

BỘ GIÁO DỤC VÀ ĐÀO TẠO

ĐẠI HỌC PHENIKAA



BÁO CÁO BÀI TẬP LỚN

HỌC PHẦN: LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

ĐỀ TÀI: Xây dựng ứng dụng quản lý bán hàng siêu thị

(Nhóm 2)

LỚP: Lập trình hướng đối tượng -1-1-25 (N03)

Sinh viên : Chu Việt Long

MSV : 22010107

Giảng viên hướng dẫn: TS. Nguyễn Lê Thu

Hà Nội, Năm 2025

Nhóm sinh viên thực hiện đề tài:

STT	Họ và tên	MSSV	Điểm (bảng số)	Điểm (bảng chữ)
1	Chu Việt Long	22010107		
2	Hà Tiên Quang	22010136		
3	Đỗ Thanh Hải	21011122		

BẢNG BÁO CÁO PHÂN CÔNG CÔNG VIỆC

STT	Họ và tên	MSSV	Công việc	Tiến độ
1	Chu Việt Long	22010107	Viết toàn bộ mã nguồn Thực hiện demo (quay video) Viết báo cáo	Hoàn thành
2	Hà Tiên Quang	22010136	Viết báo cáo	Hoàn thành
3	Đỗ Thanh Hải	21011122	Viết báo cáo	Hoàn thành

Link:

- Github: https://github.com/Mickeychus/market_managent.git
- Youtube: https://youtu.be/c8jo_qKA878
- Host: <http://localhost:8080>

MỤC LỤC

LỜI NÓI ĐẦU

CHƯƠNG I: CƠ SỞ LÝ THUYẾT6

1. Tổng quan về lập trình hướng đối tượng (OOP).....	6
1.1. Khái niệm lập trình hướng đối tượng.....	6
1.2. Các đặc trưng cơ bản của OOP.....	6
1.3. Ưu điểm của lập trình hướng đối tượng.....	6
2. Ngôn ngữ lập trình Java.....	7
2.1. Giới thiệu về Java.....	7
2.2. Đặc điểm của Java.....	7
2.3. Ưu điểm khi sử dụng Java trong phát triển ứng dụng quản lý bán hàng	7
3. Giới thiệu về Spring Boot.....	7
3.1. Khái niệm.....	7
3.2. Ưu điểm của Spring Boot.....	8
3.3. Các thành phần chính trong Spring Boot.....	8
4. Mô hình kiến trúc MVC trong Spring Boot.....	8
5. Cơ sở dữ liệu và ORM (Object-Relational Mapping).....	8
5.1. Khái niệm cơ sở dữ liệu.....	9
5.2. ORM trong Java.....	9
6. Tổng quan về ứng dụng quản lý bán hàng siêu thị.....	9
CHƯƠNG II: HỆ THỐNG QUẢN LÝ BÁN HÀNG SIÊU THỊ.....	10
1. Giới thiệu về hệ thống quản lý bán hàng siêu thị.....	10

2. Phân tích hệ thống.....	10
2.1. Sơ đồ hệ thống.....	12
2.2. Chức năng của hệ thống.....	15
2.3. Xây dựng chương trình.....	18
2.4. Cơ sở dữ liệu.....	21
3. Giao diện ứng dụng và hướng dẫn sử dụng ứng dụng.....	23

KẾT LUẬN

TÀI LIỆU THAM KHẢO

LỜI NÓI ĐẦU

Ngày nay, cuộc cách mạng công nghiệp 4.0 đang bắt đầu bùng nổ trên toàn cầu đã mang đến những thách thức cho thế giới nói chung và cho Việt Nam trên tất cả các lĩnh vực: kinh tế, kỹ thuật, y tế,...trong đó có lĩnh vực công nghệ thông tin. Việc phát triển công nghệ thông tin, ứng dụng trí tuệ nhân tạo trong đời sống là một trong những xu hướng mới mà xã hội đang hướng tới cũng và đòi hỏi những người kỹ sư, những sinh viên đại học phải nắm vững, hiểu rõ những kiến thức về các ngôn ngữ lập trình, cơ sở dữ liệu, lập trình hướng đối tượng,... và cách ứng dụng chúng để có thể bắt kịp được sự phát triển của thời đại. Khi cuộc sống của mỗi người được nâng cao thì nhu cầu về mua sắm, vui chơi giải trí ngày càng được chú trọng. Do đó, các doanh nghiệp cũng quan tâm ưu tiên cho việc phát triển các khu thương mại mua sắm nhằm đáp ứng nhu cầu của người dân. Hệ thống quản lý bán hàng siêu thị sẽ là công cụ quản lý bán hàng hiệu quả, làm đơn giản việc lập các hóa đơn giấy tờ lưu trữ chúng, cung cấp thông tin chuẩn xác làm cơ sở cho kinh doanh, phục vụ hữu ích cho việc quản lý và phát triển siêu thị.

Được sự hướng dẫn của **TS. Nguyễn Lê Thu** – giảng viên trực tiếp giảng dạy học phần **Lập trình hướng đối tượng**, với sự nhiệt tình và tâm huyết đã trang bị những kiến thức quý giá, giải đáp thắc mắc giúp chúng em có được những kỹ năng trong việc ứng dụng lập trình hướng đối tượng vào việc phát triển đề tài **Hệ thống quản lý bán hàng siêu thị**. Chúng em mong rằng đề tài sau khi thực hiện sẽ được hoàn thiện và phát triển giúp cho việc quản lý cũng như vận hành bán hàng siêu thị sẽ trở nên số hóa, tiện lợi và dễ dàng hơn.

Do hạn chế về mặt thời gian và hiểu biết, đề tài của chúng em có thể còn nhiều thiếu sót, rất mong sẽ nhận được sự góp ý, bổ sung từ thầy và các bạn để nhóm

chúng em có thể hoàn thiện vốn kiến thức của mình, tạo hành trang vững chắc cho việc phát triển trong tương lai.

CHƯƠNG I: CƠ SỞ LÝ THUYẾT

1. Tổng quan về lập trình hướng đối tượng (OOP)

1.1. Khái niệm lập trình hướng đối tượng

Lập trình hướng đối tượng (Object-Oriented Programming - OOP) là một mô hình lập trình dựa trên việc mô phỏng các thực thể trong thế giới thực dưới dạng các đối tượng (object) trong chương trình. Mỗi đối tượng có thuộc tính (attribute) và phương thức (method), thể hiện trạng thái và hành vi của đối tượng đó. Mục tiêu của lập trình hướng đối tượng là giúp chương trình dễ hiểu, dễ bảo trì, dễ mở rộng và tái sử dụng mã nguồn thông qua việc tổ chức dữ liệu và chức năng một cách hợp lý.

1.2. Các đặc trưng cơ bản của OOP

- Tính đóng gói (Encapsulation): Là khả năng che giấu thông tin bên trong đối tượng và chỉ cung cấp những gì cần thiết ra bên ngoài thông qua các phương thức (getter, setter). Điều này giúp bảo vệ dữ liệu khỏi bị truy cập hoặc thay đổi trái phép.
- Tính kế thừa (Inheritance): Cho phép một lớp (class) con kế thừa thuộc tính và phương thức của lớp cha, từ đó giúp giảm thiểu việc lặp lại mã và tăng khả năng tái sử dụng.
- Tính đa hình (Polymorphism): Cho phép một phương thức hoặc đối tượng có thể hoạt động theo nhiều cách khác nhau tùy thuộc vào ngữ cảnh. Trong Java, điều này thường thể hiện qua việc ghi đè phương thức (method overriding) và nạp chồng phương thức (method overloading).
- Tính trừu tượng (Abstraction): Là khả năng ẩn đi những chi tiết cài đặt phức tạp, chỉ hiển thị những đặc điểm cần thiết. Lập trình viên có thể sử dụng lớp trừu tượng hoặc giao diện (interface) để định nghĩa các hành vi chung mà không cần quan tâm đến chi tiết triển khai.

1.3. Ưu điểm của lập trình hướng đối tượng

- Dễ dàng mở rộng và bảo trì hệ thống.
- Tái sử dụng mã nguồn hiệu quả thông qua kế thừa và mô-đun hóa.
- Giúp mô phỏng các mối quan hệ trong thế giới thực, trực quan hơn so với lập trình hướng thủ tục.
- Tăng tính bảo mật và giảm thiểu lỗi do sự cô lập của các đối tượng.

2. Ngôn ngữ lập trình Java

2.1. Giới thiệu về Java

Java là một ngôn ngữ lập trình bậc cao, hướng đối tượng, được phát triển bởi Sun Microsystems (nay thuộc Oracle). Java có khẩu hiệu nổi tiếng “Write once, run anywhere” — viết một lần, chạy được ở mọi nơi — nhờ vào cơ chế Java Virtual Machine (JVM).

2.2. Đặc điểm của Java

- Hướng đối tượng hoàn toàn, hỗ trợ đầy đủ các khái niệm như class, object, inheritance, polymorphism.
- Đa nền tảng, chạy được trên Windows, Linux, macOS,...
- Bảo mật cao nhờ cơ chế kiểm tra bytecode và sandbox.
- Quản lý bộ nhớ tự động bằng Garbage Collector.
- Thư viện phong phú, hỗ trợ mạnh mẽ cho lập trình mạng, đa luồng, giao diện và cơ sở dữ liệu.

2.3. Ưu điểm khi sử dụng Java trong phát triển ứng dụng quản lý bán hàng

- Dễ dàng tổ chức hệ thống thành các lớp mô phỏng: sản phẩm, hóa đơn, người dùng,...
- Hỗ trợ Spring Framework – nền tảng mạnh mẽ giúp phát triển ứng dụng web và hệ thống doanh nghiệp.
- Tích hợp tốt với cơ sở dữ liệu và các công nghệ như REST API, Hibernate, JPA.

3. Giới thiệu về Spring Boot

3.1. Khái niệm

Spring Boot là một framework mở rộng của Spring Framework, giúp lập trình viên tạo và triển khai ứng dụng Java nhanh chóng, đơn giản mà không cần cấu hình phức tạp.

Spring Boot cung cấp sẵn các cấu hình mặc định và thư viện tích hợp (starter dependencies), giúp giảm thời gian cài đặt và cấu hình ban đầu.

3.2. Ưu điểm của Spring Boot

- Tự động cấu hình (Auto Configuration): Không cần khai báo thủ công file XML.
- Tích hợp sẵn máy chủ nhúng như Tomcat hoặc Jetty, giúp chạy ứng dụng trực tiếp.
- Hỗ trợ RESTful API dễ dàng xây dựng các dịch vụ web.
- Quản lý phụ thuộc tiện lợi bằng Maven hoặc Gradle.
- Tích hợp tốt với cơ sở dữ liệu thông qua Spring Data JPA, Hibernate.

3.3. Các thành phần chính trong Spring Boot

- Controller: Xử lý các yêu cầu từ người dùng, điều hướng dữ liệu giữa các lớp Model và View.
- Service: Thực hiện các logic nghiệp vụ của ứng dụng.
- Repository: Tương tác với cơ sở dữ liệu (CRUD).
- Model: Đại diện cho dữ liệu trong hệ thống.
- View: Giao diện hiển thị cho người dùng (có thể dùng Thymeleaf, HTML, CSS, JS).

4. Mô hình kiến trúc MVC trong Spring Boot

Mô hình MVC (Model – View – Controller) là nền tảng của hầu hết các ứng dụng Spring Boot.

- Model: Đại diện cho dữ liệu (ví dụ: SanPham, HoaDon, NguoiDung), chứa các thuộc tính và phương thức xử lý dữ liệu.
- View: Hiển thị thông tin cho người dùng (HTML, CSS, JavaScript hoặc Thymeleaf).
- Controller: Trung gian điều phối luồng xử lý, tiếp nhận request từ người dùng, tương tác với Model, và trả kết quả về View.

Ưu điểm của mô hình MVC:

- Dễ mở rộng và bảo trì.
- Tách biệt rõ ràng giữa giao diện, dữ liệu và xử lý logic.

- Dễ dàng kiểm thử từng phần độc lập.

5. Cơ sở dữ liệu và ORM (Object-Relational Mapping)

5.1. Khái niệm cơ sở dữ liệu

Cơ sở dữ liệu là nơi lưu trữ có tổ chức của thông tin, giúp người dùng dễ dàng truy xuất, thêm, sửa, xóa và quản lý dữ liệu. Trong ứng dụng quản lý bán hàng, cơ sở dữ liệu chứa thông tin về sản phẩm, hóa đơn, khách hàng, doanh thu,...

5.2. ORM trong Java

ORM (Object Relational Mapping) là kỹ thuật ánh xạ giữa đối tượng trong lập trình và bảng dữ liệu trong cơ sở dữ liệu.

Spring Boot sử dụng JPA (Java Persistence API) kết hợp Hibernate để ánh xạ tự động giữa lớp Java và bảng trong cơ sở dữ liệu.

Ví dụ:

- Lớp SanPham.java ↔ Bảng san_pham
- Lớp HoaDon.java ↔ Bảng hoa_don

Điều này giúp giảm việc viết truy vấn SQL thủ công, đồng thời tăng tính linh hoạt khi thay đổi cấu trúc dữ liệu.

6. Tổng quan về ứng dụng quản lý bán hàng siêu thị

Ứng dụng quản lý bán hàng siêu thị là hệ thống giúp:

- Quản lý danh sách sản phẩm (thêm, sửa, xóa, tìm kiếm, sắp xếp).
- Quản lý hóa đơn và quá trình bán hàng.
- Thống kê doanh thu, hàng tồn kho.
- Quản lý người dùng, tài khoản đăng nhập và quyền truy cập.

Hệ thống được xây dựng theo hướng mô-đun, áp dụng chặt chẽ các nguyên lý của lập trình hướng đối tượng, đảm bảo hiệu năng, khả năng mở rộng và dễ dàng bảo trì.

CHƯƠNG II: HỆ THỐNG QUẢN LÝ BÁN HÀNG SIÊU THỊ

1. Giới thiệu về hệ thống quản lý bán hàng siêu thị

Hệ thống quản lý bán hàng siêu thị là một trong những giải pháp quan trọng để giúp cho các siêu thị quản lý hoạt động kinh doanh một cách hiệu quả hơn. Dự án thiết kế hệ thống quản lý bán hàng sẽ tập trung vào việc phát triển một phần mềm chuyên nghiệp, đáp ứng các yêu cầu quản lý của các nhà hàng, đồng thời giúp cho chủ nhà hàng có thể quản lý hoạt động kinh doanh của mình một cách chuyên nghiệp và tiện lợi hơn. Một siêu thị muốn xây dựng một hệ thống phần mềm để quản lý việc mua bán hàng của siêu thị. Hệ thống cần phải ghi nhận các hoạt động bán hàng và xử lý các công việc thanh toán với khách hàng mua lẻ. Ngoài ra hệ thống còn giúp người quản lý nắm được tình hình mua bán, doanh thu của siêu thị, việc thống kê được thực hiện giúp thủ kho quản lý số lượng hàng trong kho theo mã số hàng hóa, kiểm kê hàng hóa trong kho; giúp nhân viên bán hàng sẽ tính tiền những mặt hàng mà khách hàng mua và lập hóa đơn cho khách. Trong dự án này sẽ bao gồm các tính năng quản lý sản phẩm, đơn hàng, bán hàng tính tiền cho khách và báo cáo doanh thu. Những tính năng này sẽ được thiết kế để đáp ứng các yêu cầu quản lý của các siêu thị, giúp cho các siêu thị có thể quản lý hoạt động kinh doanh của mình một cách chuyên nghiệp và dễ dàng hơn. Từ đó tăng khả năng quản lý và tránh được những rủi ro không đáng có trong nhà hàng.

Dự án này sẽ đóng góp một phần quan trọng trong việc nâng cao năng suất kinh doanh của các siêu thị, giúp cho siêu thị có thể quản lý và tối ưu hóa hoạt động kinh doanh một cách hiệu quả hơn. Bên cạnh đó, hệ thống cũng giúp khách hàng dễ dàng thực hiện đặt hàng và thanh toán, tạo ra trải nghiệm người dùng tốt hơn.

2. Phân tích hệ thống

Phần này, sẽ trình bày về công nghệ và phương pháp thiết kế và phát triển ứng dụng quản lý bán hàng siêu thị của nhóm.

Công nghệ sử dụng là ngôn ngữ Java theo hướng đối tượng. Java là một ngôn ngữ lập trình phổ biến và được sử dụng rộng rãi trong phát triển các ứng dụng web và

phần mềm. Đặc biệt, Java có hỗ trợ mô hình đối tượng, giúp cho việc xây dựng ứng dụng trở nên dễ dàng, linh hoạt và hiệu quả hơn.

Giao diện người dùng (View) được xây dựng bằng Thymeleaf, là một "template engine" phía máy chủ (server-side) tích hợp trực tiếp với **Spring Boot**. Spring Boot (với Spring MVC) đóng vai trò là **back-end**, xử lý logic và nghiệp vụ, sau đó đưa dữ liệu (Model) ra cho Thymeleaf để hiển thị (render) thành các tệp HTML động. Thymeleaf, kết hợp với HTML/CSS, cho phép tạo ra các giao diện đẹp mắt và dễ dàng sử dụng trên trình duyệt web.

Trong phiên bản hiện tại, ứng dụng sử dụng cơ sở dữ liệu **Cloud MySQL** (được host trên nền tảng Aiven) và kết nối thông qua **Spring Data JPA** của Spring Boot.

Việc sử dụng MySQL giúp đảm bảo dữ liệu được lưu trữ lâu dài, có thể truy cập và thống kê chính xác doanh thu, hàng hóa và hóa đơn. Ứng dụng được xây dựng trên nền tảng Spring Boot, sử dụng ngôn ngữ Java. Toàn bộ dữ liệu của hệ thống được lưu trữ bền vững trên Cloud MySQL (được host trên nền tảng Aiven) và được quản lý thông qua Spring Data JPA. Giao diện người dùng (View) được xây dựng bằng Thymeleaf và HTML/CSS, đảm bảo ứng dụng có thể truy cập qua trình duyệt web.

Ứng dụng được thiết kế theo mô hình **MVC (Model-View-Controller)**, đây là một kiến trúc phần mềm phổ biến trong phát triển các ứng dụng web và phần mềm. Mô hình MVC phân tách ứng dụng thành ba phần: Phần Model (mô hình dữ liệu), Phần View (giao diện người dùng) và Phần Controller (bộ điều khiển). Mô hình này giúp cho việc phát triển và bảo trì ứng dụng trở nên dễ dàng cũng như hiệu quả hơn, đồng thời cũng giúp tách biệt các phần của ứng dụng và làm cho ứng dụng dễ dàng tái sử dụng.

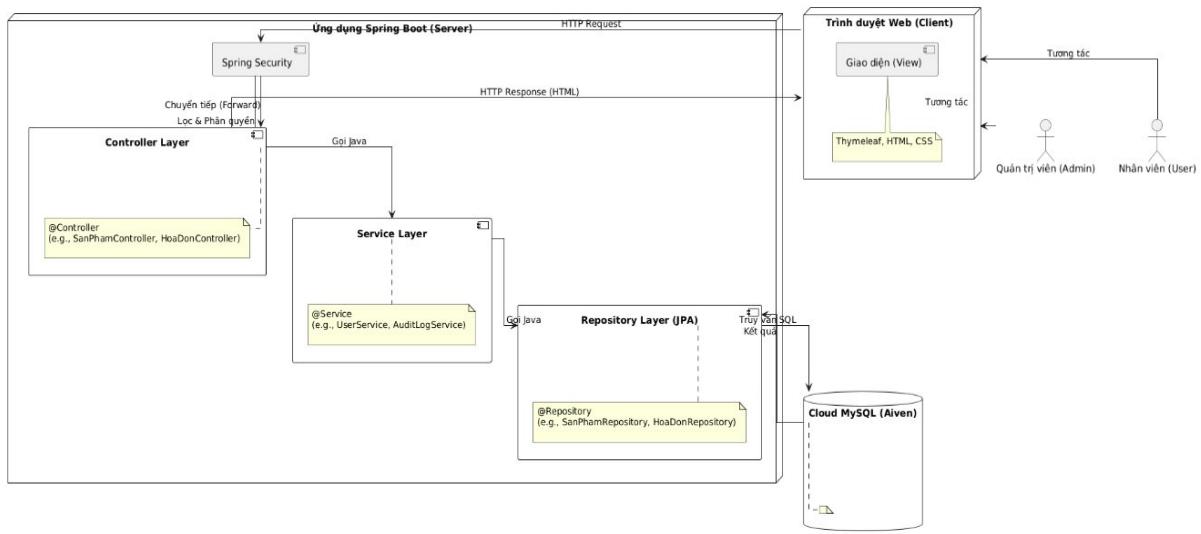
Mô hình MVC (Model-View-Controller) là một kiến trúc thiết kế phổ biến được sử dụng trong phát triển ứng dụng Java (và nhiều ngôn ngữ khác) để tách biệt các phần khác nhau của ứng dụng và làm cho mã nguồn dễ quản lý và bảo trì. MVC chia ứng dụng thành ba thành phần chính:

- **Model (M):** Phần này đại diện cho dữ liệu và logic thao tác với dữ liệu. Model là nơi bạn định nghĩa cách dữ liệu được lưu trữ, truy vấn, và cập nhật. Điều này bao gồm các lớp Java thể hiện dữ liệu (các lớp trong Gói com.store.quanlysieuthi.Model), cơ sở dữ liệu, và các phương thức xử lý dữ liệu (trong các lớp Service).
- **View (V):** View đại diện cho giao diện người dùng và hiển thị thông tin cho người dùng. Nó không chứa logic kinh doanh, chỉ hiển thị dữ liệu từ Model. Trong dự án này, View là các tệp **Thymeleaf HTML** (ví dụ: banhang.html).
- **Controller (C):** Controller quản lý tương tác giữa Model và View. Nó nhận các sự kiện (HTTP request) từ View, xử lý chúng (bằng cách gọi Service hoặc Repository), và sau đó cập nhật Model hoặc hiển thị thông tin mới trên View. Controller là các lớp trong Gói com.store.quanlysieuthi.Controller.

Trong mô hình MVC, sự tách biệt giữa các thành phần cho phép bạn dễ dàng thay đổi hoặc bổ sung một phần mà không ảnh hưởng đến các phần khác. Điều này rất hữu ích trong việc phát triển và bảo trì ứng dụng lớn.

2.1. Sơ đồ hệ thống

Sau khi đăng nhập, người dùng có thể sử dụng các chức năng của hệ thống: Nhập hàng (Quản lý sản phẩm), bán hàng tính tiền (Bán hàng), thống kê cho khách và các chức năng mà hệ thống đã cấp phát (dựa trên vai trò ROLE_ADMIN hoặc ROLE_USER).



Để xây dựng một ứng dụng Quản lý bán hàng siêu thị hiệu quả và đáp ứng được nhu cầu của siêu thị và khách hàng, chúng ta cần xác định rõ các chức năng cần thiết. Dưới đây là mô tả chi tiết về các yêu cầu cơ bản cho ứng dụng:

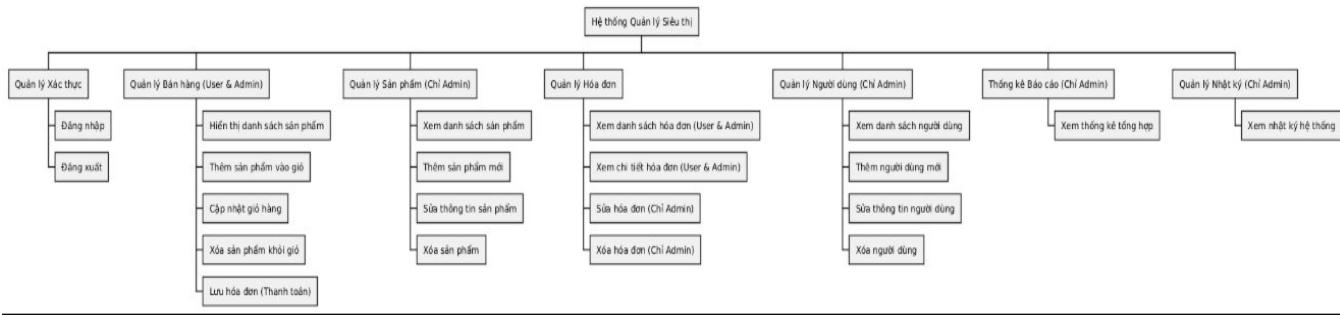
Mô tả các Use Case (UC) :

- **Use Case "Đăng nhập":**
 - **Mô tả:** Người dùng nhập tên đăng nhập và mật khẩu tại trang /login để đăng nhập vào hệ thống.
 - **Thành phần chính:** Giao diện đăng nhập, tên đăng nhập, mật khẩu.
 - **Kết quả:** Nếu tên đăng nhập thành công, người dùng được chuyển đến giao diện chính của ứng dụng (/banhang). Nếu không thành công, người dùng sẽ nhận được thông báo lỗi tương ứng.
- **Use Case "Trang chủ" (Menu điều hướng):**
 - **Mô tả:** Cho phép người dùng chọn các chức năng tiếp theo của hệ thống: Sản phẩm, Bán hàng, Hóa đơn, Thông kê. Nếu là Admin, sẽ có thêm Quản lý User và Xem Nhật ký.
 - **Thành phần chính:** Giao diện (Thymeleaf fragments), các liên kết đến các chức năng.
- **Use Case "Sản Phẩm" (Admin):**

- **Mô tả:** Cho phép người dùng (Admin) thêm, chỉnh sửa, xóa sản phẩm.
 - **Thành phần chính:** Giao diện quản lý sản phẩm, danh sách bảng thông tin sản phẩm, chức năng thêm, chỉnh sửa, xóa.
 - **Kết quả:** Người dùng có thể thực hiện các chức năng quản lý sản phẩm thành công. (Lưu ý: Chức năng tìm kiếm và sắp xếp chưa được triển khai trong mã nguồn SanPhamController hiện tại).
- **Use Case "Hóa đơn" (Bán hàng):**
 - **Mô tả:** Cho phép người dùng thêm đơn hàng (thêm sản phẩm vào giỏ hàng), sửa, xóa mặt hàng (khỏi giỏ hàng), tính tiền cho khách và **lưu hóa đơn** (chức năng "in hóa đơn" chưa được triển khai).
 - **Use Case “Thống kê” (Admin):**
 - **Mô tả:** Cho phép thống kê tổng số sản phẩm nhập, tổng tiền nhập, tổng lượng bán, tổng tiền bán, tổng lãi (dựa trên dữ liệu từ ThongKeService).

Các đối tượng trên có mối quan hệ với nhau thông qua các phương thức (methods) và thuộc tính (attributes), tạo thành một mô hình đối tượng phức tạp trong hệ thống quản lý bán hàng siêu thị. Chúng tương tác với nhau để thu thập, lưu trữ và quản lý thông tin sản phẩm, hóa đơn, doanh thu và người dùng một cách hiệu quả và chính xác. Việc phân tích và thiết kế các đối tượng này giúp xác định rõ ràng các chức năng và quy trình trong hệ thống quản lý bán hàng, đảm bảo ứng dụng hoạt động ổn định, linh hoạt và dễ mở rộng trong tương lai.

2.2. Chức năng của hệ thống

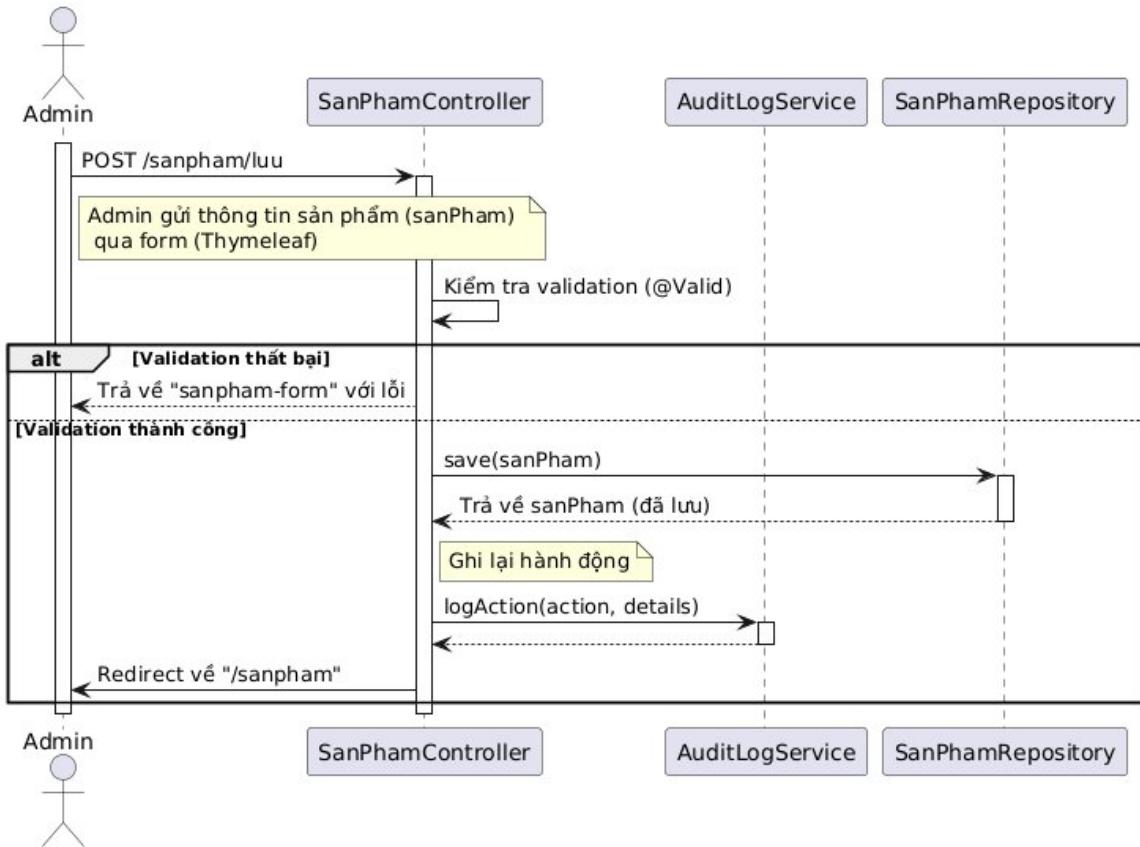


2.2.1. Quản lý sản phẩm

- Chức năng này cho phép người quản lý (Admin) thực hiện các thao tác thêm mới, chỉnh sửa, xóa và tìm kiếm sản phẩm trong hệ thống.
- Dữ liệu sản phẩm được lưu trữ trong cơ sở dữ liệu MySQL và được quản lý thông qua Spring Data JPA. Khi người dùng thêm mới hoặc cập nhật sản phẩm, dữ liệu được gửi đến SanPhamController, xử lý (qua AuditLogService) và lưu vào cơ sở dữ liệu bởi SanPhamRepository.
- Các chức năng nâng cao như sắp xếp và lọc sản phẩm chưa được triển khai trong phiên bản hiện tại và sẽ được bổ sung trong bản cập nhật tiếp theo.

Danh sách Sản phẩm						
Đã thêm sản phẩm thành công!						
Thêm Sản phẩm Mới						
ID	Tên Sản Phẩm	Ngày Nhập	Số Lượng	Phân Loại	Giá Nhập	Giá Đè Xuất
1	sữa xích cộ	31/10/2025	8	Thực phẩm	45,000.00	50,000.00
2	sữa milo	03/11/2025	5	Thực phẩm	300.00	320.00

-Sắp xếp theo thứ tự ID tự động điền vào



Sơ đồ tuần tự chức năng “Thêm sản phẩm” Sơ đồ thể hiện quy trình thêm sản phẩm mới. Nhân viên (Admin) nhập thông tin → gửi yêu cầu **POST /sanpham/luu** → Controller nhận và chuyển tiếp đến Repository để lưu (đồng thời gọi AuditLogService để ghi log) → phản hồi kết quả (redirect) về trang danh sách /sanpham.

2.2.2. Quản lý bán hàng:

Bao gồm các chức năng thêm, sửa, xóa (sản phẩm trong giỏ hàng), tính tổng tiền và lưu hóa đơn.

- Cho phép sử dụng các tính năng thêm, sửa, xóa để chỉnh sửa thông tin sản phẩm trong giỏ hàng.
- Lưu trữ và cập nhật dữ liệu vào hệ thống (khi nhấn "Lưu Hóa Đơn").
- Chức năng "in hóa đơn".

Danh sách Sản phẩm				
Tên	Tồn Kho	Giá Bán	Số Lượng	Thêm
xúc xích cp	7	50.000.00	1	<button>Thêm</button>
sữa milo	4	320.00	1	<button>Thêm</button>

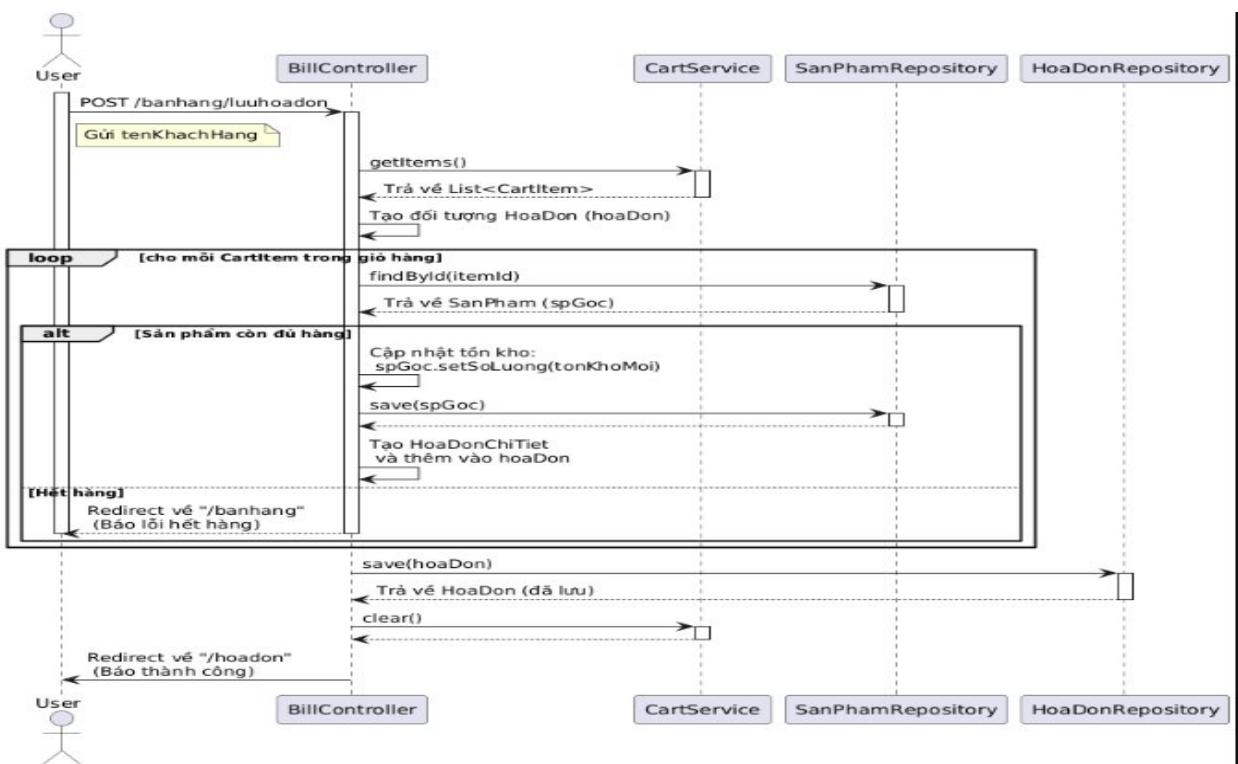
Hóa đơn Tạm / Giỏ Hàng				
Tên Hàng	Số Lượng	Đơn Giá	Thành Tiền	Hành động
xúc xích cp	1	50.000.00	50.000.00	<button>Xóa</button>
sữa milo	1	320.00	320.00	<button>Xóa</button>

Tên Khách Hàng:
Nhập tên khách hàng...

Lưu Hóa Đơn

Giao diện tổng quan:

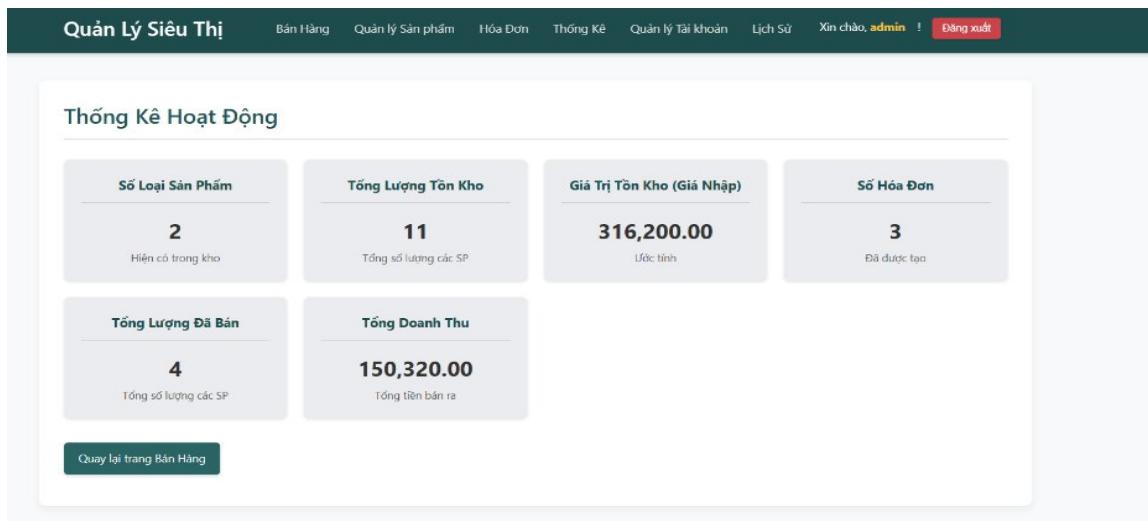
The screenshot shows the main dashboard of the Supermarket Management System. At the top, there is a navigation bar with links: Quản Lý Siêu Thị, Bán Hàng, Quản lý Sản phẩm, Hóa Đơn, Thống Kê, Quản lý Tài khoản, Lịch Sử, Xin chào, admin !, and Đăng xuất. Below the navigation bar, there are two main sections: "Danh sách Sản phẩm" and "Hóa đơn Tạm / Giỏ Hàng". The "Danh sách Sản phẩm" section displays a table with product information (Tên, Tồn Kho, Giá Bán, Số Lượng, Thêm). The "Hóa đơn Tạm / Giỏ Hàng" section displays a table with items in the cart (Tên Hàng, Số Lượng, Đơn Giá, Thành Tiền, Hành động). A large camera icon is centered below these sections.



Sơ đồ mô tả quá trình nhân viên thực hiện tạo hóa đơn: Giao diện (Trang /banhang) gửi yêu cầu **POST /banhang/luuhoadon** → BillController gọi CartService để xử lý giỏ hàng → Service kiểm tra tồn kho qua SanPhamRepository → lưu HoaDon và HoaDonChiTiet → cập nhật tồn kho và trả về kết quả.

2.2.3. Thống kê

Giao diện này sẽ hiện thống kê tổng số sản phẩm nhập, tổng tiền nhập, tổng lượng bán, tổng tiền bán, tổng doanh thu (dựa trên dữ liệu từ ThongKeService và hiển thị tại /thongke).



2.3. Xây dựng chương trình

Toàn bộ ứng dụng được xây dựng dựa trên mô hình 3 lớp **Controller – Service – Repository**, là kiến trúc chuẩn trong Spring Boot.

- **Controller** chịu trách nhiệm tiếp nhận request.
- **Service** thực hiện xử lý nghiệp vụ.
- **Repository** giao tiếp với MySQL qua JPA. Nhờ đó, ứng dụng dễ mở rộng, kiểm thử và bảo trì.

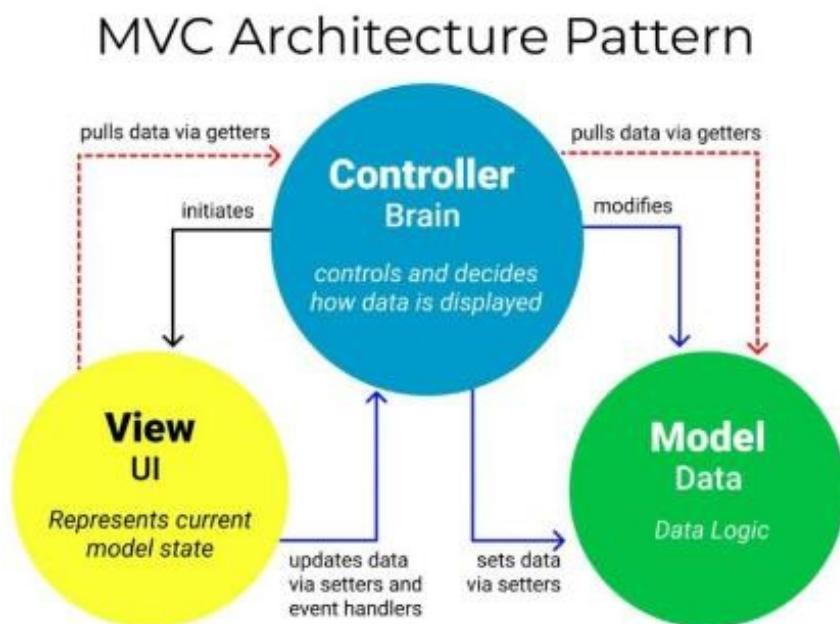
2.3.1. Mô hình MVC

MVC là viết tắt của cụm từ “Model-View-Controller”. Đây là mô hình thiết kế được sử dụng trong kỹ thuật phần mềm. MVC là một mẫu kiến trúc phần mềm để tạo lập giao diện người dùng trên máy tính. MVC chia thành ba phần được kết nối với nhau và mỗi thành phần đều có một nhiệm vụ riêng của nó và độc lập với các thành phần khác. Tên gọi 3 thành phần:

- Model (dữ liệu): Quản lý xử lý các dữ liệu.
- View (giao diện): Nơi hiển thị dữ liệu cho người dùng.
- Controller (bộ điều khiển): Điều khiển sự tương tác của hai thành phần Model và View.

Mô hình MVC (MVC pattern) thường được dùng để phát triển giao diện người dùng. Nó cung cấp các thành phần cơ bản để thiết kế một chương trình cho máy tính hoặc điện thoại di động, cũng như là các ứng dụng web.

Các thành phần trong mô hình MVC:



Mô hình MVC

Mô hình MVC gồm 3 loại chính là thành phần bên trong không thể thiếu khi áp dụng mô hình này:

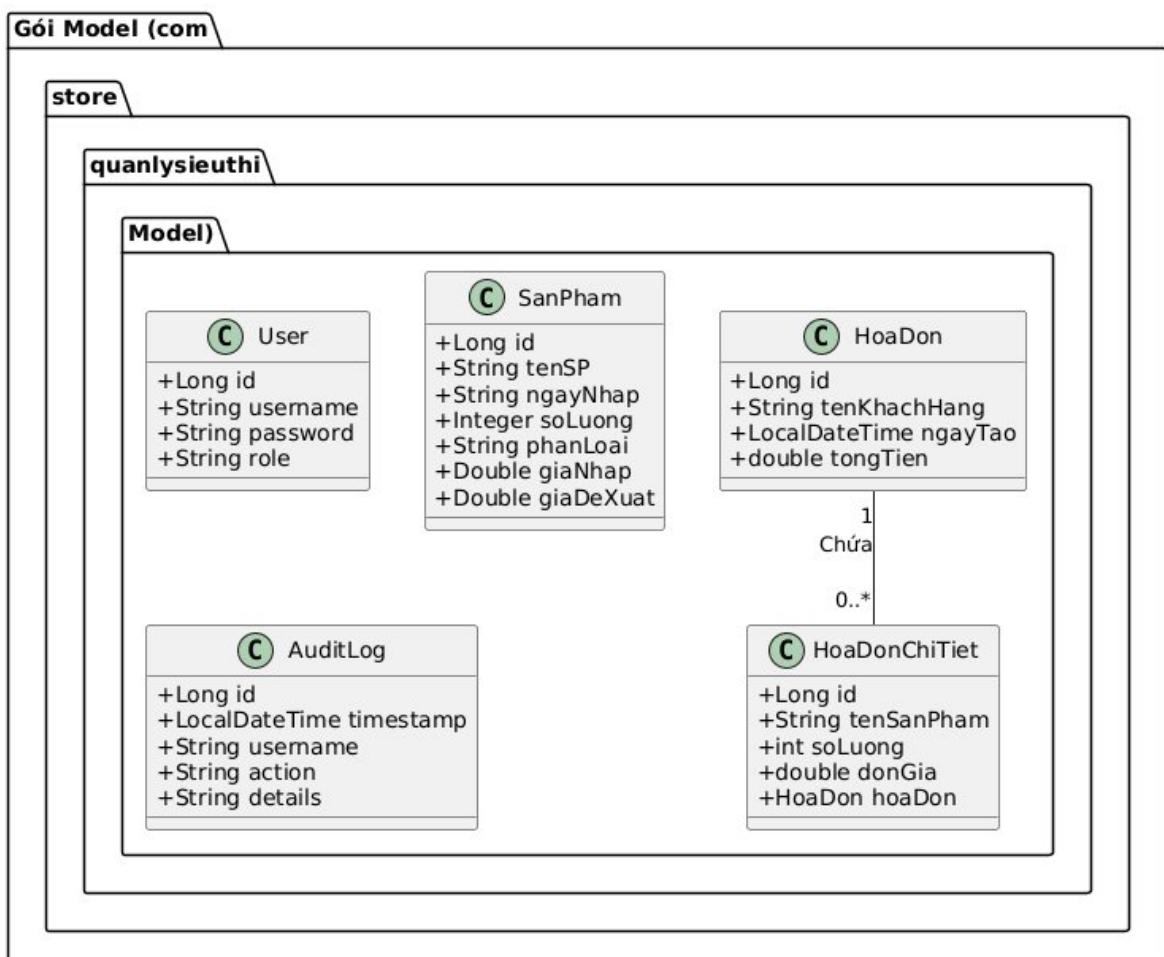
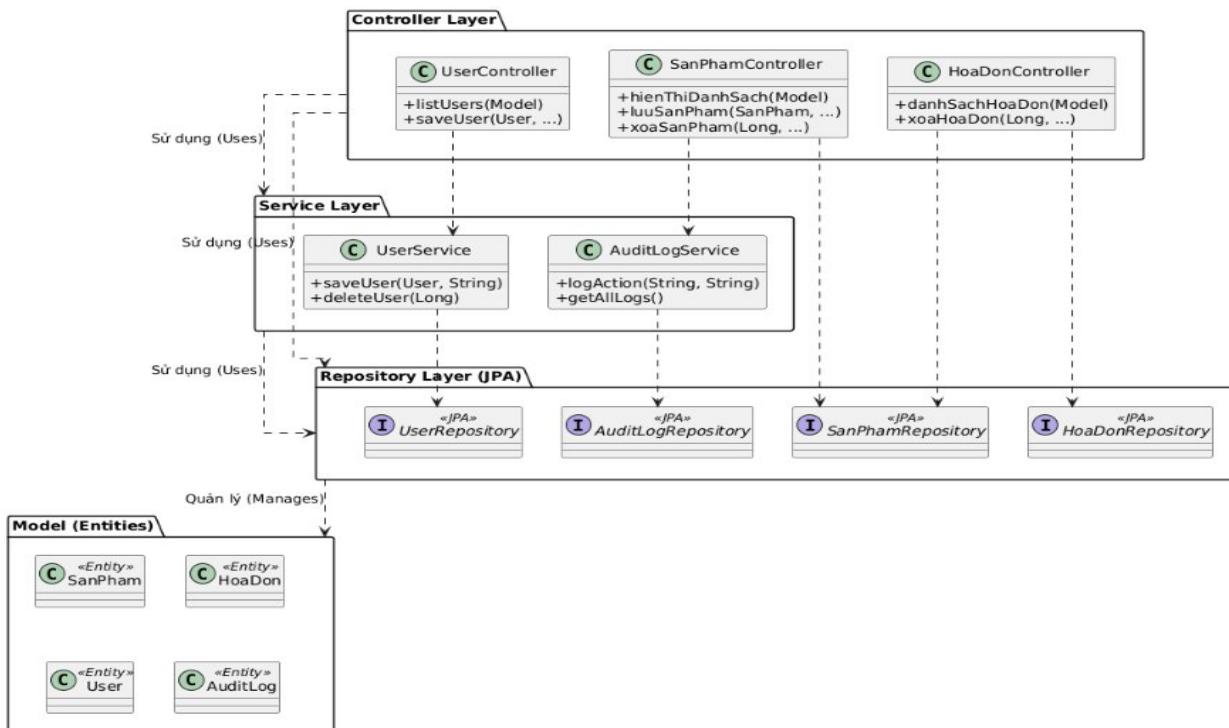
- **Model:** Là bộ phận có chức năng lưu trữ toàn bộ dữ liệu của ứng dụng và là cầu nối giữa 2 thành phần bên dưới là View và Controller. Một model là dữ liệu được sử dụng bởi chương trình. Đây có thể là cơ sở dữ liệu, hoặc file XML bình thường hay một đối tượng đơn giản. Chẳng hạn như biểu tượng hay là một nhân vật trong game.

- **View:** Đây là phần giao diện (theme) dành cho người sử dụng. View là phương tiện hiển thị các đối tượng trong một ứng dụng. Chẳng hạn như hiển thị một cửa sổ, nút hay văn bản trong một cửa sổ khác. Nó bao gồm bất cứ thứ gì mà người dùng có thể nhìn thấy được.
- **Controller:** Là bộ phận có nhiệm vụ xử lý các yêu cầu người dùng đưa đến thông qua View. Một controller bao gồm cả Model lẫn View. Nó nhận input và thực hiện các update tương ứng. Ví dụ: một Controller có thể cập nhật một Model bằng cách thay đổi các thuộc tính của nhân vật trong game. Và nó có thể sửa đổi view bằng cách hiển thị nhân vật được cập nhật trong game đó.

Trong chương trình của nhóm, mô hình MVC được áp dụng như sau:

- Bộ phận View : Đây là phần giao diện người dùng của hệ thống, được thiết kế trong thư mục resources/static (và templates nếu sử dụng Thymeleaf). Thành phần này bao gồm các tệp HTML, CSS và JavaScript, chẳng hạn như tệp style.css, giúp hiển thị thông tin sản phẩm, hóa đơn và người dùng trên trình duyệt một cách trực quan, dễ sử dụng.
- Bộ phận Model : Bộ phận này bao gồm các lớp nằm trong thư mục Model như SanPham.java, HoaDon.java, User.java, v.v... Các lớp này có chức năng lưu trữ và quản lý toàn bộ dữ liệu của ứng dụng, đồng thời được ánh xạ với cơ sở dữ liệu thông qua các lớp trong thư mục Repository. Nhờ đó, việc truy xuất, thêm, sửa, xóa dữ liệu trở nên dễ dàng và hiệu quả hơn so với cách lưu trữ bằng file XML.
- Bộ phận Controller : Các lớp trong thư mục Controller như UserController.java, SanPhamController.java, HoaDonController.java chịu trách nhiệm tiếp nhận các yêu cầu (request) từ người dùng, xử lý logic nghiệp vụ thông qua các lớp trong Service và Repository, sau đó trả kết quả về cho phần View. Controller đóng vai trò trung gian giúp kết nối giữa giao diện và dữ liệu, đảm bảo luồng xử lý của ứng dụng được thống nhất và mạch lạc.

2.3.2. Các lớp đối tượng



Biểu đồ lớp trên mô tả các lớp trong hệ thống quản lý bán hàng siêu thị.

2.4. Cơ sở dữ liệu.

Trong hệ thống quản lý bán hàng siêu thị hiện tại, nhóm sử dụng cơ sở dữ liệu **MySQL** được triển khai trên nền tảng **Cloud MySQL (Aiven)** để lưu trữ và quản lý dữ liệu thực tế. Ứng dụng được kết nối với cơ sở dữ liệu qua **Spring Data JPA**, là một phần của Spring Boot, giúp thực hiện các thao tác CRUD (Create, Read, Update, Delete) một cách tự động và hiệu quả thông qua các interface Repository.

Các lớp dữ liệu trong thư mục model (như SanPham.java, HoaDon.java, User.java, ...) được đánh dấu bằng `@Entity`, `@Table`, `@Id`, giúp ánh xạ trực tiếp giữa đối tượng Java và bảng dữ liệu trong MySQL. Ví dụ:

- SanPham.java ↔ bảng san_pham
- HoaDon.java ↔ bảng hoa_don
- User.java ↔ bảng users

Các interface trong thư mục repository như SanPhamRepository, HoaDonRepository, UserRepository mở rộng từ JpaRepository để Spring Boot tự động tạo các truy vấn SQL phù hợp.

Cấu hình kết nối được khai báo trong tệp application.properties:

Properties

```
spring.datasource.url=jdbc:mysql://<host>:<port>/<database>
```

```
spring.datasource.username=<username>
```

```
spring.datasource.password=<password>
```

```
spring.jpa.hibernate.ddl-auto=update
```

```
spring.jpa.show-sql=true
```

Ngoài ra, dữ liệu tạm thời của giỏ hàng (Cart) được quản lý bằng CartService, lưu trong session thay vì database, giúp giữ thông tin sản phẩm khách hàng đã chọn trong quá trình mua hàng.

Nhờ việc kết hợp Spring Data JPA và Cloud MySQL, hệ thống đạt được:

- Lưu trữ bền vững, an toàn.
- Giảm thiểu lỗi khi thao tác dữ liệu.
- Dễ dàng mở rộng hoặc triển khai thực tế trên môi trường server.

3. Giao diện ứng dụng và hướng dẫn sử dụng ứng dụng.

Dưới đây là các bước sử dụng ứng dụng quản lý bán hàng siêu thị:

1. Đăng nhập vào hệ thống:

- Nhập UserName và Password
- Sau khi nhập xong click Đăng nhập

Có 2 tài khoản đăng nhập chính:

Người dùng : tài khoản (user) và mật khẩu (user)
Quản lý: tài khoản (admin) và mật khẩu (admin)

Đăng nhập Hệ thống

Bạn đã đăng xuất thành công.

Tên đăng nhập:

Nhập tên đăng nhập

Mật khẩu:

Nhập mật khẩu

Đăng nhập

2. Tiếp theo giao diện ứng dụng sẽ hiện ra menu cho người dùng chọn

Danh sách Sản phẩm					Hóa đơn Tạm / Giỏ Hàng				
Tên	Tồn Kho	Giá Bán	Số Lượng	Thêm	Tên Hàng	Số Lượng	Đơn Giá	Thành Tiền	Hành động
xúc xích cp	7	50.000.00	1	<button>Thêm</button>	xúc xích cp	1	50.000.00	50.000.00	<button>Xóa</button>
sữa milo	4	320.00	1	<button>Thêm</button>	sữa milo	1	320.00	320.00	<button>Xóa</button>

Tổng Tiền: 50.320.00
Tên Khách Hàng: Nhập tên khách hàng...
Lưu Hóa Đơn

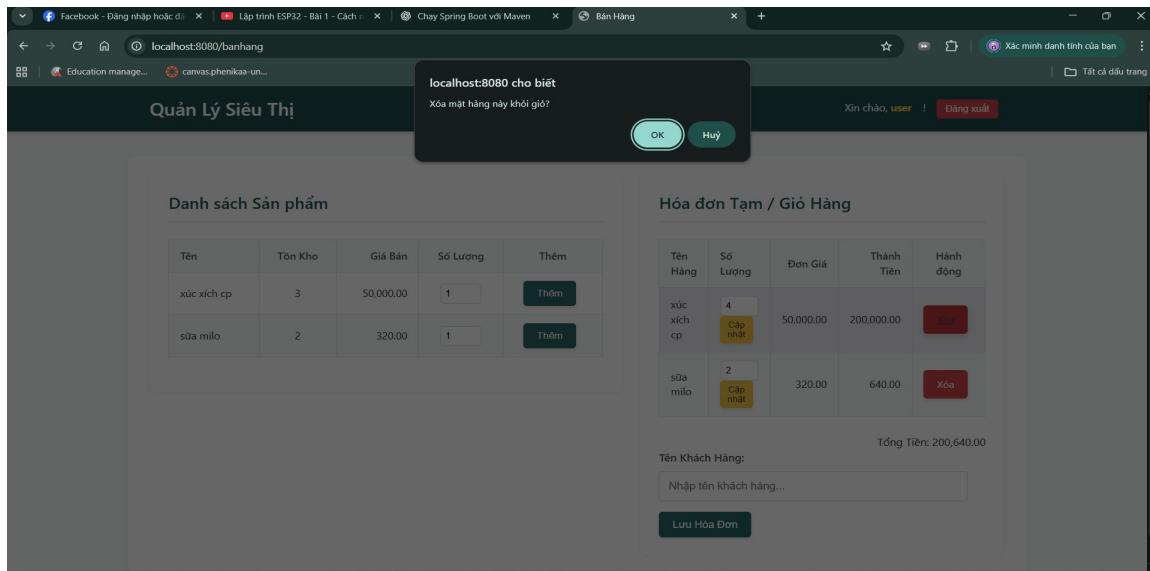
3. Nếu người dùng click Sản phẩm sẽ như sau:

- Khi người dùng chọn các sản phẩm muốn mua các sản phẩm sẽ tự động thêm vào hóa đơn tạm/giỏ hàng và sẽ tính số tiền các sản phẩm ở bên dưới sẽ có lựa chọn in hóa đơn và nhập tên khách hàng .

Danh sách Sản phẩm					Hóa đơn Tạm / Giỏ Hàng				
Tên	Tồn Kho	Giá Bán	Số Lượng	Thêm	Tên Hàng	Số Lượng	Đơn Giá	Thành Tiền	Hành động
xúc xích cp	3	50.000.00	1	<button>Thêm</button>	xúc xích cp	4	50.000.00	200.000.00	<button>Xóa</button>
sữa milo	2	320.00	1	<button>Thêm</button>	sữa milo	2	320.00	640.00	<button>Xóa</button>

Tổng Tiền: 200.640.00
Tên Khách Hàng: Nhập tên khách hàng...
Lưu Hóa Đơn

- Lỗi có thể xảy ra khi người dùng muốn mua số lượng nhiều hơn ở kho tồn tại:
 - Sẽ hiện thông báo ở giỏ hàng không đủ số lượng tồn kho.
 - Khách hàng có thể thay đổi số lượng hàng mua ở phần số lượng sau đó ấn cập nhật thì hóa đơn sẽ dựa vào số lượng hàng khách hàng mua để tính tiền lại .
 - Chức năng xóa: người dùng chọn sản phẩm cần xóa và click “ Xóa” .
 - Người dùng có thể xóa hàng mình không muốn mua thì sẽ hiện thông báo mình có xác nhận xóa hay không.



4. Khi người dùng click “Hóa đơn” để thực hiện bán hàng như sau:
- Khi người dùng đã thực hiện chọn sản phẩm cần mua và thanh toán tiền sau đó lựa chọn in hóa đơn hệ thống sẽ xuất hóa đơn cho người dùng .

Chi tiết Hóa Đơn #1

ID Hóa Đơn:	1
Tên Khách Hàng:	quang
Ngày Tạo:	31/10/2025 18:01:44
Tổng Tiền:	50,000.00

Các mặt hàng đã mua

Tên Sản Phẩm	Số Lượng	Đơn Giá	Thành Tiền
xúc xích cp	1	50,000.00	50,000.00
Tổng cộng:			50,000.00

5. Khi quản lí hệ thống click “Thống kê” tại Trang chủ thì ứng dụng sẽ thống kê các thông tin dưới đây:

Thống Kê Hoạt Động

Số Loại Sản Phẩm

2

Hiện có trong kho

Tổng Lượng Tồn Kho

5

Tổng số lượng các SP

Giá Trị Tồn Kho (Giá Nhập)

135,600.00

Ước tính

Số Hóa Đơn

3

Đã được tạo

Tổng Lượng Đã Bán

4

Tổng số lượng các SP

Tổng Doanh Thu

150,320.00

Tổng tiền bán ra

Quay lại trang Bán Hàng

- Ở trang này sẽ thống kê các giá trị sản phẩm trong kho , số lượng đang tồn kho ở thời điểm hiện tại , giá trị hàng tồn kho , hiển thị số hóa đơn đã thực hiện mua hàng , tổng số lượng hàng đã bán và thống kê doanh thu .

6. Khi quản lí hệ thống click vào hóa đơn sẽ thống kê ra các hóa đơn đã mua như sau :

Danh sách Hóa Đơn Đã Lưu

ID Hóa Đơn	Tên Khách Hàng	Ngày Tạo	Tổng Tiền	Hành động
1	quang	31/10/2025 18:01:44	50,000.00	Xem Chi tiết Sửa Xóa
2	quang	31/10/2025 18:03:50	50,000.00	Xem Chi tiết Sửa Xóa
3	quang	03/11/2025 16:12:18	50,320.00	Xem Chi tiết Sửa Xóa

Quay lại trang Bán Hàng

Ở đây hệ thống sẽ giám sát và quản lí được trước đó khách hàng đã mua những gì ở cửa hàng dựa vào hóa đơn sẽ thống kê lại sản phẩm trong kho.

7. Quản lý tài khoản của người dùng:

Quản lý Người Dùng

[Thêm Người Dùng Mới](#)

ID	Tên Đăng Nhập	Vai Trò	Hành động
1	admin	ROLE_ADMIN	Sửa (Không thể tự xóa)
2	user	ROLE_USER	Sửa Xóa

[Quay lại trang Bán Hàng](#)

Phần này cho phép người quản lý có thể sửa hoặc xóa các thông tin tài khoản của người dùng để truy cập vào trang Quản lý siêu thị.

8. Lịch sử hoạt động hệ thống:

Lịch Sử Hành Động Hệ Thống

Thời gian	Người thực hiện	Hành động	Chi tiết
03/11/2025 16:12:18	user	CREATE_HOADON	ID Hóa đơn: 3, Khách hàng: quang, Tổng tiền: 50320.00, Chi tiết: [{SP: xúc xích cp, SL: 1, ĐG: 50000.00}, {SP: sữa milo, SL: 1, ĐG: 320.00}]
03/11/2025 16:04:37	admin	CREATE_PRODUCT	ID: 2, Tên: sữa milo, SL: 5, Giá nhập: 300.00, Giá bán: 320.00
31/10/2025 18:03:51	user	CREATE_HOADON	ID Hóa đơn: 2, Khách hàng: quang, Tổng tiền: 50000.00, Chi tiết: [{SP: xúc xích cp, SL: 1, ĐG: 50000.00}]
31/10/2025 18:01:45	admin	CREATE_HOADON	ID Hóa đơn: 1, Khách hàng: quang, Tổng tiền: 50000.00, Chi tiết: [{SP: xúc xích cp, SL: 1, ĐG: 50000.00}]
31/10/2025 18:00:04	admin	CREATE_PRODUCT	ID: 1, Tên: xúc xích cp, SL: 10, Giá nhập: 45000.00, Giá bán: 50000.00

[Quay lại trang Bán Hàng](#)

Thông qua lịch sử sẽ giúp người quản lý biết được thời gian mà khách hàng truy cập vào trang mua sắm vào lúc nào và khách hàng đã sử dụng gì.

KẾT LUẬN

Như vậy với đề tài thiết kế ứng dụng quản lý bán hàng siêu thị, nhóm chúng em đã hiểu được cách hình thành, vận dụng lập trình hướng đối tượng để đưa ra được cách xây dựng một ứng dụng mang tính khả thi và thực tiễn cao. Báo cáo của chúng em đề cập đến nghiên cứu và triển khai một hệ thống quản lý bán hàng siêu thị, nhằm giúp quản lý hoạt động kinh doanh của siêu thị một cách hiệu quả hơn. Mục tiêu hàng đầu của chúng em là xây dựng một hệ thống an toàn, thông minh, giúp người dùng dễ dàng tiếp cận và sử dụng mà không gặp phải những rắc rối không đáng có...

Chúng em hi vọng rằng hệ thống sẽ đáp ứng được nhu cầu của cả người dùng và các siêu thị. Chân thành cảm ơn sự quan tâm và hỗ trợ từ cô Nguyễn Lê Thu đối với dự án của chúng em, và chúng em sẵn sàng tiếp tục phát triển và cải thiện hệ thống để mang lại lợi ích tốt nhất cho cộng đồng. Do hạn chế về mặt thời gian và hiểu biết, đề tài của chúng em có thể còn nhiều thiếu sót, rất mong sẽ nhận được sự góp ý, bổ sung từ cô và các bạn để nhóm chúng em có thể hoàn thiện vốn kiến thức của mình, tạo hành trang vững chắc cho việc phát triển trong tương lai.

TÀI LIỆU THAM KHẢO

- [1]. PGS.TS. Đỗ Trung Tuấn,(2011), Đại học quốc gia Hà Nội, Chuyển đổi giữa Cơ sở dữ liệu quan hệ và XML.
- [2]. XML là gì? Tìm hiểu về XML <https://topdev.vn/blog/xml-la-gi/> , truy cập ngày 15/03/2024.
- [3]. VietNix(26/07/2022), Tìm hiểu mô hình MVC là gì? Ví dụ về cách sử dụng mô hình MVC <https://vietnix.vn/tim-hieu-mo-hinh-mvc-la-gi/> , truy cập ngày 15/03/2024.
- [4]. Nguyễn Thanh Tùng (2020). Giáo trình Lập trình hướng đối tượng với Java. NXB Thông tin và Truyền thông.
- [5]. Oracle (2024). The Java™ Platform, Standard Edition Documentation.
URL: <https://docs.oracle.com/javase/>
- [6]. Spring.io (2024). Spring Boot Reference Documentation.
URL: <https://docs.spring.io/spring-boot/docs/current/reference/html/>
- [7]. GeeksforGeeks (2024). Introduction to Object-Oriented Programming in Java.
URL: <https://www.geeksforgeeks.org/object-oriented-programming-oops-concept-in-java/>
- [8]. TopDev (2024). Giới thiệu về Spring Framework và Spring Boot.
URL: <https://topdev.vn/blog/spring-framework-la-gi/>