**BỘ THÔNG TIN VÀ TRUYỀN THÔNG**

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**KHOA CÔNG NGHỆ THÔNG TIN 2**

---------------------------

**Báo cáo bài tập**

**KIẾN TRÚC VÀ THIẾT KẾ PHẦN MỀM**

***Đề tài:***

**XÂY DỰNG MODULE ĐẶT HÀNG TRONG TRANG MUA QUẦN ÁO VỚI MÔ HÌNH EJB**

|  |  |
| --- | --- |
| **Giảng viên hướng dẫn:** | **ThS. NGUYỄN VĂN HỮU HOÀNG** |
| **Sinh viên thực hiện:** | **Nguyễn Nhật Minh – N21DCCN053** |
|  | **Trần Vũ Phương Nam – N21DCCN151** |
|  | **Đàm Tuấn Phát – N21DCCN171** |

*Thành phố Hồ Chí Minh, ngày 02 tháng 04 năm 2025*

LỜI CẢM ƠN

Lời đầu tiên, nhóm xin gửi lời cảm ơn chân thành đến **thầy Nguyễn Văn Hữu Hoàng**, đã tận tình giảng dạy, hướng dẫn và tạo điều kiện thuận lợi để nhóm có thể hoàn thành bài báo cáo cuối kỳ này. Sự hỗ trợ quý báu của thầy không chỉ giúp nhóm có thêm nhiều kiến thức bổ ích về **chuyên đề Công nghệ Phần mềm**, mà còn mang lại những kinh nghiệm thực tế trong quá trình nghiên cứu và phát triển sản phẩm.

Trong suốt quá trình thực hiện bài báo cáo, nhóm đã cố gắng hết sức để hoàn thiện sản phẩm và nội dung trình bày. Tuy nhiên, do giới hạn về thời gian và kinh nghiệm, nhóm không tránh khỏi những thiếu sót. Vì vậy, nhóm rất mong nhận được sự góp ý và phản hồi từ thầy để có thể chỉnh sửa, hoàn thiện bài báo cáo một cách tốt nhất.

*Thành phố Hồ Chí Minh, ngày 02 tháng 04 năm 2025*

MỤC LỤC

[CHƯƠNG 1. Giới thiệu đề tài 1](#_Toc194679186)

[1.1. Tên đề tài 1](#_Toc194679187)

[1.2. Mục tiêu 1](#_Toc194679188)

[1.3. Phạm vi và đối tượng nghiên cứu 1](#_Toc194679189)

[CHƯƠNG 2. Phân tích yêu cầu hệ thống 2](#_Toc194679190)

[2.1. Xác định tác nhân 2](#_Toc194679191)

[2.2. Yêu cầu chức năng 2](#_Toc194679192)

[2.3. Yêu cầu phi chức năng 3](#_Toc194679193)

[CHƯƠNG 3. Mô hình hóa hệ thống với UML 4](#_Toc194679194)

[3.1. Sơ đồ usecase tổng quát 4](#_Toc194679195)

[3.2. Sơ đồ hoạt động 5](#_Toc194679196)

[3.3. Sơ đồ tuần tự 8](#_Toc194679197)

[CHƯƠNG 4. Thiết kế hệ thống 9](#_Toc194679198)

[4.1. Kiến trúc hệ thống 9](#_Toc194679199)

[4.2. Thiết kế kiến trúc EJB 12](#_Toc194679200)

[4.3. Thiết kế giao diện 14](#_Toc194679201)

# Giới thiệu đề tài

## Tên đề tài

Đề tài: Xây dựng module đặt hàng trong trang mua quần áo với mô hình EJB.

Link GitHub: [github.com/nnminh-sam/clothing-store-ejb](https://github.com/nnminh-sam/clothing-store-ejb).

## Mục tiêu

Mục tiêu của đề tài là phát triển một module đặt hàng trong trang mua quần áo dựa trên kiến trúc Enterprise JavaBeans (EJB). Module này cho phép người dùng xem danh sách sản phẩm, xem chi tiết sản phẩm, thêm sản phẩm vào giỏ hàng và thực hiện thao tác đặt hàng.

Sản phẩm thành quả của đề tài sẽ là ứng dụng J2EE thỏa mãn các tiêu chí:

* Trang mua quần áo đơn giản sử dụng công nghệ thành phần EJB.
* Cung cấp tính năng đăng kí, đăng nhập tài khoản cho người dùng.
* Xem danh sách sản phẩm và chi tiết sản phẩm.
* Xem thông tin giỏ hàng, thêm sản phẩm vào giỏ hàng.
* Tạo đơn hàng từ sản phẩm trong giỏ hàng.

## Phạm vi và đối tượng nghiên cứu

**Phạm vi nghiên cứu:**

* Xây dựng ứng dụng J2EE sử dụng công nghệ thành phần EJB.
* Ứng dụng cho phép người dùng đặt hàng.
* Kết nối cơ sở dữ liệu với ứng dụng để lưu trữ thông tin.

**Đối tượng nghiên cứu:**

* Mô hình EJB trong J2EE.
* Công nghệ JPA, Hibernate, JTA.
* Hệ quản trị cơ sở dữ liệu quan hệ PostgreSQL.

# Phân tích yêu cầu hệ thống

## Xác định tác nhân

Hệ thống có một tác nhân: người dùng có các chức năng:

* Đăng kí, đăng nhập tài khoản với email và mật khẩu.
* Xem danh sách sản phẩm hiện có.
* Xem chi tiết sản phẩm.
* Xem giỏ hàng của người dùng.
* Thêm sản phẩm vào giỏ hàng.
* Đặt hàng.

## Yêu cầu chức năng

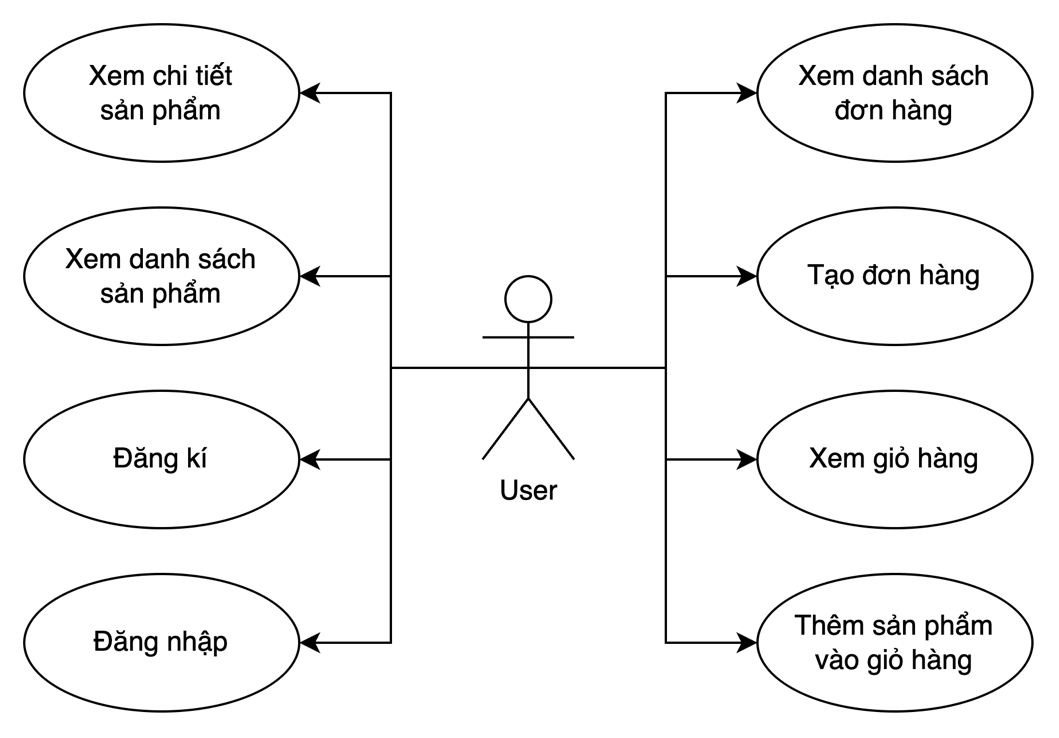
* Đăng kí tài khoản: Trang web sẽ cho người dùng đăng kí tài khoản thông qua biểu mẫu bao gồm địa chỉ email và mật khẩu.
* Đăng nhập: Người dùng sử dụng thông tin tài khoản đã đăng kí để đăng nhập vào trang mua hàng.
* Đăng xuất: Người sử dụng sau khi đăng nhập sẽ thực hiện đăng xuất thông qua nút đăng xuất trên trang web.
* Xem danh sách sản phẩm: Người sử dụng trang web có thể xem danh sách sản phẩm đang kinh doanh ngay khi truy cập vào trang web (bất kể chưa đăng nhập vào trang web).
* Xem chi tiết sản phẩm: Người sử dụng trang web có thể xem chi tiết sản phẩm đang kinh doanh khi bấm chọn một sản phẩm cụ thể (bất kể chưa đăng nhập vào trang web).
* Thêm sản phẩm vào giỏ hàng: Người sử dụng có thể thêm sản phẩm vào giỏ hàng thông qua nút “Thêm vào giỏ hàng” trong trang chi tiết sản phẩm sau khi chọn phân loại sản phẩm mong muốn.
* Xem giỏ hàng: Người sử dụng có thể xem giỏ hàng thông qua nút giỏ hàng trên trang web sau khi đăng nhập.
* Xem đơn đặt hàng: Người sử dụng có thể xem danh sách đơn đặt hàng đã đặt thông qua nút xem danh sách đơn hàng.

## Yêu cầu phi chức năng

* Hiệu suất: Trang web cần đảm bảo phản hồi nhanh chóng với người dùng. Hỗ trợ xem hình ảnh rõ ràng, giao diện thân thiện dễ dùng.
* Bảo mật: Áp dụng cơ chế mã hóa mật khẩu. Lưu trữ thông tin người dùng trong cơ sở dữ liệu để sử dụng lâu dài.

# Mô hình hóa hệ thống với UML

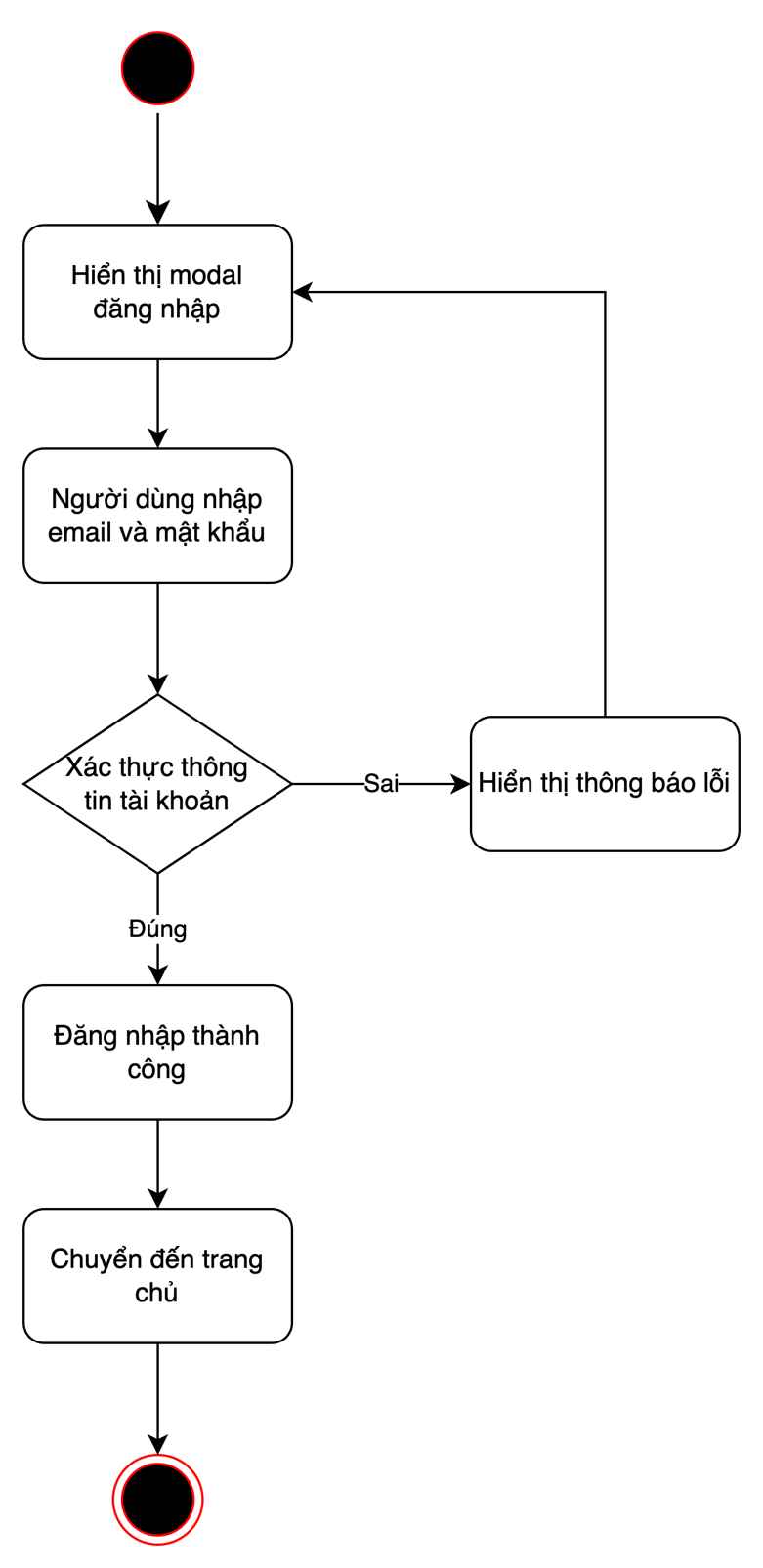
## Sơ đồ usecase tổng quát



Hình 1: Sơ đồ usecase toàn hệ thống

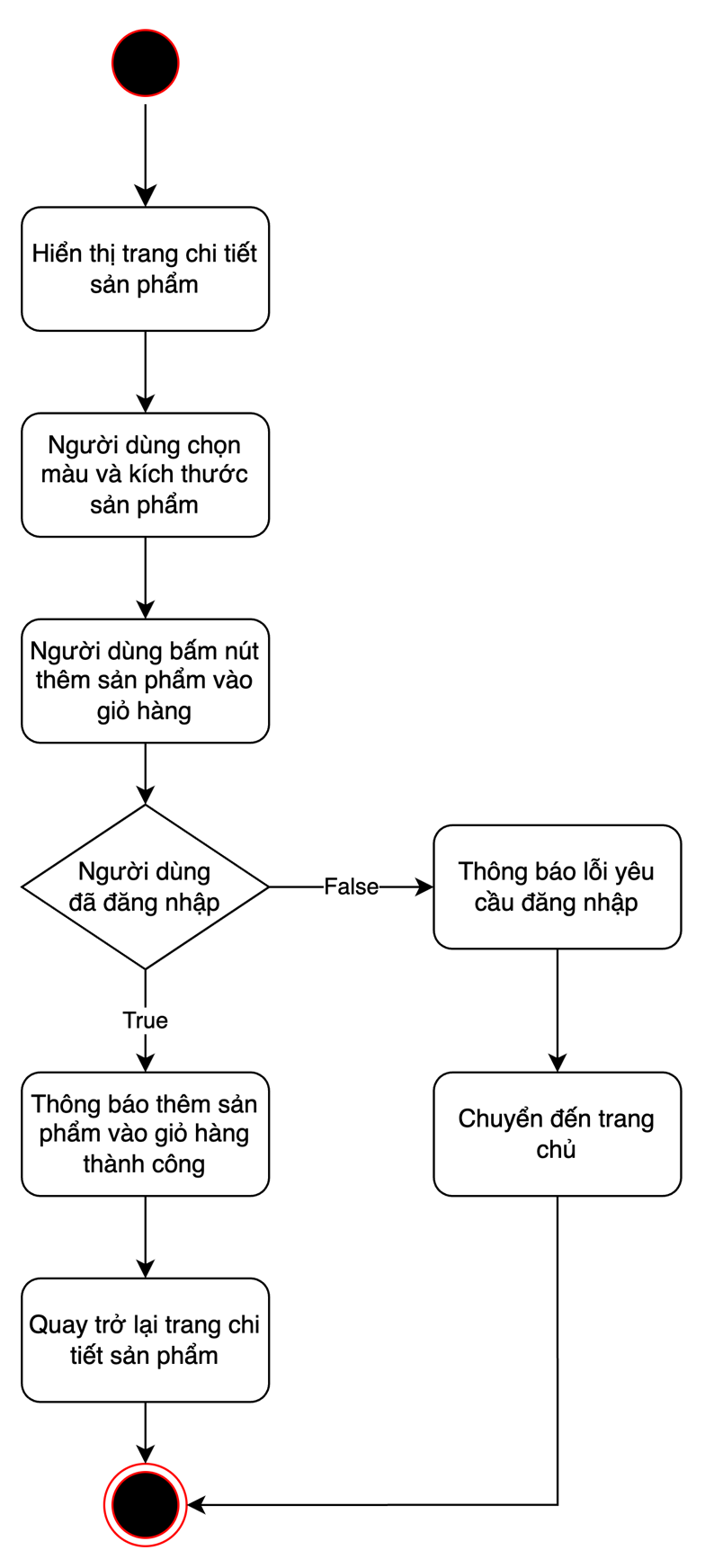
## Sơ đồ hoạt động

**Sơ đồ hoạt động đăng nhập:**



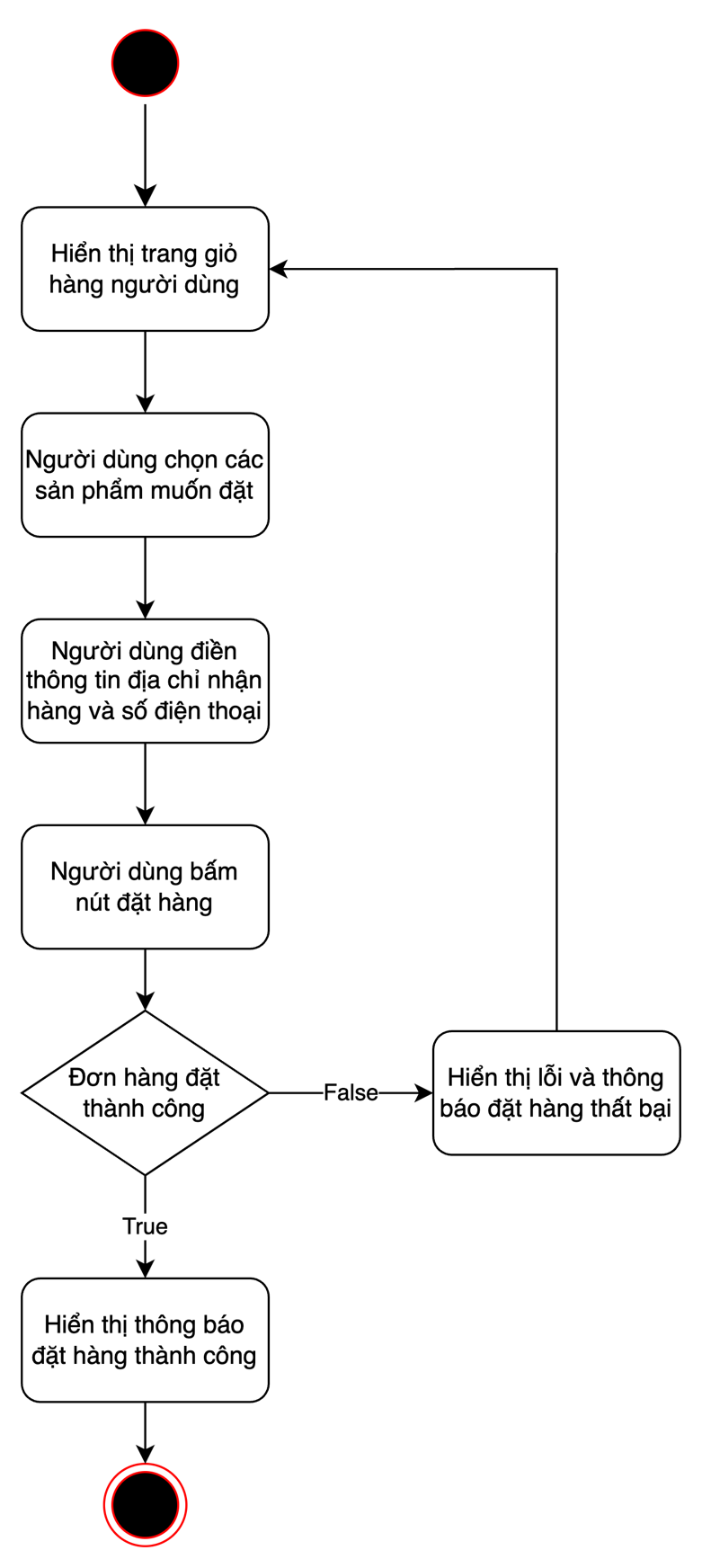
Hình 2: Sơ đồ hoạt động chức năng đăng nhập

**Sơ đồ hoạt động thêm sản phẩm vào giỏ hàng:**



Hình 3: Sơ đồ hoạt động chức năng thêm sản phẩm vào giỏ hàng

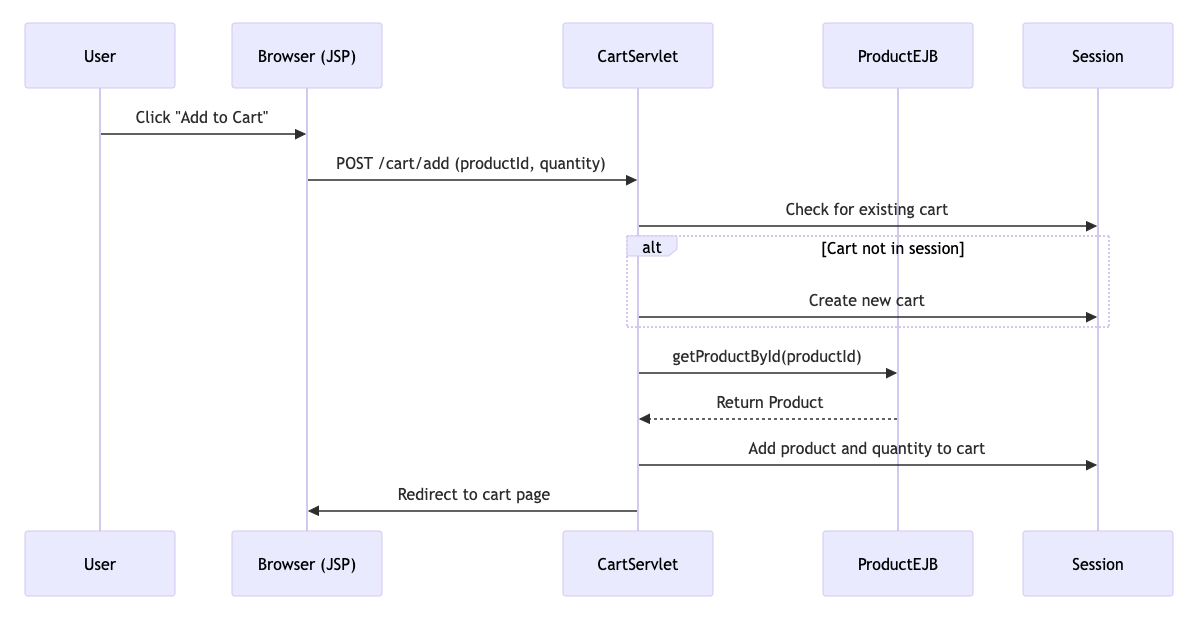
**Sơ đồ hoạt động đặt hàng:**



Hình 4: Sơ đồ hoạt động chức năng đặt hàng

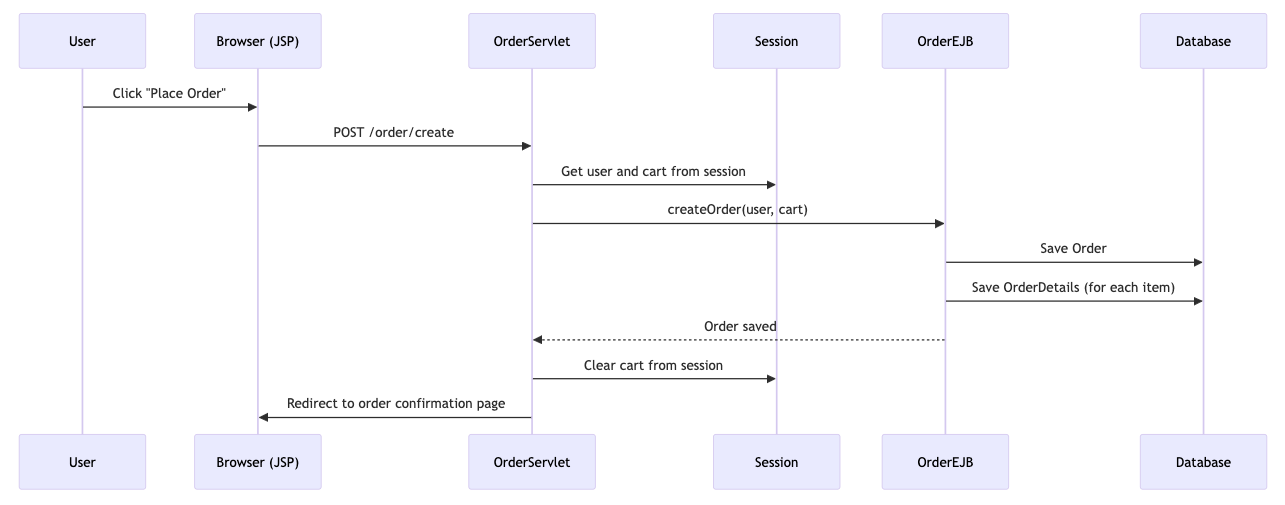
## Sơ đồ tuần tự

**Sơ đồ tuần tự chức năng thêm sản phẩm vào giỏ hàng:**



Hình 5: Sơ đồ tuần tự chức năng thêm sản phẩm vào giỏ hàng

**Sơ đồ tuần tự chức năng tạo đơn hàng:**

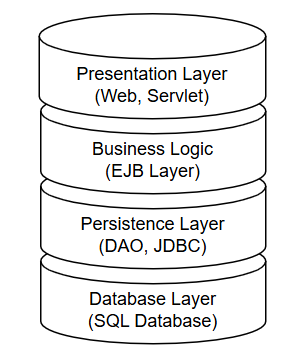


Hình 6: Sơ đồ tuần tự chức năng tạo đơn hàng

# Thiết kế hệ thống

## Kiến trúc hệ thống

**Mô hình kiến trúc:**



Hình 7: Kiến trúc hệ thống

**Công nghệ sử dụng:**

1. **Java EE (Java Enterprise Edition)**

Java EE (Java Enterprise Edition) J2EE (Java 2 Platform Enterprise Edition) là phiên bản Java dành cho doanh nghiệp, giúp cho việc lập trình ứng dụng được đơn giản hóa và giảm thiểu lập trình bằng cách tạo các thành phần module, có thể tái sử dụng và tiêu chuẩn hóa.

Các dịch vụ do J2EE Server cung cấp:

* + J2EE Server cung cấp nhiều dịch vụ quan trọng giúp quản lý và vận hành các ứng dụng doanh nghiệp hiệu quả. Một số dịch vụ chính bao gồm:
  + JNDI (Java Naming and Directory Interface):
    - JNDI là một API quan trọng cho phép các client tìm kiếm và xác định vị trí các thành phần EJB đã đăng ký trong container.
    - Nó cung cấp cơ chế đặt tên và tra cứu tài nguyên trong ứng dụng, giúp các thành phần trong hệ thống có thể giao tiếp và sử dụng lại tài nguyên một cách linh hoạt.
* Dịch vụ bảo mật và xác thực:
  + J2EE Server hỗ trợ các cơ chế bảo mật tiên tiến như xác thực (authentication), phân quyền (authorization), mã hóa dữ liệu, và quản lý danh tính người dùng.
  + Nó đảm bảo rằng chỉ những người dùng hoặc hệ thống được cấp quyền mới có thể truy cập vào các tài nguyên hoặc thực hiện các hành động nhất định trong ứng dụng.
* Hỗ trợ HTTP và các dịch vụ liên quan:
  + J2EE Server tích hợp sẵn dịch vụ HTTP để xử lý các yêu cầu từ trình duyệt web hoặc các ứng dụng client khác.
  + Nó hỗ trợ các công nghệ như Servlet, JSP, JSF, giúp phát triển và triển khai các ứng dụng web dễ dàng hơn.
* Hỗ trợ các dịch vụ EJB (Enterprise JavaBeans):
  + J2EE Server cung cấp môi trường thực thi cho EJB, giúp xử lý logic nghiệp vụ, quản lý giao dịch, bảo mật, và truy cập dữ liệu một cách tự động.
  + Nhờ đó, các nhà phát triển có thể tập trung vào logic nghiệp vụ mà không cần lo lắng về việc quản lý tài nguyên hệ thống.
* Hỗ trợ dịch vụ đa luồng (Multithreading):
  + J2EE Server cung cấp cơ chế xử lý đồng thời, cho phép nhiều tác vụ chạy song song mà không gây xung đột tài nguyên.
  + Điều này rất quan trọng đối với các ứng dụng có lưu lượng truy cập lớn hoặc cần xử lý nhiều yêu cầu cùng lúc.

Nhờ các dịch vụ trên, J2EE Server giúp các ứng dụng doanh nghiệp vận hành ổn định, bảo mật và hiệu suất cao, đồng thời giảm bớt gánh nặng cho lập trình viên trong việc quản lý tài nguyên hệ thống.

Dự án này là một ứng dụng web được xây dựng bằng công nghệ Java EE, sử dụng Enterprise Java Beans (EJB) để xử lý logic nghiệp vụ và quản lý dữ liệu. Ứng dụng được thiết kế theo mô hình module hóa, bao gồm các thành phần chính:

* Bean: Chịu trách nhiệm xử lý logic nghiệp vụ của hệ thống. Stateless Session Beans được sử dụng để tối ưu xử lý giao dịch và đảm bảo hiệu suất cao.
* JPA (Jakarta Persistence API) và Hibernate ORM để kết nối với hệ quản trị cơ sở dữ liệu PostgreSQL.
* Servlet: Tiếp nhận và xử lý các yêu cầu từ client. Giao tiếp với Bean để lấy dữ liệu từ PostgreSQL và trả về kết quả phù hợp.

1. **Intellij IDE**

IntelliJ IDEA là một môi trường phát triển tích hợp (IDE) được phát triển bởi JetBrains, nổi tiếng với khả năng hỗ trợ phát triển ứng dụng Java mạnh mẽ. Ngoài Java, IntelliJ IDEA còn hỗ trợ nhiều ngôn ngữ lập trình khác như Kotlin, Groovy, Scala, JavaScript, TypeScript, SQL, và các công nghệ web khác. IDE này được đánh giá cao nhờ vào giao diện thân thiện, khả năng hỗ trợ lập trình thông minh và tích hợp sâu với các công cụ hiện đại.

**Tính năng chính:**

1. Hỗ trợ thông minh cho Java và các ngôn ngữ JVM
   * Tự động hoàn thành mã, phân tích mã tĩnh, và refactor thông minh.
   * Hỗ trợ framework phổ biến như Spring, Hibernate, Java EE, Maven, Gradle.
2. Giao diện người dùng thân thiện và dễ sử dụng
   * Điều hướng mã nhanh, hỗ trợ tìm kiếm mạnh mẽ, tổ chức mã hợp lý.
   * Hệ thống cửa sổ, tab và layout linh hoạt.
3. Hỗ trợ phát triển web và frontend
   * Tích hợp HTML, CSS, JavaScript, React, Angular.
   * Kiểm tra và sửa lỗi trực tiếp trong quá trình viết mã.
4. Tích hợp công cụ và hệ sinh thái DevOps
   * Hỗ trợ Git, Docker, Kubernetes, CI/CD, và các dịch vụ cloud.
   * Có sẵn terminal tích hợp và hỗ trợ tương tác với cơ sở dữ liệu.
5. Khả năng mở rộng qua plugin
   * Kho plugin phong phú giúp mở rộng chức năng theo nhu cầu cá nhân hoặc dự án.

**Ưu điểm:**

* Hiệu suất cao, thông minh: IntelliJ cung cấp các tính năng tự động hóa và phân tích rất mạnh, giúp tiết kiệm thời gian và giảm lỗi.
* Hỗ trợ framework phong phú: Đặc biệt mạnh trong các ứng dụng Spring Boot, Hibernate, Java EE.
* Tích hợp tốt với hệ thống quản lý mã nguồn và build tool: Git, Maven, Gradle, giúp quy trình phát triển mượt mà.
* Giao diện hiện đại, dễ sử dụng: Người mới có thể làm quen nhanh, lập trình viên có kinh nghiệm có thể tối ưu hiệu suất làm việc.

**Nhược điểm:**

* Yêu cầu cấu hình máy cao: IntelliJ có thể ngốn nhiều RAM và CPU, đặc biệt với các dự án lớn.
* Bản Ultimate có phí: Phiên bản miễn phí (Community Edition) bị giới hạn tính năng so với bản Ultimate (mất phí).
* Đôi khi khởi động chậm: Nếu cài nhiều plugin hoặc làm việc với dự án lớn, tốc độ khởi động có thể bị ảnh hưởng.

**Kết luận:**

IntelliJ IDEA là một trong những IDE tốt nhất hiện nay dành cho lập trình viên Java và các ngôn ngữ JVM. Với hệ thống tính năng phong phú, khả năng tự động hóa thông minh, và hỗ trợ sâu rộng cho nhiều công nghệ, IntelliJ giúp tăng năng suất và trải nghiệm lập trình. Dù có một số điểm trừ về hiệu suất và chi phí, nhưng với những dự án quan trọng và lập trình viên chuyên nghiệp, IntelliJ IDEA vẫn là lựa chọn hàng đầu xứng đáng đầu tư.

1. **Wildfly Server**

WildFly là một máy chủ ứng dụng mã nguồn mở, mạnh mẽ và linh hoạt, được phát triển bởi Red Hat. Trước đây được biết đến với tên gọi JBoss Application Server, WildFly hỗ trợ đầy đủ các tiêu chuẩn Java EE (Jakarta EE), cho phép triển khai và quản lý các ứng dụng Java doanh nghiệp một cách hiệu quả. Với kiến trúc modular hiện đại, WildFly cho phép khởi động nhanh, sử dụng tài nguyên tối ưu và dễ dàng mở rộng hoặc tùy biến theo nhu cầu. Nó tích hợp nhiều tính năng như quản lý kết nối cơ sở dữ liệu, giao dịch phân tán, bảo mật, clustering và hỗ trợ phát triển microservices. Nhờ sự ổn định, khả năng mở rộng cao và cộng đồng hỗ trợ rộng lớn, WildFly là một lựa chọn lý tưởng cho các doanh nghiệp và nhà phát triển trong việc xây dựng các hệ thống ứng dụng quy mô lớn.

## Thiết kế kiến trúc EJB

1. **Thành phần trong EJB**
2. Session Bean

Session Bean đại diện cho một logic nghiệp vụ tạm thời, không lưu trạng thái lâu dài trong cơ sở dữ liệu. Có 3 loại:

* + Stateless Session Bean: Không lưu trạng thái giữa các lời gọi (ví dụ: tính toán, kiểm tra xác thực).
  + Stateful Session Bean: Lưu trạng thái giữa các tương tác với client (ví dụ: giỏ hàng).
  + Singleton Session Bean: Chia sẻ một phiên bản duy nhất trong toàn bộ ứng dụng.

1. Message-Driven Bean (MDB)

MDB được thiết kế để xử lý các thông điệp không đồng bộ từ hàng đợi JMS (Java Messaging Service). Chúng hoạt động như các listener, không tương tác trực tiếp với client.

1. Entity Bean (đã bị thay thế)

Entity Bean từng được dùng để đại diện cho dữ liệu trong cơ sở dữ liệu, nhưng hiện tạiđã bị thay thế bởi JPA (Java Persistence API) vì Entity Bean phức tạp và khó quản lý.

1. **Các thành phần khác trong kiến trúc EJB**
2. EJB Container

Đây là môi trường thực thi cho các EJB, do server cung cấp. Nó chịu trách nhiệm về vòng đời EJB, quản lý tài nguyên, bảo mật, giao dịch, và các dịch vụ khác. EJB Container giúp lập trình viên chỉ cần tập trung vào logic nghiệp vụ, không cần lo về các vấn đề nền tảng.

1. EJB Client

Client là thành phần sử dụng EJB để gọi các phương thức dịch vụ từ xa hoặc cục bộ. Client có thể là một ứng dụng Java, servlet, JSP, hoặc một ứng dụng bên ngoài.

1. Interfaces
   * Remote Interface: Dành cho client ở xa, giao tiếp qua mạng.
   * Local Interface: Dành cho client trong cùng JVM (ví dụ: trong cùng một ứng dụng web).
2. **Quản lý giao dịch trong EJB**

EJB hỗ trợ hai hình thức quản lý giao dịch:

* Container-Managed Transactions (CMT): Giao dịch được quản lý tự động bởi EJB Container. Lập trình viên chỉ cần chú thích bằng các annotation như @TransactionAttribute.
* Bean-Managed Transactions (BMT): Giao dịch được lập trình viên tự điều khiển bằng cách sử dụng API UserTransaction. Linh hoạt hơn nhưng phức tạp.

1. **Các dịch vụ EJB cung cấp**

* Quản lý vòng đời Bean
* Quản lý giao dịch
* Quản lý bảo mật
* Quản lý đồng thời (Concurrency)
* Quản lý tài nguyên (connection pooling, thread pooling)
* Hỗ trợ truy cập từ xa
* Tích hợp dễ dàng với JMS, JPA, JTA

1. **Ưu điểm của EJB**

* Tách biệt rõ logic nghiệp vụ và các chức năng nền tảng
* Tự động quản lý giao dịch và bảo mật
* Tái sử dụng và mở rộng dễ dàng
* Tích hợp tốt với các công nghệ Java EE khác
* Hỗ trợ phát triển các ứng dụng phân tán, đa tầng

1. **Nhược điểm của EJB**

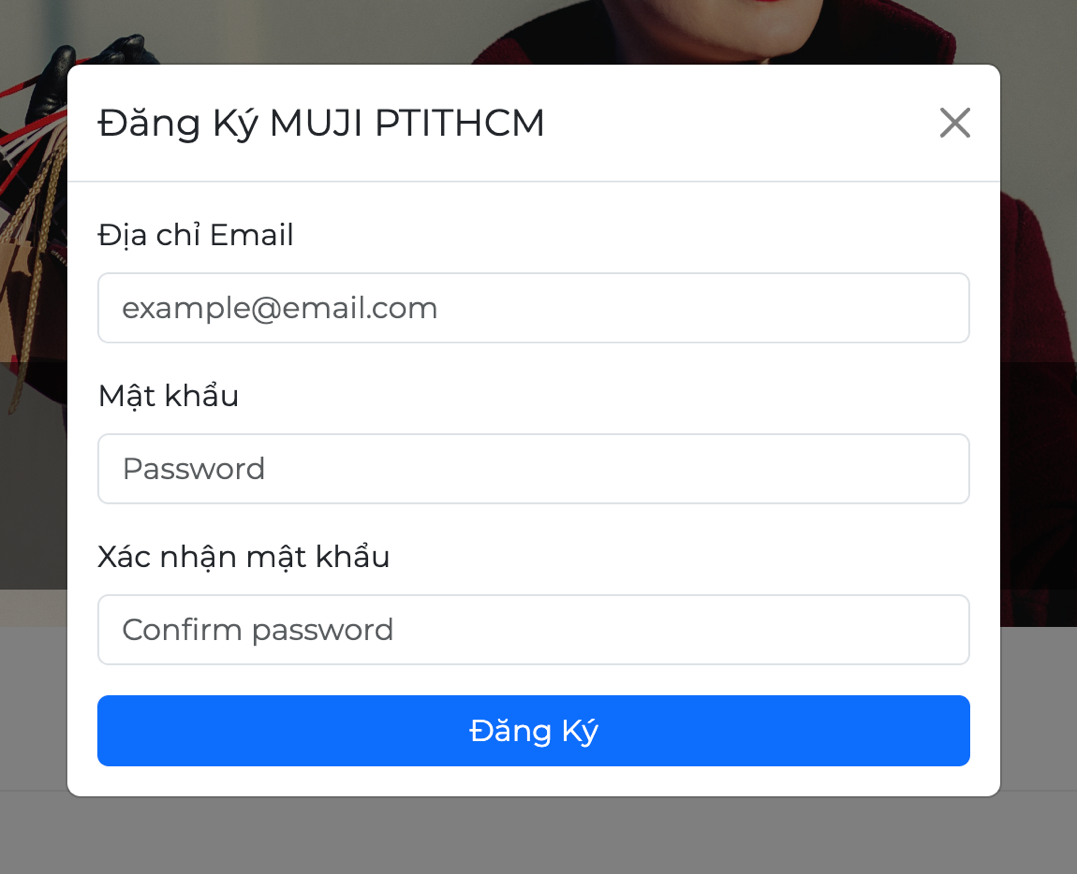
* Phức tạp và khó học đối với người mới
* Tốn nhiều tài nguyên hệ thống (nặng nề hơn so với Spring hoặc microservices nhẹ)
* Triển khai và cấu hình đôi khi rườm rà
* Quá nhiều annotation và khái niệm có thể gây khó hiểu

1. **Kết luận**

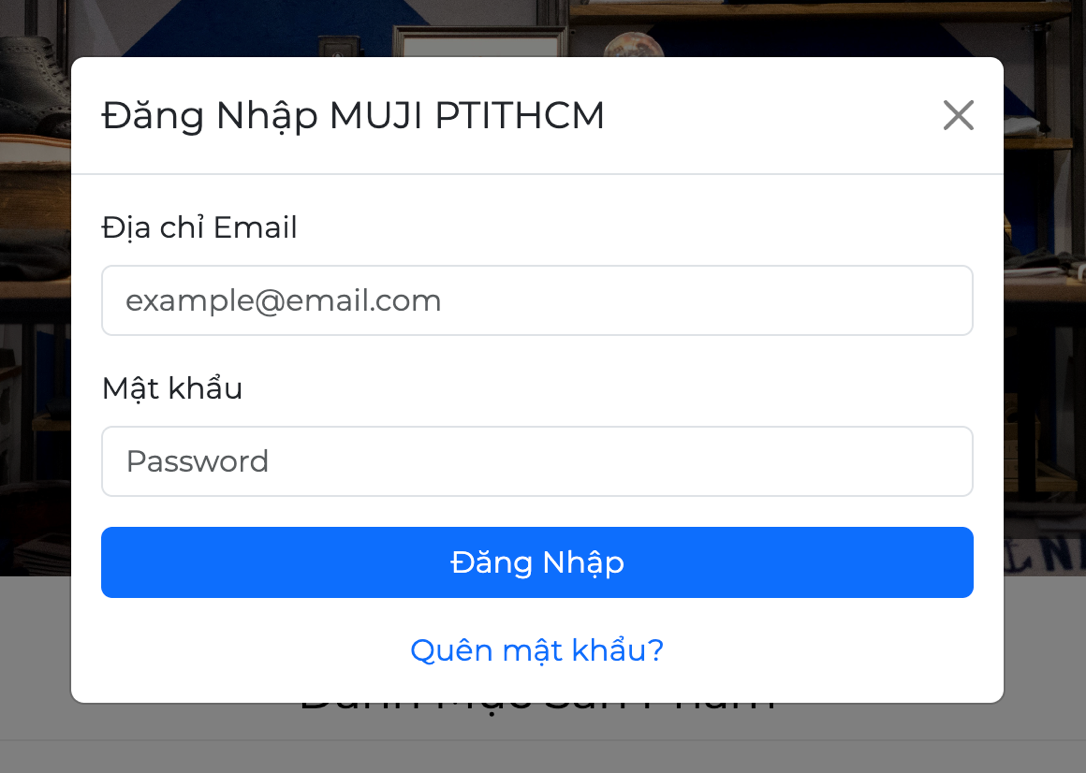
EJB là một mô hình mạnh mẽ để phát triển ứng dụng doanh nghiệp, đặc biệt khi cần xử lý logic nghiệp vụ phức tạp, bảo mật và giao dịch. Tuy nhiên, do tính chất phức tạp và yêu cầu tài nguyên cao, EJB ngày càng được thay thế bởi các giải pháp nhẹ hơn như Spring Framework, đặc biệt trong các ứng dụng microservices hiện đại. Tuy vậy, trong các hệ thống doanh nghiệp lớn, EJB vẫn là một giải pháp đáng tin cậy và chuẩn mực.

## Thiết kế giao diện

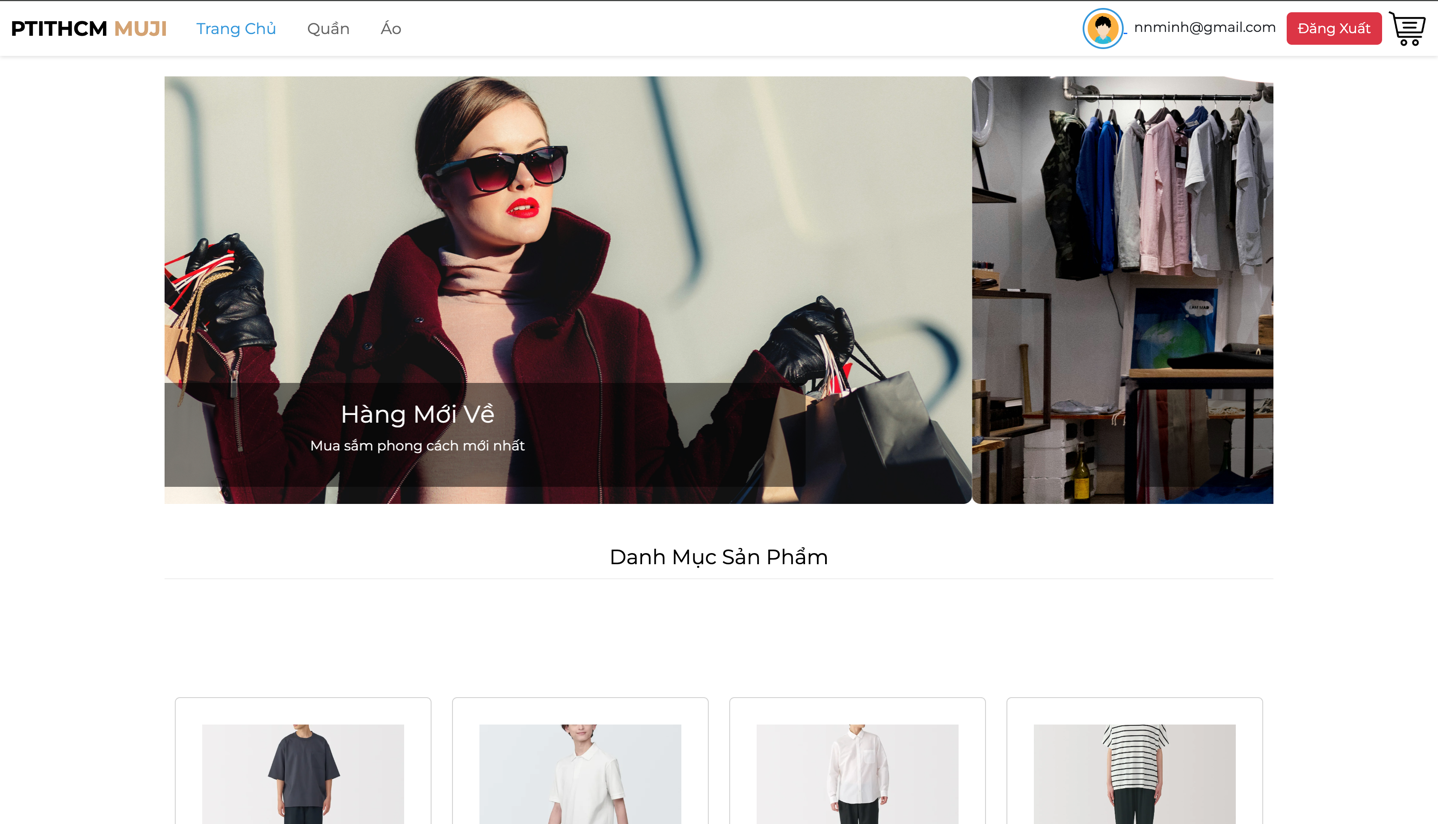
Dưới đây là một số hình ảnh về sản phẩm:



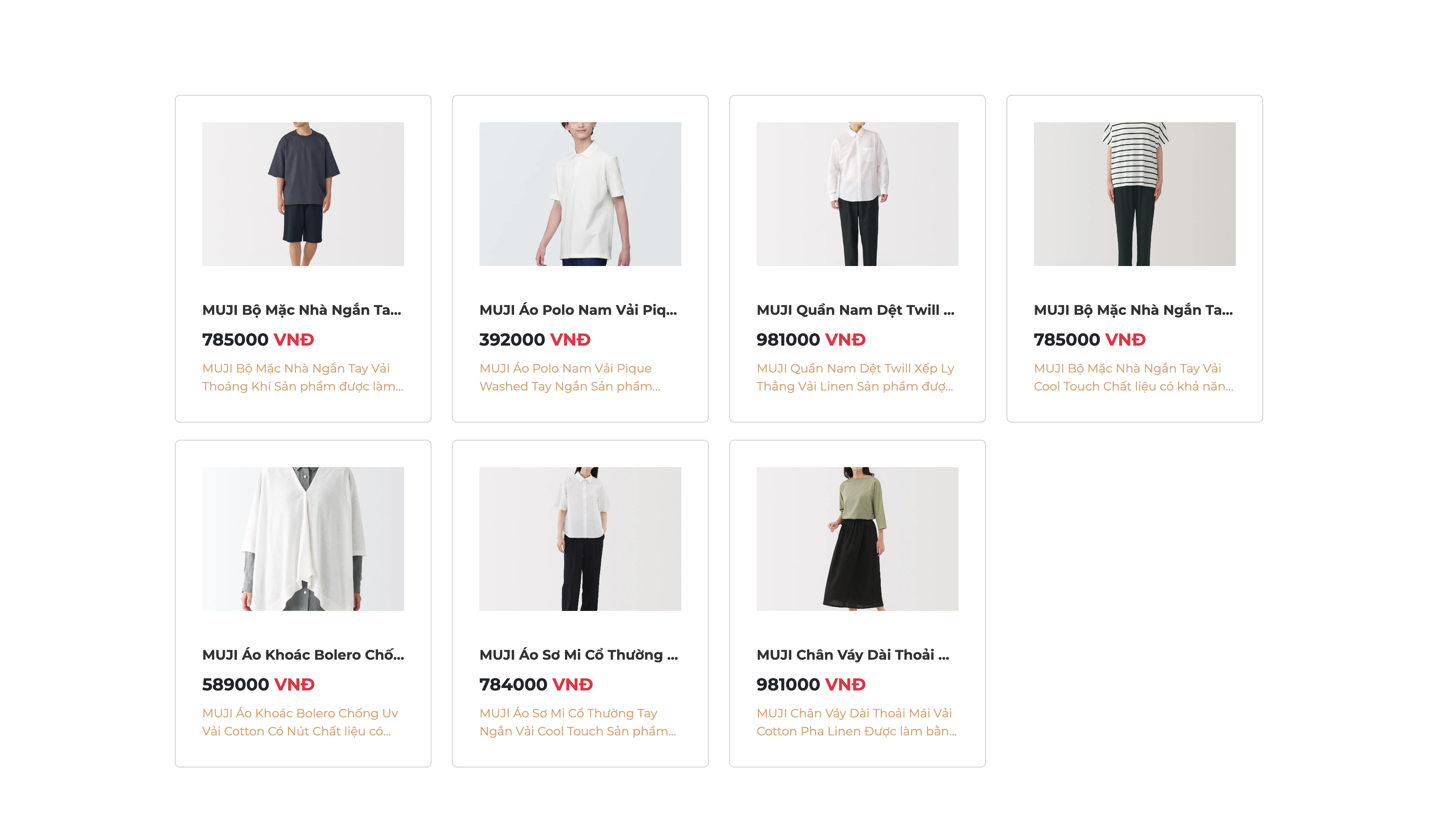
Hình 8: Giao diện đăng kí



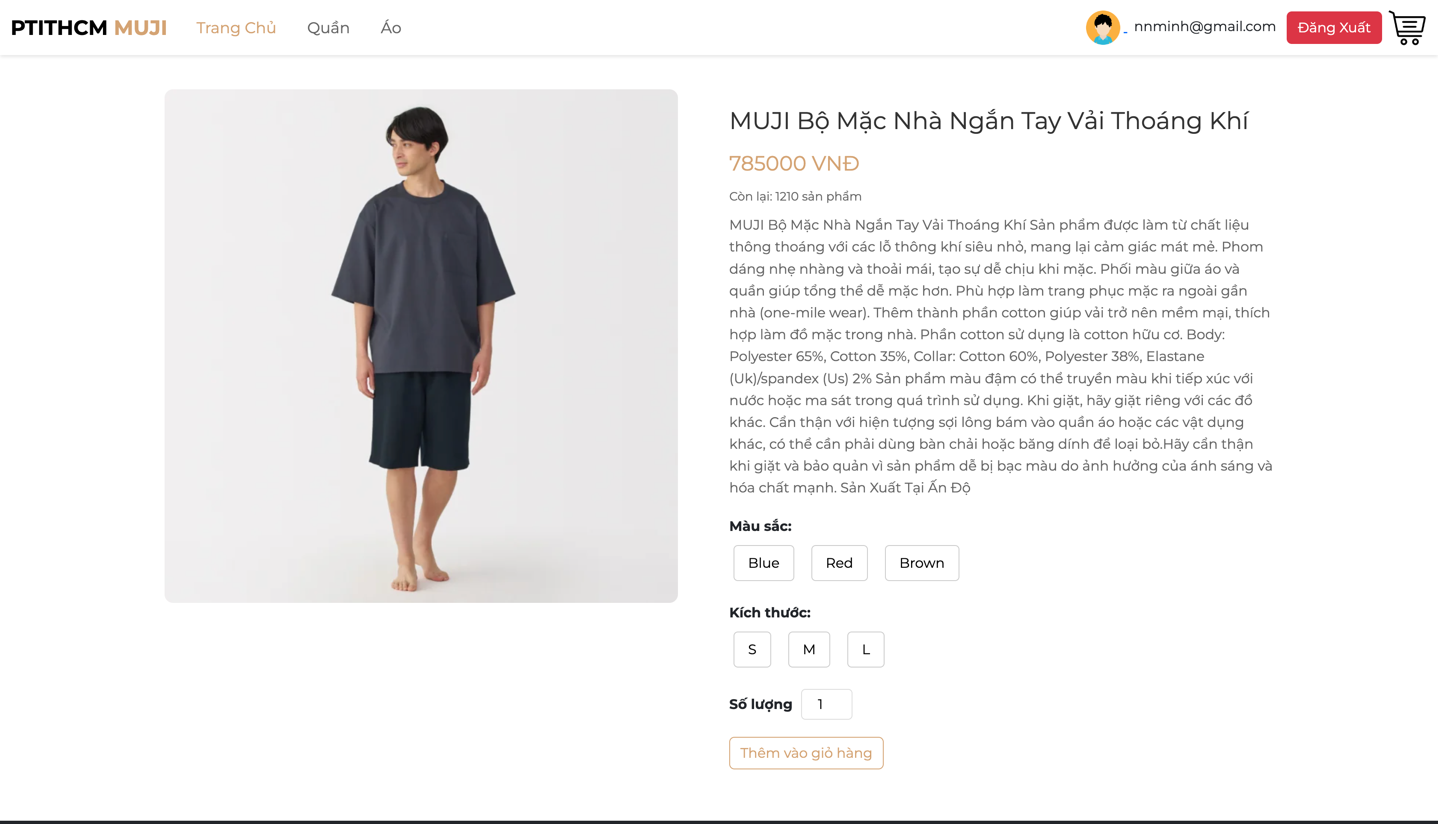
Hình 9: Giao diện đăng nhập



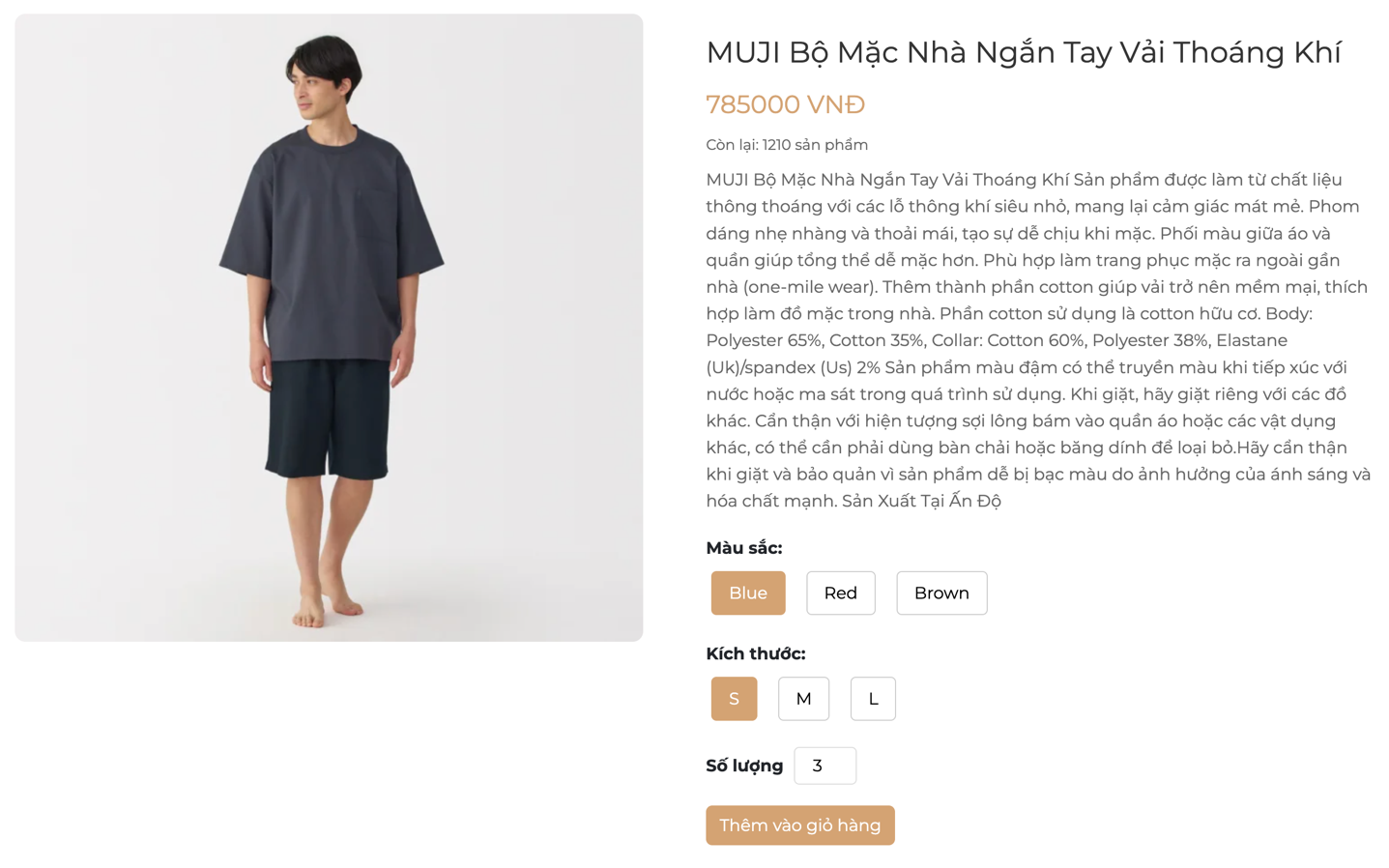
Hình 10: Giao diện trang chủ



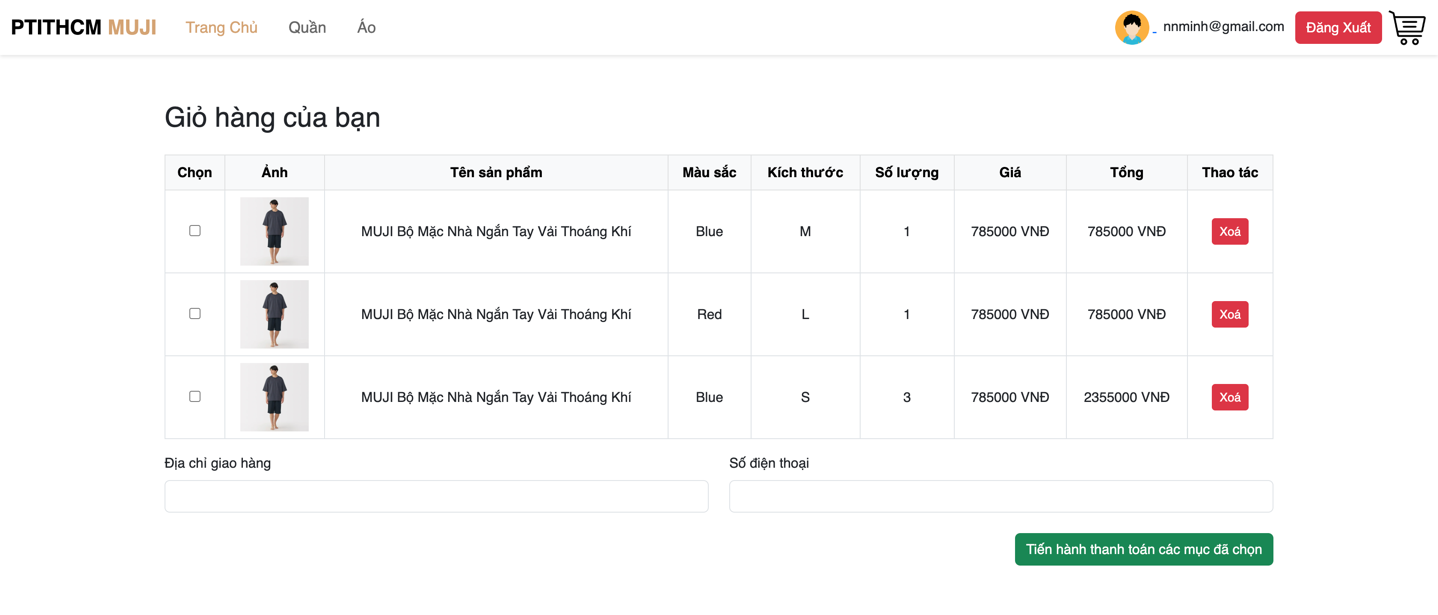
Hình 11: Giao diện xem danh sách sản phẩm



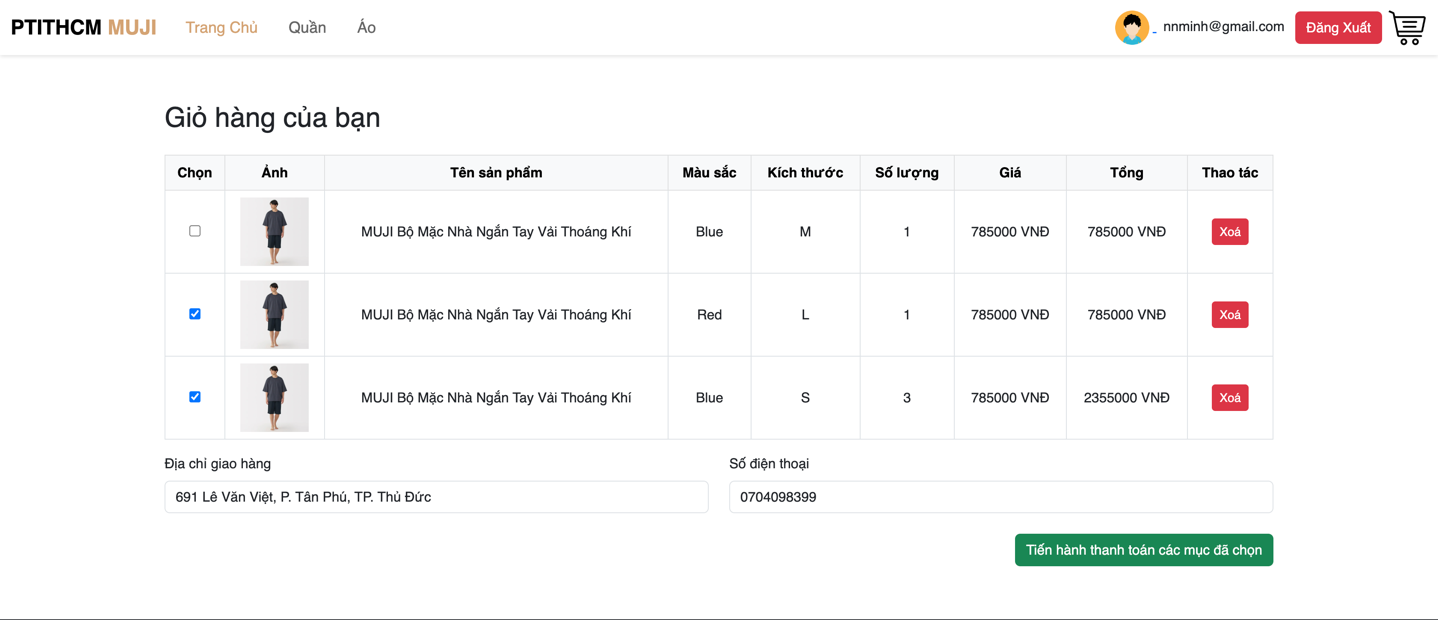
Hình 12: Giao diện chi tiết sản phẩm



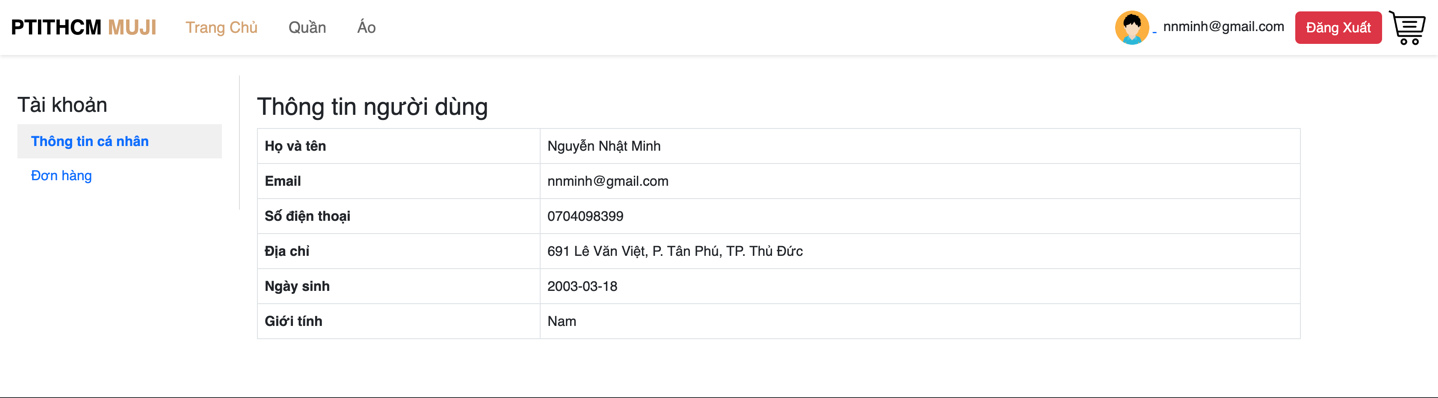
Hình 13: Giao diện chi tiết sản phẩm khi chọn màu và kích thước và bấm nút thêm vào giỏ hàng



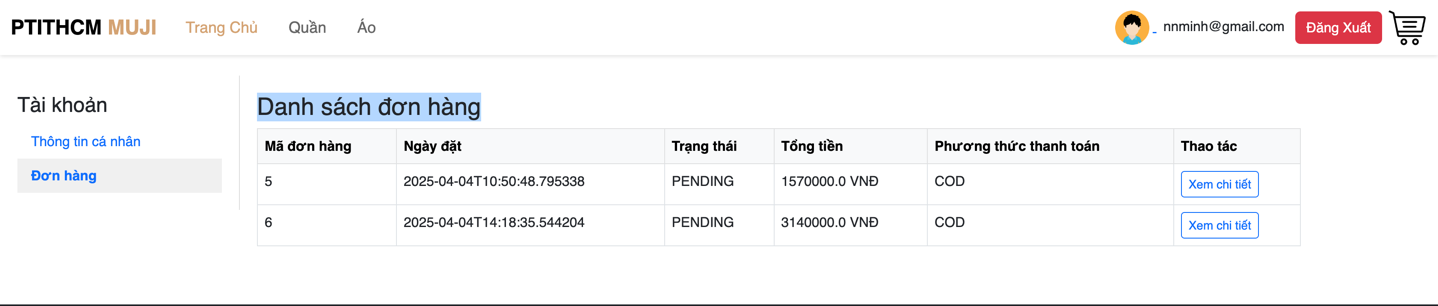
Hình 14: Giao diện giỏ hàng



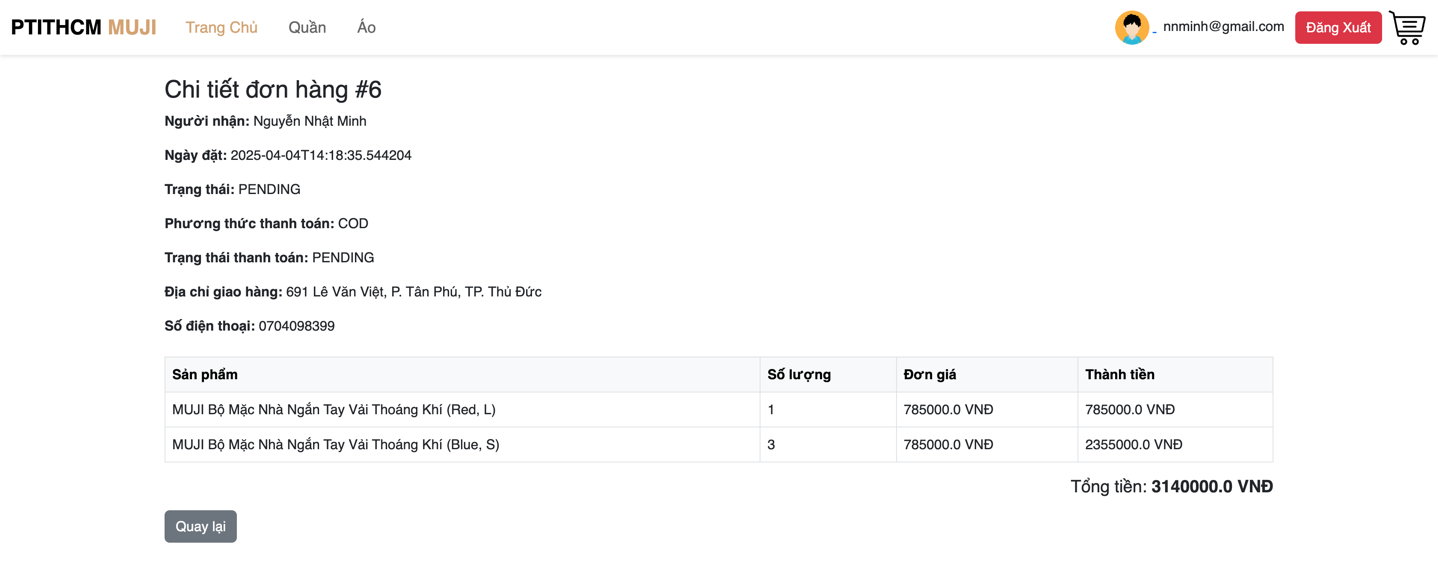
Hình 15: Giao diện giỏ hàng chi chọn sản phẩm để tạo đơn hàng



Hình 16: Giao diện chi tiết thông tin người dùng



Hình 17: Giao diện danh sách đơn hàng



Hình 18: Giao diện chi tiết đơn hàng