

HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

SCHOOL OF INFORMATION  
COMMUNICATION TECHNOLOGY



**SOICT**

**GRADUATION RESEARCH 1**

---

**E-COMMERCE WEBSITE SELLING TECHNOLOGY  
PRODUCTS**

---

**INSTRUCTOR:** Trần Hải Anh

**Student:**

Phạm Trường Sang – 20225996

## Table of Contents

<b>I.</b>	<b>Introduction .....</b>	5
1.	<b>Context and Motivation.....</b>	5
2.	<b>Project Objectives .....</b>	5
3.	<b>Project Scope.....</b>	6
<b>II.</b>	<b>System Technology Stack .....</b>	6
1.	<b>Overall System Architecture.....</b>	6
2.	<b>Backend Environment (Server-side).....</b>	6
3.	<b>Frontend Environment (Client-side).....</b>	6
4.	<b>Data Management &amp; Storage .....</b>	7
5.	<b>Integration &amp; Communication Services .....</b>	7
<b>III.</b>	<b>System Analysis &amp; Design.....</b>	7
1.	<b>Requirement Analysis.....</b>	7
2.	<b>Use Case Diagram .....</b>	7
3.	<b>Database Design (ERD) .....</b>	9
4.	<b>Activity Diagrams.....</b>	10
5.	<b>Sequence Diagrams.....</b>	16
6.	<b>Class Diagrams.....</b>	19
<b>IV.</b>	<b>Implementation .....</b>	20
1.	<b>System Architecture and Project Structure .....</b>	20
2.	<b>Extension Feature .....</b>	21
<b>V.</b>	<b>Result.....</b>	22
1.	<b>Home Page &amp; Product Catalog.....</b>	22
2.	<b>Product Detail Page .....</b>	23
3.	<b>Shopping Cart Page.....</b>	24
4.	<b>Checkout Page.....</b>	26
5.	<b>Login Page.....</b>	27
6.	<b>Register Page .....</b>	28
7.	<b>Order History Page.....</b>	31
8.	<b>VNPAY Gateway (Sandbox) .....</b>	32
9.	<b>User Profile Page.....</b>	34
<b>VI.</b>	<b>Discussion.....</b>	35
1.	<b>Limitations and Missing Features.....</b>	35

<b>2. Experiment Gains .....</b>	36
<b>VII. Conclusion .....</b>	36

## **Abstract:**

This report presents the design and implementation of a technology-focused e-commerce website. The project aims to develop a user-friendly platform that supports the online sales and management of technology products. By applying modern web technologies, the system is designed to be efficient, scalable, and capable of meeting current e-commerce requirements while enhancing the overall user experience.

# I. Introduction

## 1. Context and Motivation

With the rapid advancement of technology and the growing popularity of online shopping, businesses are increasingly required to adopt modern strategies to promote and sell their products more efficiently than traditional methods. Online shopping allows customers to search for and purchase products conveniently through smartphones or other digital devices without the need to visit physical stores. As a result, e-commerce platforms have become an essential solution for both businesses and consumers in the digital era.

This trend is particularly evident in the technology product market, where customers often seek convenience, detailed product information, and quick access to a wide range of items. An e-commerce website enables users to browse, compare, and purchase various technology products with just a few clicks, making the shopping process more efficient and accessible regardless of time and location.

However, building a trusted e-commerce platform requires more than just displaying products, it demands robust security measures and reliable payment integration to protect user data and financial transactions. This project is motivated by the need to create a complete, secure, and modern e-commerce solution that addresses these challenges.

## 2. Project Objectives

The primary goal of this project is to design and develop a comprehensive full-stack e-commerce web application tailored for technology products. The specific objectives are as follows:

- **Full-stack System Development:** To construct a complete system architecture including a responsive Frontend (ReactJS), a robust Backend (Spring Boot), and a relational Database (PostgreSQL).
- **Adoption of Modern Technologies:** To implement the latest software standards by utilizing Java 22 and Spring Boot 3.4, ensuring the system is high-performing, scalable, and secure.
- **Real-world Payment Integration:** To solve the challenge of online transactions by integrating the VNPay Payment Gateway, allowing users to perform secure electronic payments with real-time verification (IPN).

- **Enhanced Security:** To implement a multi-layer security mechanism using Spring Security and JWT (JSON Web Token), combined with verification token via Email during registration and password recovery processes to ensure user authenticity.

### 3. Project Scope

The project focuses on the Business-to-Consumer (B2C) model, facilitating the entire life cycle of e-commerce transactions. The scope encompasses two main functional areas: the Storefront for customers and the Back-office for management.

- **Target Audience:**
  - **End-Users (Customers):** Individual retail customers looking to search, compare, and purchase technology devices online.
  - **Administrators & Staff (Admin/PM):** System owners and product managers responsible for managing inventory, processing orders, and overseeing user accounts.

## II. System Technology Stack

### 1. Overall System Architecture

The system follows a decoupled Client-Server architecture, ensuring independent development and deployment of the frontend and backend components.

### 2. Backend Environment (Server-side)

- **Java Development Kit (JDK) 22:** The primary execution environment for the server-side application.
- **Spring Boot Framework 3.4.0:** The core framework providing managed infrastructure for the web layer, dependency injection, and data persistence.
- **Spring Security:** A framework focusing on providing both authentication and authorization to the Java application.
- **JSON Web Token (JWT):** Used for securely transmitting information between parties as a JSON object, facilitating stateless authentication.

### 3. Frontend Environment (Client-side)

- **ReactJS:** A JavaScript library for building user interfaces based on a component-oriented approach.

- **Axios:** A promise-based HTTP client for the browser used to interact with the Backend RESTful APIs.

## 4. Data Management & Storage

- **PostgreSQL:** The primary relational database management system (RDBMS) used for persistent data storage.
- **Spring Data JPA:** A part of the larger Spring family that makes it easy to easily implement JPA-based repositories.

## 5. Integration & Communication Services

- **VNPAY Gateway API:** The external payment interface used to process online transactions.
- **SMTP (Simple Mail Transfer Protocol):** The protocol used for automated email communication.

# III. System Analysis & Design

## 1. Requirement Analysis

Based on the project goals, the system must support three main user roles with specific functional requirements:

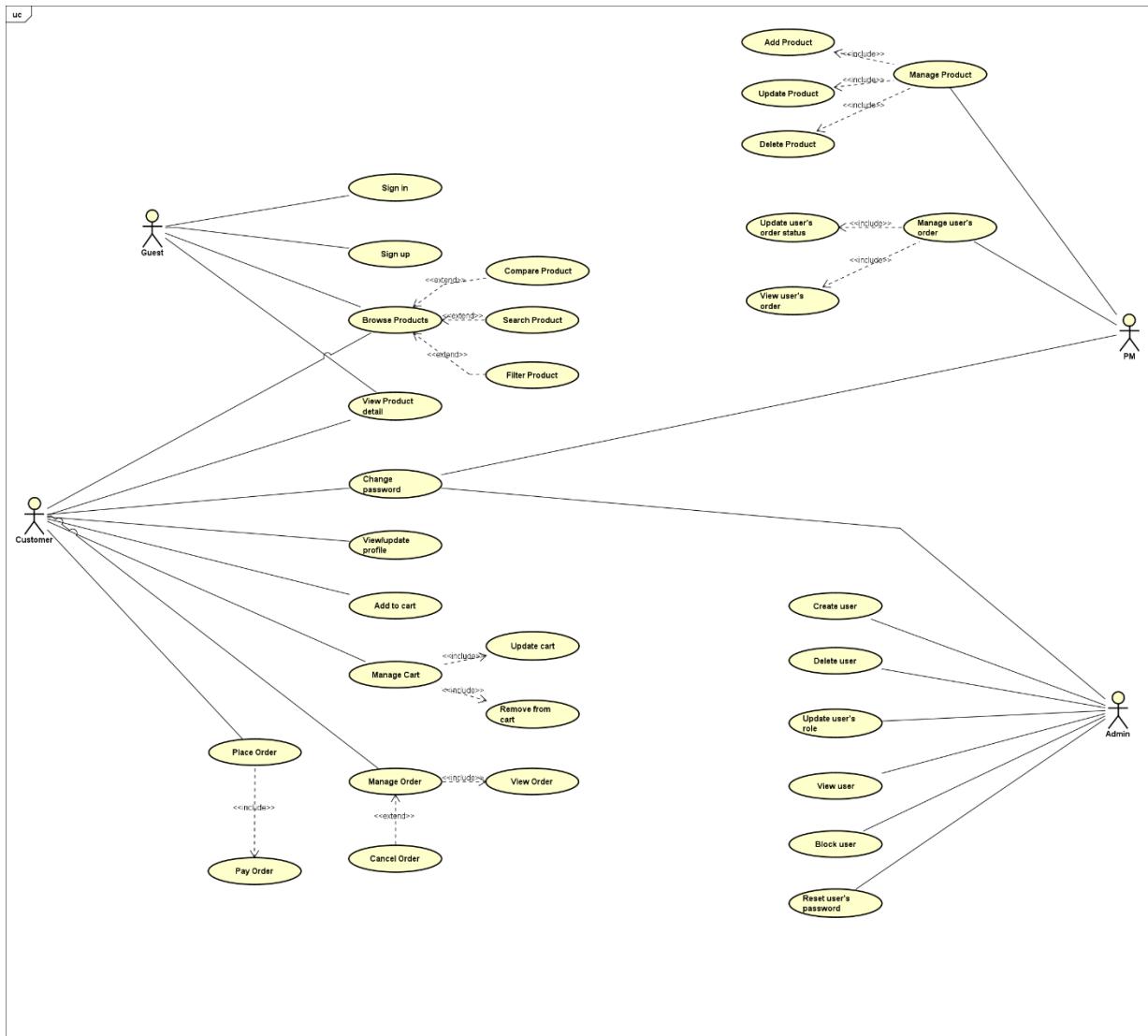
- **Customer:**
  - Browse and search for technology products by name, category, price, or filter by product components.
  - Manage personal shopping carts (Add, update quantity, remove items).
  - Secure checkout with delivery information and integrated payment via VNPAY.
  - Track order history and cancel pending orders.
  - Register with secure email-based token verification and login, change password.
- **Product Manager (PM):**
  - Manage product catalog (CRUD operations: Create, Read, Update, Delete).
  - Monitor inventory levels and update product availability.
  - Oversee all system orders and payment transactions.
- **Administrator (Admin):**
  - Manage user accounts and roles.

## 2. Use Case Diagram

The Use Case diagram provides a visual representation of the system's functional requirements, showing the relationships between actors and the various actions they can perform within the e-commerce platform.

The system identifies four primary actors, each with distinct responsibilities and access levels:

- **Guest:** An unauthenticated user who can browse products, search, compare, and sign up for an account.
- **Customer:** A registered user with full access to shopping features, including cart management, checkout, order tracking, and profile updates.
- **Product Manager (PM):** A staff member responsible for catalog maintenance, including adding, updating, and deleting technology products, as well as managing user orders.
- **Administrator (Admin):** The system owner with high-level privileges to manage user accounts, roles, and administrative tasks like resetting passwords or blocking users.



### 3. Database Design (ERD)

The system utilizes a relational database structure to ensure data integrity and efficient querying.

#### User & Security Management:

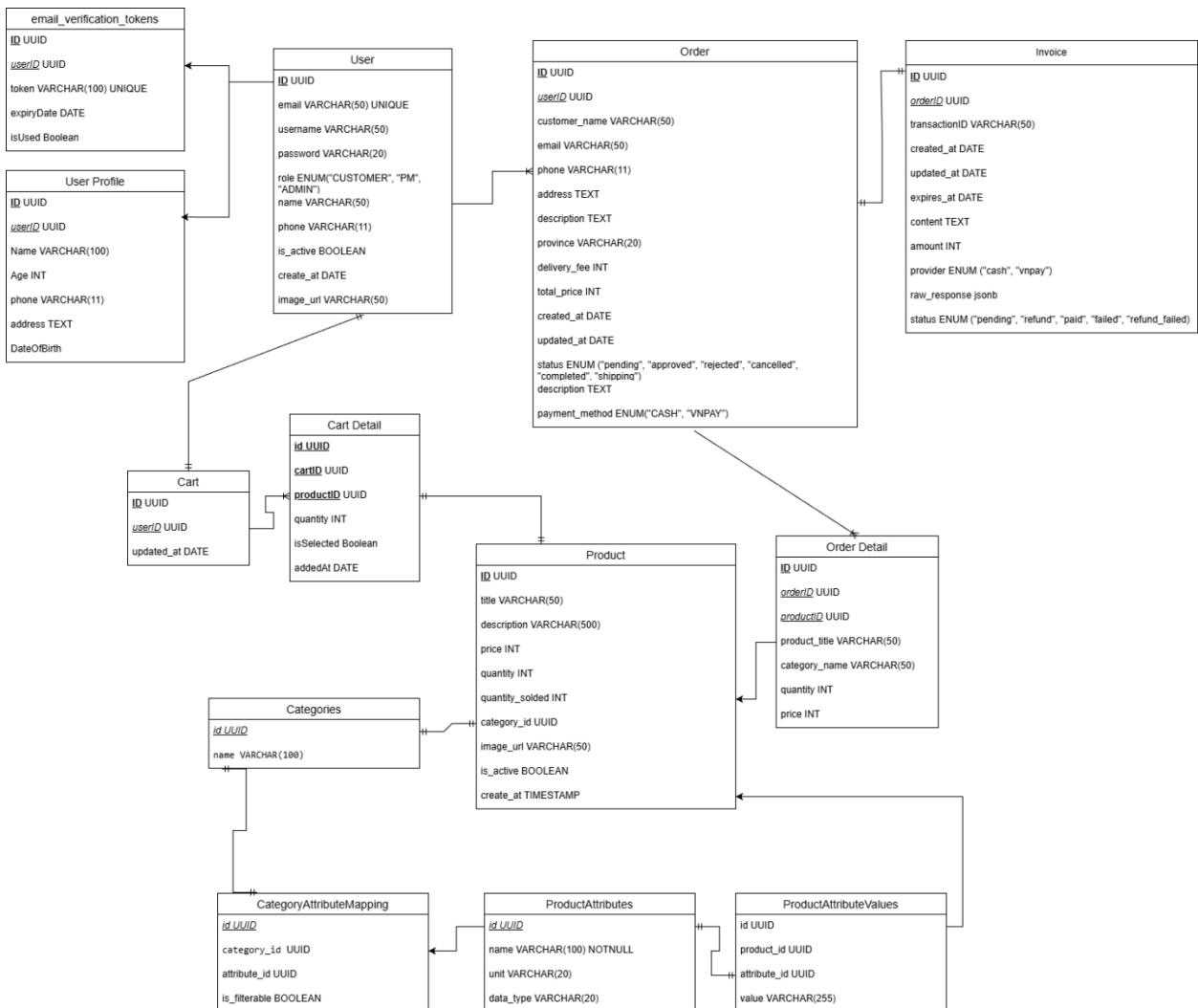
- **User:** Stores core credentials including hashed passwords, roles (Customer, PM, Admin), and the account activation status (is\_active).
- **Email Verification Tokens:** A specialized table to manage Tokens for registration and password recovery, including expiryDate and isUsed flags to prevent token reuse.
- **User Profile:** Separated from the core User table to store extended personal information like address, phone, and dateOfBirth.

#### Product & Catalog System:

- **Product:** The central entity for inventory, including pricing, quantity in stock, and images.
- **Categories:** Organizes products into logical groups.
- **Attributes & Values (EAV Model):** This advanced design (ProductAttributes, ProductAttributeValue) allows for flexible product specifications without modifying the table schema.

#### Shopping Cart & Ordering:

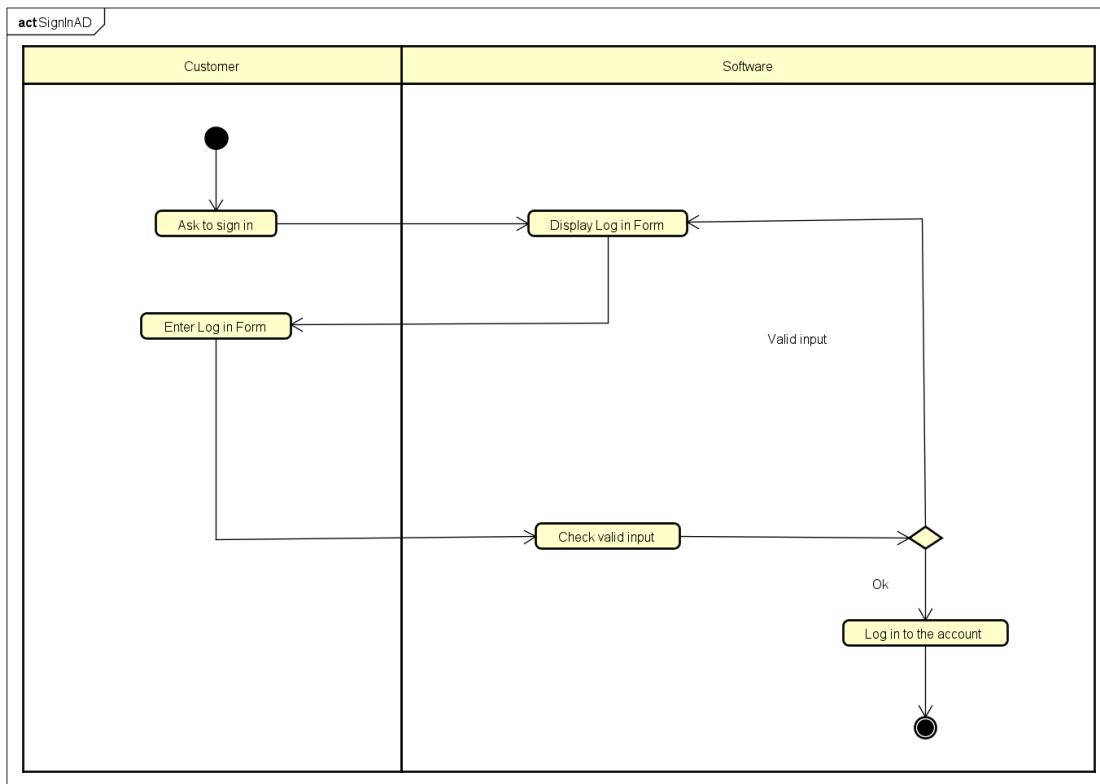
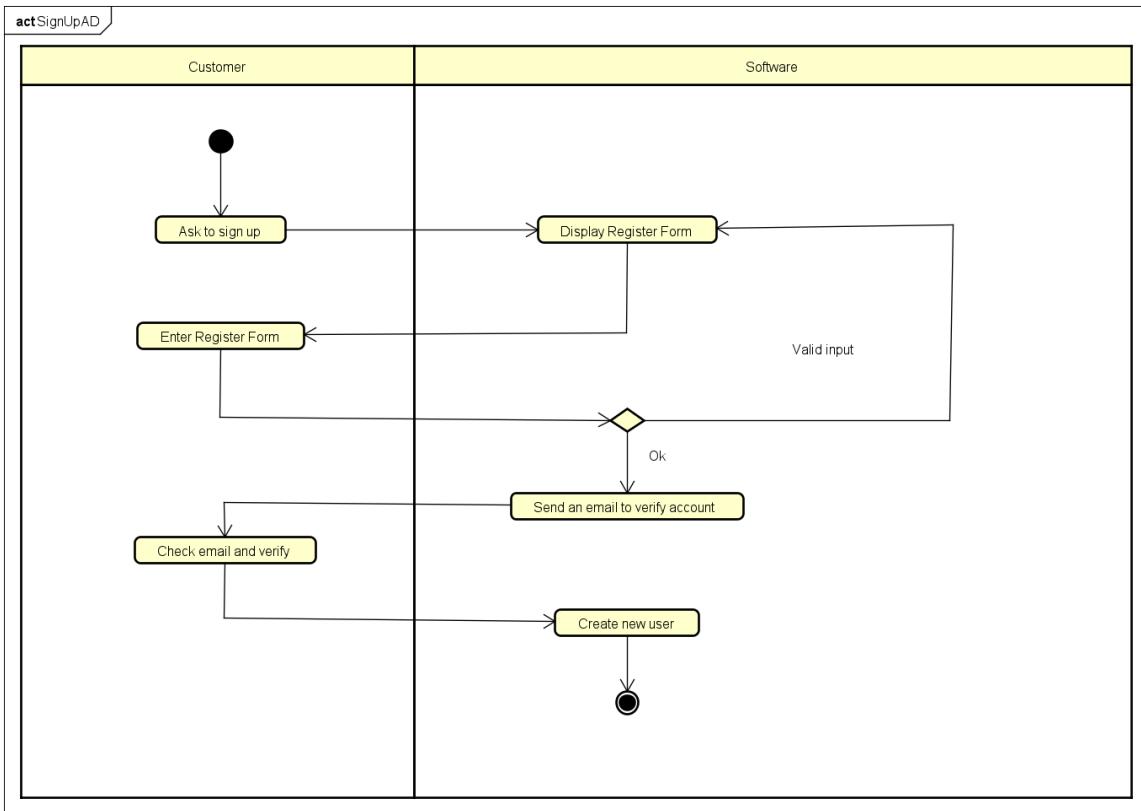
- **Cart & Cart Detail:** Manages items added by customers before purchase, including a flag (isSelected) to support partial checkout.
- **Order & Order Detail:** Captures a snapshot of the transaction, including shipping addresses, total price, and the status of the order.
- **Invoice:** Linked to the Order to track specific payment transactions, including transactionID from VNPay, payment provider details, and raw responses for auditing purposes.



## 4. Activity Diagrams

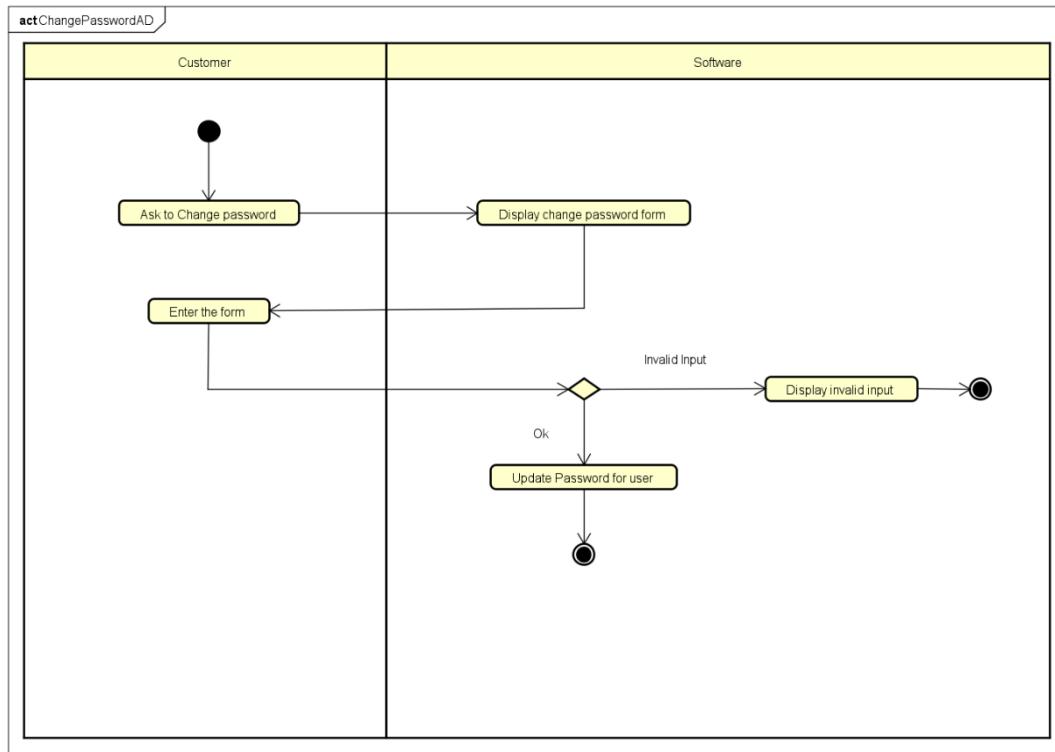
- **Authentication Process (Sign In & Sign Up):**

- **Sign In Flow:** The process begins when the customer requests to sign in. The software displays a login form, and upon submission, it validates the input credentials. If the information is correct, the system logs the user into their account; otherwise, it prompts for re-entry.
  - **Sign Up Flow:** When a guest asks to sign up, the system provides a registration form. Once the input is validated, the software initiates an email verification process to ensure account authenticity before creating the new user record in the database.

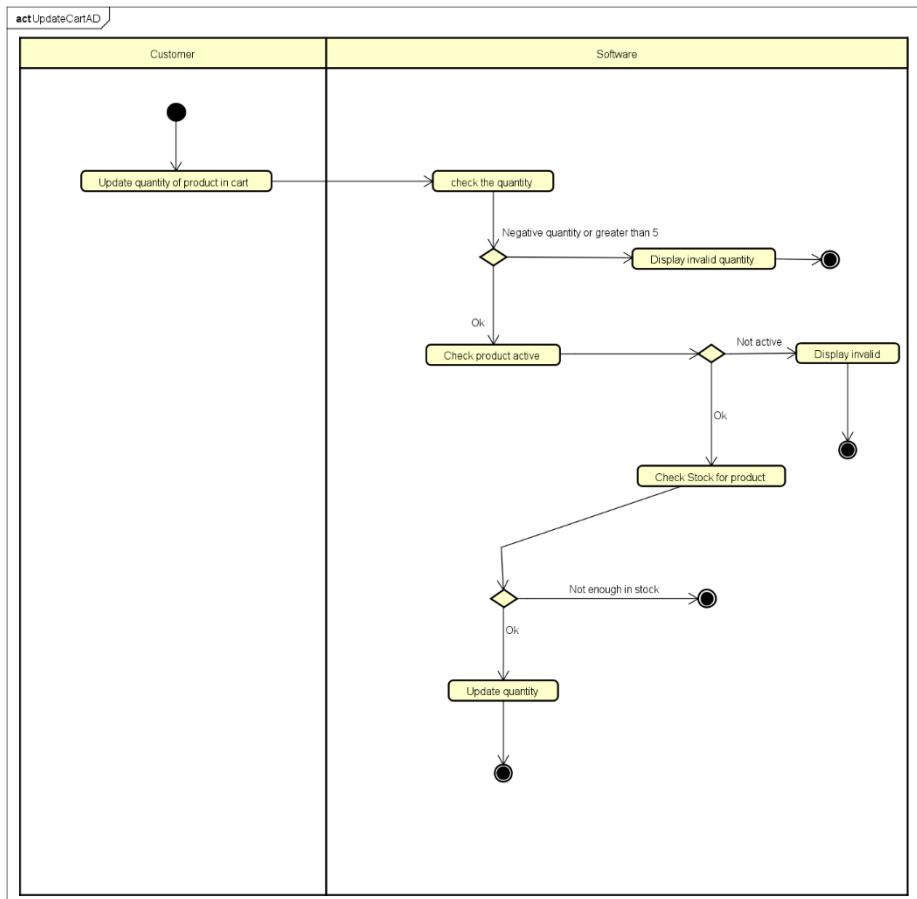
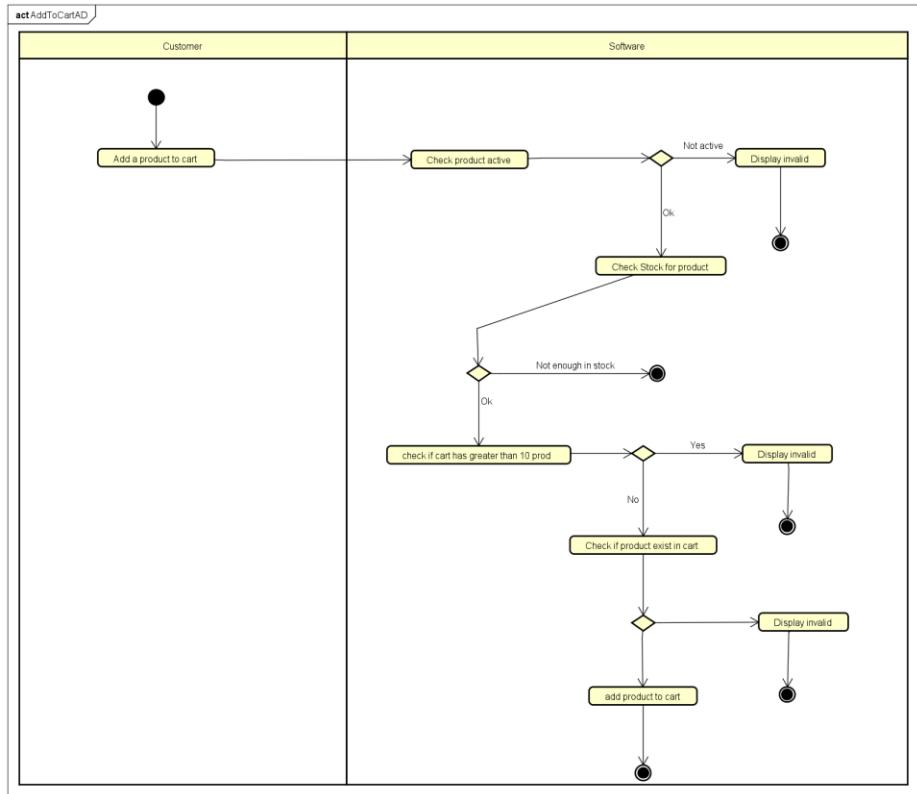


- **Account & Security (Change Password):**

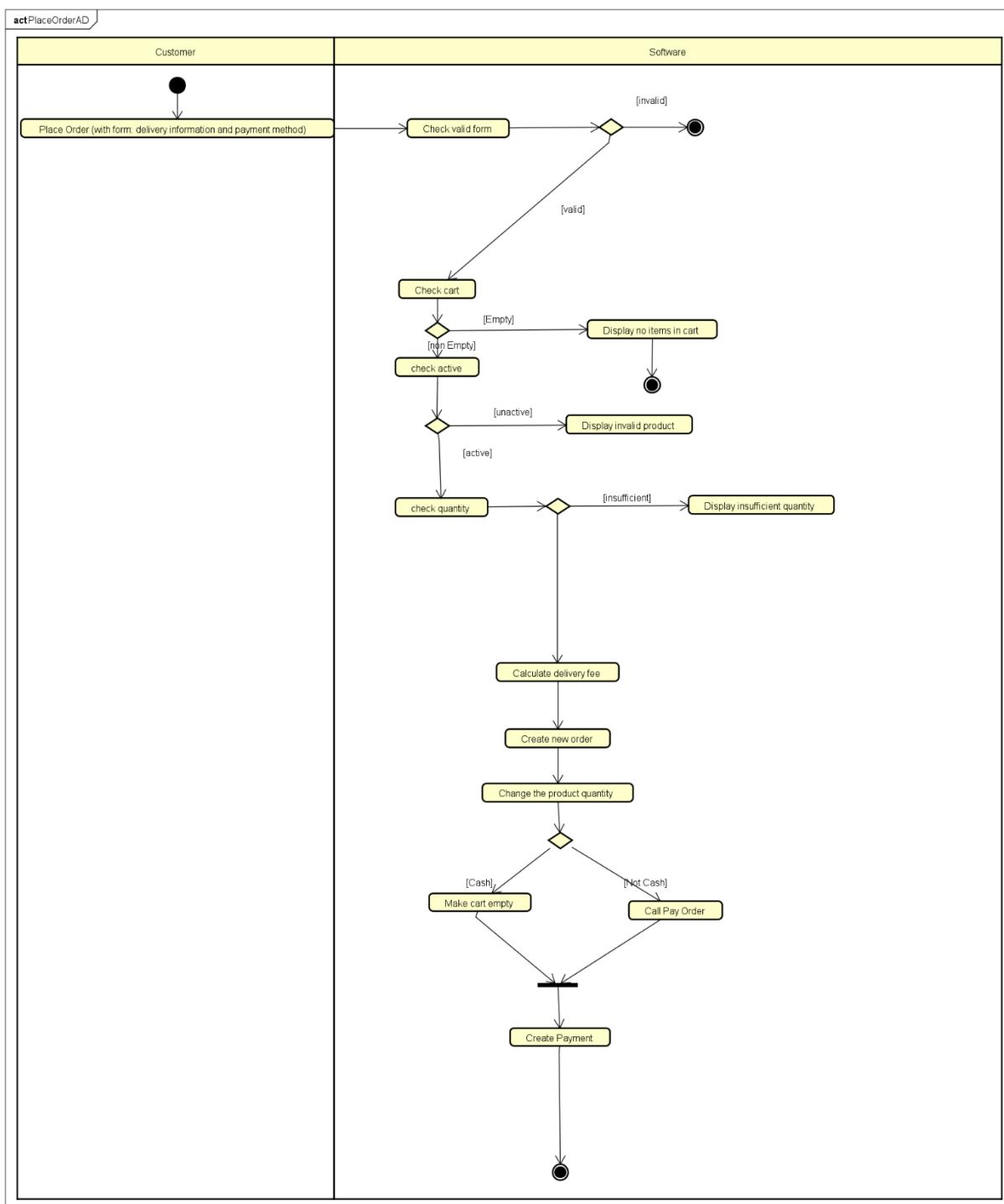
The system displays a change password form and performs strict input validation. Upon successful verification of the new data, the software updates the user's password in the persistent storage.



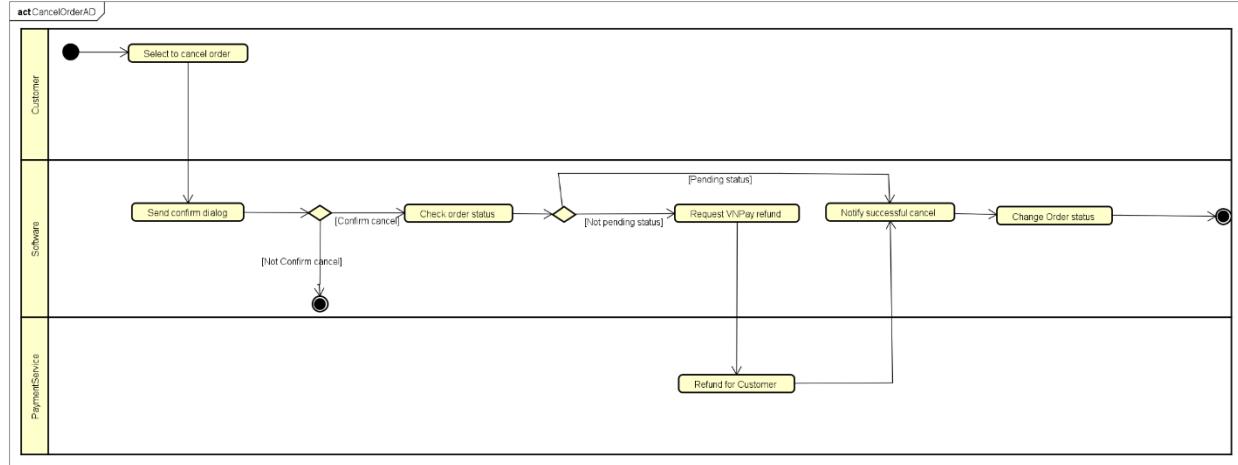
- **Shopping Cart Management (Add & Update Cart):**
  - **Add to Cart:** The software performs multiple critical checks before adding a product: verifying if the product is active, checking if sufficient stock is available, and ensuring the cart does not exceed a limit of 10 products per item type.
  - **Update Cart:** When a customer modifies a quantity, the system validates the request against a specific range (e.g., non-negative and not exceeding 5 units per update), checks product status, and verifies stock levels before committing the change.



- **Order Lifecycle (Place Order & Cancel Order):**
  - **Place Order:** This comprehensive flow involves validating the delivery form, checking cart contents, and verifying product availability. The system calculates delivery fees, creates orders, and updates inventory. If a non-cash payment method is selected, it triggers the "Pay Order" sub-process.



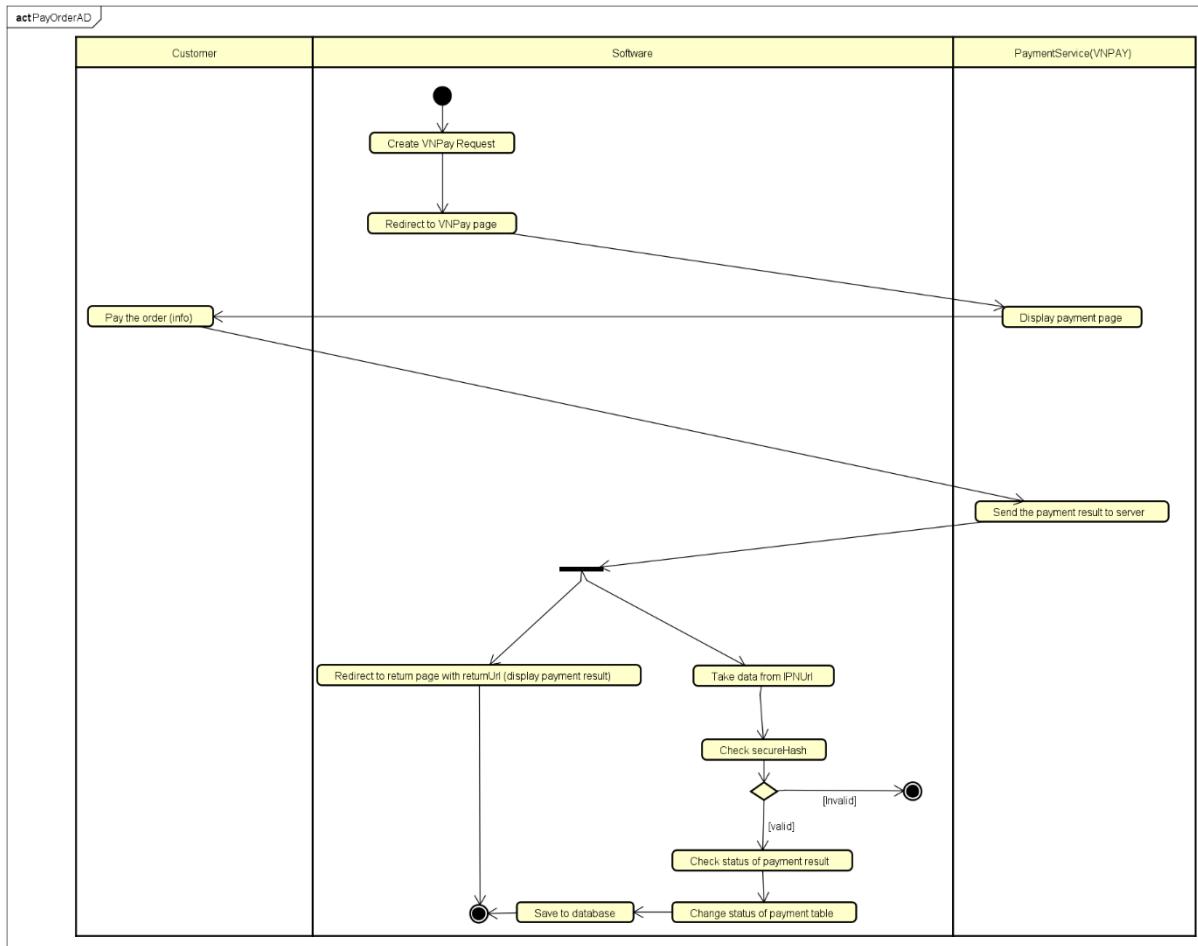
- **Cancel Order:** A customer can request to cancel an order. The software sends a confirmation dialog and checks the current order status. If the order is already paid via VNPAY, the system initiates a refund request through the PaymentService before successfully changing the order status to "Cancelled".



- **Electronic Payment (Pay Order with VNPAY):**

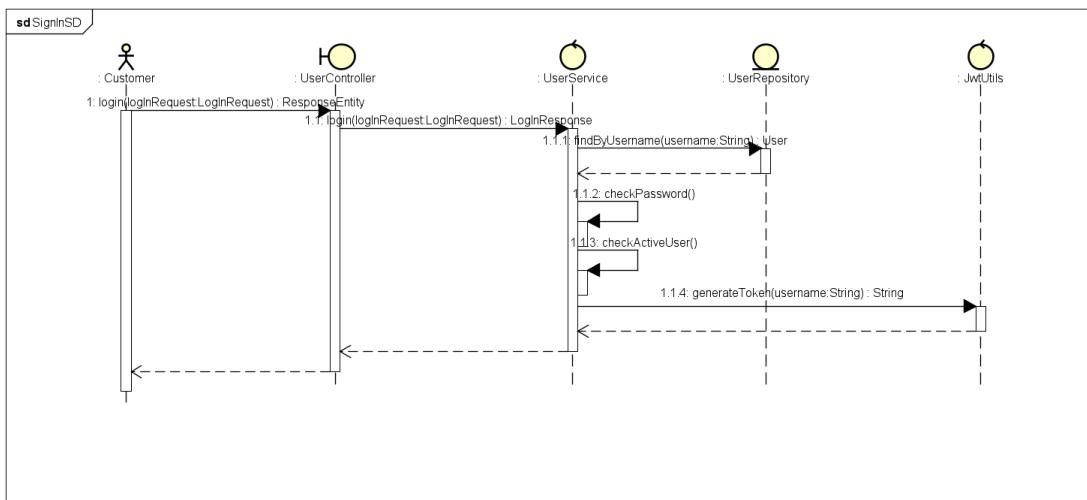
This diagram illustrates the complex interaction between the Customer, Software, and VNPAY PaymentService. After creating a request and redirecting the user to VNPAY, the system handles the response through two parallel paths:

- **User Redirect:** Returns the user
- **IPN Handling:** The software captures data from the IPN URL, performs a secureHash check for security, verifies the payment result status, and updates both the payment and database records to ensure data consistency before displaying immediate results.

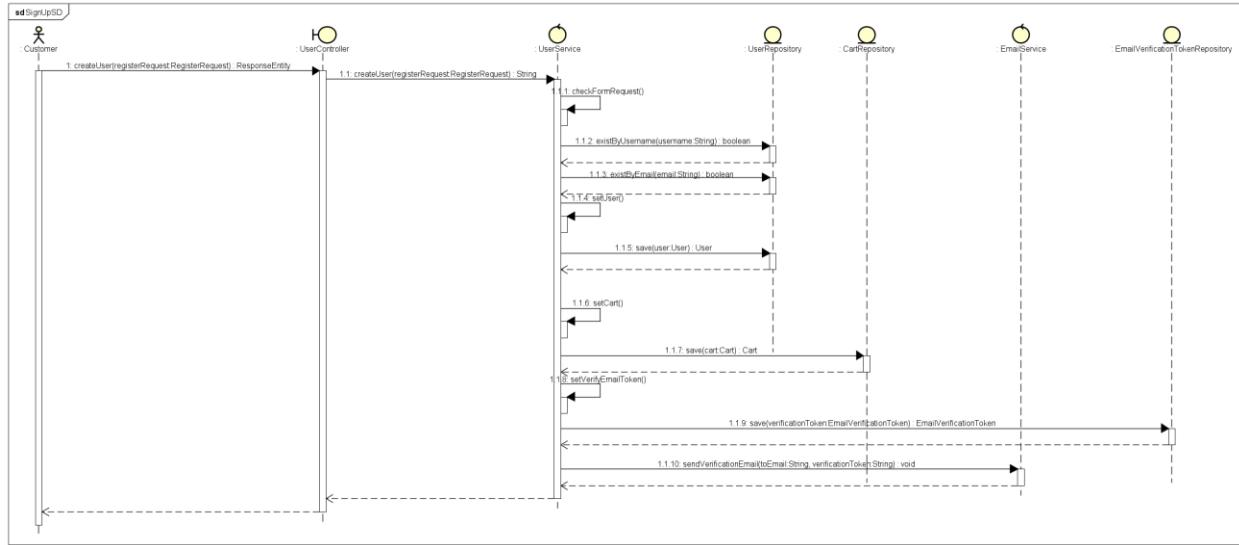


## 5. Sequence Diagrams

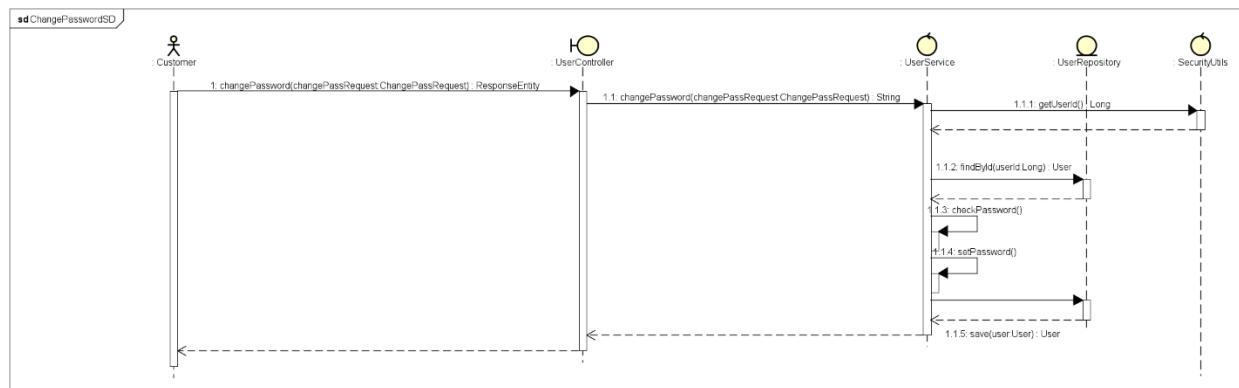
- Sign In:** UserController → UserService → UserRepository, JwtUtils



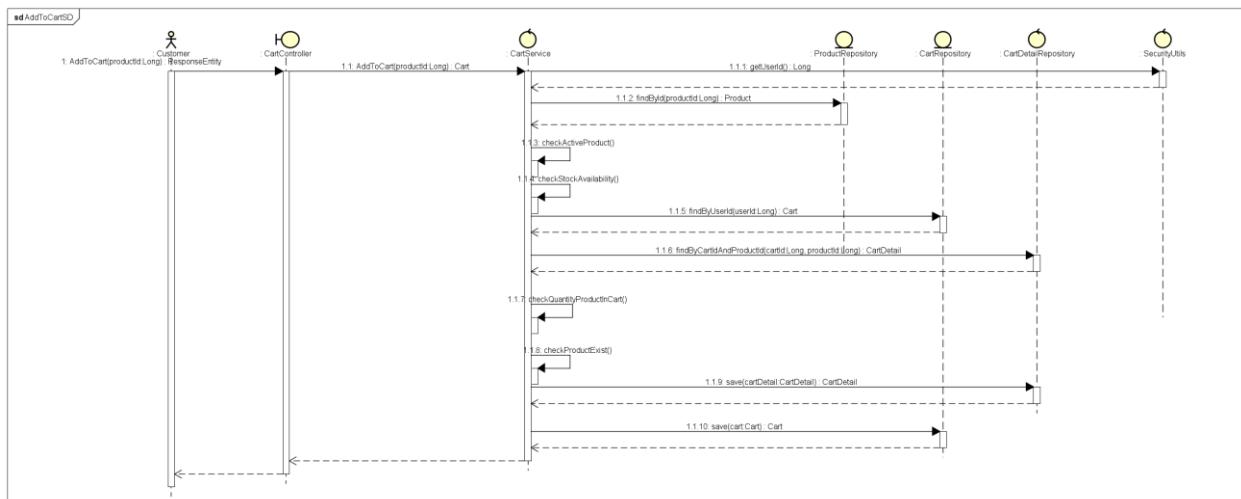
- Sign Up:** UserController → UserService → UserRepository, CartRepository, EmailService, EmailVerificationTokenRepository.



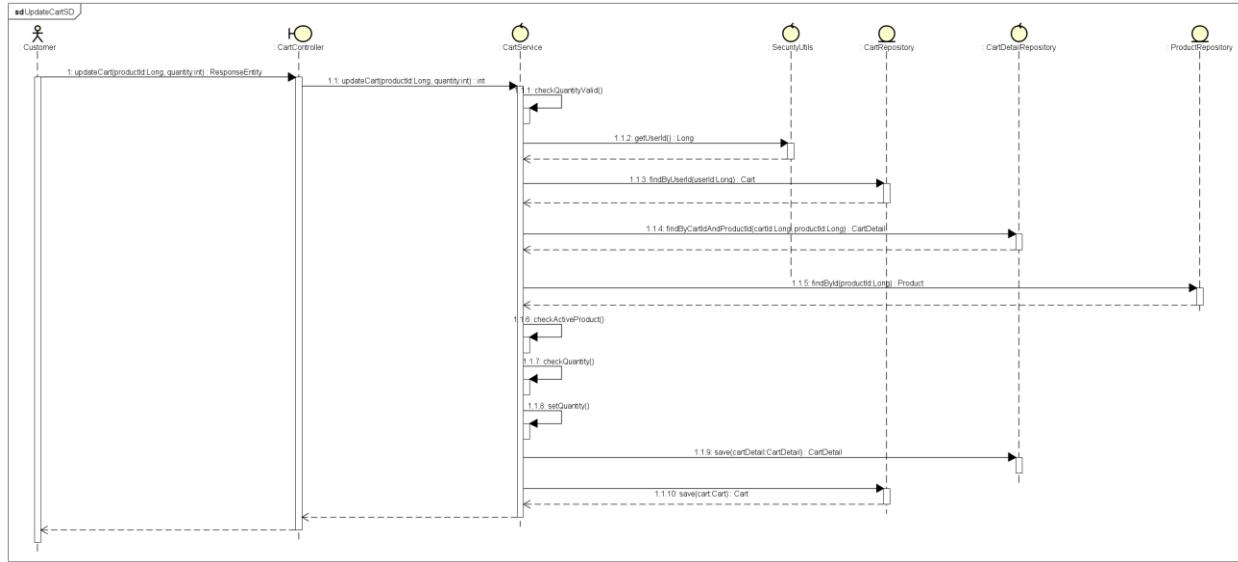
- **Change Password:** UserController → UserService → UserRepository, UserService, SecurityUtils.



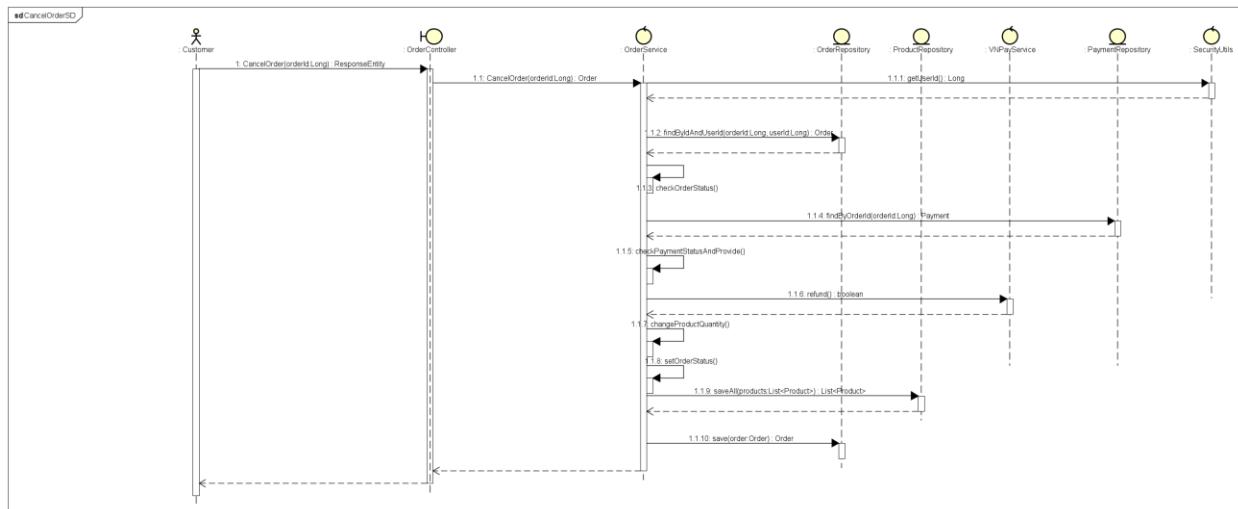
- **Add to Cart:** CartController → CartService → ProductRepository, CartRepository, CartDetailRepository, SecurityUtils.



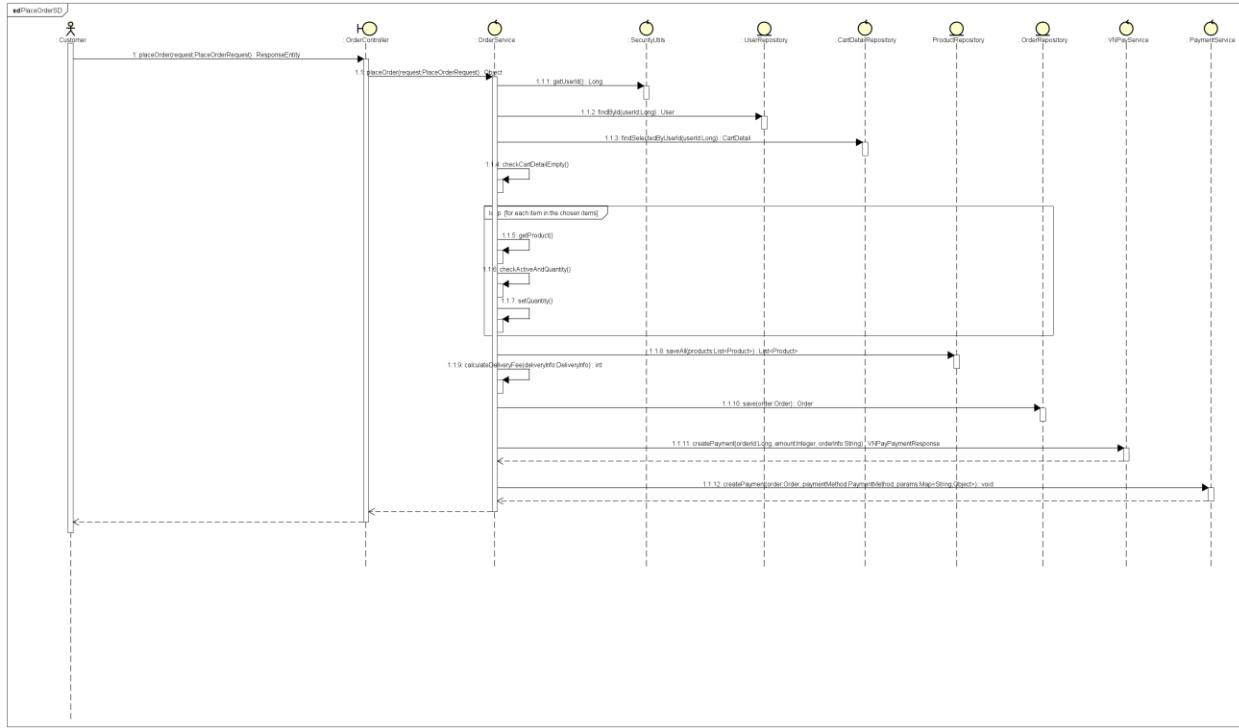
- **Update Cart:** Cart Controller → Cart Service → Security Utils, Cart Repository, Product Repository, CartDetail Repository.



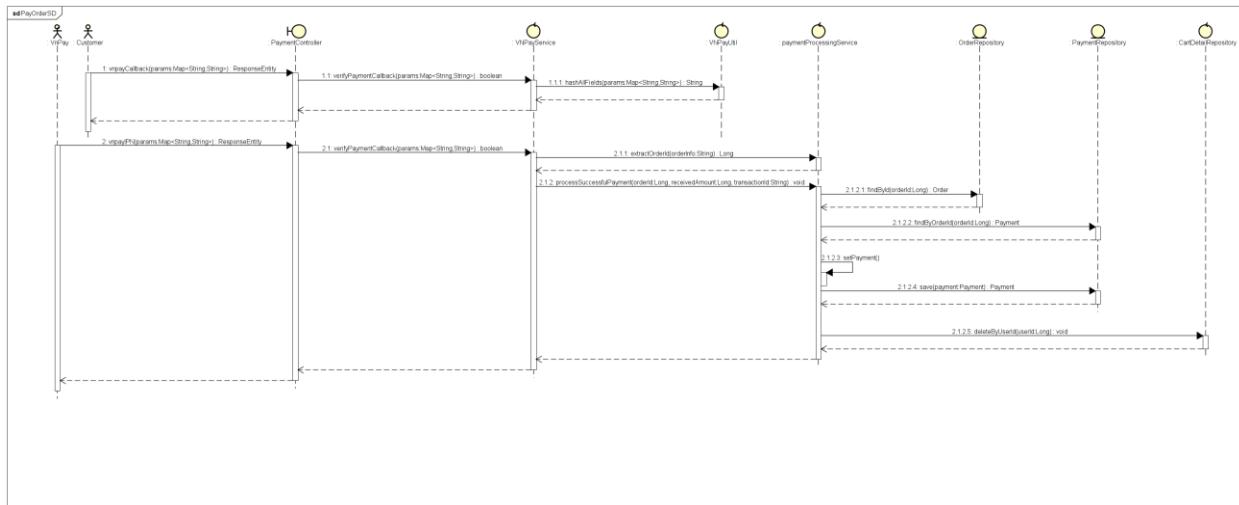
- **Order Cancel:** Order Controller → Order Service → Order Repository, Product Repository, VNPay Service, Payment Repository, Security Utils.



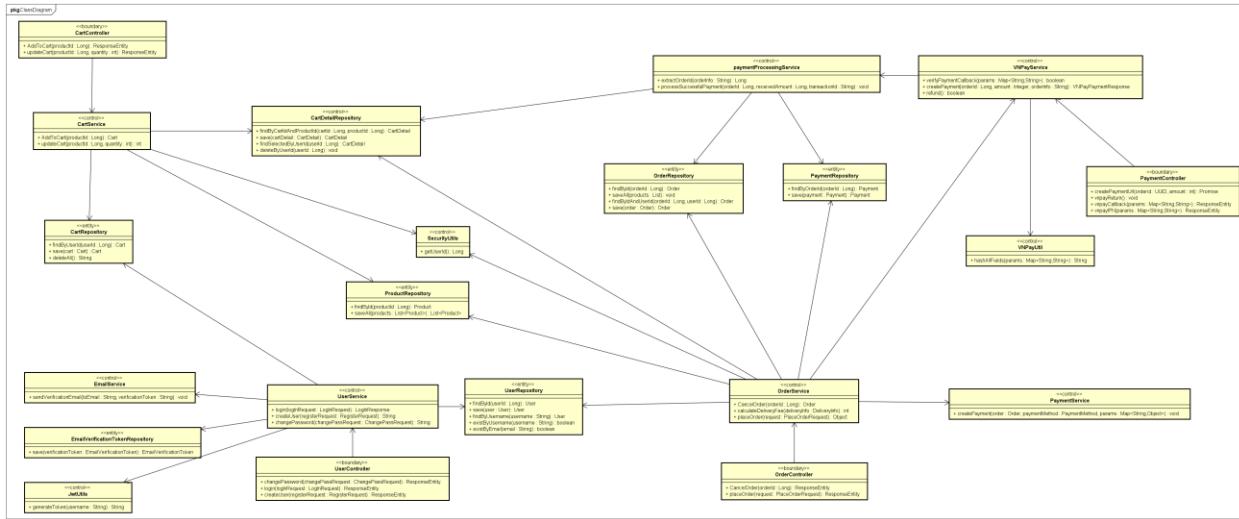
- **Place Order:** OrderController → OrderService → Other Service or Repository.



- **Pay Order:** PaymentController → VNPayService → paymentProcessingService → Other Service or Repository.



## 6. Class Diagrams



## IV. Implementation

### 1. System Architecture and Project Structure

The application utilizes a decoupled architecture with a clear separation between the ReactJS frontend and the Spring Boot backend. This ensures independent scalability and simplifies the maintenance process.

- Backend Implementation Structure (Spring Boot):**

The source code is organized logically to improve readability and maintainability:

- controller**: Handles incoming HTTP requests.
- service**: Contains business logic.
- repository**: Manages database access.
- entity**: Represents database tables.
- security**: Handles authentication and authorization logic.
- config**: Stores application configuration settings.

The backend source code is organized into a modular package structure, following the Layered Architecture pattern:

- config/**: Contains global configurations, specifically Spring Security.
  - WebSecurityConfig.java**: Manages CORS, Session management (Stateless), and URL security access.

- AuthTokenFilter.java: Intercepts requests to validate JWTs before they reach controllers.
- **controller/**: Acts as the main entry point for REST APIs.
- **exception/**: Implements global error handling via ExceptionController and standardized ErrorResponse objects.
- **feature/**:
  - **cart/**: Manages the shopping cart.
  - **email/**: Manages the automated delivery of EmailVerificationToken via Google SMTP.
  - **jwt/**: Security utilities including JwtUtils (token generation/validation) and SecurityUtils.
  - **order/**: Manages the order lifecycle.
  - **payment/**: A complex module for transaction processing.
  - **product/**: Advanced product management.
  - **users/**: Manages user accounts.

- **Frontend Implementation Structure (ReactJS):**

The Frontend is built with React and SASS, following a component-driven architecture:

- **api/**: Abducts Axios calls, sets up the base URL and interceptors.
- **router/**: Manages client-side routing and ensures unauthenticated users cannot access restricted pages.
- **components/**: Reusable UI blocks.
- **styles/**: Global SASS management.
- **verifyemail/**: Handles the UX flow for account activation (WaitingVerification.js and VerifyEmail.js).
- **pages/**: Manages all the pages to display in the web.
- **contexts**: Provides global state.
- **utils/**: Helper functions.

## 2. Extension Feature

- **Automated Email Verification**: Upon registration, the EmailService triggers a verification email. The user is redirected to a WaitingVerification page on the frontend until the token is verified via the verifyemail page logic.

- **VNPAY Payment Integration and IPN Handling:**
  - **URL Generation:** The VNPayService calculates a secure hash using HmacSHA512 to generate a redirect URL for the customer.
  - **IPN (Instant Payment Notification):** The PaymentController features a /vnpay-ipn endpoint. This allows the backend to receive payment results directly from VNPAY's server, ensuring order status is updated even if the user closes their browser during redirection. The VNpay will send the IPN to Ngrok and Ngrok will forward it to localhost.
- **Schedule to scan payment expired and verify email expired.**

## V. Result

### 1. Home Page & Product Catalog

The Home Page acts as the main landing screen, providing users with immediate access to navigation and product discovery tools. It features a top navigation bar, and a central grid displaying available technology items.

#### Key Features:

- **Product Grid Display:** Showcases a comprehensive list of products with images, names, and prices.
- **Global Navigation:** A persistent top bar providing quick access.
- **Advanced Filter Sidebar:** Enables users to refine the product list by specific criteria such as Price Range, and technical specifications.
- **Search Bar:** Allows for real-time product lookup by keywords.

#### Home Page View:

# Product

Search products...

Search

Category: All products



## Category-Specific View:

# Product

Search products...

Search

Category: Máy tính xách tay

**Filters**
[Clear filters](#)

**Sort by price**
[Price: low → high](#)

**Price range**

From:  To:

**Công nghệ CPU**

- AMD Ryzen 7
- Apple M2
- Intel Core i5



**HP Victus 16**  
19.500.000 đ  
[Compare](#)



**ASUS TUF Gaming A15**  
21.990.000 đ  
[Compare](#)



**MacBook Air M2**  
26.500.000 đ  
Sold: 1  
[Compare](#)



**Dell XPS 13**  
35.000.000 đ  
[Compare](#)



**Samsung Galaxy S24 Ultra**  
27.900.000 đ  
[Compare](#)

## 2. Product Detail Page

This page provides a dedicated view for a single product, presenting in-depth information to assist the user's purchase decision.

### Key Features:

- Detailed Specifications:** Displays technical attributes (CPU, RAM, Color) and high-resolution images.

- **Add to Cart Action:** Allows users to select a quantity and add the item to their shopping cart.

## Product Detail

[← Back](#)



**R5-7640HS | RTX 3050**

**32GB | 512GB | 16.1" Full HD**

Máy tính xách tay

**HP Victus 16**

**19.500.000 đ**

**In Stock**

**Description**  
Cấu hình mạnh, tản nhiệt tốt

**Specifications**

Công nghệ CPU	Intel Core i5
Tên đầy đủ CPU	Intel Core i5-13500H
Số nhân/Số luồng	8 nhân / 16 luồng
Card đồ họa (rời)	NVIDIA RTX 4050 6GB
Dung lượng RAM	16 GB
Dung lượng SSD	512 GB
Kích thước màn hình	16.1 inch
Độ phân giải	1920 x 1080
Tần số quét	144 Hz
Trọng lượng	2.31 kg
Hệ điều hành	Windows
Dung lượng Pin	48Wh

[Add to Cart](#)

## 3. Shopping Cart Page

The Cart page displays a tabular summary of all items the user intends to purchase, serving as a review step before checkout.

Key Features:

- **Item Management:** Lists selected products with options to adjust quantities or remove items.
- **Total Calculation:** Automatically updates and displays the total provisional price based on current quantities.
- **Empty State:** Shows a visual prompt to "Buy Now" if the cart has no items.

- **Select Product:** User can select the product to buy.

### Empty Cart:

---

Magic Shop

About Me

Shop



Cart(0)

## Cart

---

Your cart is empty.

Buy Now!

### Cart Page:

---

Magic Shop

About Me

Shop



Cart(1)

## Cart

Select All

<input checked="" type="checkbox"/>		Dell XPS 13	x
		35.000.000 đ	
	-	1	+
i5-1135G7 Intel Iris Xe 8GB 256GB 13.3" Full HD			

Total (Selected)

35.000.000 đ

Proceed to Checkout →

### Selected Product:

# Cart

The screenshot shows a shopping cart interface. At the top left is a checkbox labeled "Select All". Below it are two items:

- Samsung Galaxy S24 Ultra**: Price 27.900.000 ₫, Quantity 1. To the left of the product image is an unchecked checkbox.
- Dell XPS 13**: Price 35.000.000 ₫, Quantity 1. To the left of the product image is a checked checkbox.

On the right side, there is a summary box with the heading "Total (Selected)" and the value "35.000.000 ₫". Below it is a green button labeled "Proceed to Checkout" with a right-pointing arrow.

## 4. Checkout Page

The finalization screen where the user reviews their order and provides necessary fulfillment details.

### Key Features:

- Delivery Information:** Form for entering or confirming the recipient's Name, Phone Number, and Address.
- Payment Method Selection:** Toggle options to choose between "Cash" and "Card (VNPay)".
- Order Summary:** A final overview of products and the total cost including shipping fees.

## Place Order

The screenshot shows the "Place Order" page. On the left is the "Delivery Information" section, which includes fields for Full Name (Sang Pham), Phone (0912345678), Email (your.email@example.com), Province (dropdown menu with placeholder "-- Select Province --"), and Address (House number, street, ward, district...).

On the right is the "Order Summary (1 items)" section, which displays the selected item (Dell XPS 13) with its details: Price 35.000.000 ₫, Quantity 1, Subtotal 35.000.000 ₫, Delivery Fee 30.000 ₫, and Total 35.030.000 ₫. Below the summary is a green "Place Order" button.

The screenshot shows a two-column form for placing an order. On the left, fields include 'Province: \*' (dropdown), 'Address: \*' (text input), 'Note:' (text area), and 'Payment: \*'. Under 'Payment:', there are two options: 'Cash on Delivery' (selected, indicated by a green border) and 'VNPay'. On the right, a 'Order Summary (1 items)' section displays a product thumbnail, price (35.000.000đ), quantity (1), and subtotal (35.000.000đ). Below this, breakdowns show Subtotal (35.000.000đ), Delivery Fee (30.000đ), and a final Total (35.030.000đ). A large green 'Place Order' button is at the bottom.

## 5. Login Page

The secure entry point for registered users to access their personal accounts.

### Key Features:

- Credential Entry:** Inputs for Username and Password.
- Navigation Links:** Quick access options to register a new account or recover a lost password.
- Error Feedback:** Visual notifications for incorrect credentials.

### Login Page:

The screenshot shows the login interface for 'Magic Shop'. At the top, there's a navigation bar with links for 'About Me', 'Shop', 'Login', and a 'Cart(2)' button. The main area features a light gray background with a central white login form. The form has a title 'Login' and two input fields: 'Username' (placeholder: 'Enter your username') and 'Password' (placeholder: 'Enter your password'). Below these is a large green 'Submit' button. At the bottom of the form, a link says 'Don't have an account? [Sign up here](#)'.

## Login Page invalid display:

The screenshot shows the Magic Shop login interface. At the top, there is a navigation bar with links for 'About Me', 'Shop', 'Login', and a green 'Cart(2)' button. The main content area has a white background with a centered 'Login' form. The form title is 'Login'. Below it, a red error message says 'Invalid username or password'. There are two input fields: 'Username' containing 'a' and 'Password' containing '.'. A green 'Submit' button is below the fields. At the bottom of the form, there is a link 'Don't have an account? [Sign up here](#)'.

## 6. Register Page

The sign-up interface allows new visitors to create an account on the platform.

### Key Features:

- **Information Collection:** Fields for Username, Email, and Password setup.
- **Password Confirmation:** Requires re-entering the password to prevent typing errors.
- **Submission Control:** Prevents submission if required fields are empty or invalid.
- **Verification via Email:** User need to verify email to register account.

### Register Page:

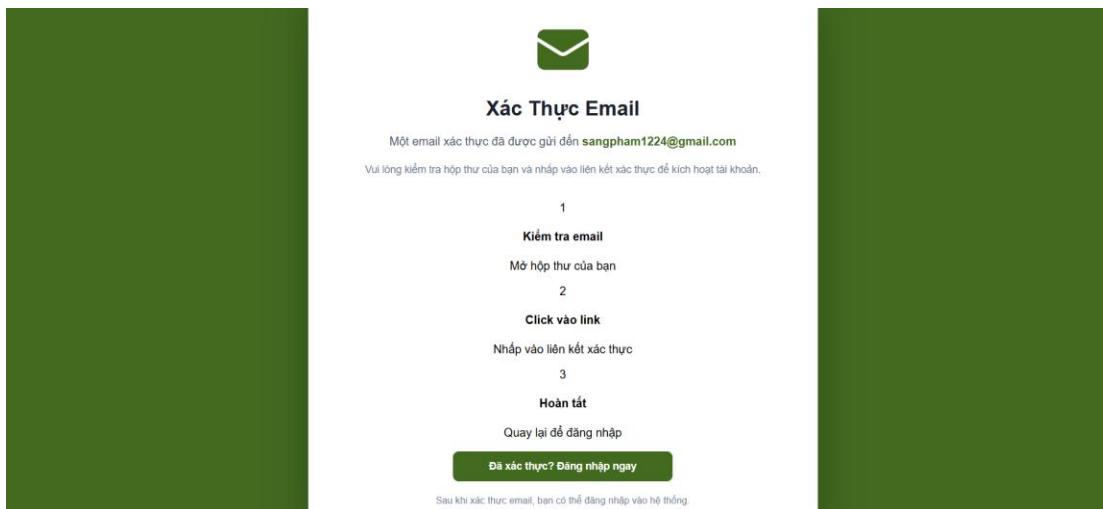
The screenshot shows the Magic Shop register interface. At the top, there is a navigation bar with links for 'About Me', 'Shop', 'Login', and a green 'Cart(2)' button. The main content area has a white background with a centered 'Register' form. The form title is 'Register'. It contains five input fields: 'Username' (placeholder 'Enter your username'), 'Full Name' (placeholder 'Enter your full name'), 'Email' (placeholder 'Enter your email'), 'Phone' (placeholder 'Enter your phone number'), and 'Password' (placeholder 'Enter your password').

A screenshot of a registration form titled "REGISTER". The form consists of several input fields: "Username" (placeholder: Enter your username), "Full Name" (placeholder: Enter your full name), "Email" (placeholder: Enter your email), "Phone" (placeholder: Enter your phone number), "Password" (placeholder: Enter your password), and "Confirm Password" (placeholder: Confirm your password). Below the fields is a green "Register" button.

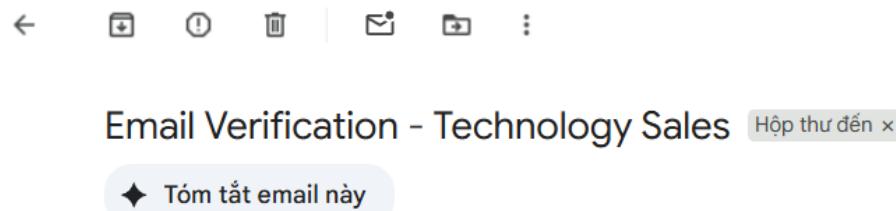
### Register Page Invalid display:

A screenshot of a registration form titled "Register". The form fields and their validation errors are: "Username" (ttffix), "Full Name" (Sang Phạm), "Email" (sangpham1224@gmail.com), "Phone" (033434343) with the error "Invalid phone number", "Password" (\*\*\*\*\*) with the error "Password must be at least 8 characters", and "Confirm Password" (\*\*\*\*\*) with the error "Passwords do not match". Below the fields is a green "Register" button.

### Verification Email Notice:



## Verification Email:



 sangpham1224@gmail.com  
đến tôi ▾

Welcome to Technology Sales!

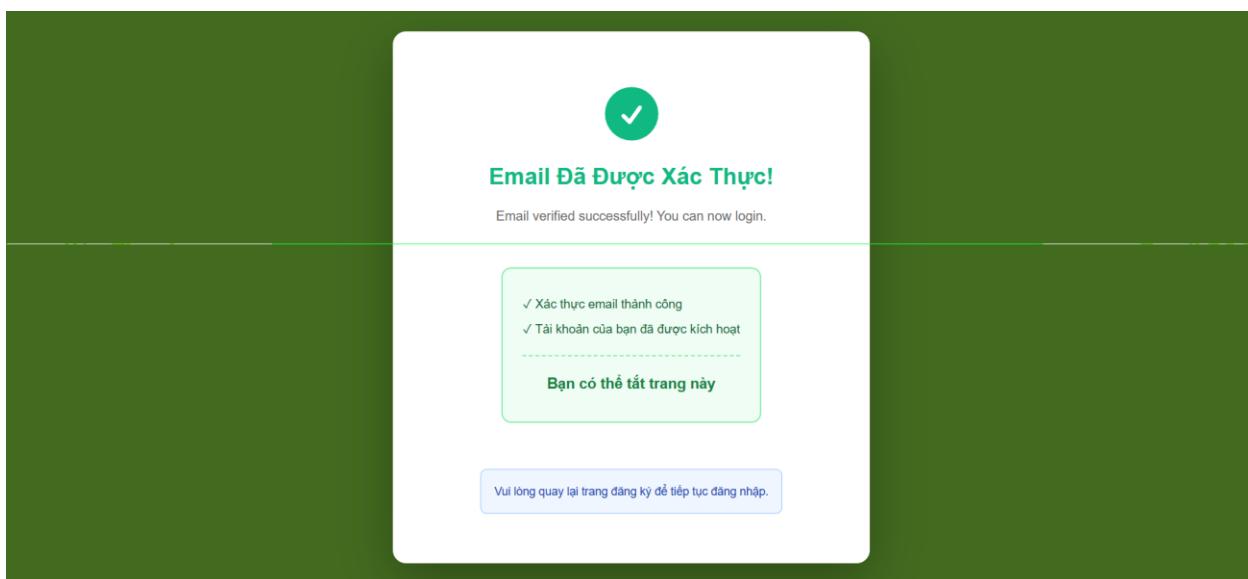
Thank you for registering with us. Please verify your email address by clicking the link below:

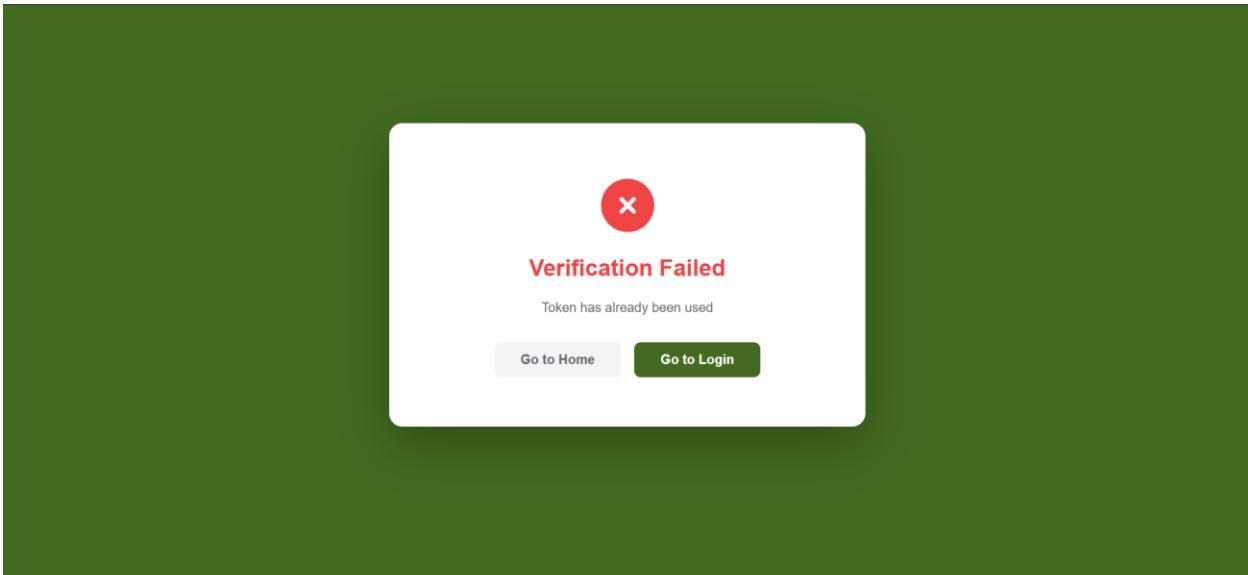
<http://localhost:3000/verify-email?token=1c8524ce-0477-4d04-a7c5-903fabdbfcb2>

This link will expire in 30 minutes.

If you did not create this account, please ignore this email.

Best regards,  
Technology Sales Team





## 7. Order History Page

A dashboard allowing customers to track and review their past purchase activities.

### Key Features:

- Order List:** Displays a chronological list of orders with Date, Order ID, Total Price, Order Status, Payment Status and action buttons.
- Status Filter:** Filter by status of orders.
- View Details:** Interaction to open a modal viewing specific items within an order.
- Cancel Order:** Cancel order with refund.

## Order History

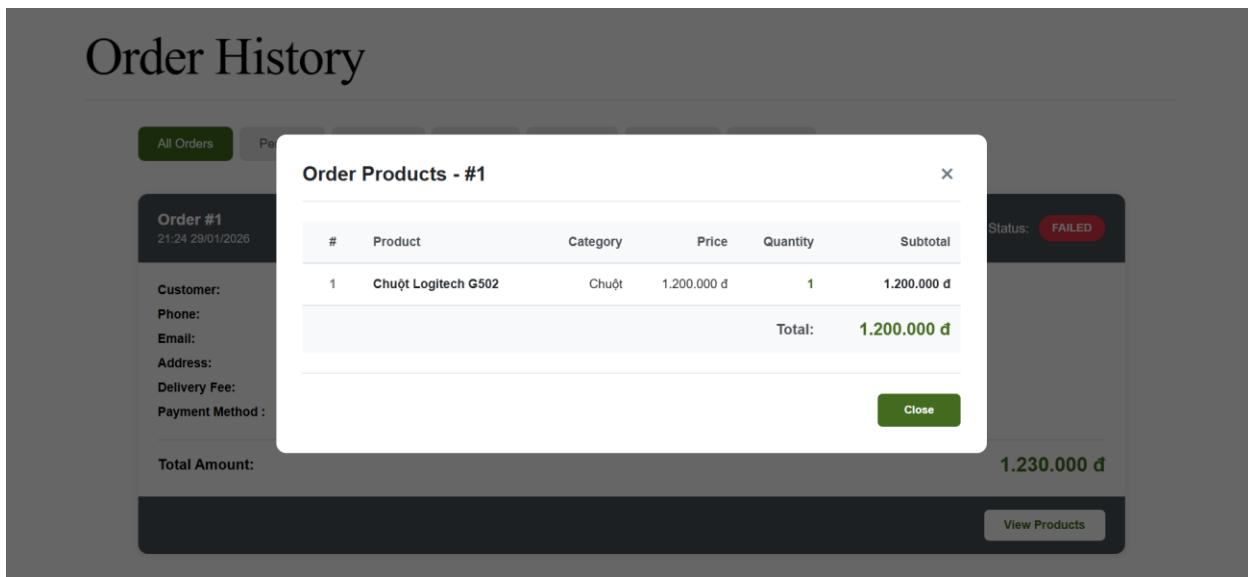
All Orders   Pending   Preparing   Shipping   Delivered   Cancelled   Rejected

**Order #1**  
21:24 29/01/2026   Order Status: PENDING   Payment Status: PENDING

Customer:	Sang Phạm
Phone:	0334343434
Email:	tranthib@example.com
Address:	Sevin Office 609 Trương Định, Bình Thuận
Delivery Fee:	30.000 đ
Payment Method:	CASH

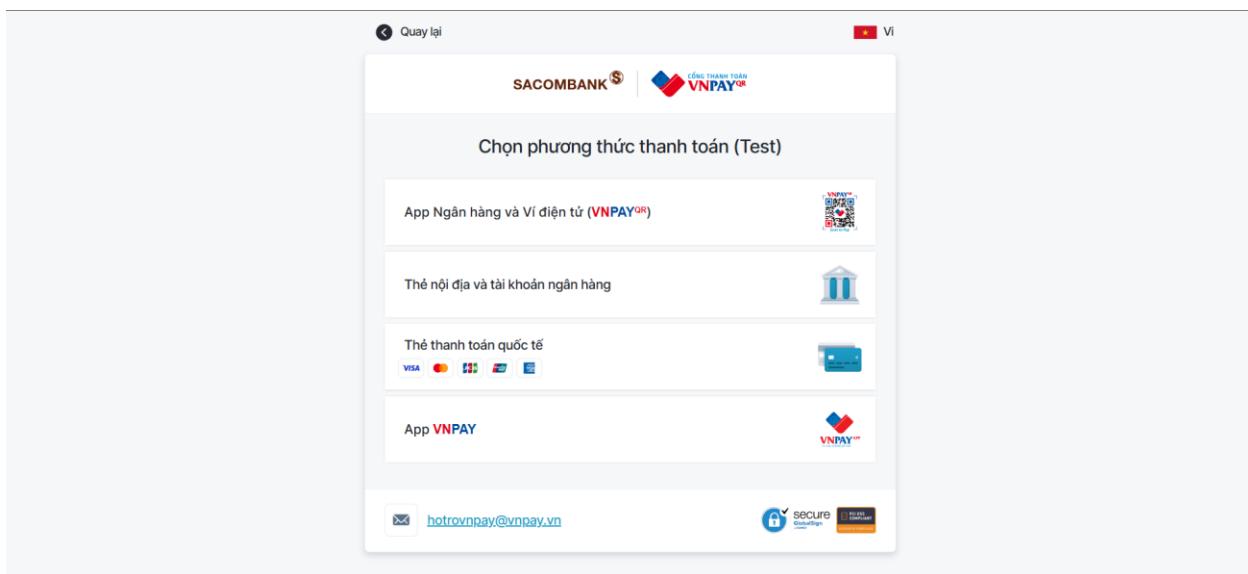
Total Amount: 1.230.000 đ

[View Products](#)   [Cancel Order](#)



## 8. VNPay Gateway (Sandbox)

The external secure interface where electronic payment transactions take place.



Quay lại

VNPay

Giao dịch hết hạn sau 00 : 28

Thanh toán qua Ngân hàng NCB

Thông tin đơn hàng (Test)

Số tiền thanh toán  
**19.530.000 VND**

Giá trị đơn hàng  
19.530.000 VND

Phí giao dịch  
0 VND

Mã đơn hàng  
64269412

Nhà cung cấp  
<https://vnshop.vn/>

Thẻ nội địa

Số thẻ  
Nhập số thẻ  NCB

Tên chủ thẻ  
Nhập tên chủ thẻ (không dấu)

Ngày phát hành

Mã khuyến mại  
Chọn hoặc nhập mã

[Điều kiện sử dụng dịch vụ](#)

Hủy thanh toán Tiếp tục

VNPay

Giao dịch hết hạn sau 00 : 14

Xác thực OTP

Thông tin đơn hàng (Test)

Số tiền thanh toán  
**19.530.000 VND**

Giá trị đơn hàng  
19.530.000 VND

Phí giao dịch  
0 VND

Mã đơn hàng  
01935336

Nhà cung cấp  
<https://vnshop.vn/>

Nhập mã OTP  NCB

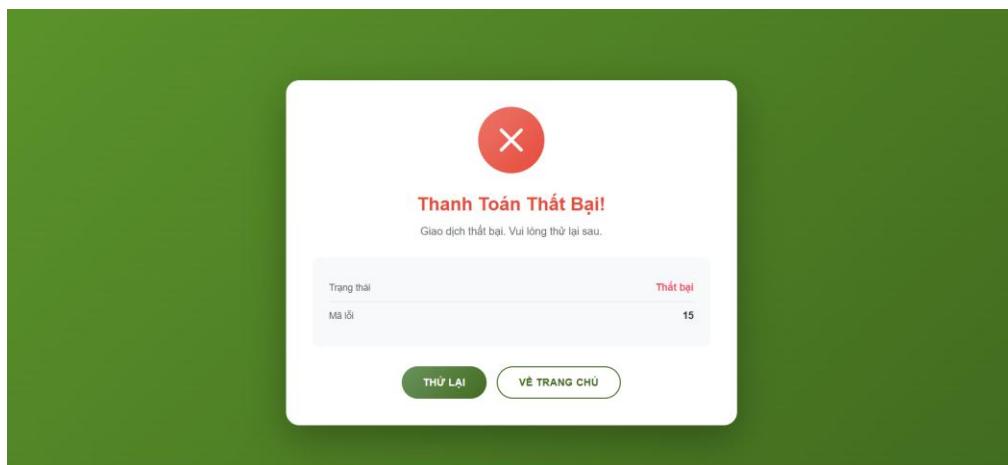
OTP đã được gửi về số điện thoại đăng ký. Quý khách vui lòng nhập OTP để tiếp tục thực hiện giao dịch.

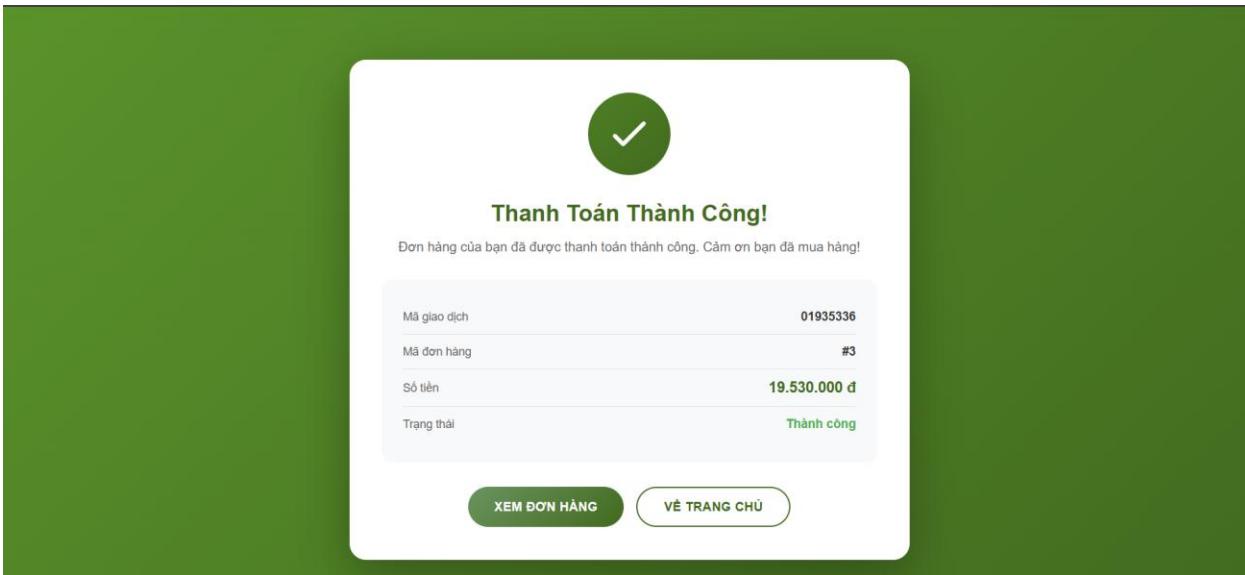
Hủy Thanh toán

[hotrovnpay@vnpay.vn](mailto:hotrovnpay@vnpay.vn)

secure

## Payment Result:





## 9. User Profile Page

A personal management screen where users can view and update their account information.

### Key Features:

- Profile Information:** Displays editable fields for Name, Phone, Address, and Date of Birth.
- Security Settings:** Option to open a "Change Password" modal to update login credentials.
- Read-Only Fields:** Displays the Username and Email as fixed identifiers.

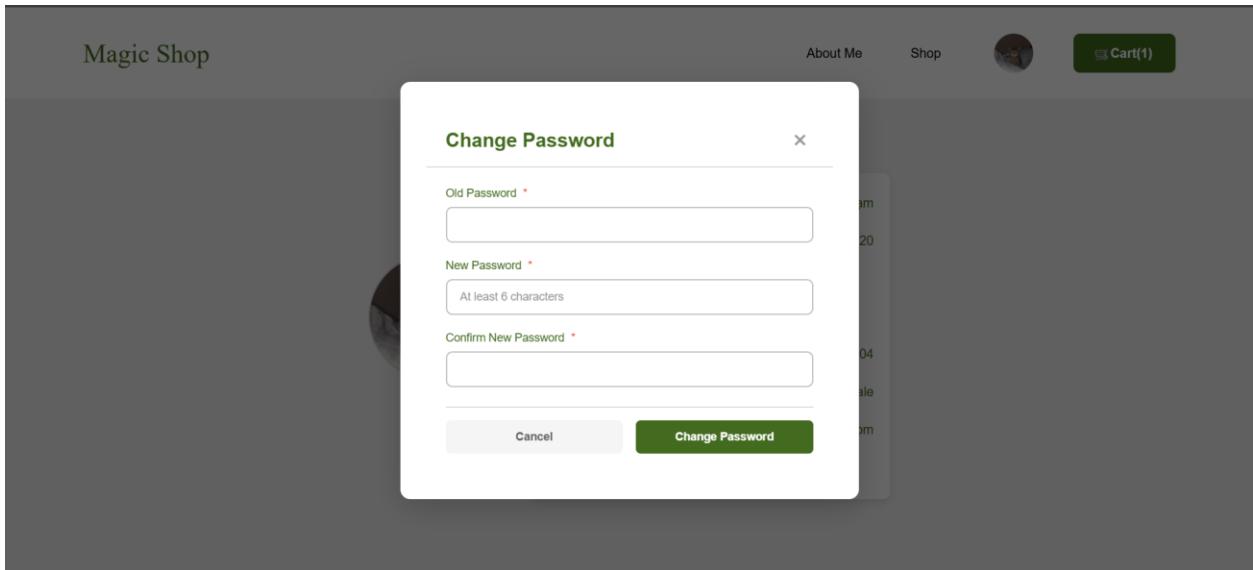
### User Profile Page:

A screenshot of a user profile page. At the top, there's a navigation bar with "Magic Shop", "About Me", "Shop", a user icon, and a "Cart(1)" button. Below the navigation is a profile section with a placeholder image and the username "ttiffiX". To the right is a detailed profile card with the following data:

Name	Sang Phạm
Age	20
Phone	
Address	
Date of Birth	10-02-2004
Gender	Male
Email	sangpham1224@gmail.com

At the bottom of the card are two buttons: "Edit" and "Change Password".

### Change Password Modal:



## VI. Discusion

### 1. Limitations and Missing Features

While the core e-commerce engine is functional, several sophisticated features remain in the development backlog to be addressed in future iterations.

- **Customer Experience Gaps:**
  - **Automated Refund Integration:** Although the payment flow via VNPay is successful, the automated refund mechanism for cancelled orders has not been fully realized. Currently, transactions require manual intervention for reversals.
  - **Profile Data Synchronization:** The User Profile interface is active. However, a complete bi-directional data bind between the frontend forms and the backend persistence layer is still being finalized.
  - **Credential Recovery:** The "Forgot Password" workflow, involving temporary password generation and secure email dispatch, is designed but has not yet been integrated into the live production environment.
- **Administrative & Management Modules:**
  - **Administrative Governance:** The dedicated interface for **Admin** users to manage system configurations, user roles, and security audits is currently under development.

- **Product & Inventory Management:** The **Product Manager (PM)** dashboard, intended for catalog maintenance and inventory tracking, remains to be implemented to provide a complete end-to-end business solution.

## 2. Experiment Gains

The project served as a rigorous practical experiment in full-stack engineering, yielding significant professional and technical growth:

- **Advanced Architecture Mastery:** Gained hands-on experience in implementing a **Feature-based Packaging** architecture in Spring Boot, significantly improving code modularity compared to traditional layered approaches.
- **Third-party Integration Proficiency:** Successfully navigated the complexities of financial integration by implementing **VNPAY**, including secure hash algorithms and the mission-critical **IPN (Instant Payment Notification)** mechanism to ensure data consistency.
- **Security & Authentication Flow:** Developed a deep understanding of stateless security using **JWT (JSON Web Tokens)** and multi-step verification via **Google SMTP**, enhancing the system's defensive posture.
- **Dynamic Data Modeling:** Mastered the **Entity-Attribute-Value (EAV)** model to support diverse technical specifications for technology products without requiring frequent database schema migrations.

## VII. Conclusion

The development of the Technology Sales Website represents a successful transition from theoretical system design to a functional, full-stack application.

The project successfully achieved its primary objective: building a high-performance e-commerce platform that handles real-time product discovery, secure cart management, and electronic payments. By utilizing Java 22 and Spring Boot 3.4, the system benefits from the latest advancements in backend performance and security. The frontend, powered by ReactJS, provides a dynamic and intuitive user experience that meets modern web standards.

Despite the identified limitations in the administrative modules and advanced customer features, the core infrastructure is robust and scalable. The lessons learned regarding IPN handling, EAV modeling, and Stateless Security provide a solid foundation for future professional development.

In conclusion, the product is not merely a shopping tool but a comprehensive technical experiment that demonstrates the power of integrating modern frameworks with secure third-party financial services to solve real-world e-commerce challenges.