

**BỘ KHOA HỌC VÀ CÔNG NGHỆ**  
**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**  
**KHOA CÔNG NGHỆ THÔNG TIN 2**

---



**Báo cáo cuối kì**  
**PHÁT TRIỂN ỨNG DỤNG**  
**CHO CÁC THIẾT BỊ DI ĐỘNG**

*Đề tài:*

**XÂY DỰNG ỨNG DỤNG MUA ĐỒNG HỒ**  
**WATCH SHOPPING APP**

**Giảng viên hướng dẫn:** Nguyễn Trung Hiếu

**Sinh viên thực hiện:** Nguyễn Nhật Minh – N21DCCN035 (Leader)

Trần Vũ Phương Nam – N21DCCN151

*Thành phố Hồ Chí Minh, ngày 10 tháng 6 năm 2025*

# LỜI CẢM ƠN

Lời đầu tiên, nhóm xin gửi lời cảm ơn chân thành đến thầy Nguyễn Trung Hiếu, đã tận tình giảng dạy, hướng dẫn và tạo điều kiện thuận lợi để nhóm có thể hoàn thành đồ án cuối kỳ này. Sự hỗ trợ quý báu của thầy không chỉ giúp nhóm có thêm nhiều kiến thức bổ ích về quá trình phát triển một sản phẩm ứng dụng cho các thiết bị di động, 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 đồ án, 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 và sản phẩm một cách tốt nhất.

*Thành phố Hồ Chí Minh, ngày 10 tháng 6 năm 2025*

# BẢNG PHÂN CÔNG CÔNG VIỆC

Họ và tên	MSSV	Công việc
Nguyễn Nhật Minh	N21DCCN053	<ul style="list-style-type: none"><li>Lập trình hệ thống Backend</li><li>Viết báo cáo</li><li>Xây dựng layout và chức năng danh mục sản phẩm</li><li>Xây dựng layout và chức năng chi tiết sản phẩm</li><li>Xây dựng layout và chức năng tài khoản người dùng</li></ul>
Trần Vũ Phương Nam	N21DCCN151	<ul style="list-style-type: none"><li>Lập trình hệ thống Backend</li><li>Viết báo cáo</li><li>Xây dựng layout và chức năng giỏ hàng và đặt mua</li><li>Xây dựng layout và chức năng theo dõi đơn hàng</li><li>Xây dựng layout và chức năng lưu địa chỉ giao hàng và lịch sử mua hàng</li></ul>

# MỤC LỤC

<b>CHƯƠNG 1. TỔNG QUAN</b>	<b>1</b>
1.1. Giới thiệu đề tài	1
1.2. Cơ sở lý thuyết	1
<b>CHƯƠNG 2. THIẾT KẾ HỆ THỐNG</b>	<b>3</b>
2.1. Xác định tác nhân	3
2.2. Sơ đồ usecase của hệ thống	4
2.3. Sơ đồ tuần tự cốt lõi của hệ thống	6
2.4. Sơ đồ trạng thái cốt lõi của hệ thống	8
2.5. Sơ đồ lớp của hệ thống	10
2.6. Sơ đồ ERD của hệ thống	11
<b>CHƯƠNG 3. TRIỂN KHAI HỆ THỐNG</b>	<b>14</b>
3.1. Chuẩn bị môi trường phát triển	14
3.1.1. Công cụ phát triển ứng dụng Android	14
3.1.2. Backend API sử dụng trong hệ thống	14
3.2. Triển khai ứng dụng Android	16
3.2.1. Khởi tạo dự án và cấu trúc thư mục	16
3.2.2. Mô hình kiến trúc MVVM	17
3.2.3. Luồng dữ liệu giữa các thành phần	17
3.2.4. Vai trò các class chính	17
3.2.5. Ưu điểm mô hình MVVM trong đồ án	18
3.2.6. Kết nối backend và xử lý dữ liệu	18
3.2.7. Xây dựng các chức năng chính	18
3.2.8. Lưu trữ dữ liệu cục bộ	19
3.2.9. Kiểm thử và hoàn thiện	19
3.3. Hình ảnh giao diện của ứng dụng	20
<b>CHƯƠNG 4. KẾT LUẬN</b>	<b>28</b>
4.1. Tổng kết quá trình thực hiện đề tài	28
4.2. Định hướng và phát triển	28
<b>TÀI LIỆU THAM KHẢO</b>	<b>30</b>

# DANH MỤC ẢNH

Hình 1: Sơ đồ usecase các chức năng bảo mật hệ thống .....	4
Hình 2: Sơ đồ usecase các chức năng liên quan đến đơn hàng .....	4
Hình 3: Sơ đồ usecase chức năng xem sản phẩm .....	5
Hình 4: Sơ đồ usecase chức năng quản lý thông tin người dùng .....	5
Hình 5: Sơ đồ usecase chức năng quản lý giỏ hàng .....	5
Hình 6: Sơ đồ tuần tự chức năng đặt hàng .....	6
Hình 7: Sơ đồ tuần tự chức năng thêm sản phẩm vào giỏ hàng .....	7
Hình 8: Sơ đồ trạng thái giỏ hàng.....	8
Hình 9: Sơ đồ trạng thái đơn hàng.....	9
Hình 10: Sơ đồ lớp của toàn hệ thống .....	10
Hình 11: Sơ đồ ERD phần giỏ hàng của hệ thống.....	11
Hình 12: Sơ đồ ERD phần đơn hàng của hệ thống.....	12
Hình 13: Sơ đồ ERD phần sản phẩm, thuộc tính sản phẩm .....	13
Hình 14: Sơ đồ ERD phần sản phẩm, nhãn hàng, và danh mục.....	13
Hình 15: Các API endpoints liên quan đến bảo mật.....	15
Hình 16: Các API endpoint liên quan đến nhãn hàng.....	15
Hình 17: Các API endpoint liên quan đến giỏ hàng .....	15
Hình 18: Các API endpoint liên quan đến danh mục .....	15
Hình 19: Các API endpoint liên quan đến đơn hàng .....	16
Hình 20: Các API endpoint liên quan đến sản phẩm.....	16
Hình 21: Các API endpoint liên quan đến người dùng.....	16
Hình 22: Giao diện trang chủ (home) của ứng dụng .....	20
Hình 23: Giao diện đăng nhập của ứng dụng .....	21
Hình 24: Giao diện đăng kí của ứng dụng.....	22
Hình 25: Giao diện giỏ hàng của ứng dụng.....	23
Hình 26: Giao diện trang cá nhân người dùng của ứng dụng.....	24
Hình 27: Giao diện danh sách đơn hàng người dùng của ứng dụng.....	25
Hình 28: Giao diện chi tiết sản phẩm của ứng dụng .....	26
Hình 29: Giao diện chi tiết đơn hàng người dùng của ứng dụng .....	27

## CHƯƠNG 1. TỔNG QUAN

### 1.1. Giới thiệu đề tài

Trong bối cảnh công nghệ ngày càng phát triển, nhu cầu mua sắm trực tuyến ngày càng trở nên phổ biến và tiện lợi với người tiêu dùng. Đặc biệt, với các sản phẩm thời trang, nhờ vào công nghệ nên khách hàng có thể dễ dàng tiếp cận và tìm kiếm sản phẩm. Cũng như các người bán hàng cũng có thể quảng cáo, trưng bày và kinh doanh sản phẩm online một cách dễ dàng hơn, tiện lợi hơn vượt qua những rào cản về vị trí địa lý, thời gian và công sức.

Xuất phát từ thực tiễn đó, nhóm chúng em lựa chọn đề tài “Xây dựng ứng dụng mua đồng hồ (Watch Shopping App)” nhằm xây dựng một nền tảng thương mại điện tử hiện đại, chuyên biệt cho các sản phẩm đồng hồ. Ứng dụng cho phép người dùng dễ dàng tìm kiếm, lựa chọn và mua sắm các loại đồng hồ như đồng hồ cơ, điện tử, thể thao, thông minh,... với đầy đủ thông tin chi tiết về sản phẩm, thương hiệu, giá cả, chế độ bảo hành cũng như hỗ trợ các tính năng lọc thông minh theo nhu cầu.

Bên cạnh đó, ứng dụng còn cung cấp các chức năng như quản lý giỏ hàng, đặt mua, theo dõi trạng thái đơn hàng, nhận thông báo cập nhật qua email và quản lý tài khoản cá nhân. Giao diện được thiết kế hiện đại, sang trọng, mang lại trải nghiệm mua sắm chuyên nghiệp và tiện lợi cho người dùng, phù hợp với đặc thù ngành hàng thời trang cao cấp.

Việc thực hiện đề tài này không chỉ giúp nhóm sinh viên áp dụng các kiến thức đã học về phát triển ứng dụng di động mà còn rèn luyện kỹ năng làm việc nhóm, phân tích, thiết kế và triển khai giải pháp phần mềm đáp ứng nhu cầu thực tế của người dùng hiện đại.

### 1.2. Cơ sở lý thuyết

#### **Thương mại điện tử và xu hướng mua sắm trực tuyến**

Thương mại điện tử (E-Commerce) là quá trình mua bán hàng hóa hoặc dịch vụ thông qua các hệ thống điện tử như Internet và các mạng máy tính. Trong bối cảnh hiện nay, xu hướng mua sắm trực tuyến ngày càng phát triển mạnh mẽ nhờ sự tiện lợi, đa dạng về sản phẩm, tiết kiệm thời gian và chi phí cho người tiêu dùng. Đặc biệt đối với các mặt hàng thời trang cao cấp như đồng hồ, khách hàng có nhu cầu tham khảo nhiều lựa chọn, so sánh thông tin sản phẩm, chính sách bảo hành và các chương trình ưu đãi trước khi quyết định mua hàng.

#### **Ứng dụng di động trong thương mại điện tử**

Với sự phổ biến của điện thoại thông minh, ứng dụng di động đóng vai trò quan trọng trong việc thúc đẩy thương mại điện tử. Ứng dụng di động mang lại trải nghiệm mua sắm cá nhân hóa, tiện lợi, nhanh chóng và dễ dàng tiếp cận ở mọi lúc, mọi nơi. Việc phát triển các ứng dụng

## *Báo cáo Đồ án Xây dựng ứng dụng mua đồng hồ Watch Shopping App*

mua sắm chuyên biệt như ứng dụng mua đồng hồ giúp doanh nghiệp tiếp cận khách hàng mục tiêu hiệu quả hơn, đồng thời nâng cao khả năng tương tác, chăm sóc khách hàng và gia tăng doanh số bán hàng.

### **Kiến trúc ứng dụng di động**

Các ứng dụng di động hiện đại thường được xây dựng theo kiến trúc phân lớp (multi-tier architecture) gồm ba lớp chính:

- Lớp giao diện người dùng (UI): Chịu trách nhiệm hiển thị thông tin, nhận và xử lý tương tác từ người dùng.
- Lớp xử lý nghiệp vụ (Business Logic): Xử lý các yêu cầu, điều hướng luồng dữ liệu giữa UI và cơ sở dữ liệu, kiểm tra logic nghiệp vụ.
- Lớp dữ liệu (Data Layer): Quản lý, lưu trữ và truy xuất dữ liệu từ hệ thống hoặc từ server thông qua các API.

### **Các công nghệ sử dụng trong phát triển ứng dụng**

- Ngôn ngữ lập trình: Java.
- Kiến trúc MVVM: Hỗ trợ tách biệt giao diện với logic xử lý, dễ dàng bảo trì, mở rộng ứng dụng.
- API và giao tiếp client-server: Sử dụng RESTful API để trao đổi dữ liệu giữa ứng dụng và server, đảm bảo tính linh hoạt và bảo mật.

### **Thiết kế giao diện người dùng**

Giao diện người dùng là yếu tố then chốt quyết định trải nghiệm mua sắm của khách hàng. Giao diện cần đảm bảo các tiêu chí: trực quan, thân thiện, hiện đại và phù hợp với sản phẩm cao cấp. Ngoài ra, giao diện còn phải đáp ứng các yêu cầu về khả năng truy cập, tối ưu hóa thao tác và tính thẩm mỹ.

## CHƯƠNG 2. THIẾT KẾ HỆ THỐNG

### 2.1. Xác định tác nhân

**Tác nhân:** Người dùng (*Khách hàng*). Các hành động người dùng có thể thực hiện:

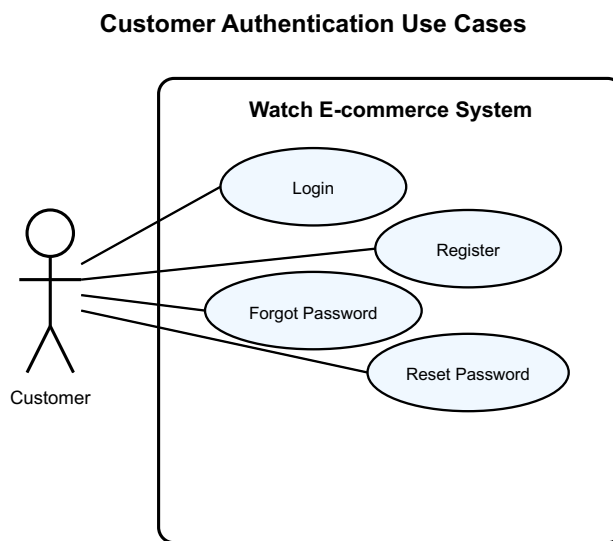
- 🔍 Xem danh sách đồng hồ (theo loại, thương hiệu, giá cả, v.v.)
- 🔍 Tìm kiếm sản phẩm bằng từ khóa
- 📄 Xem chi tiết một sản phẩm cụ thể
- 📄 Đăng ký tài khoản
- 🗝 Đăng nhập / Đăng xuất
- 🛒 Thêm sản phẩm vào giỏ hàng
- ✎ Cập nhật số lượng sản phẩm trong giỏ
- ❌ Xóa sản phẩm khỏi giỏ hàng
- 📦 Đặt hàng (Checkout)
- 👁 Theo dõi trạng thái đơn hàng
- 👤 Cập nhật thông tin cá nhân

**Yêu cầu phi chức năng:**

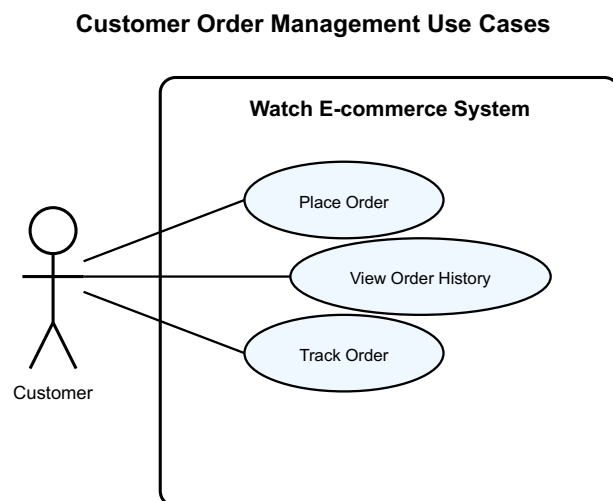
- Hiệu suất: Trang web cần đảm bảo phản hồi nhanh chóng với người dùng. Hỗ trợ xem hình ảnh rõ ràng, giao diện thân thiện dễ dùng.
- Bảo mật: Áp dụng cơ chế mã hóa mật khẩu. Lưu trữ thông tin người dùng trong cơ sở dữ liệu để sử dụng lâu dài.



## 2.2. Sơ đồ usecase của hệ thống

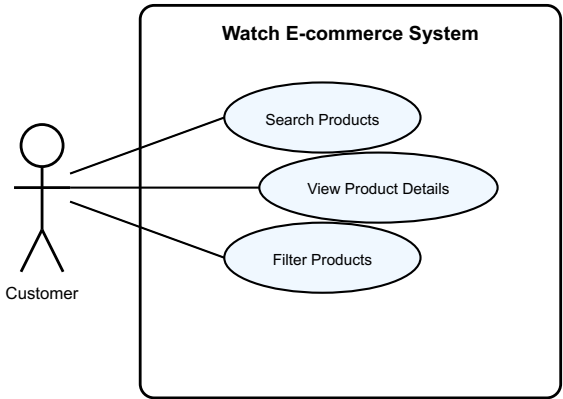


Hình 1: Sơ đồ usecase các chức năng bảo mật hệ thống



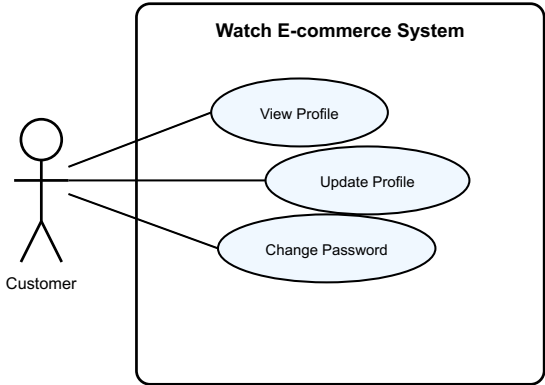
Hình 2: Sơ đồ usecase các chức năng liên quan đến đơn hàng

Customer Product Browsing Use Cases



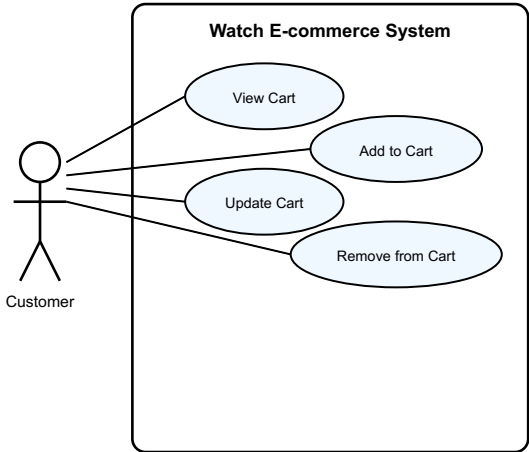
Hình 3: Sơ đồ usecase chức năng xem sản phẩm

Customer Profile Management Use Cases



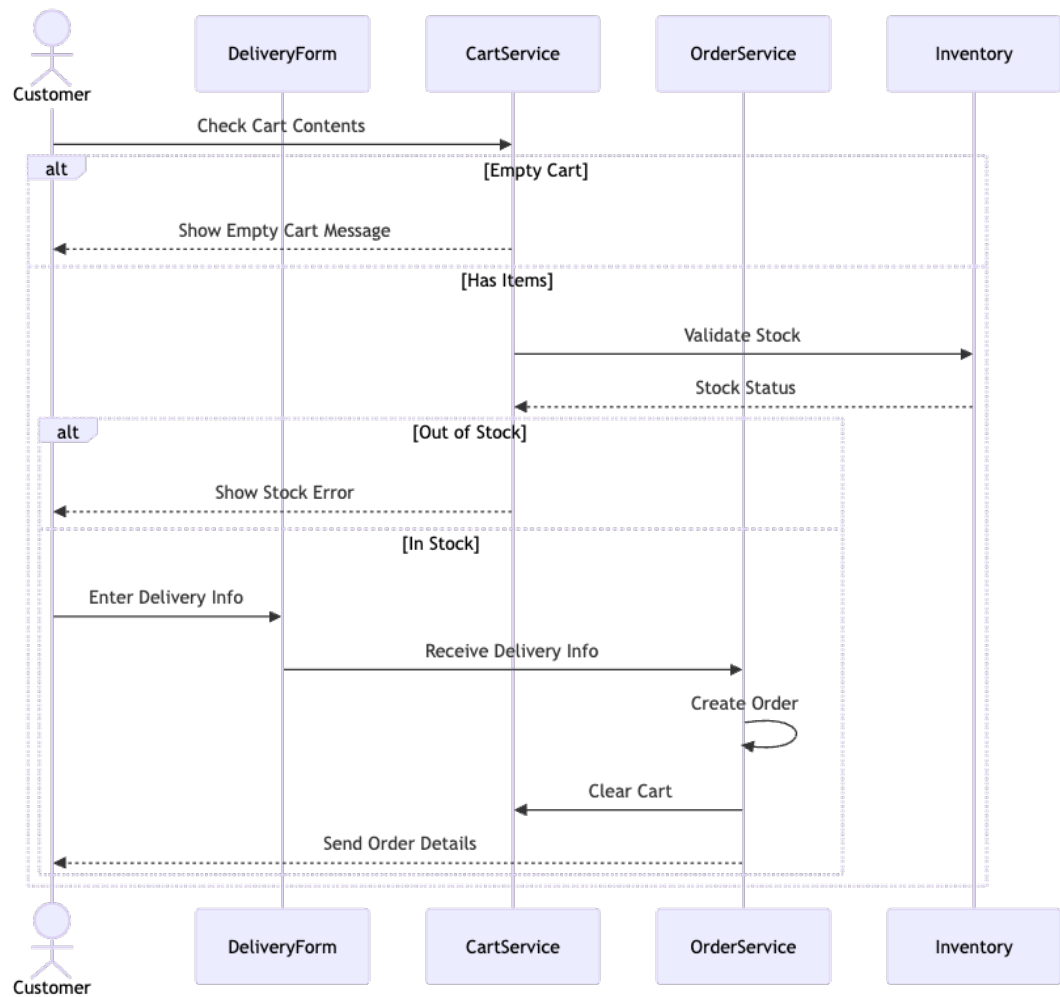
Hình 4: Sơ đồ usecase chức năng quản lí thông tin người dùng

Customer Shopping Cart Use Cases

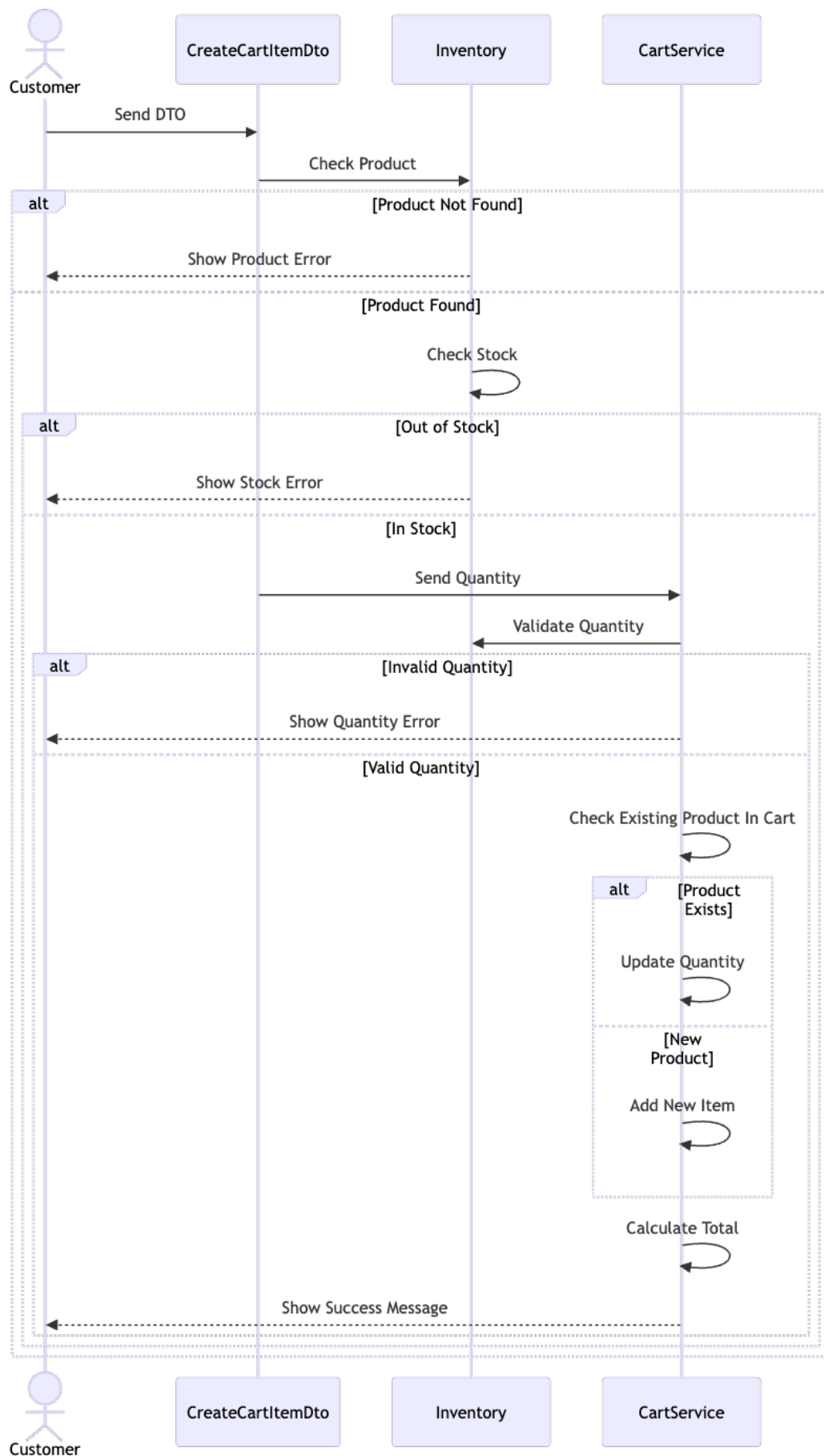


Hình 5: Sơ đồ usecase chức năng quản lí giỏ hàng

## 2.3. Sơ đồ tuần tự cốt lõi của hệ thống

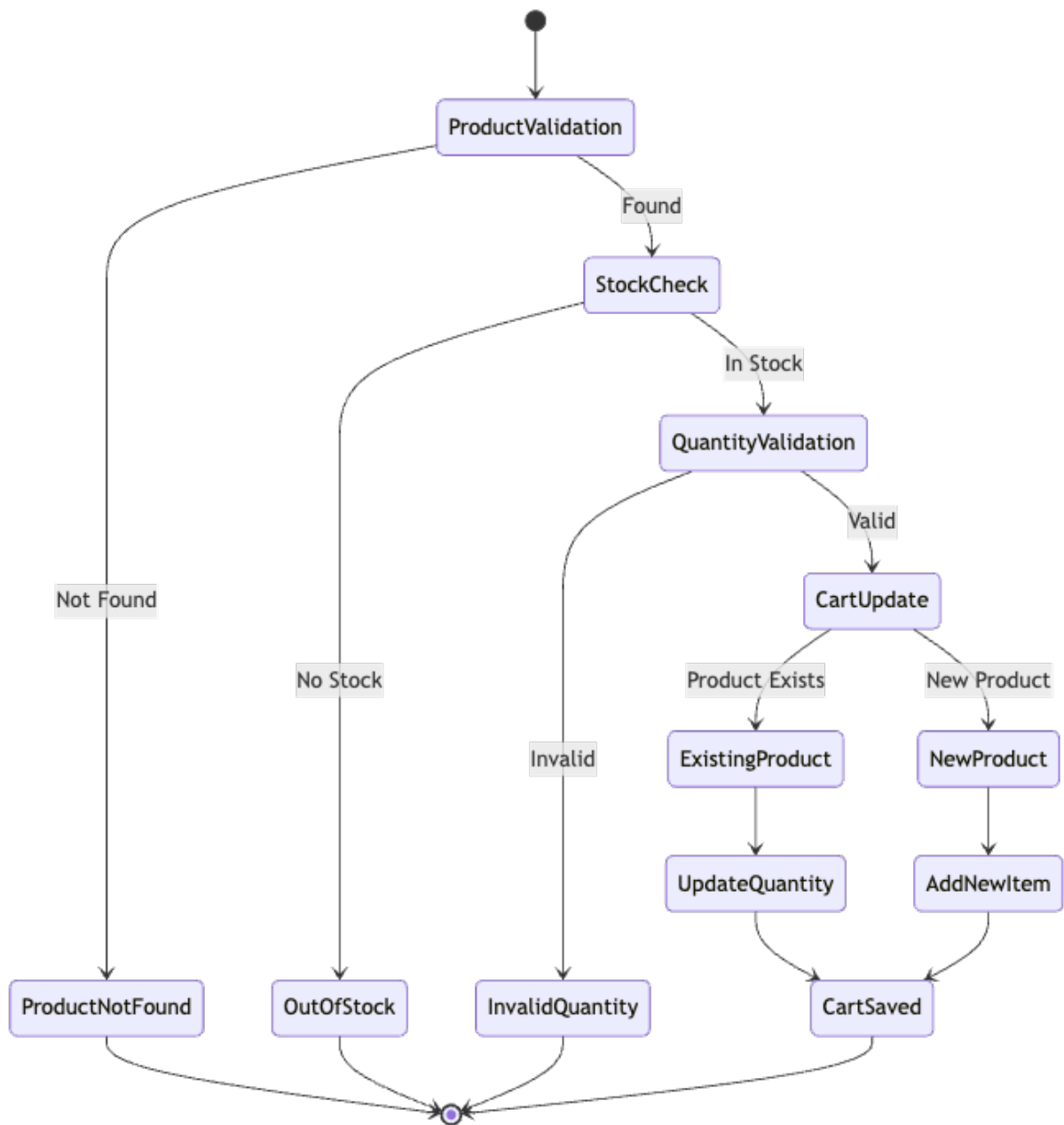


Hình 6: Sơ đồ tuần tự chức năng đặt hàng

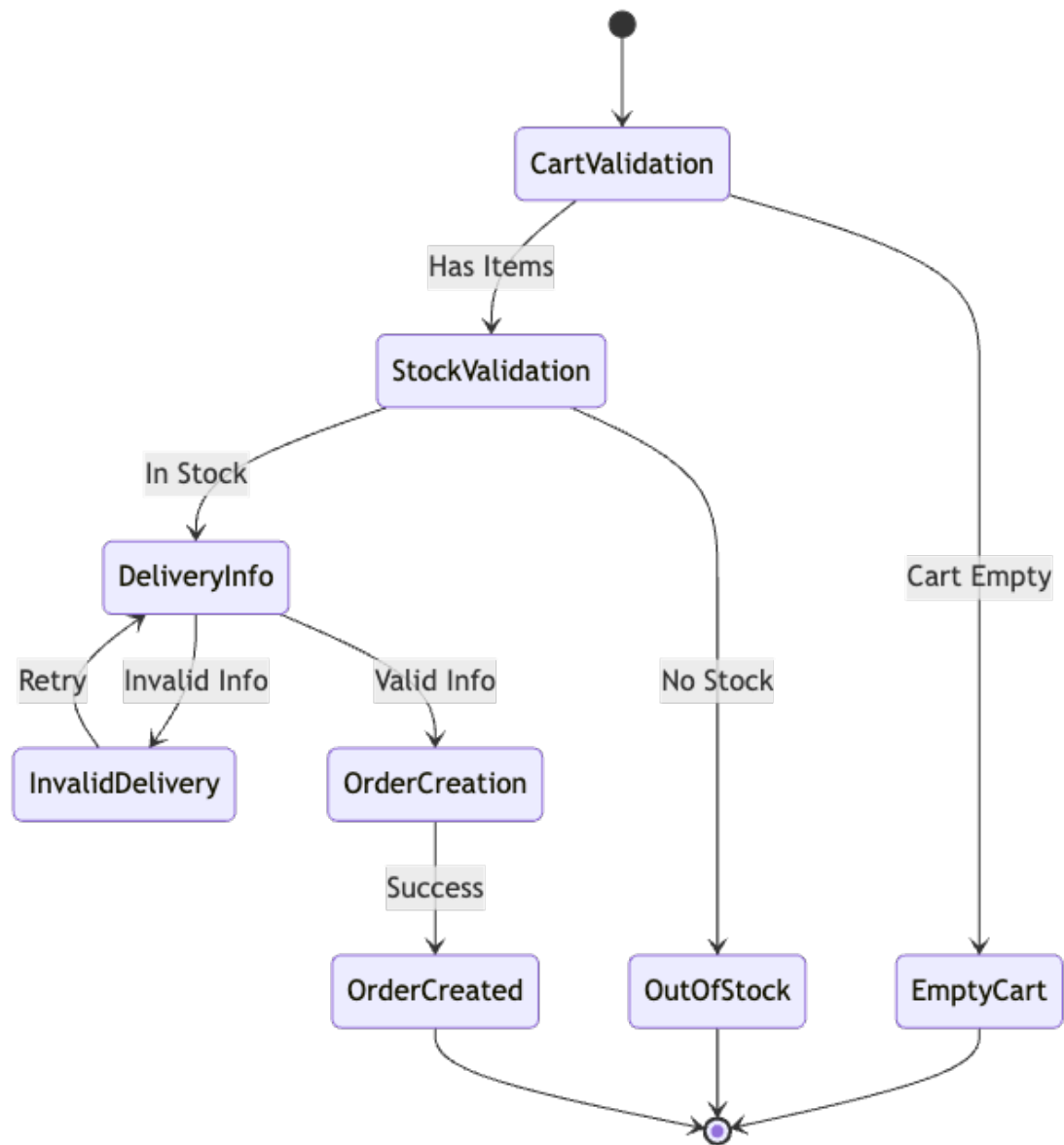


Hình 7: Sơ đồ tuần tự chức năng thêm sản phẩm vào giỏ hàng

## 2.4. Sơ đồ trạng thái cốt lõi của hệ thống

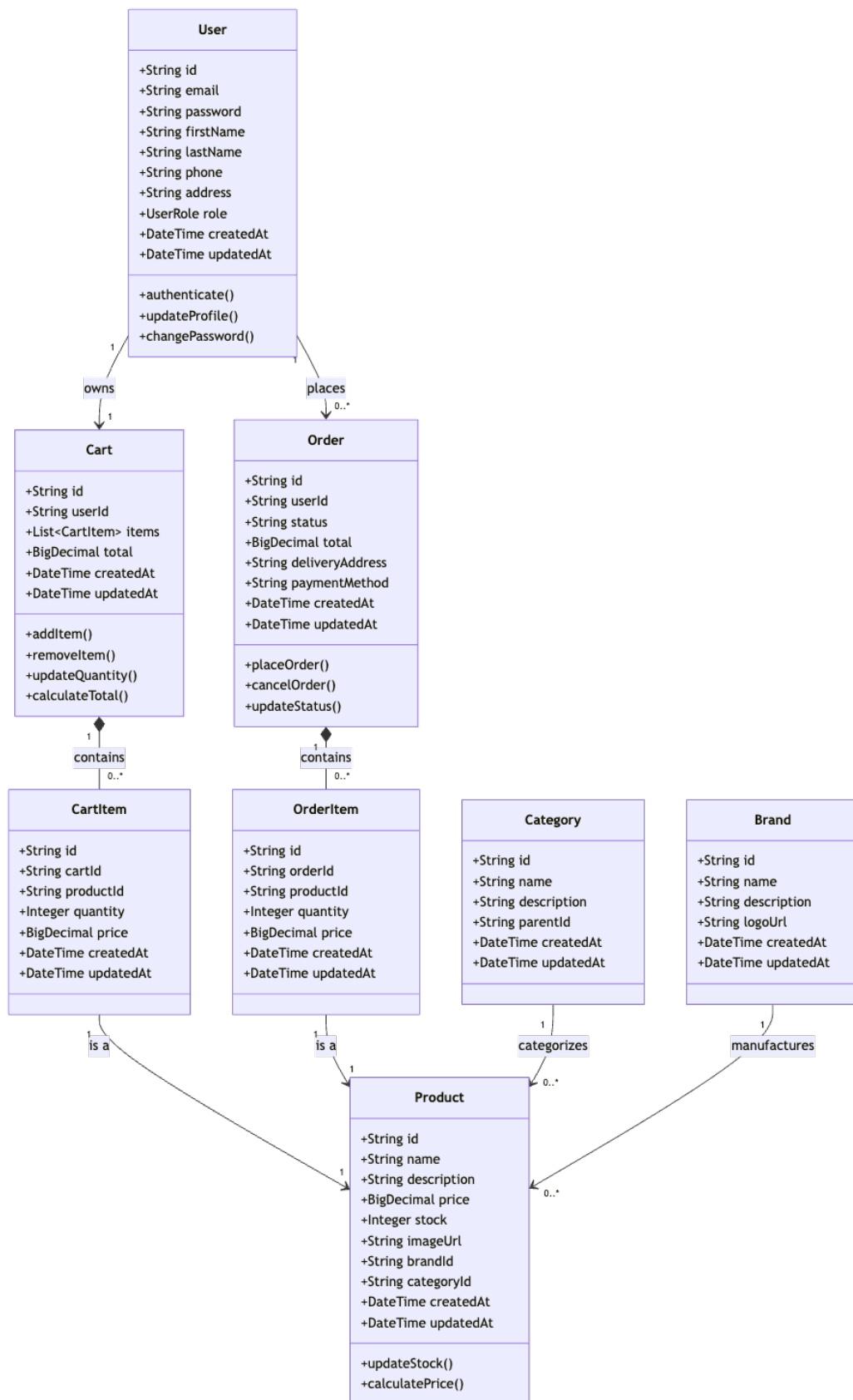


Hình 8: Sơ đồ trạng thái giỏ hàng



Hình 9: Sơ đồ trạng thái đơn hàng

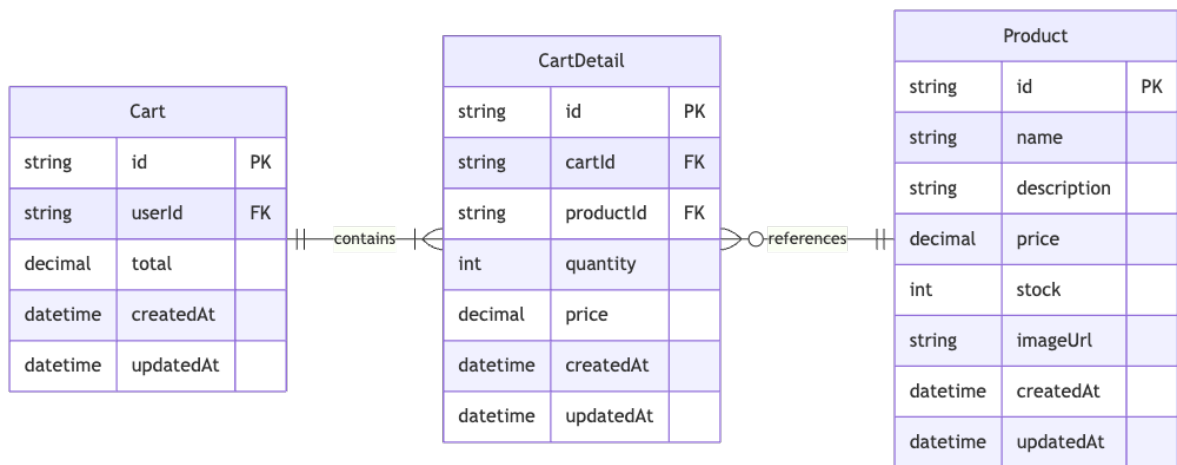
## 2.5. Sơ đồ lớp của hệ thống



Hình 10: Sơ đồ lớp của toàn hệ thống

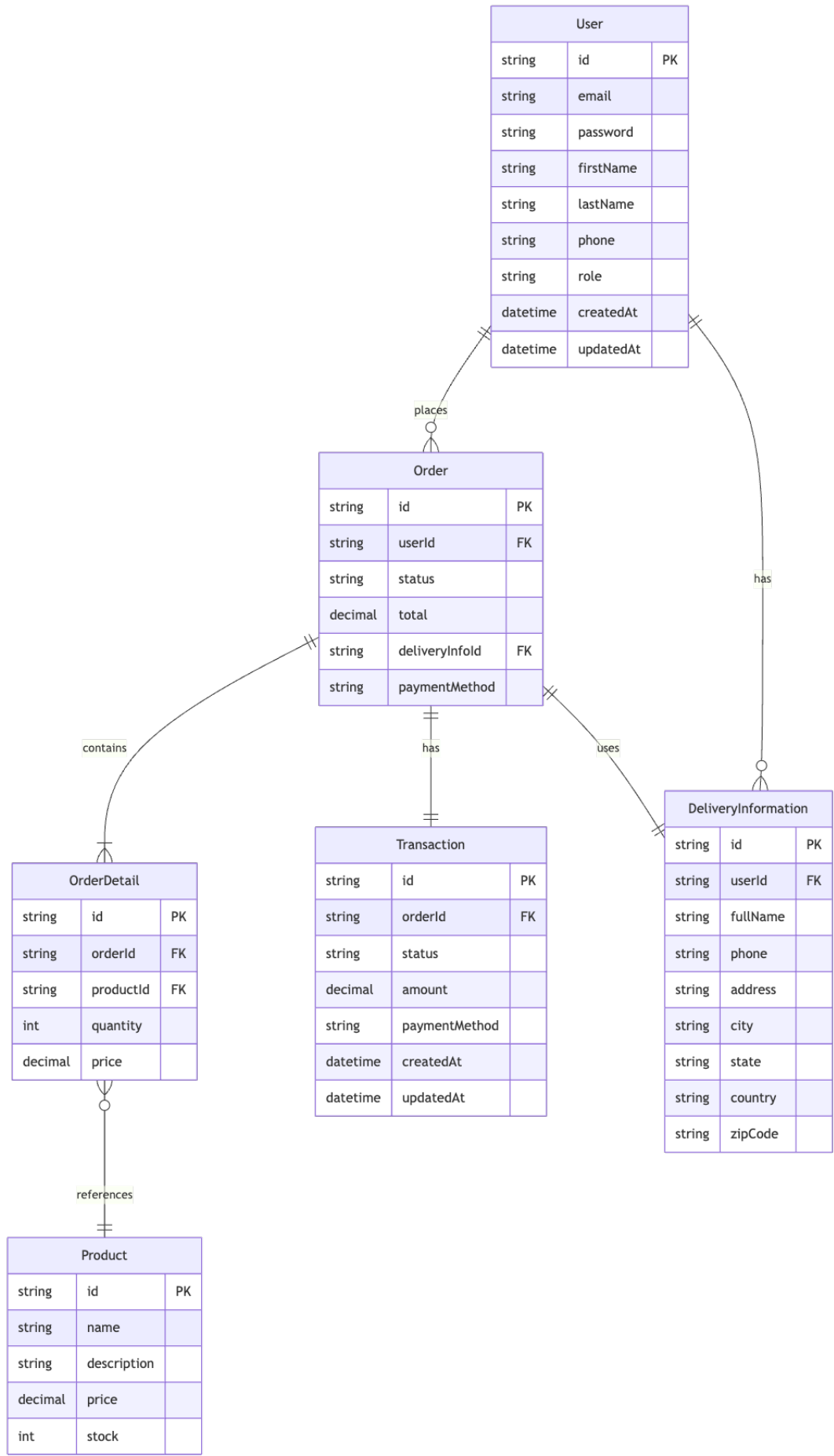
## 2.6. Sơ đồ ERD của hệ thống

Để thể hiện chi tiết, thì sơ đồ ERD của toàn hệ thống được chia ra thành các sơ đồ ERD nhỏ hơn như sau:



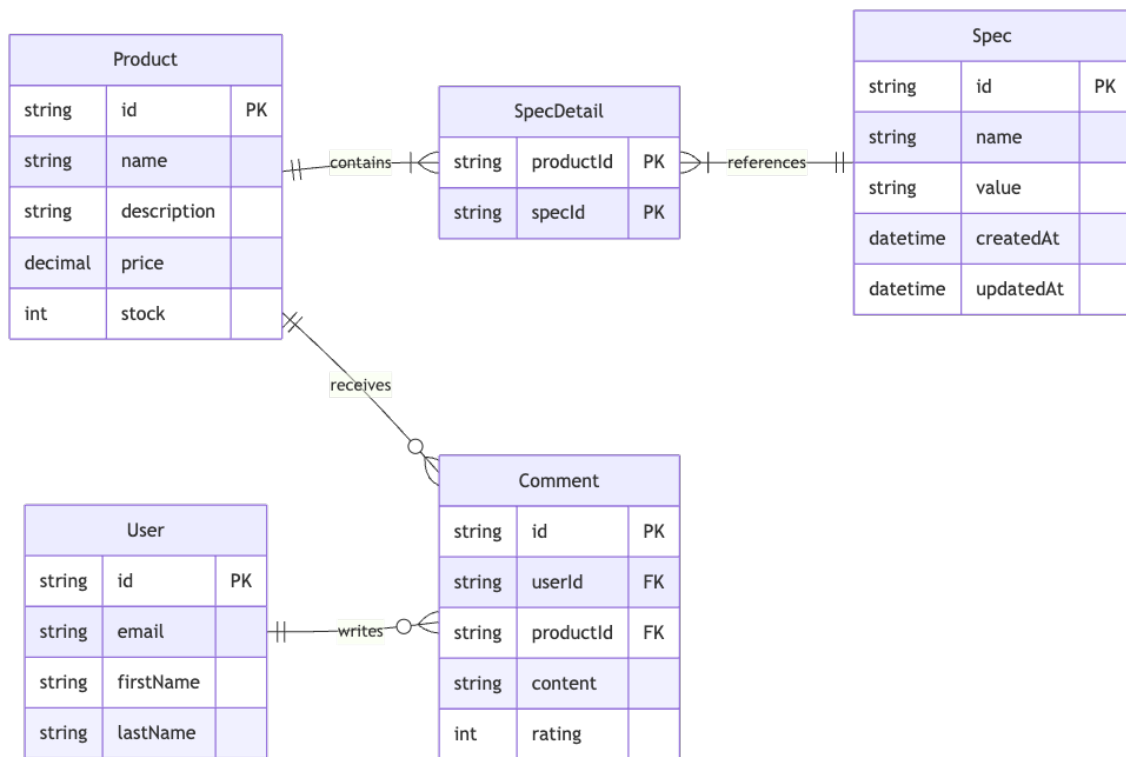
Hình 11: Sơ đồ ERD phân giỏ hàng của hệ thống



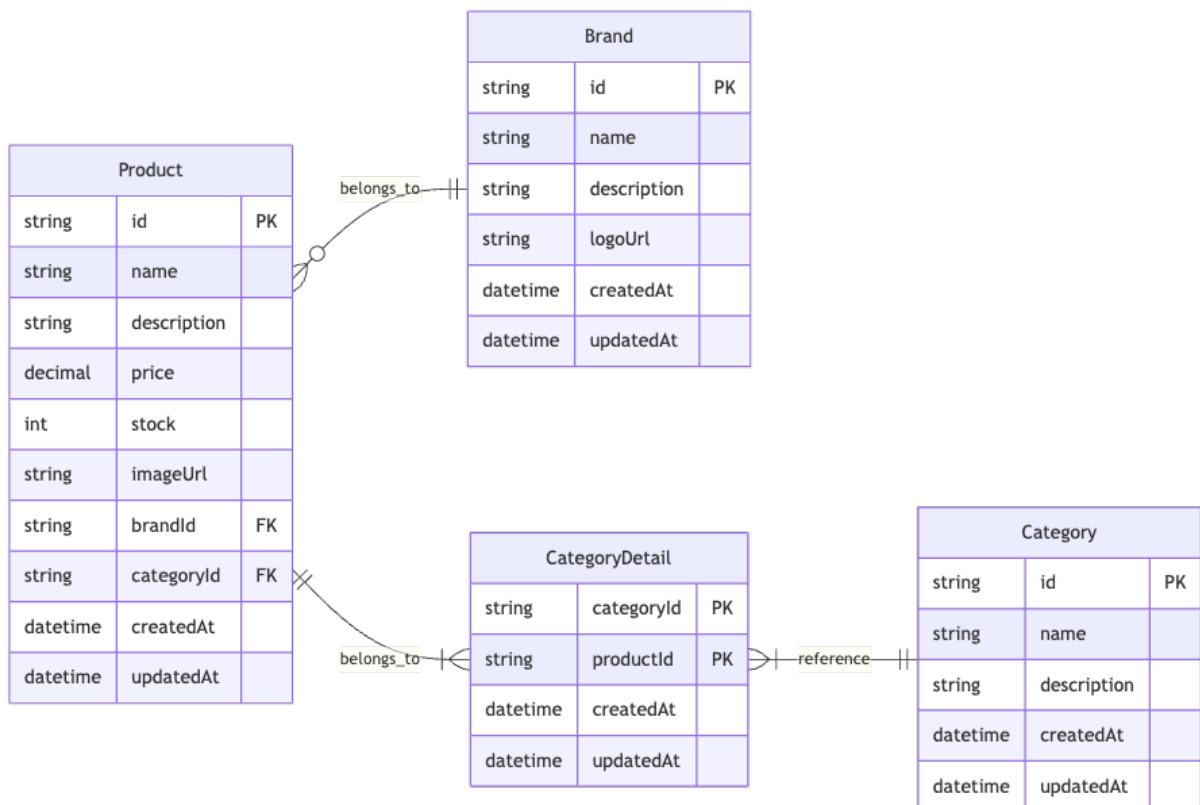


Hình 12: Sơ đồ ERD phần đơn hàng của hệ thống

Báo cáo Đồ án Xây dựng ứng dụng mua đồng hồ Watch Shopping App



Hình 13: Sơ đồ ERD phần sản phẩm, thuộc tính sản phẩm



Hình 14: Sơ đồ ERD phần sản phẩm, nhãn hàng, và danh mục

## CHƯƠNG 3. TRIỂN KHAI HỆ THỐNG

### 3.1. Chuẩn bị môi trường phát triển

#### 3.1.1. Công cụ phát triển ứng dụng Android

Để phát triển ứng dụng mua đồng hồ trên nền tảng Android, nhóm sử dụng Android Studio – môi trường phát triển tích hợp (IDE) chính thức do Google phát triển dành cho lập trình Android. Android Studio hỗ trợ lập trình viên thiết kế giao diện, viết mã nguồn, biên dịch, kiểm thử, giả lập thiết bị và đóng gói ứng dụng một cách hiệu quả. Ứng dụng được phát triển bằng ngôn ngữ Java, giúp tận dụng tối đa các thư viện và tài nguyên phong phú của hệ sinh thái Android.

Một số tính năng nổi bật của Android Studio gồm:

- Hỗ trợ kéo thả, thiết kế giao diện trực quan (Layout Editor)
- Công cụ giả lập nhiều cấu hình thiết bị Android (AVD Manager)
- Tích hợp hệ thống quản lý gói và phụ thuộc Gradle
- Hỗ trợ kiểm thử, debug và profiling ứng dụng
- Cộng đồng hỗ trợ lớn, nhiều plugin tiện ích

#### 3.1.2. Backend API sử dụng trong hệ thống

Ứng dụng Android kết nối và tương tác với hệ thống backend thông qua các API do dịch vụ NestJS cung cấp. NestJS là một framework Node.js hiện đại, mạnh mẽ, phù hợp xây dựng các hệ thống API RESTful với khả năng mở rộng cao, bảo mật tốt và dễ dàng bảo trì. Trong đồ án này, backend NestJS đã được xây dựng và triển khai sẵn trên server, cung cấp các API chính phục vụ cho các nghiệp vụ như: quản lý người dùng, sản phẩm, giỏ hàng, đơn hàng, xác thực, v.v.

Việc sử dụng backend đã được triển khai giúp nhóm tập trung vào phát triển và hoàn thiện ứng dụng Android, đồng thời đảm bảo ứng dụng có thể giao tiếp ổn định với server thông qua các giao thức HTTP.

Dưới đây là hình ảnh các API endpoints của backend mà hệ thống sử dụng:

Watch E-Commerce API Service <span>1.0</span> <span>OAS 3.0</span>		
API Service for Watch E-Commerce store		
Authentications		
POST	/api/v1/auth/sign-in	User sign-in
POST	/api/v1/auth/sign-up	User registration
GET	/api/v1/auth/sign-out	User sign-out
POST	/api/v1/auth/revoke-tokens	Revoke tokens
PATCH	/api/v1/auth/update-password	Update user password
POST	/api/v1/auth/forgot-password	

Hình 15: Các API endpoints liên quan đến bảo mật

Brands		
POST	/api/v1/brands	Allowed user roles: [ADMIN, EMPLOYEE] - Create a new brand
GET	/api/v1/brands	Find brands
GET	/api/v1/brands/slug/{slug}	Find brand by slug
GET	/api/v1/brands/{id}	Find brand by ID
PATCH	/api/v1/brands/{id}	Allowed user roles: [ADMIN, EMPLOYEE] - Update a brand
DELETE	/api/v1/brands/{id}	Allowed user roles: [ADMIN, EMPLOYEE] - Delete a brand
PATCH	/api/v1/brands/assets/{id}	Update brand asset

Hình 16: Các API endpoint liên quan đến nhãn hàng

Carts		
GET	/api/v1/carts	Find user cart
POST	/api/v1/carts/details	Create new detail in user cart
PATCH	/api/v1/carts/details/{id}	Update cart detail in user cart
DELETE	/api/v1/carts/details/{id}	Delete cart detail in user cart

Hình 17: Các API endpoint liên quan đến giỏ hàng

Categories		
POST	/api/v1/categories	Allowed user roles: [ADMIN, EMPLOYEE] - Create a new category
GET	/api/v1/categories	Find categories
GET	/api/v1/categories/slug/{slug}	Find category by slug
GET	/api/v1/categories/{id}	Find category by ID
PATCH	/api/v1/categories/{id}	Allowed user roles: [ADMIN, EMPLOYEE] - Update a category
DELETE	/api/v1/categories/{id}	Allowed user roles: [ADMIN, EMPLOYEE] - Delete a category
PATCH	/api/v1/categories/assets/{id}	Update category asset

Hình 18: Các API endpoint liên quan đến danh mục

Orders				^
POST	/api/v1/orders	Create a new order	🔒	▼
GET	/api/v1/orders	Find orders by filters	🔒	▼
GET	/api/v1/orders/{id}	Find an order by ID	🔒	▼
PATCH	/api/v1/orders/{id}	Allowed user roles: [ADMIN, EMPLOYEE] - Update order status	🔒	▼

Hình 19: Các API endpoint liên quan đến đơn hàng

Products				^
POST	/api/v1/products	Allowed user roles: [ADMIN, EMPLOYEE] - Create a new product	🔒	▼
GET	/api/v1/products	Find products with filters		▼
GET	/api/v1/products/specifications	Find all product specification for building product filter		▼
GET	/api/v1/products/{id}	Find a product by ID		▼
PATCH	/api/v1/products/{id}	Allowed user roles: [ADMIN, EMPLOYEE] - Update a product	🔒	▼
PATCH	/api/v1/products/assets/{id}	Allowed user roles: [ADMIN, EMPLOYEE] - Update product assets	🔒	▼

Hình 20: Các API endpoint liên quan đến sản phẩm

Users				^
GET	/api/v1/users/my	Get user profile	🔒	▼
GET	/api/v1/users	Find users	🔒	▼
PATCH	/api/v1/users	Update user profile	🔒	▼
PATCH	/api/v1/users/avatar	Update user avatar	🔒	▼

Hình 21: Các API endpoint liên quan đến người dùng

## 3.2. Triển khai ứng dụng Android

### 3.2.1. Khởi tạo dự án và cấu trúc thư mục

Ứng dụng được phát triển bằng ngôn ngữ Java trên nền tảng Android Studio. Khi khởi tạo project, nhóm lựa chọn cấu trúc project dạng “Empty Activity” để linh hoạt xây dựng các module chức năng. Thư mục dự án được tổ chức theo mô hình phân lớp rõ ràng, bao gồm các package chính như:

- activities/: Chứa các activity chính (trang chủ, đăng nhập, chi tiết sản phẩm, giỏ hàng, v.v.)
- fragments/: Quản lý các màn hình con và luồng chuyển đổi giao diện
- adapters/: Quản lý dữ liệu hiển thị cho các RecyclerView/ListView
- models/: Định nghĩa các lớp dữ liệu (sản phẩm, người dùng, đơn hàng, v.v.)
- network/: Quản lý các lớp gọi API đến backend
- utils/: Các tiện ích chung (validate, format, xử lý ảnh, v.v.)

### 3. 2. 2. Mô hình kiến trúc MVVM

Ứng dụng Android được xây dựng theo mô hình kiến trúc MVVM (Model – View – ViewModel) nhằm tách biệt rõ ràng giữa giao diện người dùng và logic xử lý dữ liệu, giúp ứng dụng dễ bảo trì, mở rộng và kiểm thử.

- View: Bao gồm các Activity, Fragment, layout XML, chịu trách nhiệm hiển thị dữ liệu, nhận và xử lý các sự kiện tương tác từ người dùng (ví dụ: nhấn nút, nhập liệu).
- ViewModel: Là lớp trung gian giữa View và Model, giữ dữ liệu cần thiết cho View và xử lý logic chuyển đổi/truy xuất dữ liệu. ViewModel không tham chiếu trực tiếp tới View nên không bị ràng buộc vòng đời UI, thuận tiện cho việc kiểm thử.
- Model: Bao gồm các class dữ liệu, xử lý truy xuất dữ liệu từ API hoặc cơ sở dữ liệu cục bộ (Repository, DataSource...). Model là nơi tổ chức, lưu trữ và cập nhật dữ liệu cho toàn ứng dụng.

### 3. 2. 3. Luồng dữ liệu giữa các thành phần

1. Người dùng thao tác trên View (ví dụ: bấm vào nút “Thêm vào giỏ hàng”).
2. View gửi sự kiện (event) tới ViewModel (gọi hàm của ViewModel).
3. ViewModel xử lý logic, gọi tới Model/Repository để lấy hoặc cập nhật dữ liệu (gửi request API, truy vấn database...).
4. Model/Repository thực hiện thao tác (ví dụ: gọi API lấy danh sách sản phẩm), trả kết quả về ViewModel.
5. ViewModel cập nhật các thuộc tính dữ liệu (thường dùng LiveData, Observable, hoặc các callback).
6. View quan sát các thuộc tính này, tự động cập nhật giao diện khi dữ liệu thay đổi.

### 3. 2. 4. Vai trò các class chính

- Activity/Fragment (View): Hiển thị giao diện, khởi tạo ViewModel, quan sát dữ liệu thay đổi để cập nhật UI.
- ViewModel: Xử lý các sự kiện logic, lưu trữ trạng thái màn hình, gọi tới Repository để thao tác dữ liệu, cung cấp dữ liệu cho View dưới dạng LiveData/Observable.
- Model/Entity: Định nghĩa các đối tượng dữ liệu (Product, User, CartItem, Order...).
- Repository: Đóng vai trò trung gian giữa ViewModel và các nguồn dữ liệu (API service, local storage), che giấu chi tiết việc lấy dữ liệu.
- ApiService/Network Layer: Định nghĩa các phương thức giao tiếp với backend (Retrofit interface).

- Adapter: Liên kết dữ liệu với các thành phần giao diện dạng danh sách (RecyclerView, ListView...).
- Utils: Các class tiện ích dùng chung (format dữ liệu, validate, xử lý hình ảnh...).

### 3. 2. 5. Ưu điểm mô hình MVVM trong đồ án

- Dễ mở rộng và bảo trì: Logic và giao diện tách rời, dễ chỉnh sửa từng phần.
- Tối ưu hiệu suất UI: Nhờ LiveData/Observable, UI cập nhật tự động khi dữ liệu thay đổi, hạn chế lỗi do bất đồng bộ.
- Dễ kiểm thử: ViewModel không phụ thuộc UI nên có thể test đơn vị logic dễ dàng.
- Phù hợp ứng dụng có nhiều màn hình, dữ liệu động như thương mại điện tử.

### 3. 2. 6. Kết nối backend và xử lý dữ liệu

Để giao tiếp với backend NestJS, nhóm sử dụng các thư viện hỗ trợ như Retrofit hoặc OkHttp giúp thực hiện các yêu cầu HTTP/HTTPS (GET, POST, PUT, DELETE). Tất cả các endpoint được backend cung cấp đều được khai báo rõ ràng, đảm bảo đúng định dạng dữ liệu và bảo mật thông qua JWT token nếu có.

Các bước kết nối API cơ bản gồm:

- Định nghĩa các interface API và phương thức tương ứng (login, lấy danh sách sản phẩm, thêm vào giỏ hàng, đặt hàng, v.v.)
- Sử dụng các model dữ liệu tương ứng với cấu trúc trả về của backend
- Xử lý response, hiển thị dữ liệu hoặc thông báo lỗi cho người dùng

### 3. 2. 7. Xây dựng các chức năng chính

Ứng dụng được xây dựng với đầy đủ các chức năng cần thiết:

- Đăng ký, đăng nhập: Cho phép người dùng tạo tài khoản mới hoặc đăng nhập để sử dụng các tính năng nâng cao. Dữ liệu xác thực được lưu trữ tạm thời (SharedPreferences) để duy trì phiên đăng nhập.
- Quản lý hồ sơ cá nhân: Người dùng có thể chỉnh sửa thông tin, đổi mật khẩu, cập nhật địa chỉ giao hàng.
- Hiển thị danh mục và chi tiết sản phẩm: Toàn bộ đồng hồ được phân loại (theo thương hiệu, loại dây, giá, v.v.), hỗ trợ tìm kiếm và lọc sản phẩm nhanh chóng.
- Quản lý giỏ hàng: Cho phép thêm, sửa, xóa sản phẩm trong giỏ; cập nhật số lượng; tính tổng tiền đơn hàng.

- Đặt hàng và theo dõi đơn hàng: Hỗ trợ nhập thông tin giao hàng, lựa chọn phương thức thanh toán và kiểm tra trạng thái đơn hàng (đang xử lý, giao hàng, hoàn tất).

### 3. 2. 8. Lưu trữ dữ liệu cục bộ

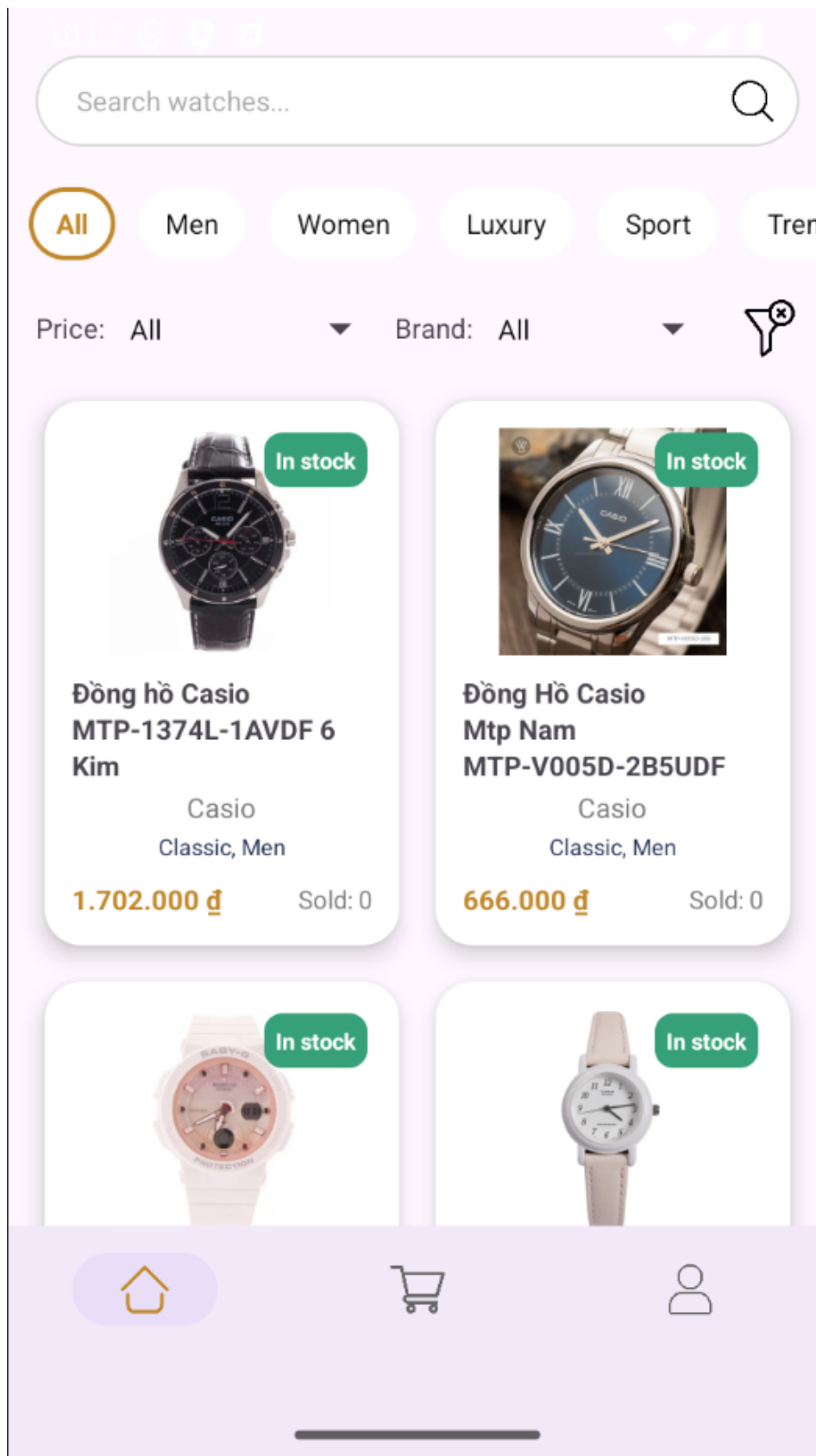
Sau khi đăng nhập, access token đã được xác thực sẽ lưu trữ tạm thời trên thiết bị người dùng thông qua SharedPreferences nhằm nâng cao trải nghiệm và bảo mật.

### 3. 2. 9. Kiểm thử và hoàn thiện

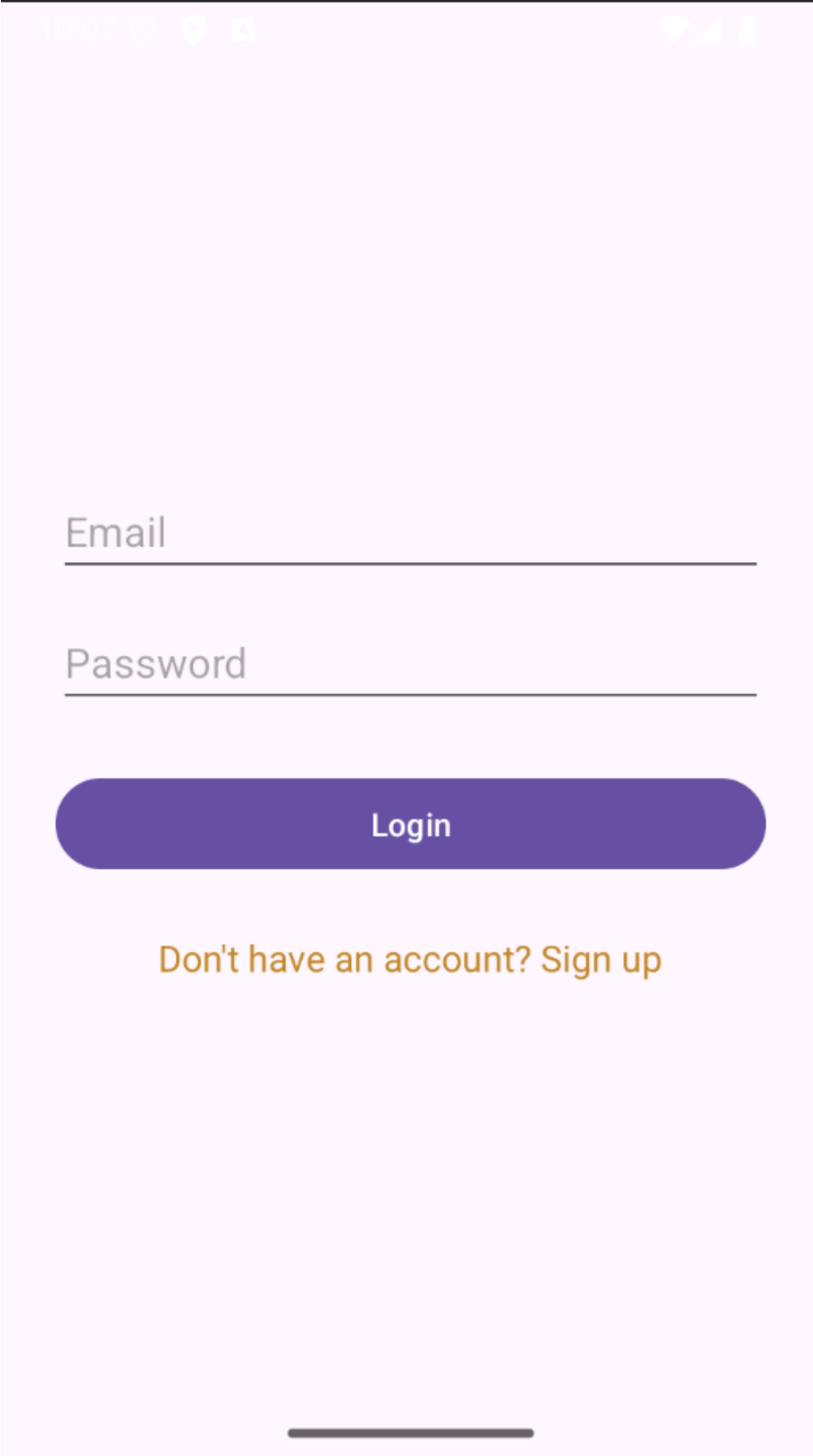
Ứng dụng được kiểm thử trên nhiều thiết bị và phiên bản Android khác nhau để đảm bảo hoạt động ổn định, giao diện hiển thị tốt và các tính năng vận hành mượt mà. Quá trình kiểm thử bao gồm kiểm thử chức năng, kiểm thử giao diện, kiểm thử hiệu năng và kiểm thử tích hợp với backend.



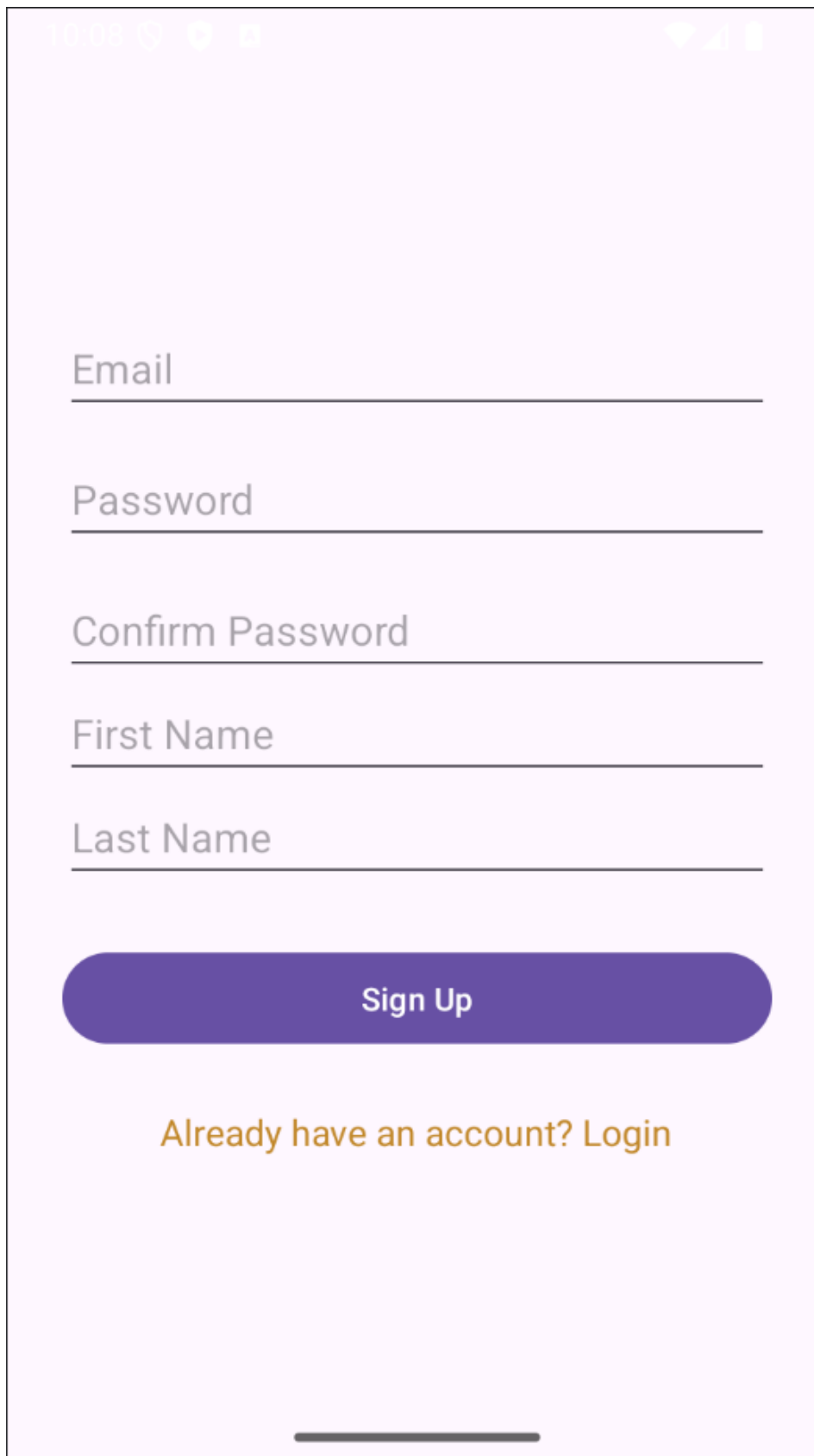
### 3.3. Hình ảnh giao diện của ứng dụng



Hình 22: Giao diện trang chủ (home) của ứng dụng

The image shows a mobile application login screen. At the top, there is a status bar with the time 10:07 and icons for signal, Wi-Fi, and battery. The main area has a light purple background. It features two input fields: 'Email' and 'Password', both with light blue borders and placeholder text. Below these fields is a large, rounded purple button with the text 'Login' in white. Under the button, there is a link that says 'Don't have an account? Sign up' in a brown color. At the bottom, there is a dark horizontal bar representing the home indicator.

Hình 23: Giao diện đăng nhập của ứng dụng

A mobile application registration screen with a light pink background. At the top, there is a status bar showing the time 10:08 and various icons. The main content area contains five text input fields with labels: 'Email', 'Password', 'Confirm Password', 'First Name', and 'Last Name'. Below these fields is a large, rounded purple button with the text 'Sign Up'. Under the button, there is a link that says 'Already have an account? Login' in a brown color. At the very bottom, there is a thin horizontal line representing the home indicator bar.

10:08

Email

Password

Confirm Password

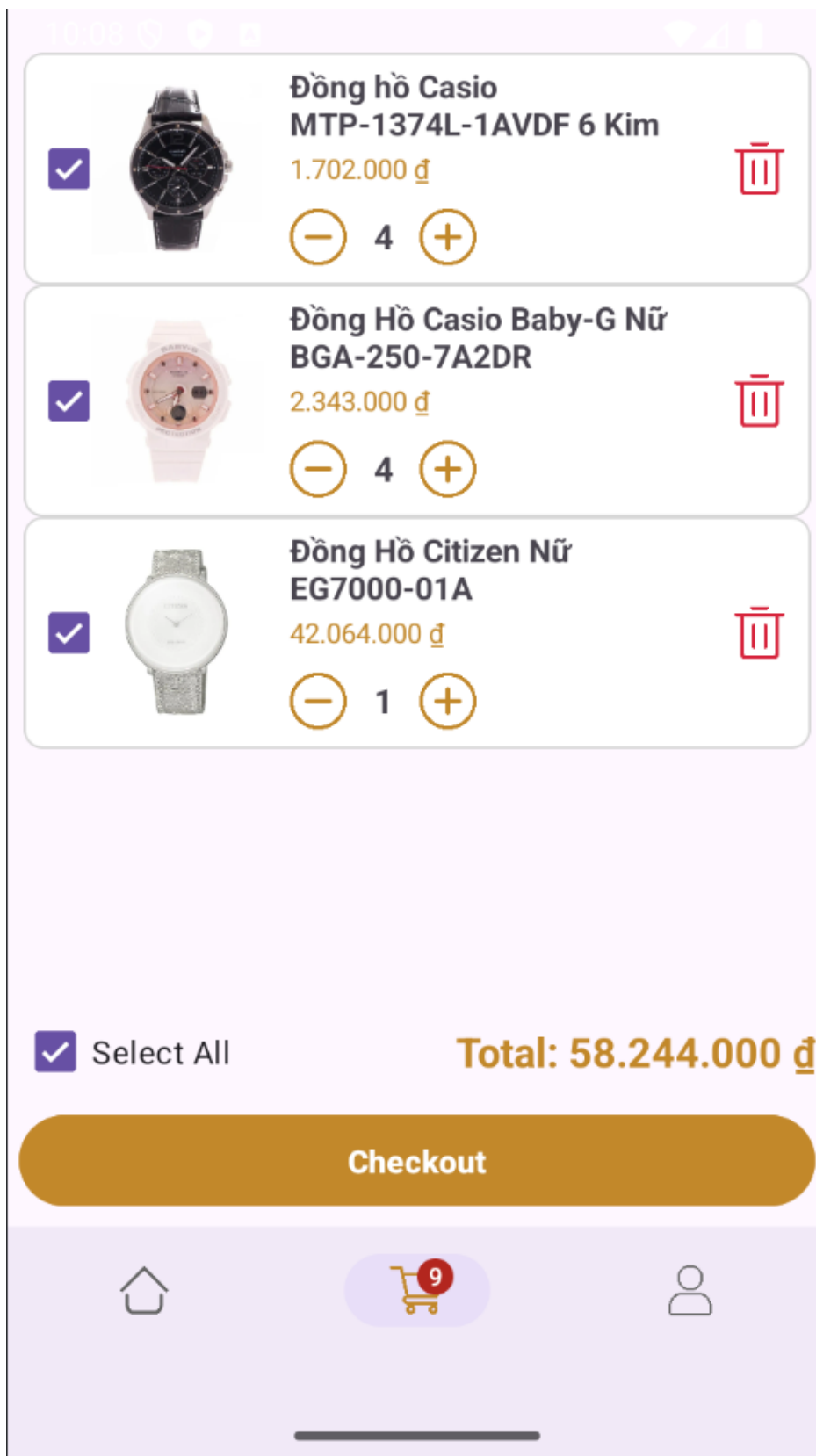
First Name

Last Name

Sign Up

Already have an account? Login

Hình 24: Giao diện đăng kí của ứng dụng



Hình 25: Giao diện giỏ hàng của ứng dụng

10:08

Change Password

Email: admin@gmail.com

Nguyễn

Nhật Minh 1

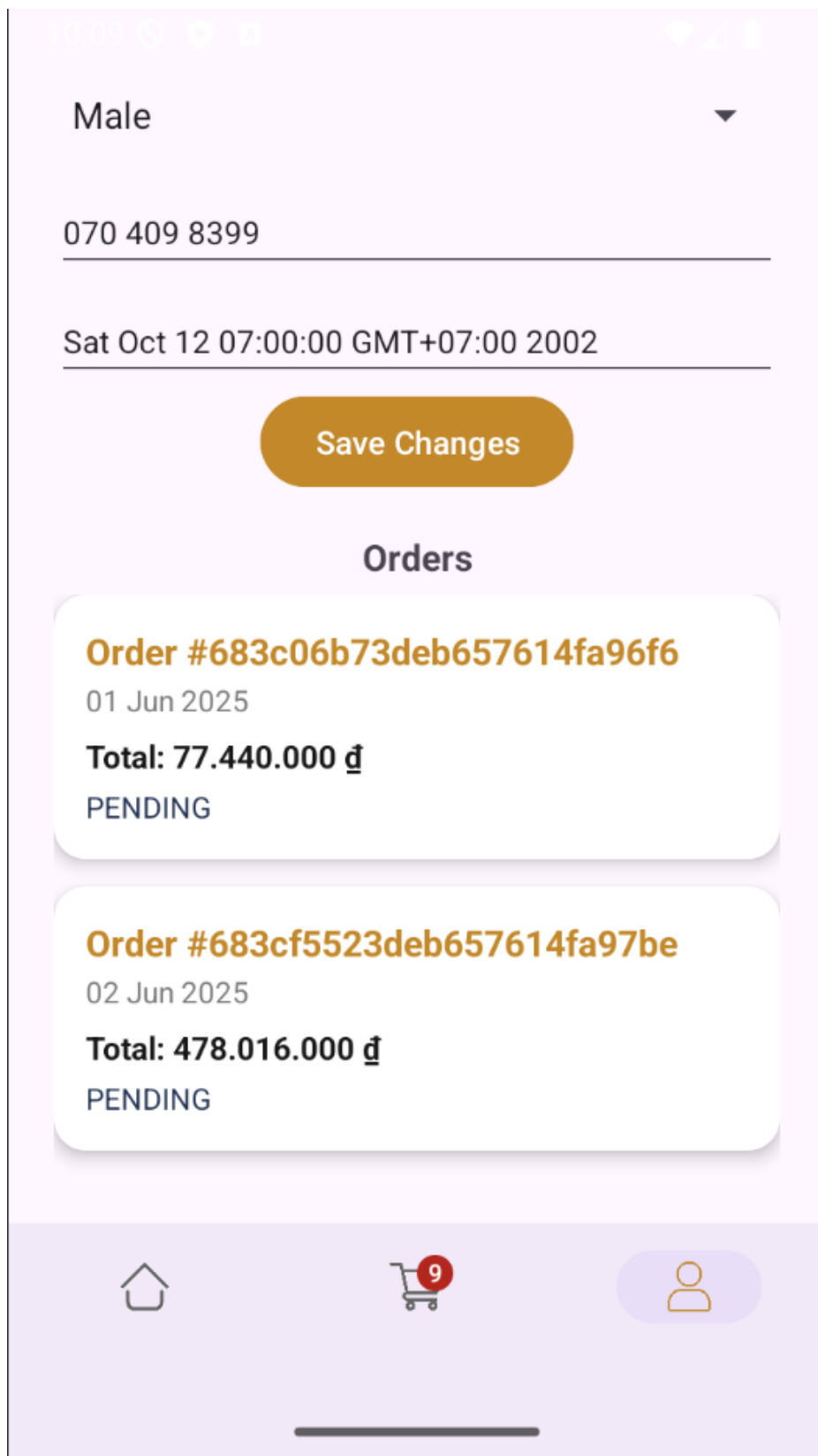
Male

070 409 8399

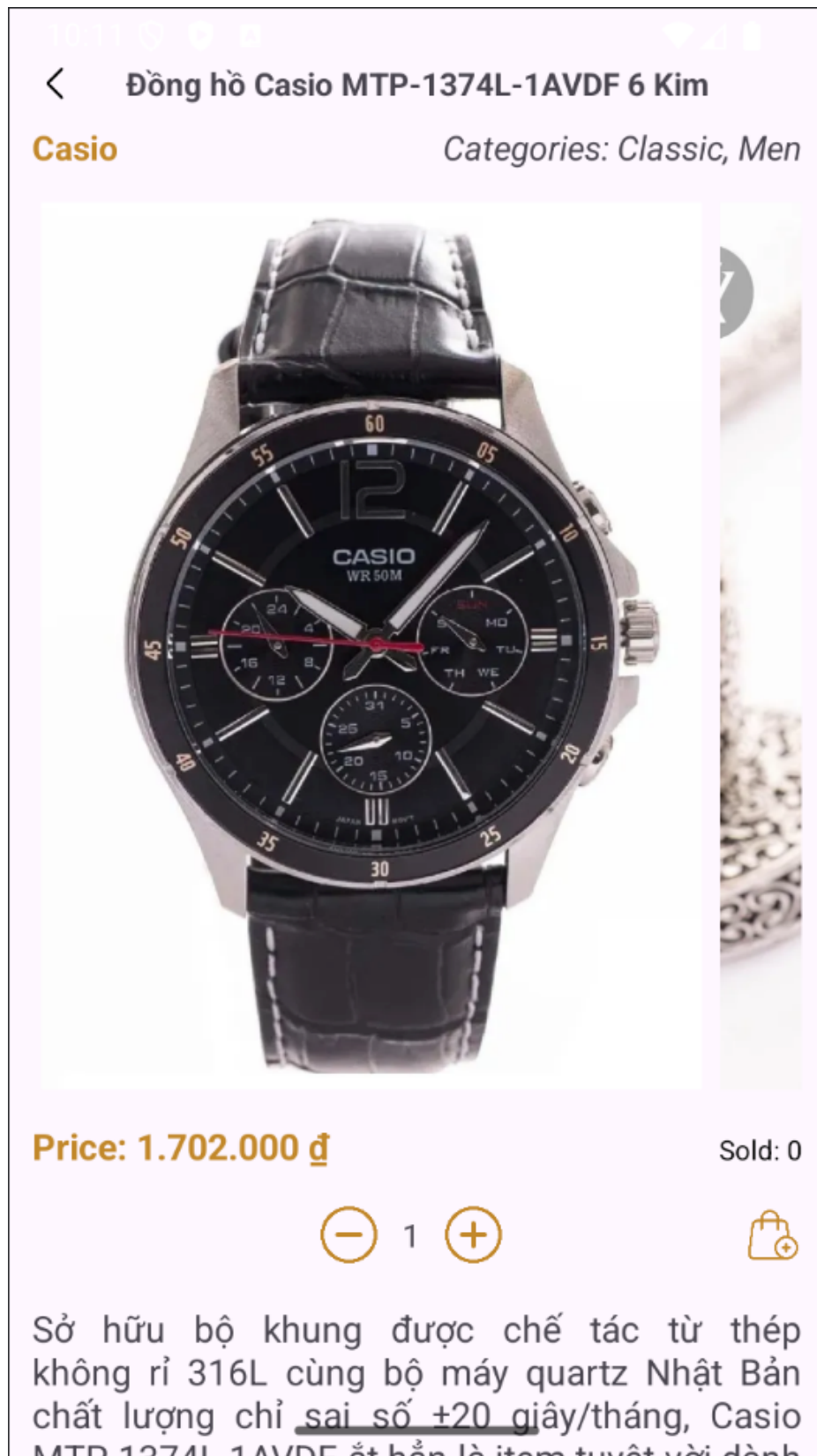
Sat Oct 12 07:00:00 GMT+07:00 2002

Save Changes

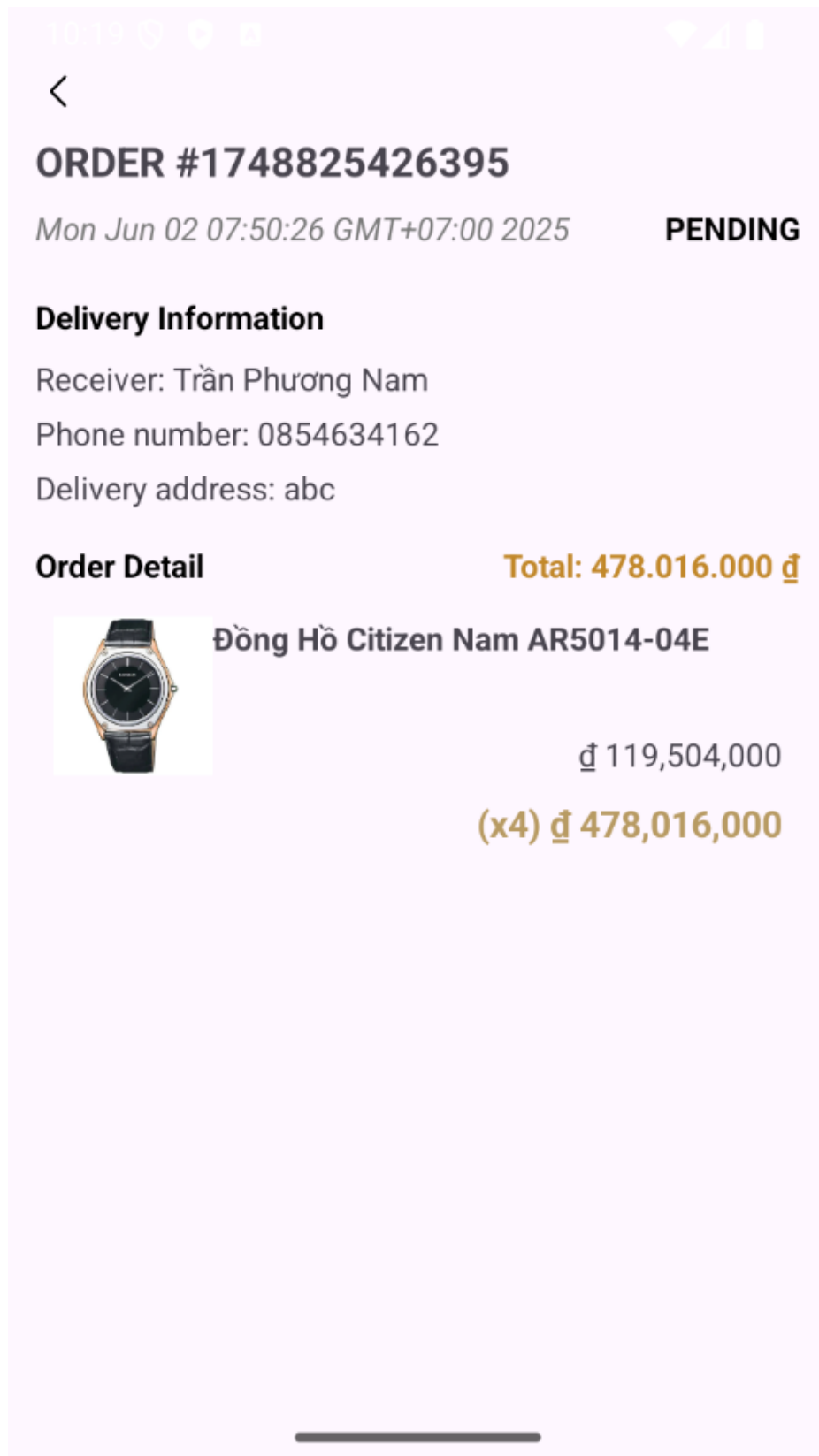
Hình 26: Giao diện trang cá nhân người dùng của ứng dụng



Hình 27: Giao diện danh sách đơn hàng người dùng của ứng dụng



Hình 28: Giao diện chi tiết sản phẩm của ứng dụng



Hình 29: Giao diện chi tiết đơn hàng người dùng của ứng dụng



## CHƯƠNG 4. KẾT LUẬN

### 4.1. Tổng kết quá trình thực hiện đề tài

Đề tài “Xây dựng ứng dụng mua đồng hồ (Watch Shopping App)” đã được nhóm nghiên cứu, thiết kế và triển khai thành công trên nền tảng Android. Ứng dụng đáp ứng đầy đủ các chức năng cơ bản của một hệ thống thương mại điện tử hiện đại như: quản lý sản phẩm, hiển thị danh mục và chi tiết sản phẩm, tìm kiếm – lọc sản phẩm, quản lý giỏ hàng, đặt mua và theo dõi trạng thái đơn hàng, quản lý tài khoản người dùng, cùng với giao diện thân thiện, hiện đại và phù hợp với ngành hàng thời trang cao cấp.

Thông qua quá trình thực hiện đề tài, nhóm đã vận dụng hiệu quả các kiến thức về lập trình Android, thiết kế giao diện người dùng, giao tiếp với backend qua API cũng như các kỹ năng làm việc nhóm, quản lý tiến độ và giải quyết vấn đề thực tiễn phát sinh. Ngoài ra, việc kết nối với backend NestJS giúp nhóm hiểu rõ hơn về mô hình client-server, xử lý dữ liệu bất đồng bộ, xác thực và bảo mật trong ứng dụng di động.

Mặc dù đã cố gắng hoàn thiện đề tài, ứng dụng vẫn còn một số hạn chế nhất định như giao diện chưa thực sự tối ưu cho mọi thiết bị, chưa tích hợp thanh toán trực tuyến hoặc các công nghệ mới như push notification,... Đây sẽ là những hướng phát triển tiếp theo để nâng cao trải nghiệm người dùng và mở rộng chức năng cho hệ thống trong tương lai.

Qua đề tài này, nhóm không chỉ nâng cao kiến thức chuyên môn mà còn rèn luyện được tư duy thiết kế hệ thống, khả năng giải quyết vấn đề và kỹ năng làm việc nhóm – những yếu tố quan trọng cho sự phát triển nghề nghiệp trong lĩnh vực công nghệ thông tin.

*Source code của đề tài:*

1. Source code [sản phẩm ứng dụng Watch Shopping App Andoird](#).
2. Source code [hệ thống Backend API](#).

Backend API đang được deploy ở địa chỉ <http://34.28.186.188/> và tài liệu API của backend đã được deploy ở địa chỉ: <http://34.28.186.188/api-document>. Địa chỉ sẽ được tồn tại đến cuối tháng 6 năm 2025.

### 4.2. Định hướng và phát triển

Mặc dù ứng dụng mua đồng hồ đã đáp ứng được các yêu cầu cơ bản của một hệ thống thương mại điện tử trên nền tảng Android, nhóm nhận thấy vẫn còn nhiều tiềm năng để mở rộng và nâng cao chất lượng sản phẩm trong thời gian tới. Một số định hướng và hướng phát triển cụ thể như sau:

**1. Tích hợp cổng thanh toán trực tuyến:**

Hiện tại, ứng dụng chỉ hỗ trợ hình thức thanh toán khi nhận hàng (COD). Trong tương lai, nhóm dự kiến tích hợp các cổng thanh toán phổ biến như Momo, ZaloPay, VNPay, hoặc thẻ ngân hàng để đa dạng hóa lựa chọn cho khách hàng và nâng cao trải nghiệm mua sắm.

**2. Phát triển tính năng thông báo đẩy (Push Notification):**

Bổ sung chức năng gửi thông báo đẩy giúp người dùng nhận được thông báo về trạng thái đơn hàng, chương trình khuyến mãi hoặc các tin tức mới nhất một cách nhanh chóng và tiện lợi.

**3. Nâng cấp giao diện người dùng (UI/UX):**

Tiếp tục hoàn thiện, tối ưu giao diện để phù hợp với nhiều thiết bị khác nhau, nâng cao tính thẩm mỹ và trải nghiệm người dùng. Xây dựng phiên bản Dark Mode hoặc giao diện tùy chỉnh theo sở thích cá nhân.

**4. Xây dựng hệ thống đánh giá, bình luận sản phẩm:**

Cho phép người dùng đánh giá, nhận xét về sản phẩm đã mua, từ đó tăng tính tương tác và cung cấp thông tin tham khảo hữu ích cho khách hàng khác.

**5. Phát triển ứng dụng trên nhiều nền tảng:**

Mở rộng ứng dụng lên hệ điều hành iOS hoặc phát triển phiên bản web giúp tiếp cận nhiều đối tượng khách hàng hơn.

**6. Tích hợp các công nghệ mới:**

Ứng dụng có thể kết hợp các công nghệ như trí tuệ nhân tạo (AI) để gợi ý sản phẩm phù hợp, hoặc thực tế tăng cường (AR) giúp người dùng thử đồng hồ trên tay thông qua camera điện thoại.

**7. Tăng cường bảo mật và hiệu năng:**

Tiếp tục rà soát, tối ưu các vấn đề bảo mật trong đăng nhập, thanh toán, lưu trữ thông tin cá nhân và nâng cao hiệu năng xử lý của ứng dụng khi dữ liệu lớn.

Với các định hướng trên, nhóm mong muốn phát triển ứng dụng ngày càng hoàn thiện, đáp ứng tốt hơn nhu cầu của người dùng, đồng thời tạo nền tảng vững chắc cho việc mở rộng và ứng dụng thực tế trong tương lai.

## TÀI LIỆU THAM KHẢO

1. Tài liệu về phát triển ứng dụng Android:
  - Android Developers: [Developer Guides](#)
  - Vogella: [Android Tutorials](#)
  - Google Codelabs: [Build your first Android app in Java](#)
2. Tài liệu về kiến trúc phần mềm Android (MVVM):
  - Android Developers: [Guide to app architecture](#)
  - Raywenderlich: [MVVM on Android](#)
3. Tài liệu về giao tiếp API và Retrofit:
  - Square: [Retrofit Documentation](#)
  - Android Developers: [Connect to the network](#)
4. Tài liệu về backend NestJS:
  - NestJS Official Documentation: [NestJS - A progressive Node.js framework](#)
5. Tài liệu về thương mại điện tử:
  - Shopify: [What is E-commerce?](#)
  - Wikipedia: [E-commerce](#)
6. Tài liệu về thiết kế UI/UX cho ứng dụng di động:
  - Google Material Design: [Material Design Guidelines](#)
7. Các bài viết chuyên sâu và thảo luận:
  - Stack Overflow: [Android Development](#)
  - Medium: [Android & Mobile Development](#)