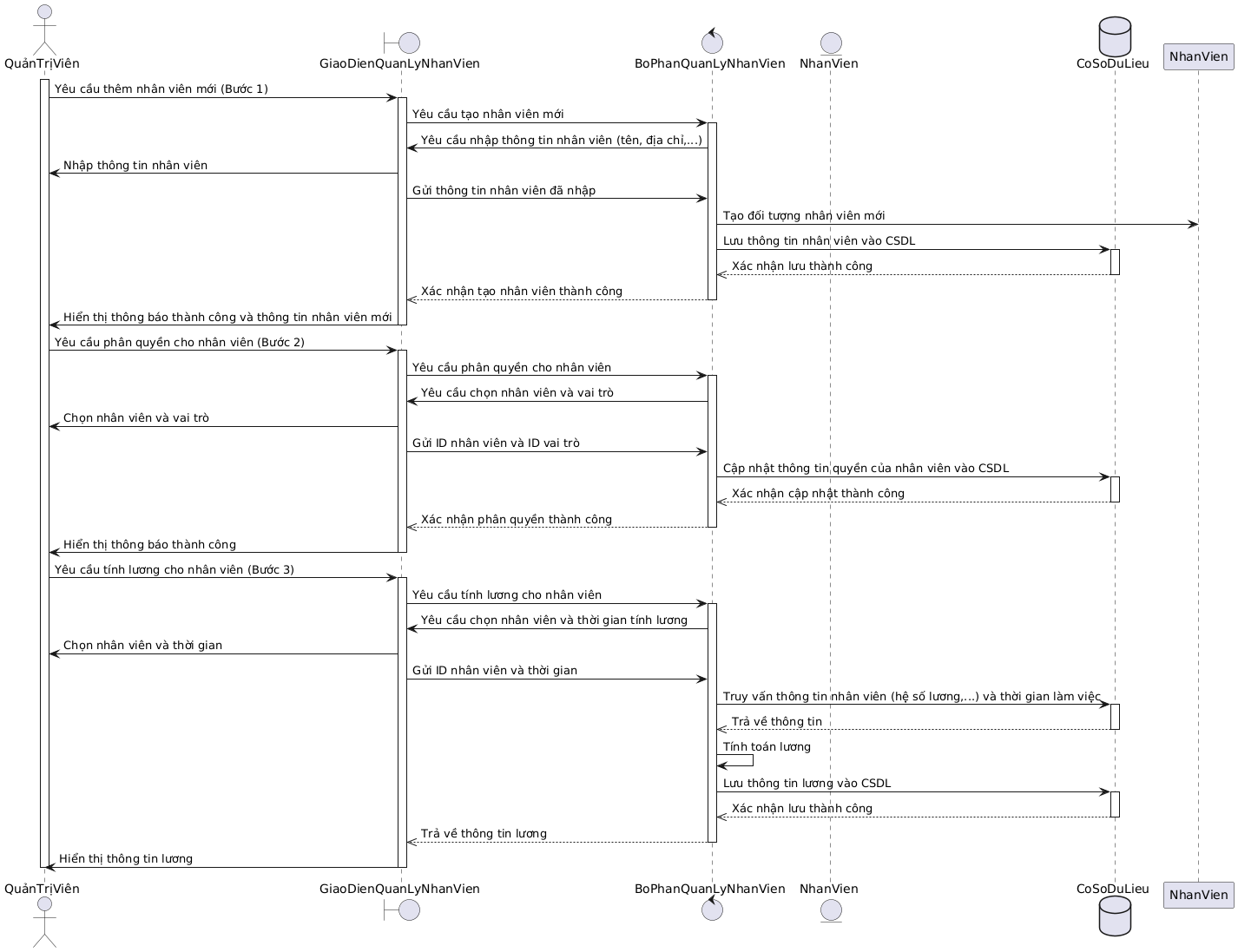
Hình 3.19: Biểu đồ tuần tự chức năng quản lý khách hàng



Hình 3.20: Biểu đồ tuần tự chức năng quản lý hóa đơn

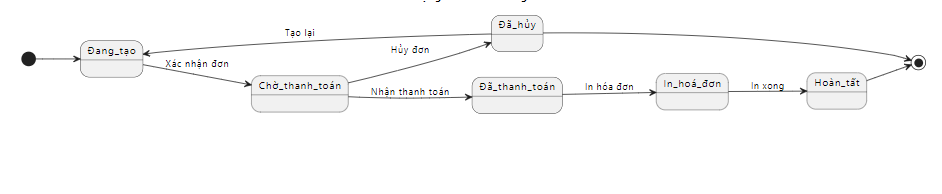


Hình 3.21: Biểu đồ tuần tự chức năng quản lý nhân viên

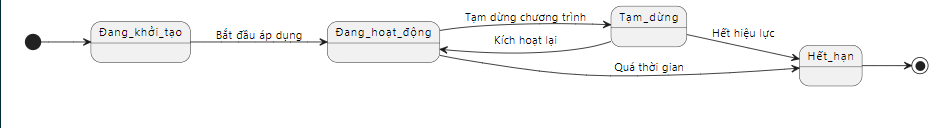


Hình 3.22: Biểu đồ tuần tự chức năng thống kê

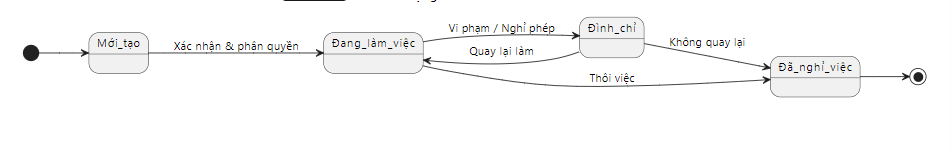
### Biểu đồ trạng thái

******

Hình 3.23: Biểu đồ trạng thái của đơn hàng

**

Hình 3.24: Biểu đồ trạng thái của khuyến mãi

**

Hình 3.25: Biểu đồ trạng thái nhân viên

## Thiết kế hệ thống

### Thiết kế lớp

**a, Lớp thực thể (Entities class)**

Bảng 3.11: Danh sách thực thể trong hệ thống

|  |  |
| --- | --- |
| **Tên thực thể** | **Mô tả** |
| Category | Thực thể danh mục sản phẩm |
| Customer | Thực thể khách hàng |
| Employee | Thực thể nhân viên |
| Order | Thực thể đơn hàng chính thức |
| OrderDetail | Thực thể chi tiết đơn hàng |
| Product | Thực thể sản phẩm |
| Promotion | Thực thể khuyến mại |
| Supplier | Thực thể nhà cung cấp |
| TempOrder | Thực thể đơn hàng tạm thời |
| TempOrderDetail | Thực thể chi tiết đơn hàng tạm thời |

**b, Mô tả chi tiết từng lớp**

Bảng 3.12: Mô tả chi tiết lớp Category

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên lớp** | | | |
| Category | | | |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Phương thức** |
| 1 | categoryId | String | Create\_Item(): String  Modified\_Item(): String  Remove\_Item(): bool  AddProducts(): void |
| 2 | name | String |
| 3 | description | String |
| 4 | products | List<Product> |

Bảng 3.13: Mô tả chi tiết lớp Customer

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên lớp** | | | |
| Customer | | | |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Phương thức** |
| 1 | customerId | String | Create\_Item(): String  Modified\_Item(): String  Remove\_Item(): bool  AddCustomer(): void  UpdatePoints(): void |
| 2 | phoneNumber | String |
| 3 | name | String |
| 4 | address | String |
| 5 | points | int |
| 6 | createdAt | LocalDateTime | SetCreatedAt(): void |
| 7 | updatedAt | LocalDateTime |  |

Bảng 3.14: Mô tả chi tiết lớp Employee

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên lớp** | | | |
| Employee | | | |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Phương thức** |
| 1 | employeeId | String | Create\_Item(): String  Modified\_Item(): String  Remove\_Item(): bool  AddEmployee(): void  UpdateRole(): void  SetPasswordHash(): void |
| 2 | name | String |
| 3 | phone | String |  |
| 4 | Email | String |  |
| 5 | Role | Role(Enum) |  |
| 6 | passwordHash | String |  |

Bảng 3.15: Mô tả chi tiết lớp Order

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên lớp** | | | |
| Order | | | |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Phương thức** |
| 1 | orderId | String | Create\_Item(): String  Modified\_Item(): String  Remove\_Item(): bool  AddOrder(): void  UpdateOrder(): void  SetPaymentMethod(): void |
| 2 | customer | Customer |
| 3 | employee | Employee |
| 4 | orderDate | LocalDateTime |
| 5 | totalAmount | BigDecimal |
| 6 | paymentMethod | String |
| 7 | status | OrderStatus (Enum) | SetStatus(): void |
| 8 | createdAt | LocalDateTime | SetCreatedAt(): void |
| 9 | updatedAt | LocalDateTime | SetUpdatedAt(): void |

Bảng 3.16: Mô tả chi tiết lớp OrderDetail

|  |  |  |  |
| --- | --- | --- | --- |
| **n lớp** | | | |
| OrderDetail | | | |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Phương thức** |
| 1 | orderDetailId | String | Create\_Item(): String  Modified\_Item(): String  Remove\_Item(): bool  AddOrderDetail(): void |
| 2 | order | Order |
| 3 | product | Product |
| 4 | quantity | Integer |
| 5 | unitPrice | BigDecimal | UpdateOrderDetail(): void |
| 6 | discount | BigDecimal | SetDiscount(): void |

Bảng 3.17: Mô tả chi tiết lớp Product

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên lớp** | | | |
| Product | | | |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Phương thức** |
| 1 | productId | String | Create\_Item(): String  Modified\_Item(): String  Remove\_Item(): bool  AddProduct(): void  UpdateProduct(): void  SetPrice(): void |
| 2 | category | Category |
| 3 | supplier | Supplier |
| 4 | name | String |
| 5 | barcode | String |
| 6 | price | BigDecimal |
| 7 | stock | int | SetStock(): void |
| 8 | imageUrl | String | SetImageUrl(): void |
| 9 | createdAt | LocalDateTime | SetCreatedAt(): void |
| 10 | updatedAt | LocalDateTime | SetUpdatedAt(): void |

Bảng 3.18: Mô tả chi tiết lớp Promotion

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên lớp** | | | |
| Promotion | | | |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Phương thức** |
| 1 | id | Long | Create\_Item(): Long  Modified\_Item(): int  Remove\_Item(): bool  AddPromotion(): void  UpdatePromotion(): void |
| 2 | discountPercentage | Int |
| 3 | applyToPhoneNumber | Boolean |

Bảng 3.19: Mô tả chi tiết lớp Supplier

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên lớp** | | | |
| Supplier | | | |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Phương thức** |
| 1 | supplierId | String | Create\_Item(): String  Modified\_Item(): String  Remove\_Item(): bool  AddSupplier(): void  UpdateSupplier(): void  GetSuppliedProducts(): List<Product> |
| 2 | name | String |
| 3 | contactPerson | String |
| 4 | phone | String |
| 5 | address | String |
| 6 | products | List<Product> |

Bảng 3.20: Mô tả chi tiết lớp TempOrder

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên lớp** | | | |
| TempOrder | | | |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Phương thức** |
| 1 | id | String | Create\_Item(): String  Modified\_Item(): String  Remove\_Item(): bool |
| 2 | customerName | String |
| 3 | createdAt | LocalDateTime |
| 4 | orderDetails | List<TempOrderDetail> | AddTempOrder(): void |
|  |  |  | AddOrderDetail(): void |
|  |  |  | RemoveOrderDetail(): void |

Bảng 3.21: Mô tả chi tiết lớp TempOrderDetail

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên lớp** | | | |
| TempOrderDetail | | | |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Phương thức** |
| 1 | id | String | Create\_Item(): String  Modified\_Item(): String  Remove\_Item(): bool  AddDetailToTempOrder(): void |
| 2 | productId | String |
| 3 | productName | String |
| 4 | quantity | int |
| 5 | price | BigDecimal | UpdateDetail(): void |
| 6 | tempOrder | TempOrder | GetTempOrder(): TempOrder |

### Thiết kế Database

**a, Thiết kế chi tiết các bảng dữ liệu**

Bảng 3.22: Mô tả chi tiết bảng Category

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tên bảng** | | **Mô tả** | | | |
| Category | | Danh mục sản phẩm | | | |
| **STT** | **Tên thuộc tính** | **Mô tả** | **Kiểu dữ liệu** | **NULL** | **Ràng buộc** |
| 1 | category\_id | Mã danh mục | VARCHAR(36) | Not | PK |
| 2 | name | Tên danh mục | VARCHAR(100) | Yes |  |
| 3 | description | Mô tả | VARCHAR(255) | Yes |  |
| 4 | category\_id | Mã danh mục | VARCHAR(36) | Not | PK |

Bảng 3.23: Mô tả chi tiết bảng Customer

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tên bảng** | | **Mô tả** | | | |
| Customer | | Khách hàng | | | |
| **STT** | **Tên thuộc tính** | **Mô tả** | **Kiểu dữ liệu** | **NULL** | **Ràng buộc** |
| 1 | Customer\_id | Mã khách hàng | VARCHAR(36) | Not | PK |
| 2 | Phone\_number | Số điện thoại | VARCHAR(255) | Not | UNIQUE |
| 3 | Name | Tên khách hàng | VARCHAR(255) | Not |  |
| 4 | Address | Địa chỉ | VARCHAR(255) |  |  |
| 5 | Points | Điểm | Int | Not |  |
| 6 | Created\_at | Ngày tạo | VARBINARY(255) | Not |  |
| 7 | Updated\_at | Ngày cập nhật | VARBINARY(255) |  |  |

Bảng 3.24: Mô tả chi tiết bảng Employee

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tên bảng** | | **Mô tả** | | | |
| Employee | | Nhân viên | | | |
| **STT** | **Tên thuộc tính** | **Mô tả** | **Kiểu dữ liệu** | **NULL** | **Ràng buộc** |
| 1 | employee\_id | Mã nhân viên | VARCHAR(36) | Not | PK |
| 2 | name | Tên nhân viên | VARCHAR(100) | Not |  |
| 3 | phone | Số điện thoại | VARCHAR(20) | Yes |  |
| 4 | email | Email | VARCHAR(100) | Yes | UNIQUE |
| 5 | role | Vai trò | ENUM | Not |  |
| 6 | password\_hash | Mật khẩu mã hóa | VARCHAR(255) | Not |  |

Bảng 3.25: Mô tả chi tiết bảng Product

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tên bảng** | | **Mô tả** | | | |
| Product | | Sản phẩm | | | |
| **STT** | **Tên thuộc tính** | **Mô tả** | **Kiểu dữ liệu** | **NULL** | **Ràng buộc** |
| 1 | product\_id | Mã sản phẩm | VARCHAR(36) | Not | PK |
| 2 | category\_id | Mã danh mục | VARCHAR(36) | Not | FK |
| 3 | supplier\_id | Mã nhà cung cấp | VARCHAR(36) | Not | FK |
| 4 | name | Tên sản phẩm | VARCHAR(255) | Yes |  |
| 5 | barcode | Mã vạch | VARCHAR(255) | Yes |  |
| 6 | price | Giá bán | DECIMAL(38,2) | Yes |  |
| 7 | stock | Tồn kho | INT | Yes |  |
| 8 | image\_url | Ảnh sản phẩm | VARCHAR(255) | Yes |  |
| 9 | created\_at | Ngày tạo | VARBINARY(255) | Yes |  |
| 10 | updated\_at | Ngày cập nhật | VARBINARY(255) | Yes |  |

Bảng 3.26: Mô tả chi tiết bảng Supplier

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tên bảng** | | **Mô tả** | | | |
| Supplier | | Nhà cung cấp | | | |
| **STT** | **Tên thuộc tính** | **Mô tả** | **Kiểu dữ liệu** | **NULL** | **Ràng buộc** |
| 1 | supplier\_id | Mã nhà cung cấp | VARCHAR(36) | Not | PK |
| 2 | name | Tên nhà cung cấp | VARCHAR(100) | Yes |  |
| 3 | contact\_person | Người liên hệ | VARCHAR(100) | Yes |  |
| 4 | phone | SĐT nhà cung cấp | VARCHAR(20) | Yes |  |
| 5 | address | Địa chỉ | VARCHAR(255) | Yes |  |

Bảng 3.27: Mô tả chi tiết bảng Orders

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tên bảng** | | **Mô tả** | | | |
| Orders | | Đơn hàng | | | |
| **STT** | **Tên thuộc tính** | **Mô tả** | **Kiểu dữ liệu** | **NULL** | **Ràng buộc** |
| 1 | order\_id | Mã đơn hàng | VARCHAR(36) | Not | PK |
| 2 | customer\_id | Mã khách hàng | VARCHAR(36) | Yes | FK |
| 3 | employee\_id | Mã nhân viên | VARCHAR(36) | Yes | FK |
| 4 | order\_date | Ngày đặt | DATETIME | Yes |  |
| 5 | total\_amount | Tổng tiền | DECIMAL(10,2) | Not |  |
| 6 | payment\_method | Phương thức TT | VARCHAR(50) | Not |  |
| 7 | status | Trạng thái | ENUM | Not |  |
| 8 | created\_at | Ngày tạo | TIMESTAMP | Yes |  |
| 9 | updated\_at | Ngày cập nhật | TIMESTAMP | Yes |  |
| 10 | discount\_percent | Giảm giá % | DECIMAL(5,2) | Yes |  |
| 11 | promotion\_id | Mã khuyến mãi | VARCHAR(36) | Yes | FK |

Bảng 3.28: Mô tả chi tiết bảng Order\_detail

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tên bảng** | | **Mô tả** | | | |
| Order\_detail | | Chi tiết đơn hàng | | | |
| **STT** | **Tên thuộc tính** | **Mô tả** | **Kiểu dữ liệu** | **NULL** | **Ràng buộc** |
| 1 | order\_detail\_id | Mã chi tiết đơn hàng | VARCHAR(36) | Not | PK |
| 2 | order\_id | Mã đơn hàng | VARCHAR(36) | Not | FK |
| 3 | product\_id | Mã sản phẩm | VARCHAR(36) | Not | FK |
| 4 | quantity | Số lượng | INT | Not |  |
| 5 | unit\_price | Đơn giá | DECIMAL(10,2) | Not |  |
| 6 | discount | Giảm giá sản phẩm | DECIMAL(10,2) | Not |  |

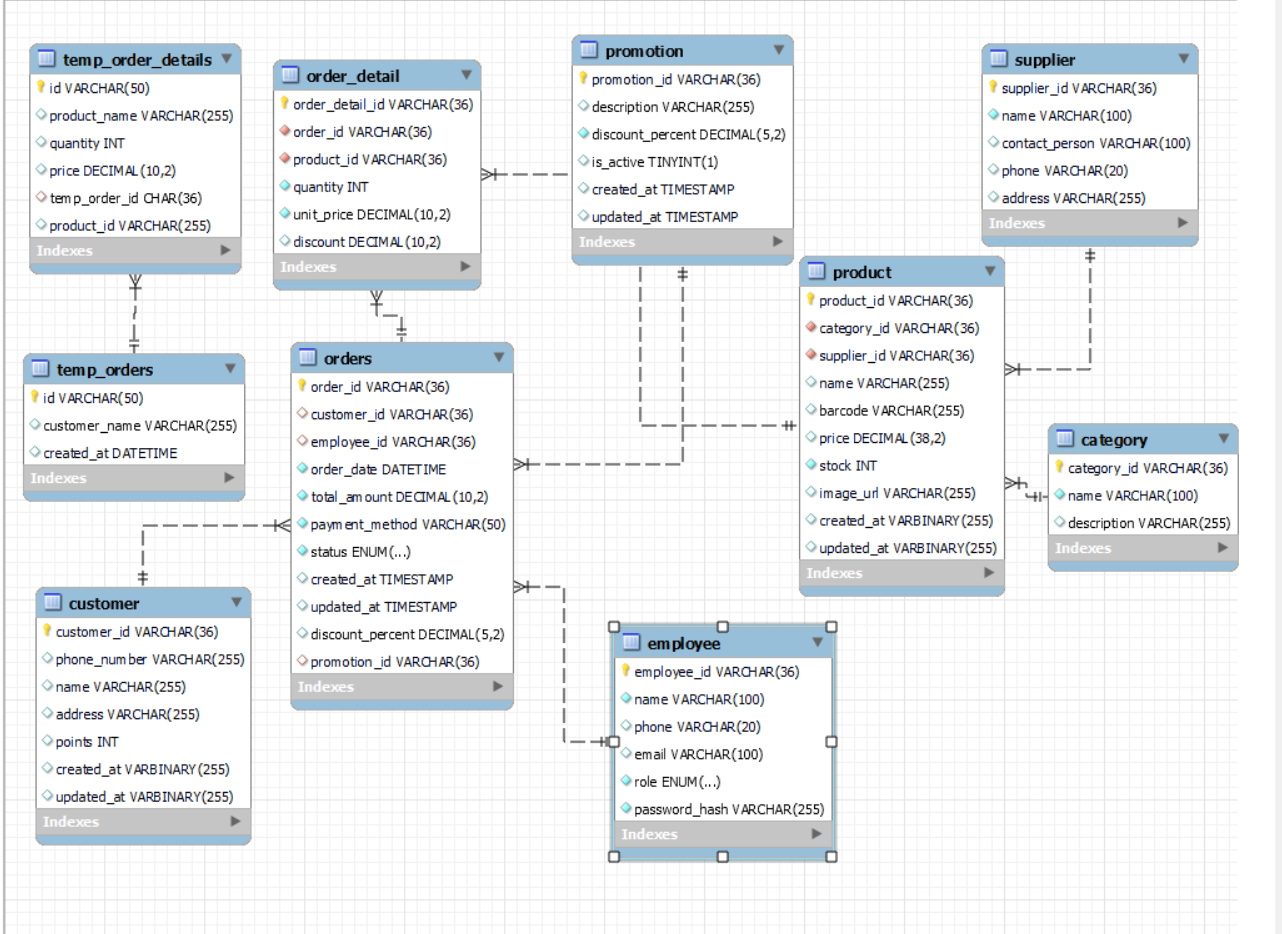
Bảng 3.29: Mô tả chi tiết bảng Temp\_orders

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tên bảng** | | **Mô tả** | | | |
| Temp\_orders | | Đơn hàng tạm thời | | | |
| **STT** | **Tên thuộc tính** | **Mô tả** | **Kiểu dữ liệu** | **NULL** | **Ràng buộc** |
| 1 | id | Mã đơn tạm | VARCHAR(50) | Not | PK |
| 2 | customer\_name | Tên khách hàng | VARCHAR(255) | Yes |  |
| 3 | created\_at | Ngày tạo | DATETIME | Yes |  |

Bảng 3.30: Mô tả chi tiết bảng Temp\_order\_details

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tên bảng** | | **Mô tả** | | | |
| Temp\_order\_details | | Chi tiết đơn tạm thời | | | |
| **STT** | **Tên thuộc tính** | **Mô tả** | **Kiểu dữ liệu** | **NULL** | **Ràng buộc** |
| 1 | id | Mã chi tiết | VARCHAR(50) | Not | PK |
| 2 | product\_name | Tên sản phẩm | VARCHAR(255) | Yes |  |
| 3 | quantity | Số lượng | INT | Not |  |
| 4 | price | Giá | DECIMAL(10,2) | Not |  |
| 5 | temp\_order\_id | Mã đơn tạm | VARCHAR(36) | Yes | FK |
| 6 | product\_id | Mã sản phẩm (gốc) | VARCHAR(255) | Yes |  |

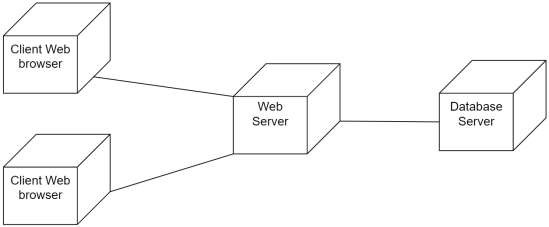
**b, Biểu đồ ERD**



Hình 3.26: Biểu đồ ERD

### Thiết kế kiến trúc hệ thống

* Biểu đồ triển khai

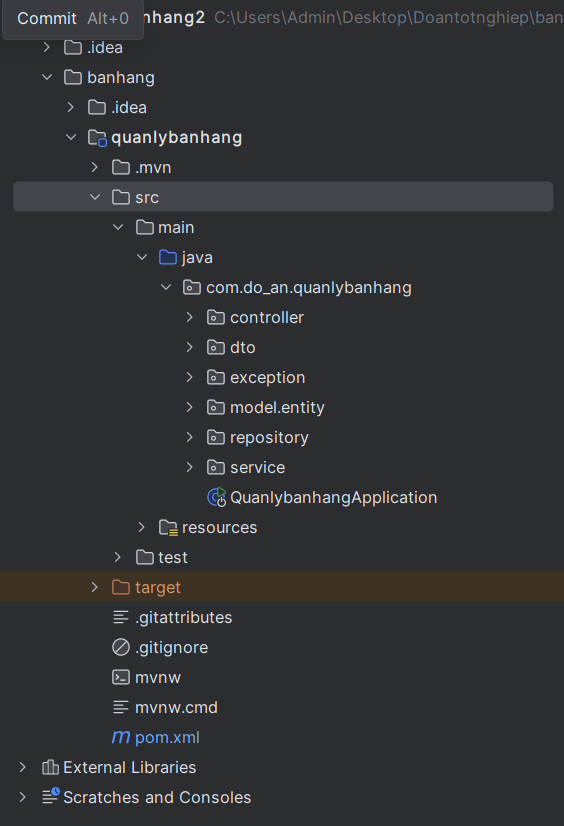
****

Hình 3.27: Biểu đồ triển khai

# CHƯƠNG IV: XÂY DỰNG HỆ THỐNG

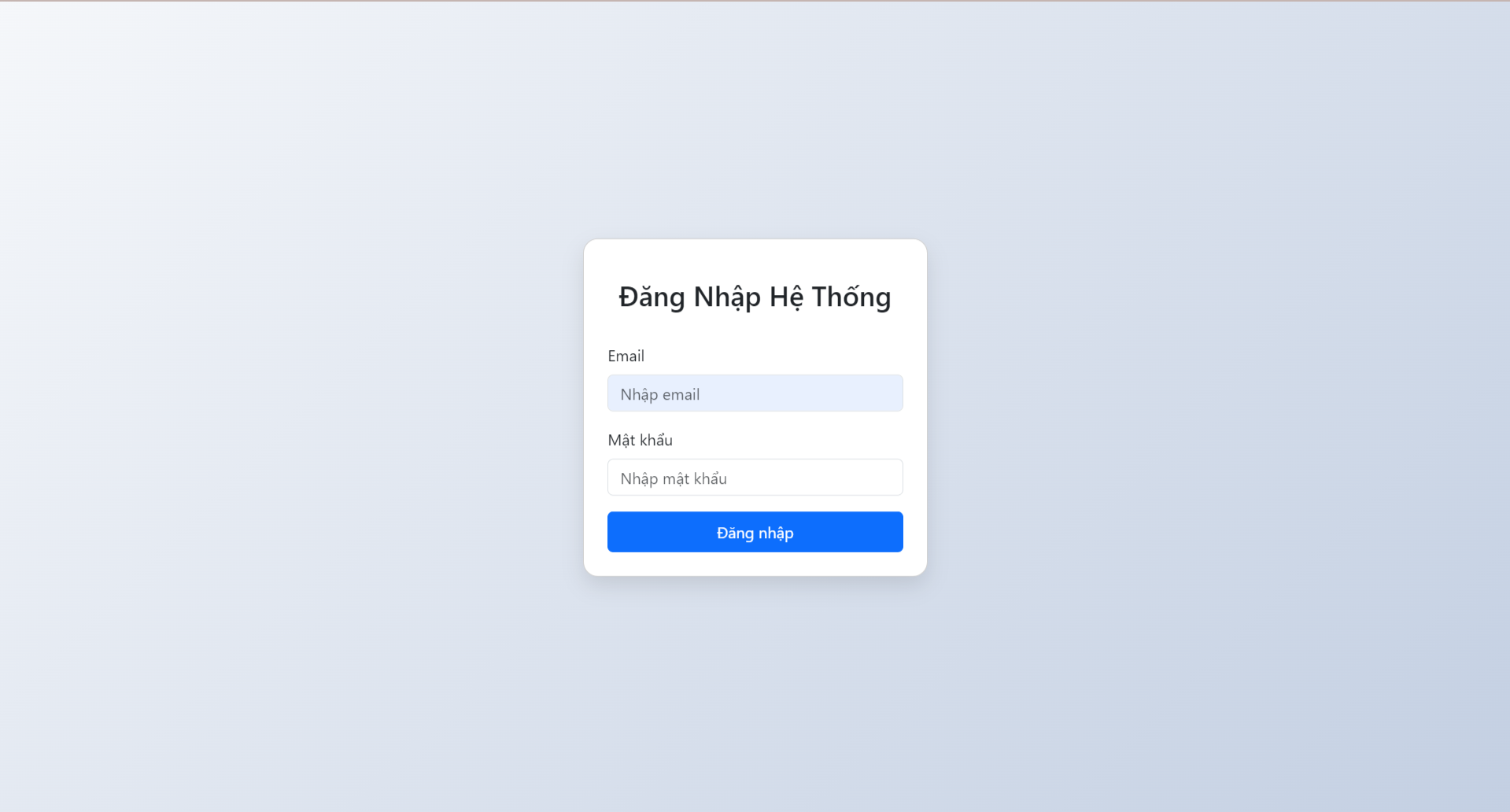
## Lập trình xây dựng hệ thống

### Cấu trúc mã nguồn

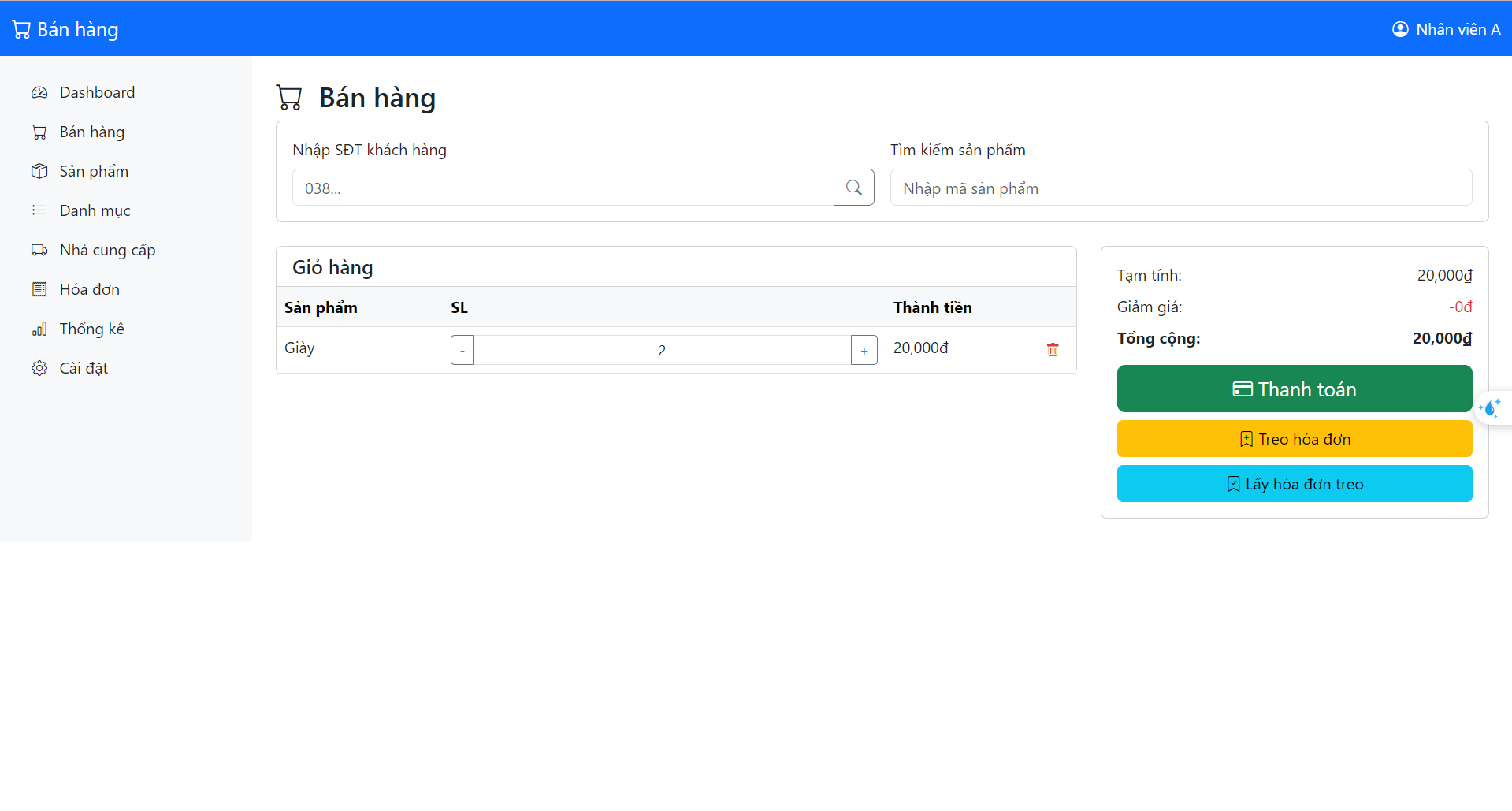
******

Hình 4.1: Folder tổng quát project

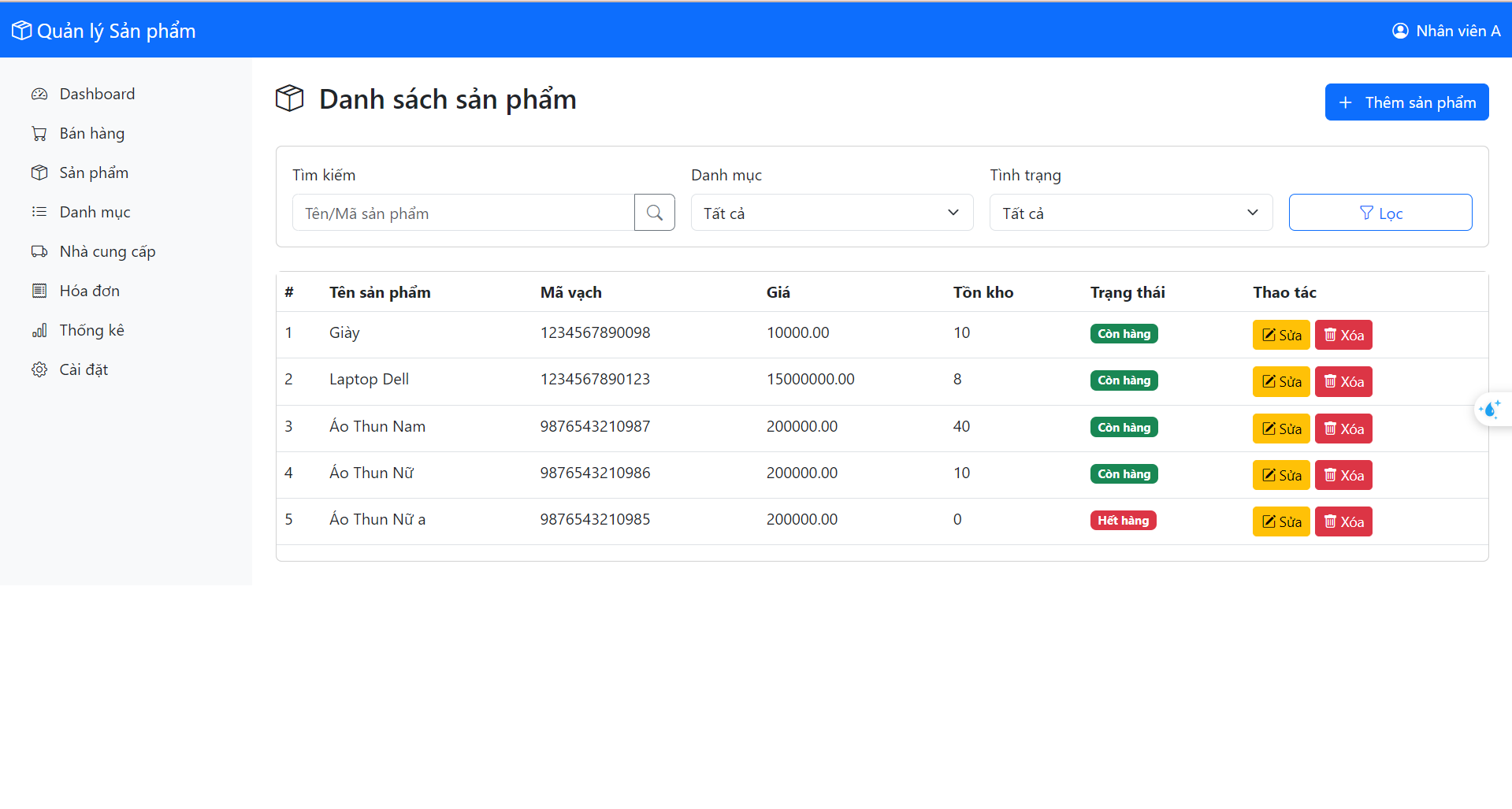
### Hình ảnh giao diện phần mềm thực tế xây dựng

******

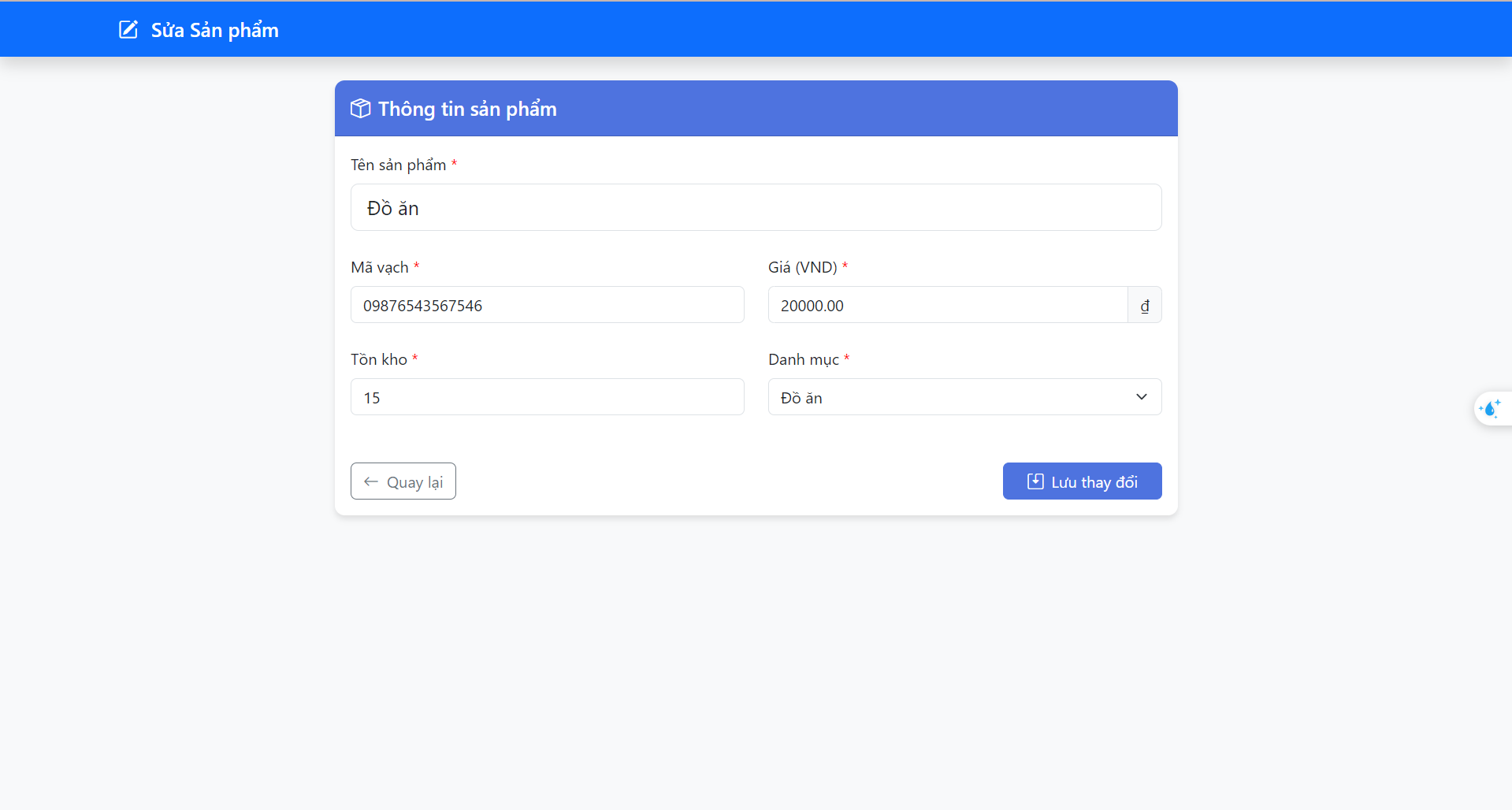
Hình 4.2: Giao diện màn hình đăng nhập thực tế



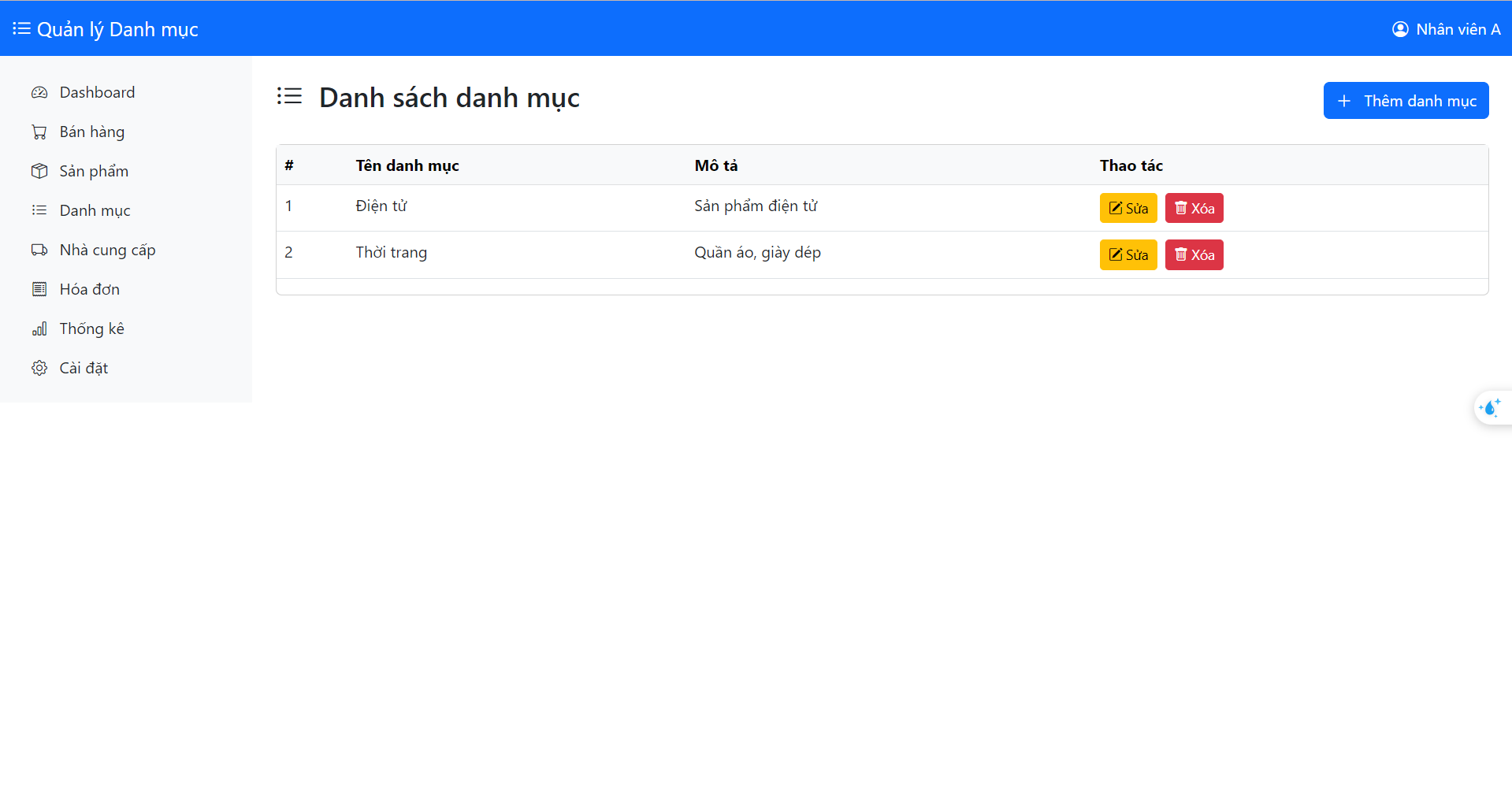
Hình 4.3: Giao diện màn hình bán hàng

**

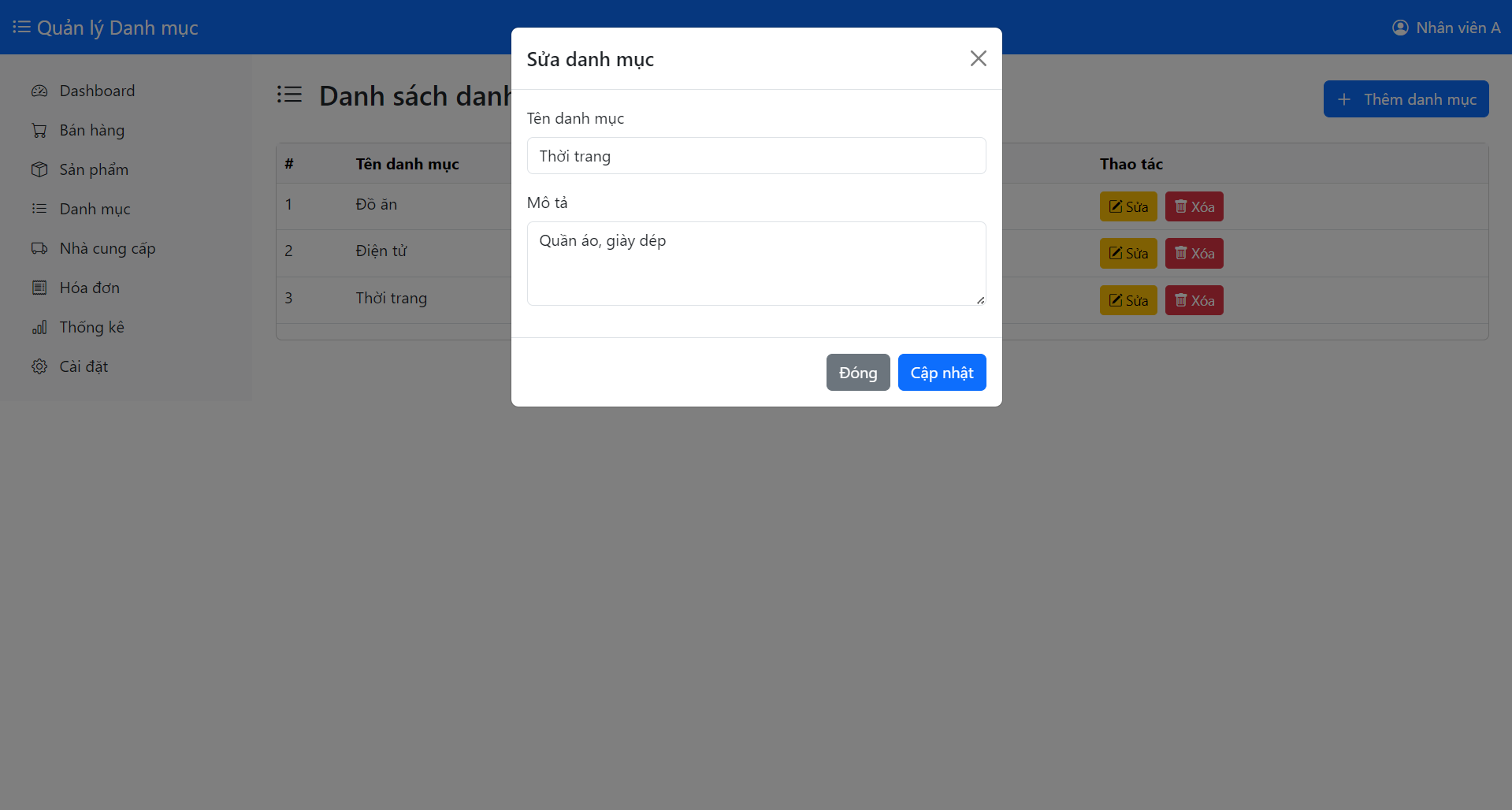
Hình 4.4: Giao diện màn hình quản lý sản phẩm



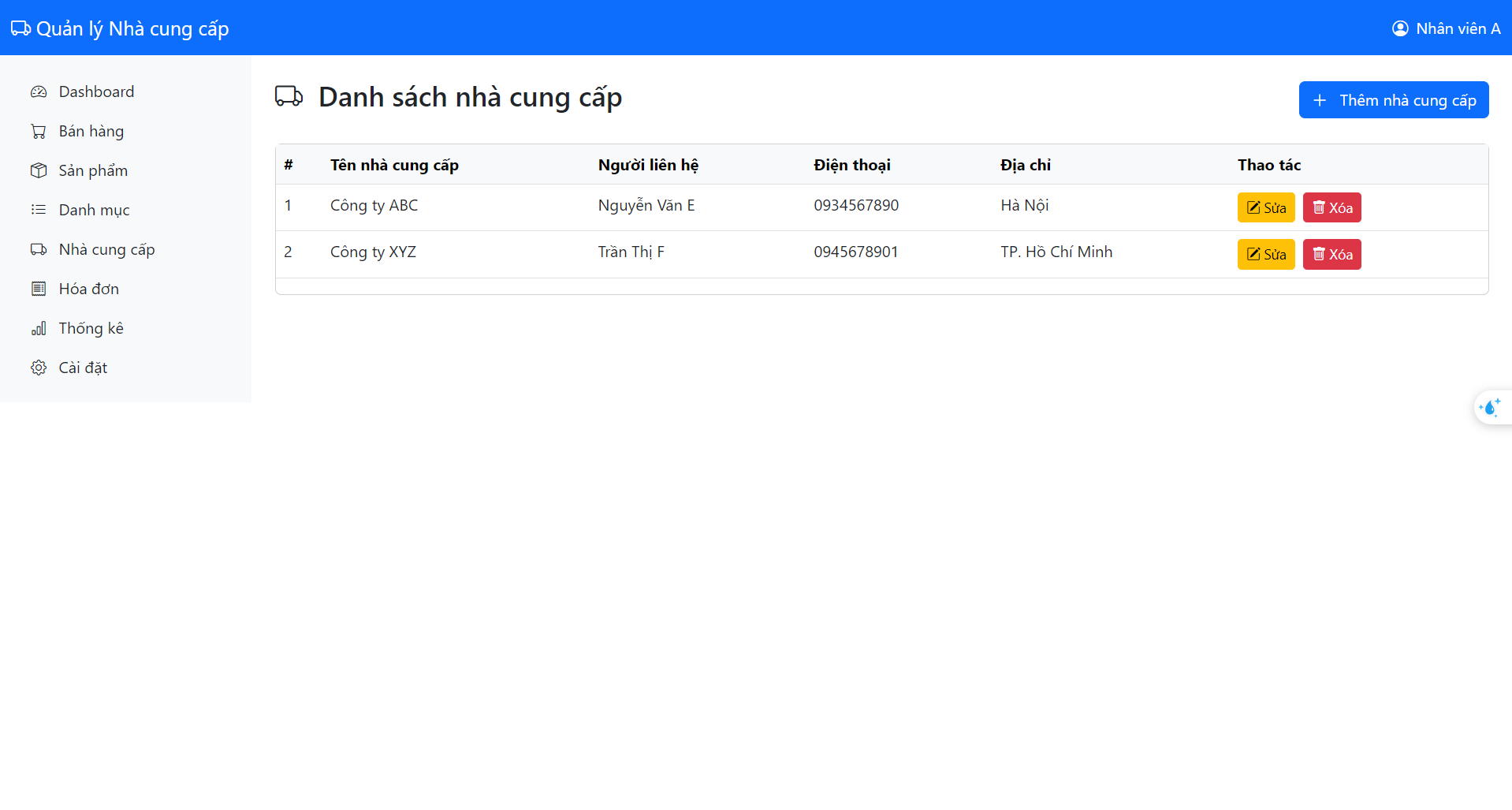
Hình 4.5: Giao diện màn hình sửa sản phẩm

**

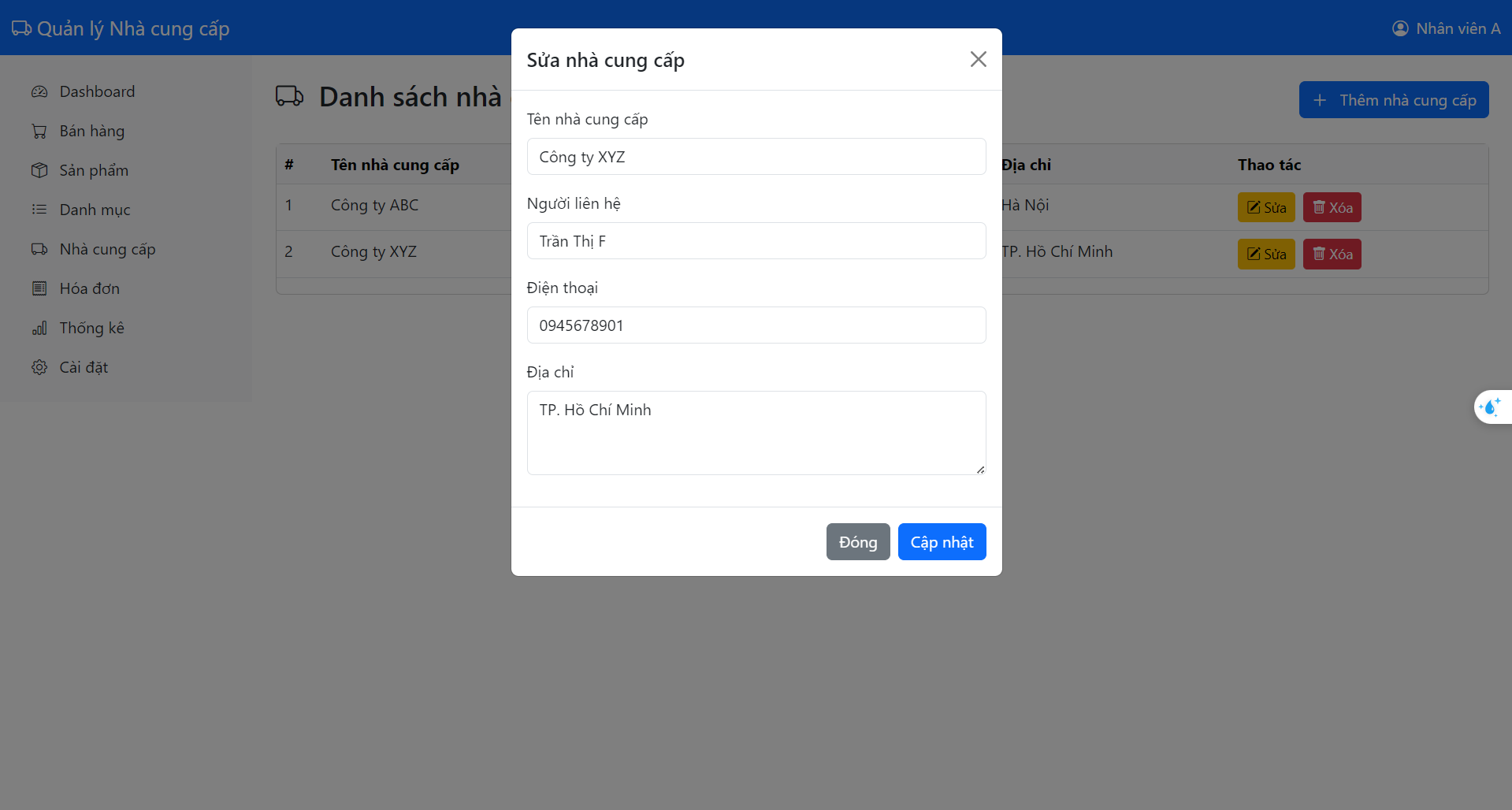
Hình 4.6: Giao diện màn hình quản lý danh mục



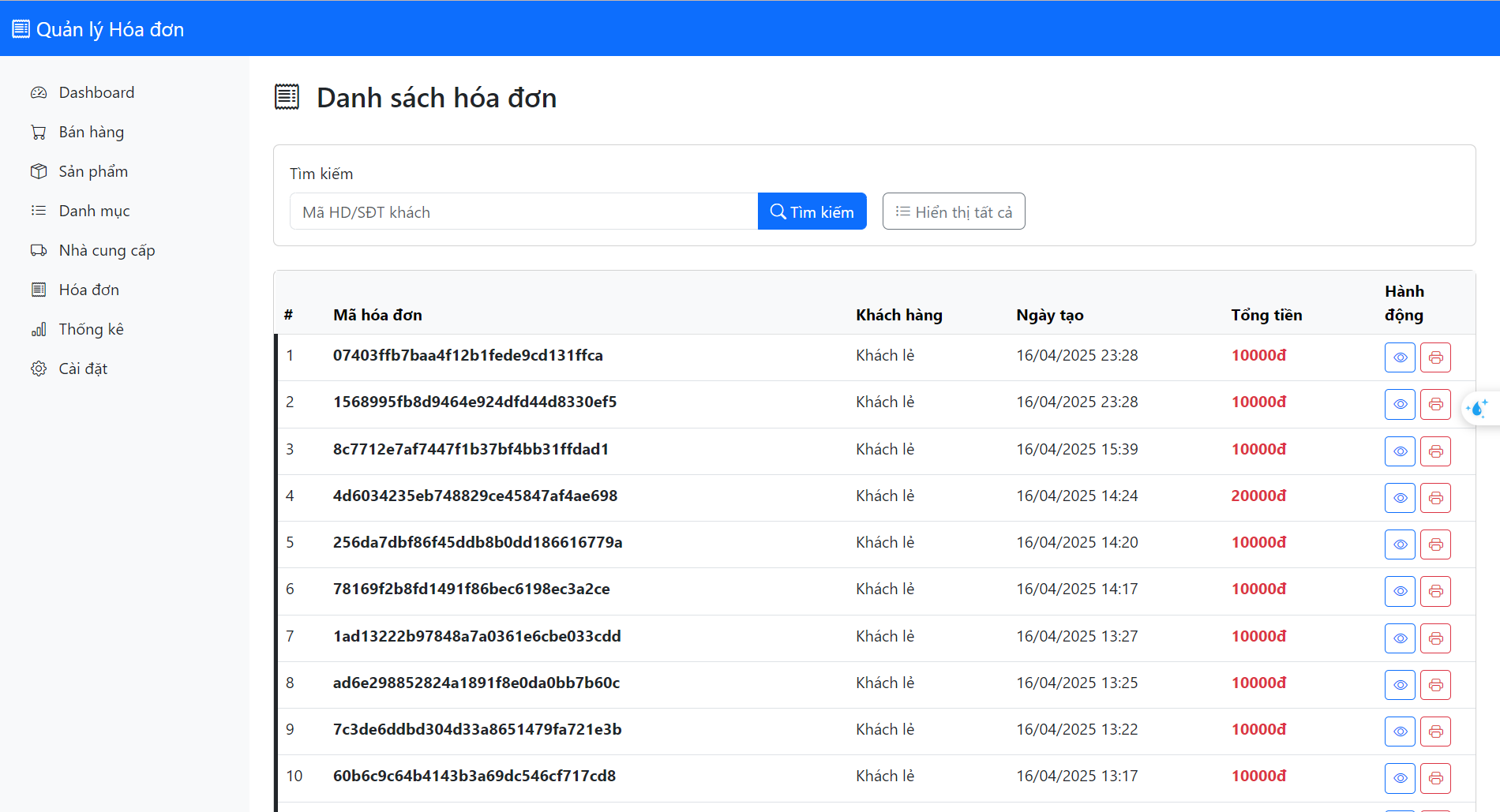
Hình 4.7: Giao diện màn hình sửa danh mục

**

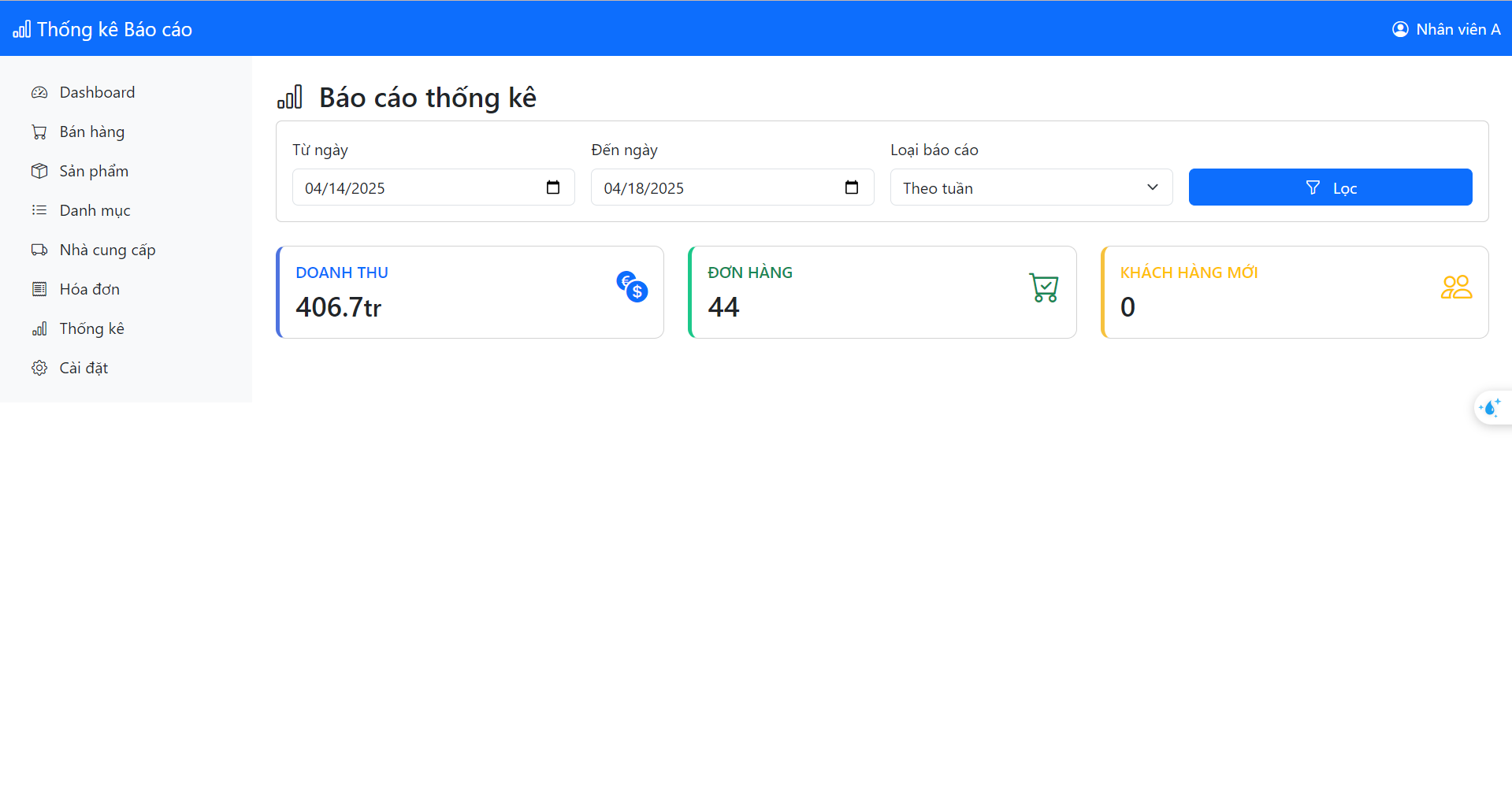
Hình 4.8: Giao diện màn hình quản lý nhà cung cấp



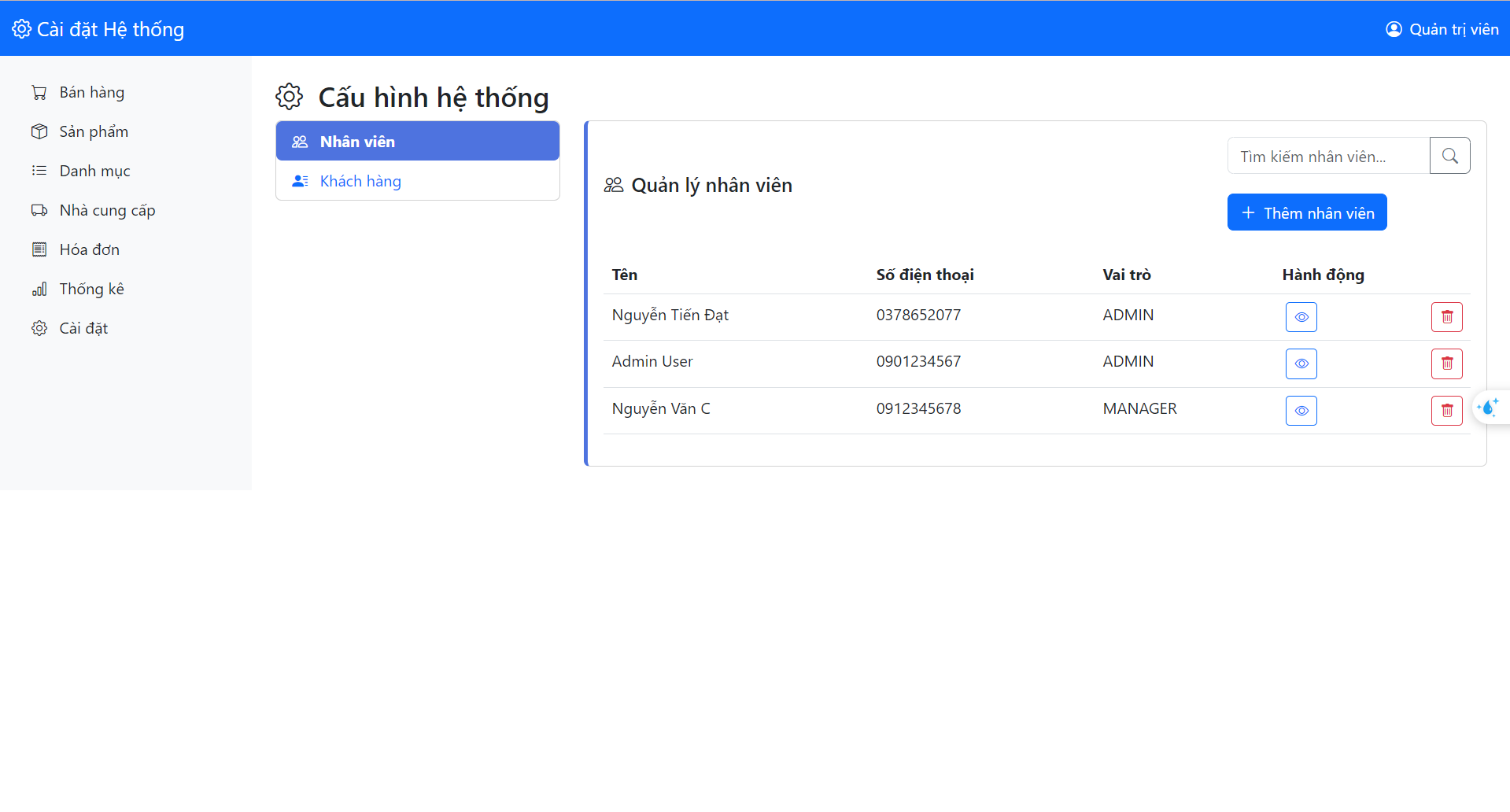
Hình 4.9: Giao diện màn hình sửa nhà cung cấp

**

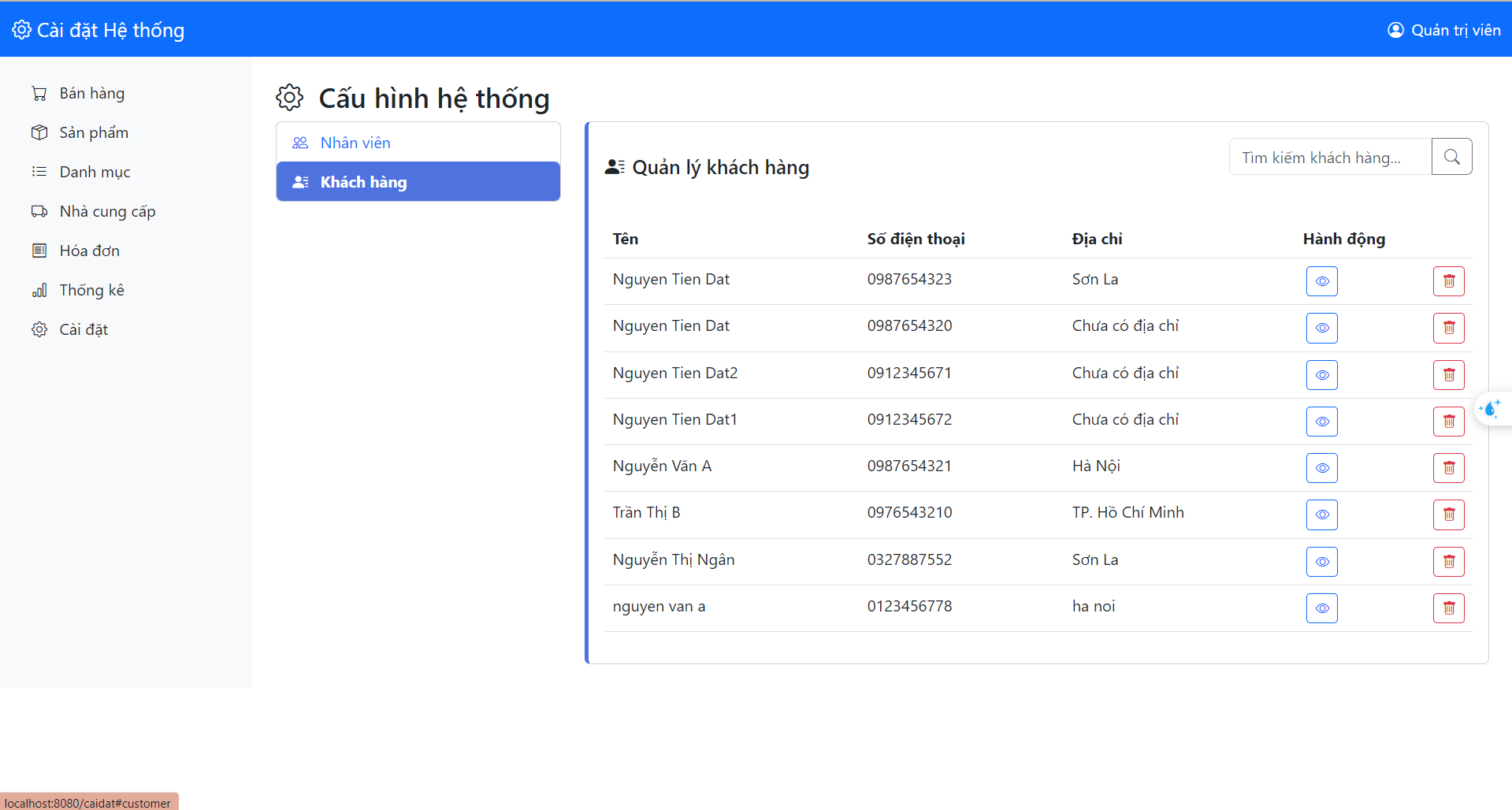
Hình 4.10: Giao diện màn hình quản lý hóa đơn



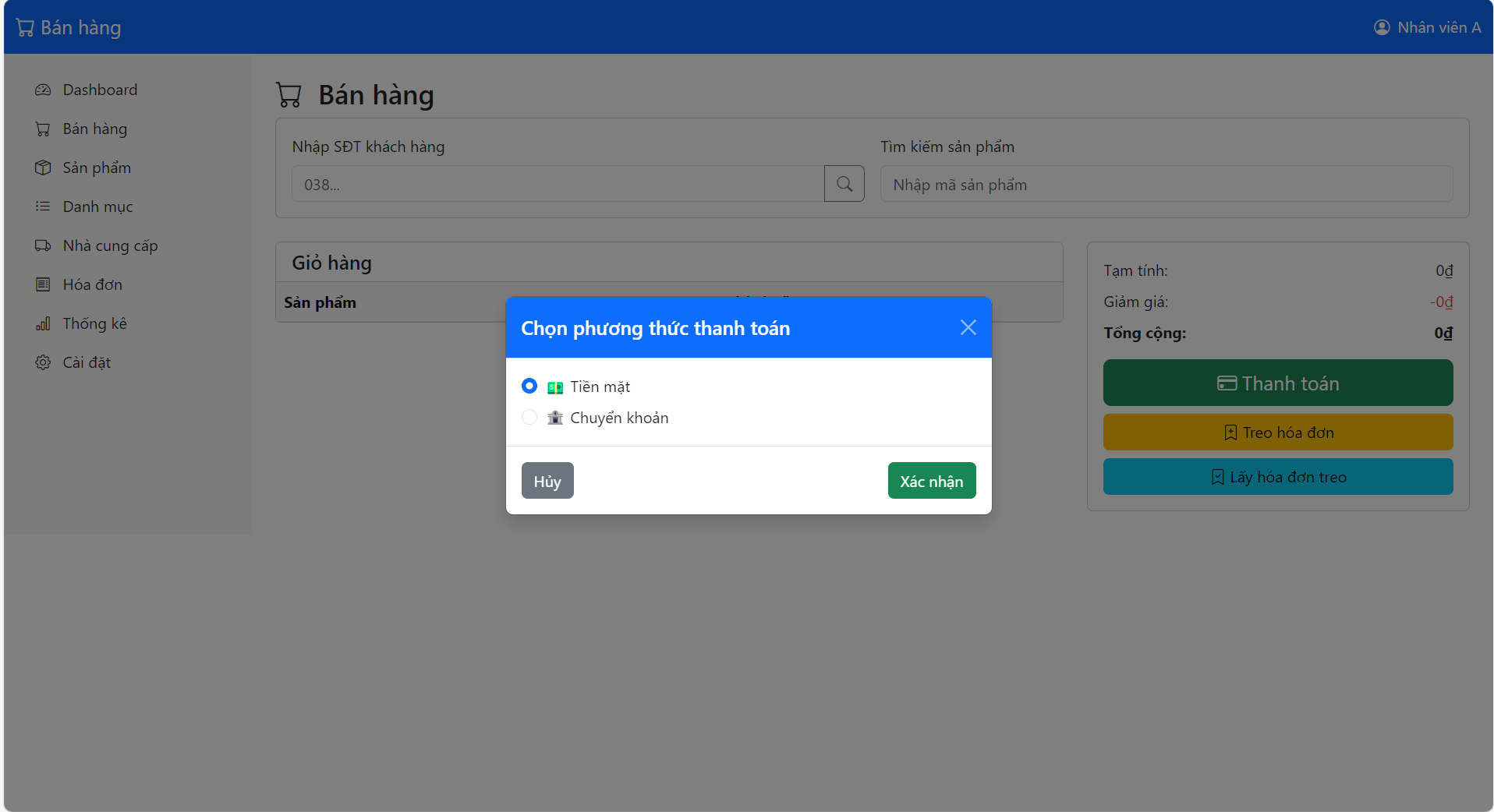
Hình 4.11: Giao diện màn hình thống kê

**

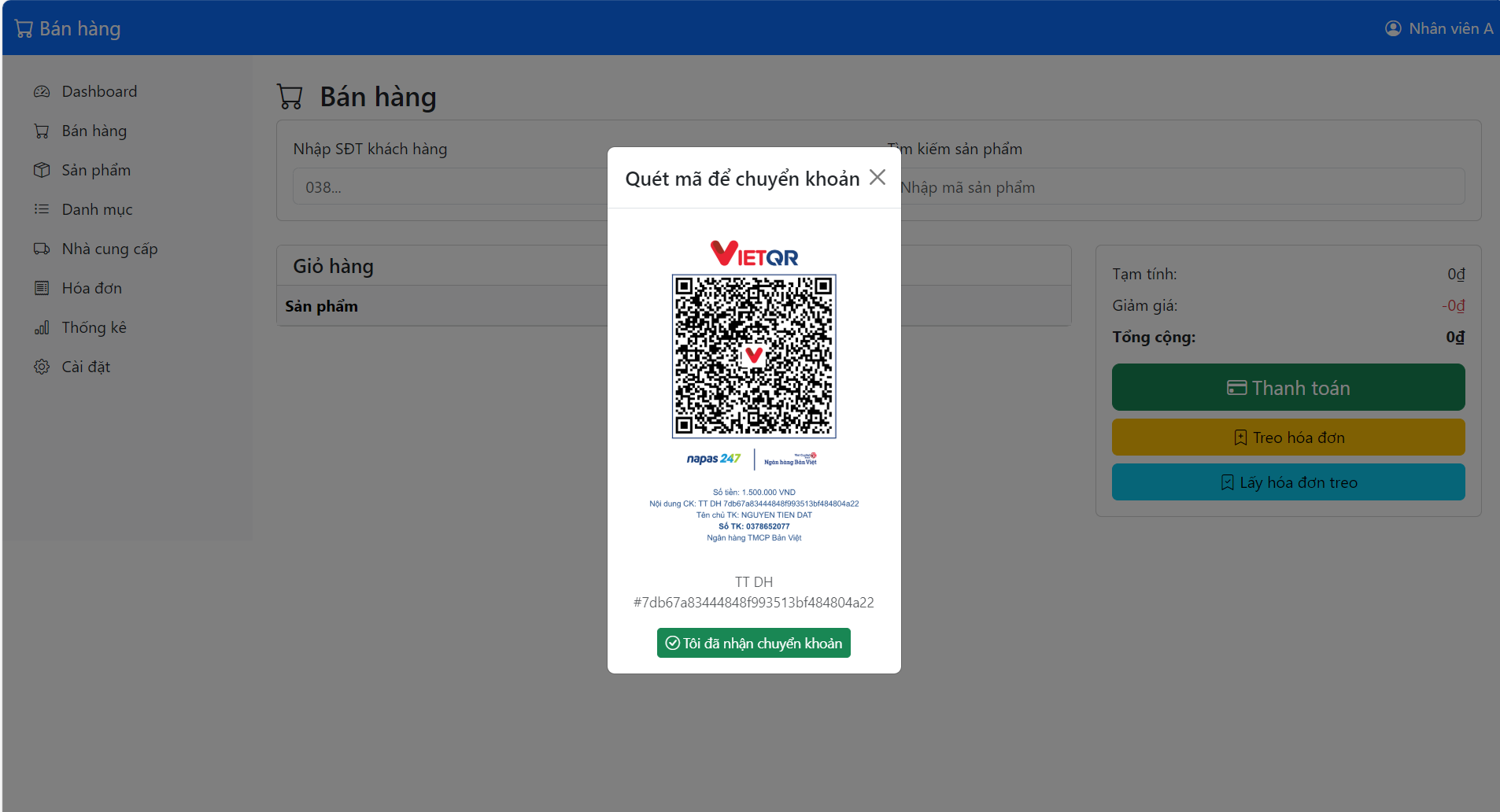
Hình 4.12: Giao diện màn hình danh sách nhân viên



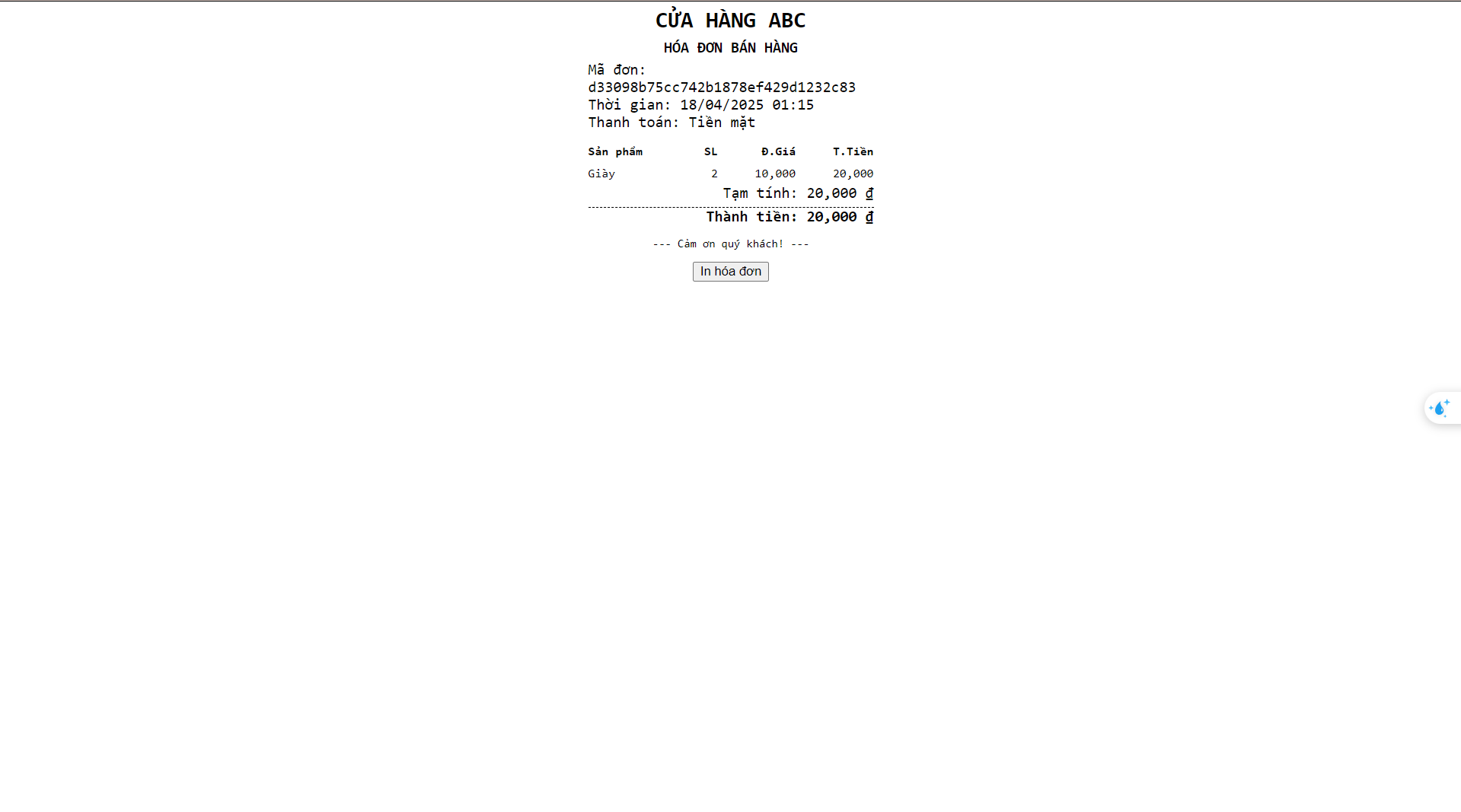
Hình 4.13: Giao diện màn hình danh sách khách hàng



Hình 4.14: Giao diện màn hình phương thức thanh toán

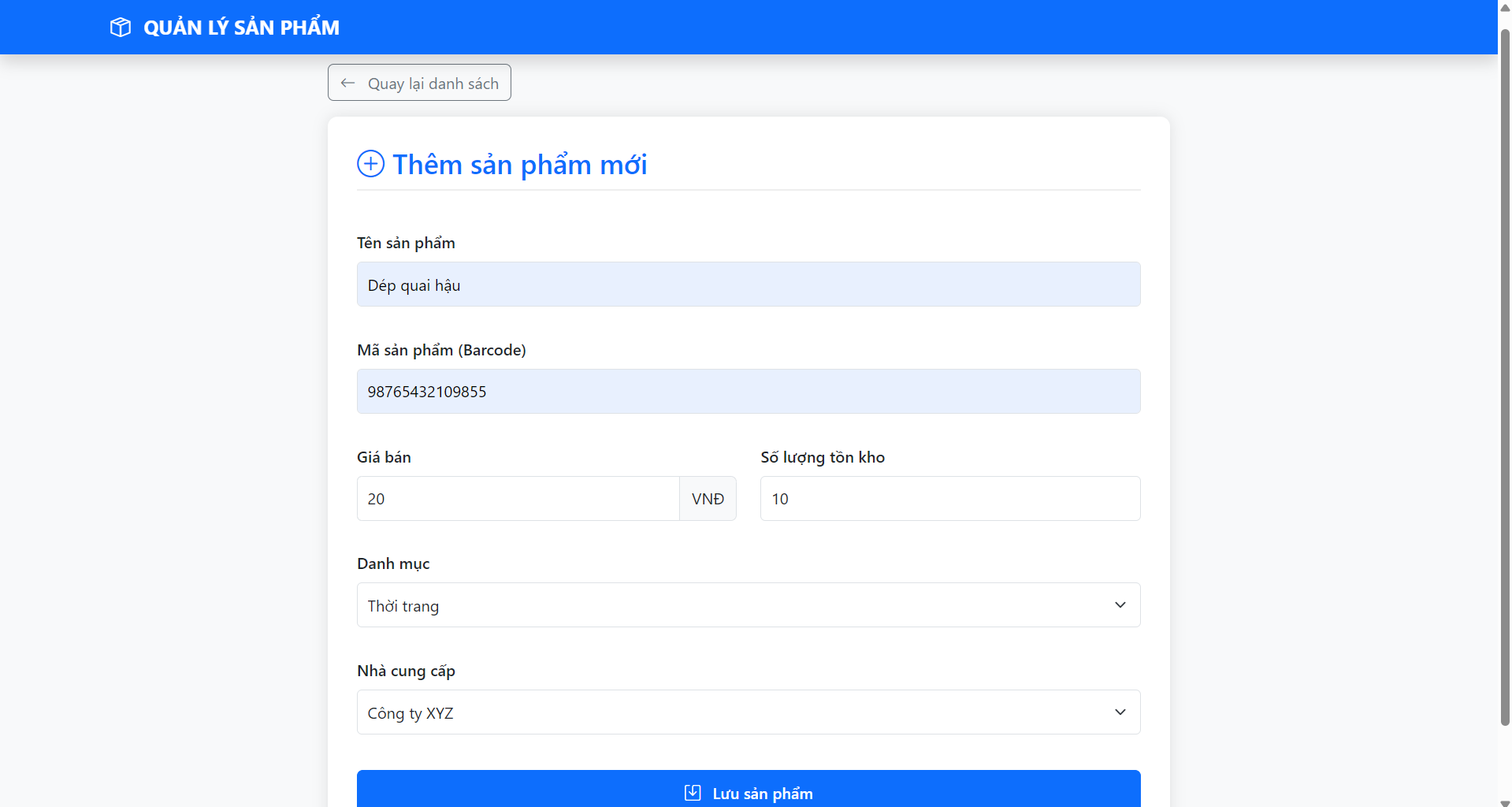


Hình 4.15: Giao diện màn hình thanh toán chuyển khoản

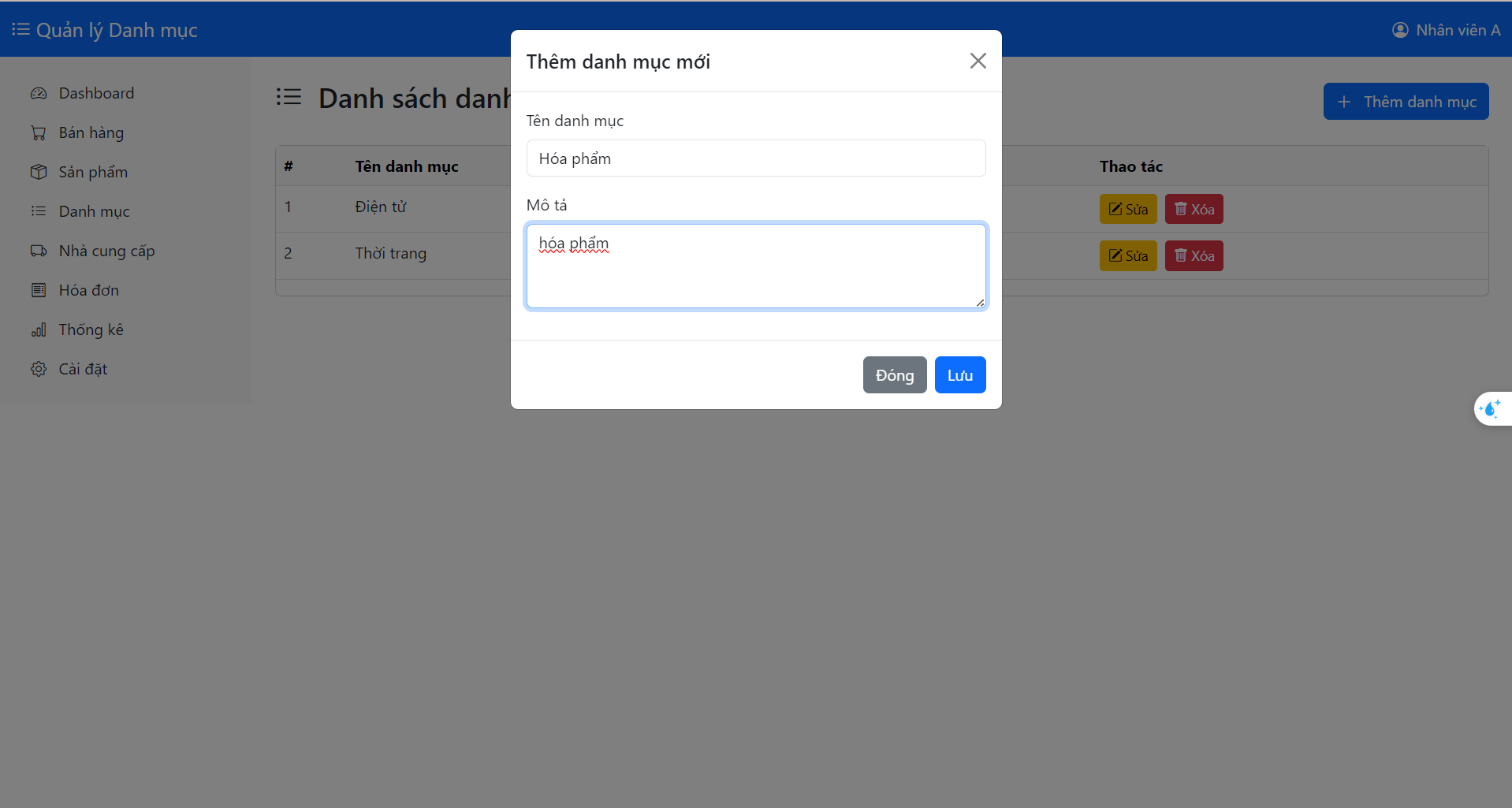
**

Hình 4.16: Giao diện màn hình hóa đơn

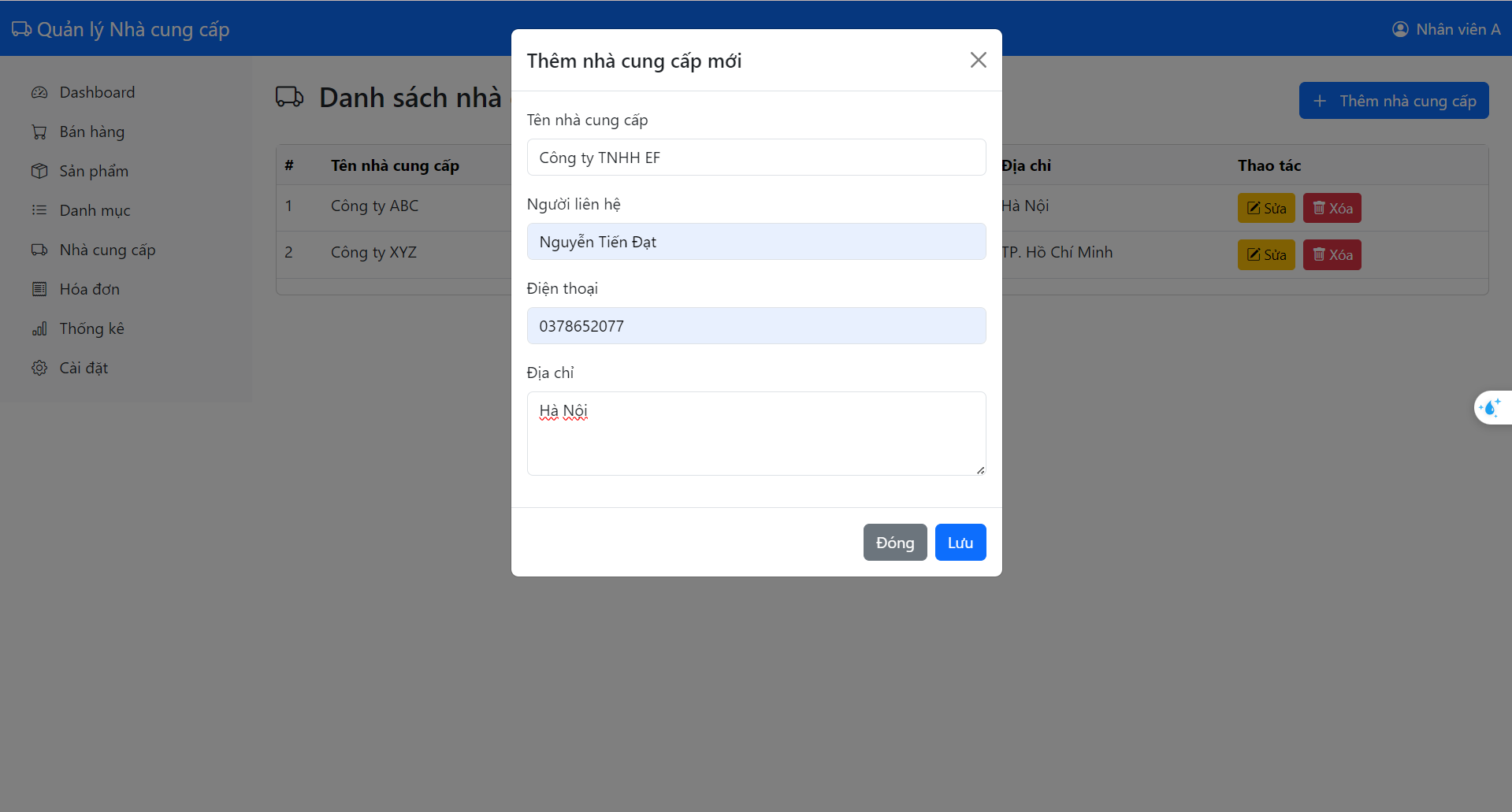
### Khởi tạo dữ liệu ban đầu cho hệ thống

****

Hình 4.17: Thêm sản phẩm mới trong hệ thống

**

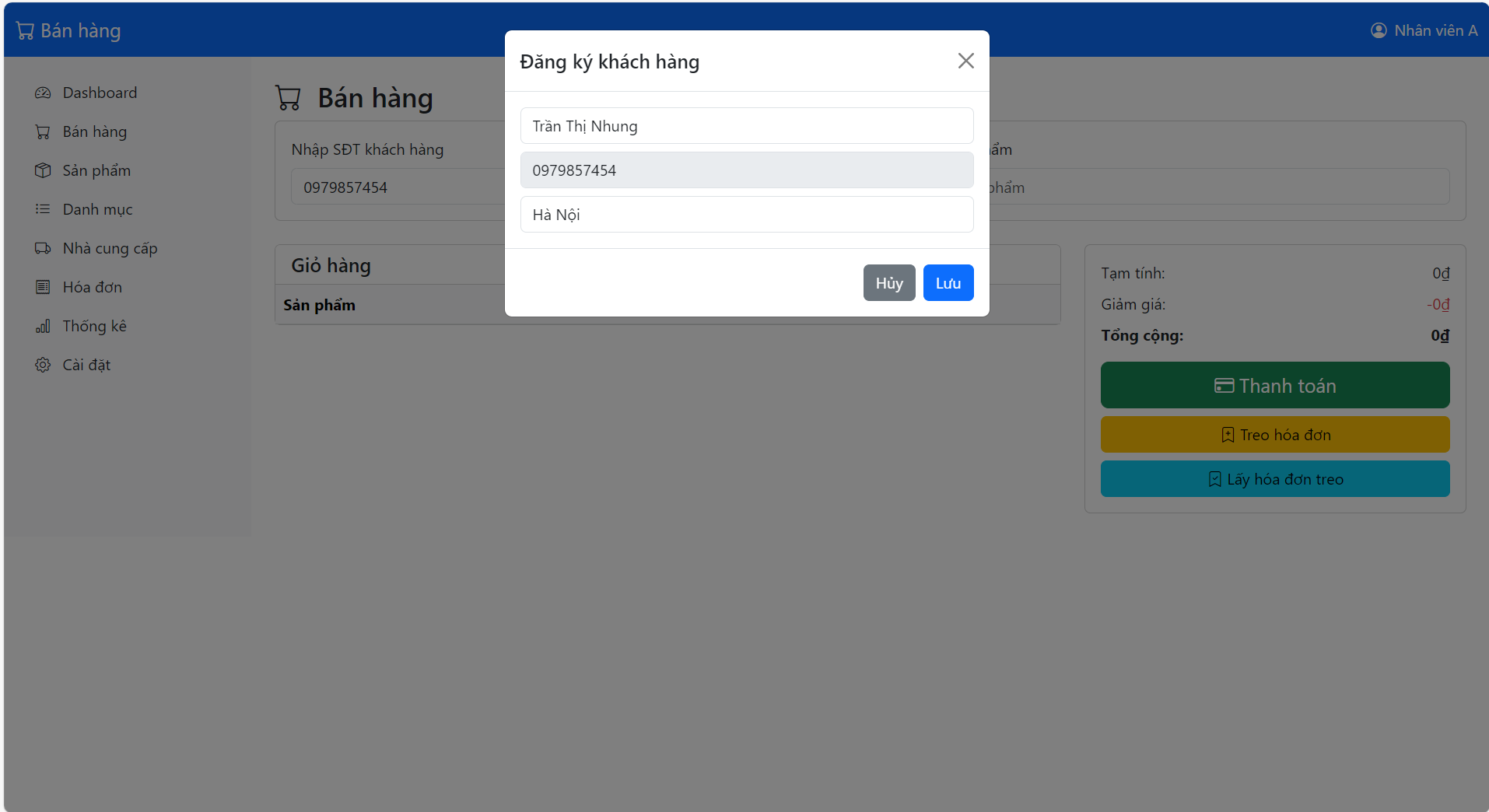
Hình 4.18: Thêm danh mục trong hệ thống



Hình 4.19: Thêm nhà cung cấp mới trong hệ thống



Hình 4.20: Thêm nhân viên mới



Hình 4.21: Thêm số điện thoại khách hàng mới

### Kiểm thử và đánh giá chương trình

Bảng 4.1: Danh sách test case kiểm thử đánh giá hệ thống

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Test case** | **Kết quả** | **Đánh giá** |
| 1 | Đăng nhập các tài khoản có vai trò khác nhau | * Đăng nhập thành công hệ thống * Hiển thị đúng giao diện trang chủ của   từng vai trò | Đạt |
| 2 | Phân vai trò tài khoản là nhân viên | * Phân vai trò nhân viên thành công | Đạt |
| 3 | Phân vai trò tài khoản là quản lý | - Phân vai trò quản lý thành công | Đạt |
| 4 | Tìm kiếm sản phẩm theo mã barcode | - Tìm kiếm sản phẩm thành công | Đạt |
| 5 | Treo đơn hàng chưa thanh toán | - Treo đơn hàng chưa thanh toán thành công | Đạt |
| 6 | Lấy đơn hàng đang treo | - Lấy đơn hàng đang treo thành công | Đạt |
| 7 | Nhập số điện thoại khách hàng để giảm giá | - Nhập số điện thoại khách hàng để giảm giá thành công | Đạt |
| 8 | Thanh toán bằng tiền mặt | - Thanh toán bằng tiền mặt thành công | Đạt |
| 9 | Thanh toán bằng chuyển khoản | - Thanh toán bằng chuyển khoản thành công | Đạt |
| 10 | Hiển thị danh sách sản phẩm | - Hiển thị danh sách sản phẩm thành công | Đạt |
| 11 | Lọc sản phẩm theo tiêu chí | - Lọc sản phẩm theo tiêu chí thành công | Đạt |
| 12 | Thêm sản phẩm mới | - Thêm sản phẩm mới thành công | Đạt |
| 13 | Sửa thông tin sản phẩm | - Sửa thông tin sản phẩm thành công | Đạt |
| 14 | Xóa sản phẩm | - Xóa sản phẩm thành công | Đạt |
| 15 | Hiển thị danh sách danh mục | - Hiển thị danh sách danh mục thành công | Đạt |
| 16 | Thêm danh mục mới | - Thêm danh mục mới thành công | Đạt |
| 17 | Sửa thông tin danh mục | Sửa thông tin danh mục thành công | Đạt |
| 18 | Xóa danh mục | - Xóa danh mục thành công | Đạt |
| 19 | Hiện thị danh sách nhà cung cấp | - Hiện thị danh sách nhà cung cấp thành công | Đạt |
| 20 | Thêm nhà cung cấp mới | - Thêm nhà cung cấp mới thành công | Đạt |
| 21 | Sửa thông tin nhà cung cấp | - Sửa thông tin nhà cung cấp thành công | Đạt |
| 22 | Xóa nhà cung cấp | - Xóa nhà cung cấp thành công | Đạt |
| 23 | Hiển thị danh sách hóa đơn | - Hiển thị danh sách hóa đơn thành công | Đạt |
| 24 | Tìm kiếm hóa đơn theo mã hóa đơn | - Tìm kiếm hóa đơn theo mã hóa đơn thành công | Đạt |
| 25 | Xem chi tiết đơn hàng | - Xem chi tiết đơn hàng thành công | Đạt |
| 26 | Xóa hóa đơn | - Xóa hóa đơn thành công | Đạt |
| 27 | In hóa đơn | - In hóa đơn thành công | Đạt |
| 28 | Thống kê theo ngày | - Thống kê theo ngày thành công | Đạt |
| 29 | Hiển thị danh sách nhân viên | - Hiển thị danh sách nhân viên thành công | Đạt |
| 30 | Tìm kiếm nhân viên theo tên | - Tìm kiếm nhân viên theo tên thành công | Đạt |
| 31 | Thêm nhân viên mới | - Thêm nhân viên mới thành công | Đạt |
| 32 | Hiển thị danh sách khách hàng | - Hiển thị danh sách khách hàng thành công | Đạt |
| 33 | Tìm kiếm khách hàng theo số điện thoại | - Tìm kiếm khách hàng theo số điện thoại thành công | Đạt |
| 34 | Hiển thị chi tiết khách hàng khi nhấn vào xem | - Hiển thị chi tiết khách hàng khi nhấn vào xem thành công | Đạt |

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## Kết quả đạt được

* + Đồ án đã thực hiện được nhiệm vụ đề ra và đạt được các kết quả sau:

+ Đã mô tả chi tiết quy trình nghiệp vụ của hệ thống quản lý bán hàng, từ khâu bán hàng, thanh toán, in hóa đơn đến quản lý sản phẩm, khách hàng, nhân viên, thống kê doanh thu, và cấu hình hệ thống. Qua đó xác định rõ yêu cầu và mục tiêu của hệ thống.

+ Ứng dụng phương pháp phân tích và thiết kế hướng đối tượng để xây dựng mô hình hệ thống. Các sơ đồ Use Case, ERD và các lớp đã được thiết kế đầy đủ, rõ ràng, đảm bảo chất lượng và khả năng mở rộng, thuận lợi cho việc triển khai thành phần chức năng.

+ Hệ thống được xây dựng trên nền tảng **Spring Boot**, sử dụng cơ sở dữ liệu **MySQL** và công nghệ giao diện **Thymeleaf**, kết hợp các thư viện hỗ trợ như **ZXing** để tạo mã QR. Giao diện người dùng được thiết kế trực quan, dễ sử dụng.

+ Hệ thống đã triển khai các chức năng chính như: tạo đơn hàng, thanh toán qua mã **VietQR**, quản lý hóa đơn, in hóa đơn, thống kê doanh thu theo thời gian, quản lý khuyến mãi theo điều kiện khách hàng có số điện thoại, và hệ thống cấu hình đa tab (Cửa hàng, Khuyến mãi, Nhân viên…)

+ Hệ thống đã được thử nghiệm với dữ liệu thực tế và cho kết quả tốt. Các chức năng hoạt động ổn định, đảm bảo tính chính xác và hiệu quả trong quá trình xử lý dữ liệu.

## Hạn chế

* + Do thời gian thực hiện có hạn nên một số chức năng vẫn chưa được hoàn thiện đầy đủ hoặc chỉ dừng lại ở mức cơ bản, ví dụ như phần cấu hình máy in, sao lưu dữ liệu, phân quyền chi tiết theo vai trò chưa được phát triển chuyên sâu.
  + Một số yếu tố về an toàn bảo mật hệ thống chưa thực sự tốt.
  + Một số giao diện người dùng còn đơn giản, chưa được tối ưu về mặt trải nghiệm (UX/UI), chưa có sự đồng bộ cao giữa các thành phần giao diện.
  + Tính năng xử lý khuyến mãi, lọc nâng cao, phân tích dữ liệu (BI) và các báo cáo tổng hợp chuyên sâu vẫn chưa được triển khai do giới hạn về thời gian.
  + Tính năng xử lý khuyến mãi, lọc nâng cao, phân tích dữ liệu (BI) và các báo cáo tổng hợp chuyên sâu vẫn chưa được triển khai do giới hạn về thời gian.

## Hướng phát triển

* + Hoàn thiện các chức năng chưa phát triển được.
  + Tối ưu giao diện hệ thống thân thiện, đẹp mắt, dễ sử dụng.
  + Tăng cường các yếu tố an toàn bảo mật của hệ thống.
  + Đưa hệ thống vào ứng dụng thử nghiệm trong thực tế.

# TÀI LIỆU THAM KHẢO

1. Nguyễn Đức Thiện (2024). *“Bài giảng giới thiệu tổng quan UML”*. Trường Đại học Công nghệ Đông Á.
2. Trung Nguyễn (2024). “Spring Boot là gì? Tại sao lập trình viên nên học Spring Boot?”. Truy cập ngày 25/03/2024, từ <https://topdev.vn/blog/spring-boot-la-gi/>
3. Võ Hoàng (2023). “Thymeleaf là gì? Hướng dẫn sử dụng Thymeleaf với Spring Boot”. Truy cập ngày 26/03/2024, từ <https://techmaster.vn/posts/38338/thymeleaf-la-g>i
4. CodeGym (2023). “MySQL là gì? Cài đặt và sử dụng cơ bản MySQL cho người mới bắt đầu”. Truy cập ngày 26/03/2024, từ <https://codegym.vn/blog/2020/02/12/mysql-la-gi/>
5. Tấn Lộc (2024). “RESTful API là gì? Cách xây dựng RESTful API trong Spring Boot”. Truy cập ngày 28/03/2024, từ <https://viblo.asia/p/restful-api-la-gi-va-cach-xay-dung-trong-spring-boot-E375zM7xZGW>
6. Tiến Duy (2023). “Giới thiệu về JSON và cách sử dụng trong giao tiếp giữa client-server”. Truy cập ngày 02/04/2024, từ <https://viblo.asia/p/json-la-gi-json-duoc-su-dung-nhu-the-nao-trong-lap-trinh-RnB5pkMNKPG>
7. Việt Trần (2023). “JPA là gì? Tổng quan về Java Persistence API trong Spring Boot”. Truy cập ngày 02/04/2024, từ <https://toidicode.com/jpa-la-gi-su-dung-jpa-trong-spring-boot-288.html>
8. FastCoding (2023). “ZXing là gì? Hướng dẫn tạo mã QR code trong Java”. Truy cập ngày 13/04/2024, từ <https://fastcoding.net/blog/zxing-tao-qr-code-trong-java/>