

ĐẠI HỌC BÁCH KHOA HÀ NỘI  
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

**BÀI TẬP LỚN**

MÔN: THIẾT KẾ VÀ XÂY DỰNG PHẦN MỀM

*(Mã học phần: IT4490)*

*Đề tài:*

**AIMS - AN INTERNET MEDIA STORE**

Nhóm :	5
Mã lớp học :	152250
Giảng viên hướng dẫn :	Nguyễn Thị Thu Trang

***Hà Nội, Tháng 1 năm 2025***

<b>1. Giới thiệu</b>	<b>4</b>
1.1. Mục đích	5
1.2. Phạm vi	5
1.3. Từ điển thuật ngữ	5
1.4. Tài liệu tham khảo	6
<b>2. Mô tả tổng quan</b>	<b>6</b>
2.1. Các tác nhân	6
2.2. Biểu đồ usecase	7
2.3. Quy trình nghiệp vụ	8
2.3.1. Quy trình đặt hàng và thanh toán	8
2.3.1.1. Sơ đồ hoạt động	8
2.3.1.2. Mô tả chi tiết	8
2.3.2. Quy trình huỷ đơn hàng	9
2.3.2.1. Sơ đồ hoạt động	9
2.3.2.2. Mô tả chi tiết	9
<b>3. Phân tích yêu cầu</b>	<b>10</b>
3.1. Đặc tả usecase UC001 “Đặt hàng”	10
3.1.1. Mã usecase	10
3.1.2. Giới thiệu	10
3.1.3. Tác nhân	10
3.1.4. Tiền điều kiện	10
3.1.5. Luồng sự kiện chính	10
3.1.6. Luồng sự kiện thay thế	11
3.1.7. Dữ liệu đầu vào	12
3.1.8. Dữ liệu đầu ra	12
3.1.9. Hậu điều kiện	14
3.2. Đặc tả usecase UC002 “Thanh toán”	14
3.2.1. Mã usecase	14
3.2.2. Giới thiệu	14
3.2.3. Tác nhân	14
3.2.4. Tiền điều kiện	14
3.2.5. Luồng sự kiện chính	15
3.2.6. Luồng sự kiện thay thế	15
3.2.7. Dữ liệu đầu vào	15
3.2.8. Dữ liệu đầu ra	15
3.2.9. Hậu điều kiện	16
3.3. Đặc tả usecase UC003 “Đặt hàng nhanh”	16
3.3.1. Mã usecase	16
3.3.2. Giới thiệu	16
3.3.3. Tác nhân	16

3.3.4. Tiền điều kiện	16
3.3.5. Luồng sự kiện chính	16
3.3.6. Luồng sự kiện thay thế	16
3.3.7. Dữ liệu đầu vào	17
3.3.8. Dữ liệu đầu ra	18
3.3.9. Hậu điều kiện	18
3.4. Đặc tả usecase UC004 “Huỷ đơn hàng”	18
3.4.1. Mã usecase	18
3.4.2. Giới thiệu	18
3.4.3. Tác nhân	18
3.4.4. Tiền điều kiện	18
3.4.5. Luồng sự kiện chính	18
3.4.6. Luồng sự kiện thay thế	19
3.4.7. Dữ liệu đầu vào	19
3.4.8. Dữ liệu đầu ra	19
3.4.9. Hậu điều kiện	19
3.5. Đặc tả usecase UC005 “Xem sản phẩm”	19
3.5.1. Mã usecase	19
3.5.2. Giới thiệu	19
3.5.3. Tác nhân	19
3.5.4. Tiền điều kiện	20
3.5.5. Luồng sự kiện chính	20
Dữ liệu đầu vào	21
Dữ liệu đầu ra	21
Hậu điều kiện	21
3.6. Đặc tả usecase UC006 “Tìm kiếm sản phẩm”	21
3.6.1. Mã usecase	21
UC006	21
3.6.2. Giới thiệu	21
3.6.3. Tác nhân	21
3.6.4. Tiền điều kiện	22
3.6.5. Luồng sự kiện chính	22
3.6.6. Luồng sự kiện thay thế	22
3.6.7. Dữ liệu đầu vào	23
3.6.8. Dữ liệu đầu ra	23
3.6.9. Hậu điều kiện	23
3.7. Đặc tả usecase UC007 “Sắp xếp sản phẩm”	23
Mã usecase	23
Giới thiệu	24
Tác nhân	24
Tiền điều kiện	24
Luồng sự kiện chính	24
Luồng sự kiện thay thế	24
Dữ liệu đầu vào	25

Dữ liệu đầu ra	25
Hậu điều kiện	25
3.8. Đặc tả usecase UC008 “Xem lịch sử mua hàng”	25
3.8.1. Mã usecase	25
3.8.2. Giới thiệu	26
3.8.3. Tác nhân	26
3.8.4. Tiền điều kiện	26
3.8.5. Luồng sự kiện chính	26
3.8.6. Luồng sự kiện thay thế	26
3.8.7. Dữ liệu đầu vào	26
3.8.8. Dữ liệu đầu ra	26
3.8.9. Hậu điều kiện	26
<b>4. Kiến trúc hệ thống và thiết kế kiến trúc</b>	<b>26</b>
4.1. Kiến trúc clean architecture	26
4.2. Biểu đồ tương tác	28
4.2.1. Usecase UC01 “Đặt hàng”	28
4.2.2. Usecase UC02 “Thanh toán”	28
4.2.3. Usecase UC03 “Đặt hàng nhanh”	29
4.2.4. Usecase UC04 “Huỷ đơn hàng”	30
4.2.5. Usecase UC06 “Tìm kiếm sản phẩm”	30
4.2.6. Usecase UC07 “Sắp xếp sản phẩm”	31
4.3. Biểu đồ phân tích lớp	31
Usecase UC01 “Đặt hàng”	31
Usecase UC01 “Thanh toán”	31
Usecase UC01 “Đặt hàng nhanh”	31
Usecase UC01 “Huỷ đơn hàng”	31
<b>5. Thiết kế chi tiết</b>	<b>32</b>
5.1. Thiết kế giao diện người dùng	32
5.1.1. Biểu đồ chuyển màn hình	32
5.1.2. Đặc tả màn hình	32
5.1.2.1. Home screen	32
5.1.2.2. Store screen	32
5.1.2.3. Product details	34
5.1.2.4. Cart screen	35
5.1.2.5. Checkout screen	35
5.1.2.6. Place order result screen	36
5.1.2.7. Order tracking screen (Màn hình theo dõi đơn hàng)	36
5.1.2.8. Order details (pop-up Xem chi tiết đơn hàng)	37
5.1.2.9. Biểu đồ lớp	37
5.1.2.10. Biểu đồ trình tự	37
5.1.2.11. Biểu đồ lớp	38
5.1.2.12. Biểu đồ trình tự	38
5.2. Mô hình dữ liệu	38
5.2.1.1. Hệ quản trị cơ sở dữ liệu	38

5.2.1.2. Biểu đồ cơ sở dữ liệu	39
5.2.1.3. Thiết kế chi tiết cơ sở dữ liệu	39
5.3. Thiết kế lớp	40
5.3.1. Tổng quan	40
5.3.2. Chi tiết	40
<b>6. Xem xét thiết kế (Design Consideration)</b>	<b>42</b>
6.1. Goals and Guidelines	42
6.2. Architectural Strategies	42
6.3. Coupling và Cohesion	43
6.3.1. Coupling	43
6.3.1.1. Content Coupling	43
6.3.1.2. Common Coupling	43
6.3.1.3. Control Coupling	43
6.3.1.4. Stamp Coupling	44
6.3.1.5. Data Coupling	44
6.3.2. Cohesion	45
6.3.2.1. Coincidental Cohesion	45
6.3.2.2. Logical Cohesion	45
6.3.2.3. Temporal Cohesion	45
6.3.2.4. Procedural Cohesion	45
6.3.2.5. Communicational Cohesion	46
6.3.2.6. Sequential Cohesion	46
6.3.2.7. Functional Cohesion	46
6.4. Design Principles: SOLID	47
6.4.1. Single Responsibility Principle (SRP)	47
6.4.2. Open-Closed Principle (OCP)	48
6.4.3. Liskov Substitution Principle (LSP)	49
6.4.4. Interface Segregation Principle (ISP)	49
6.4.5. Dependency Inversion Principle (DIP)	50
6.5. Design Pattern	51
6.5.1. Factory Pattern	51
<b>7. Đánh giá công việc</b>	<b>51</b>

# 1. Giới thiệu

**AIMS (An Internet Media Store)** là một phần mềm thương mại điện tử dành cho nền tảng desktop, được thiết kế nhằm đáp ứng nhu cầu giao dịch và quản lý các sản phẩm truyền thông vật lý như sách, CD, DVD, và đĩa LP. Phần mềm mang lại trải nghiệm thân thiện với người dùng, kết hợp hiệu năng cao và độ tin cậy, đồng thời tích hợp các dịch vụ như thanh toán qua VNPay, giao hàng nhanh, và thông báo qua email, mang lại giải pháp toàn diện và hiệu quả cho người tiêu dùng.

## 1.1. Mục đích

Dự án AIMS (An Internet Media Store) được thực hiện với các mục đích chính như sau:

- Cung cấp một nền tảng thương mại điện tử hiện đại:
  - Hỗ trợ khách hàng tìm kiếm, lựa chọn, và mua các sản phẩm truyền thông vật lý một cách tiện lợi.
  - Đảm bảo trải nghiệm người dùng tối ưu với giao diện thân thiện, dễ sử dụng.
- Tăng cường hiệu quả vận hành:
  - Tích hợp thanh toán an toàn qua VNPay.
  - Hỗ trợ giao hàng nhanh và tiêu chuẩn.
- Đảm bảo tính tin cậy và minh bạch:
  - Lưu trữ đầy đủ thông tin đơn hàng, thanh toán, và lịch sử mua sắm.
  - Gửi thông báo trạng thái đơn hàng qua email.

## 1.2. Phạm vi

Phạm vi của dự án AIMS bao gồm:

- Đối tượng sử dụng:
  - Khách hàng (Customer): Người mua sản phẩm có thể duyệt, tìm kiếm, sắp xếp sản phẩm, thêm sản phẩm vào giỏ hàng, đặt hàng, và thanh toán qua VNPay. Hệ thống cũng cung cấp chức năng cho phép khách hàng xem thông tin chi tiết của đơn hàng.
- Hạ tầng công nghệ:
  - Backend: Spring Boot
  - Frontend: ReactJS
  - Cơ sở dữ liệu: MySQL
- Thanh toán và giao dịch: VNPay Sandbox API.

## 1.3. Từ điển thuật ngữ

- Customer: Khách hàng, người mua sản phẩm trên nền tảng AIMS.
- Product: Sản phẩm truyền thông vật lý (sách, CD, DVD, LP).
- Cart: Giỏ hàng, nơi khách hàng quản lý các sản phẩm muốn mua.
- Order: Đơn hàng được tạo sau khi khách hàng xác nhận thanh toán.
- VNPay: Cổng thanh toán trực tuyến được tích hợp trong AIMS để xử lý giao dịch.
- Delivery: Giao hàng, bao gồm giao hàng tiêu chuẩn và giao hàng nhanh.
- Entity: Lớp ánh xạ dữ liệu giữa cơ sở dữ liệu và ứng dụng.
- Repository: Giao tiếp giữa ứng dụng và cơ sở dữ liệu để thực hiện CRUD.
- Service: Tầng xử lý logic nghiệp vụ, bao gồm xử lý đơn hàng, giỏ hàng, và giao hàng.
- Controller: Tầng tiếp nhận yêu cầu từ người dùng và trả về kết quả.
- Response: Dữ liệu được định dạng để gửi trả cho khách hàng dưới dạng JSON/XML.
- Hibernate/JPA: Công nghệ ánh xạ đối tượng quan hệ để tương tác với cơ sở dữ liệu.
- Sandbox: Môi trường thử nghiệm cho các giao dịch VNPay.

## 1.4. Tài liệu tham khảo

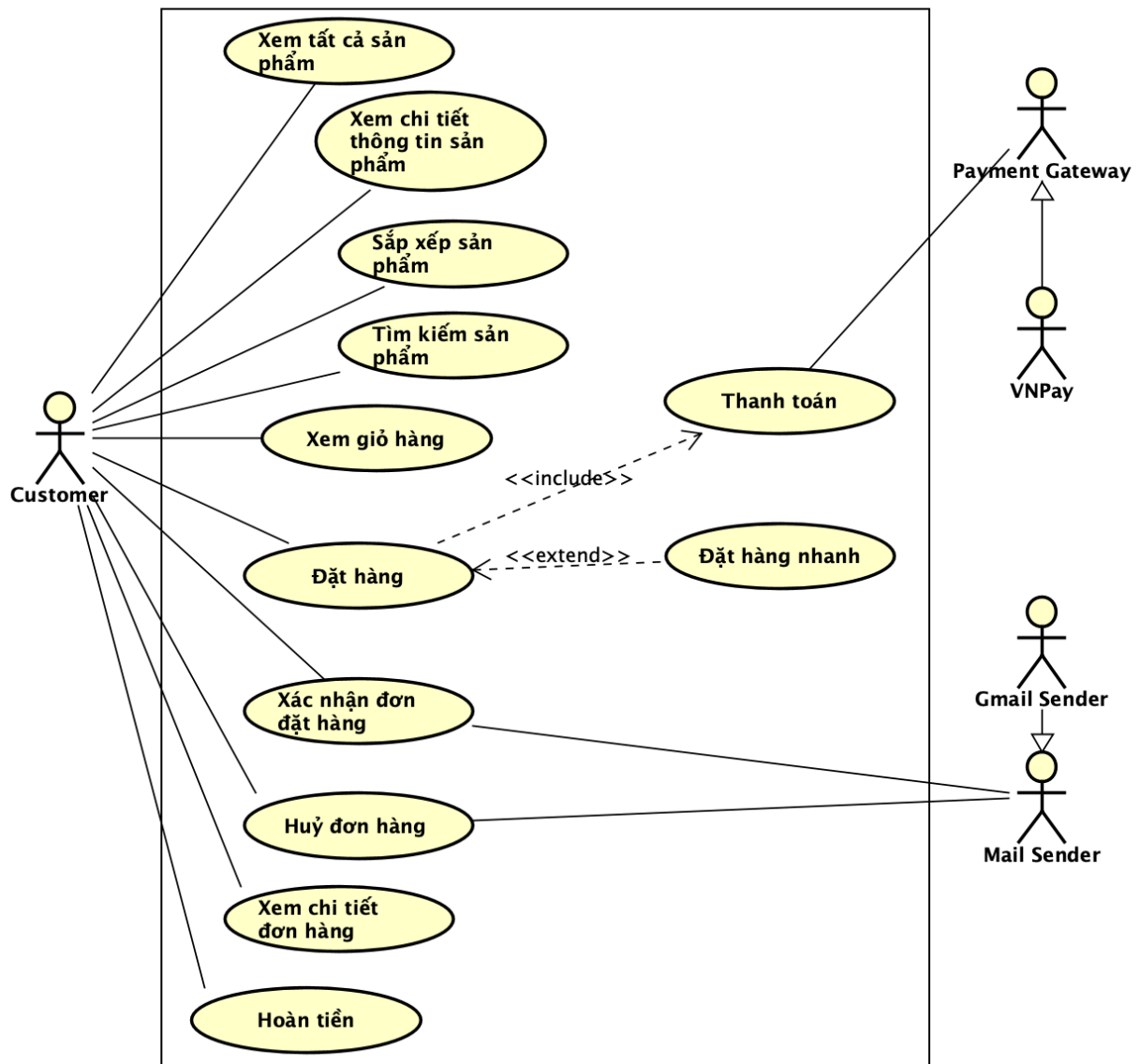
Tên tài liệu	Đường dẫn
Tài liệu chính thức của SpringBoot	<a href="#"><u>link</u></a>
Tài liệu giảng dạy môn Thiết kế xây dựng phần mềm của cô Nguyễn Thị Thu Trang	

## 2. Mô tả tổng quan

### 2.1. Các tác nhân

- Khách hàng (Customer)
  - Người sử dụng chính của hệ thống, có thể duyệt, tìm kiếm, thêm sản phẩm vào giỏ hàng, đặt hàng và thanh toán.
- Hệ thống Thanh toán (Payment Gateway)
  - Cung cấp dịch vụ thanh toán và hoàn tiền cho các giao dịch của khách hàng thông qua cổng thanh toán.
- Hệ thống thông báo (Mail Sender):
  - Tác nhân này chịu trách nhiệm gửi email thông báo đến khách hàng, ví dụ: xác nhận đơn đặt hàng, thông báo hủy đơn hàng, hoặc hoàn tiền.

## 2.2. Biểu đồ usecase

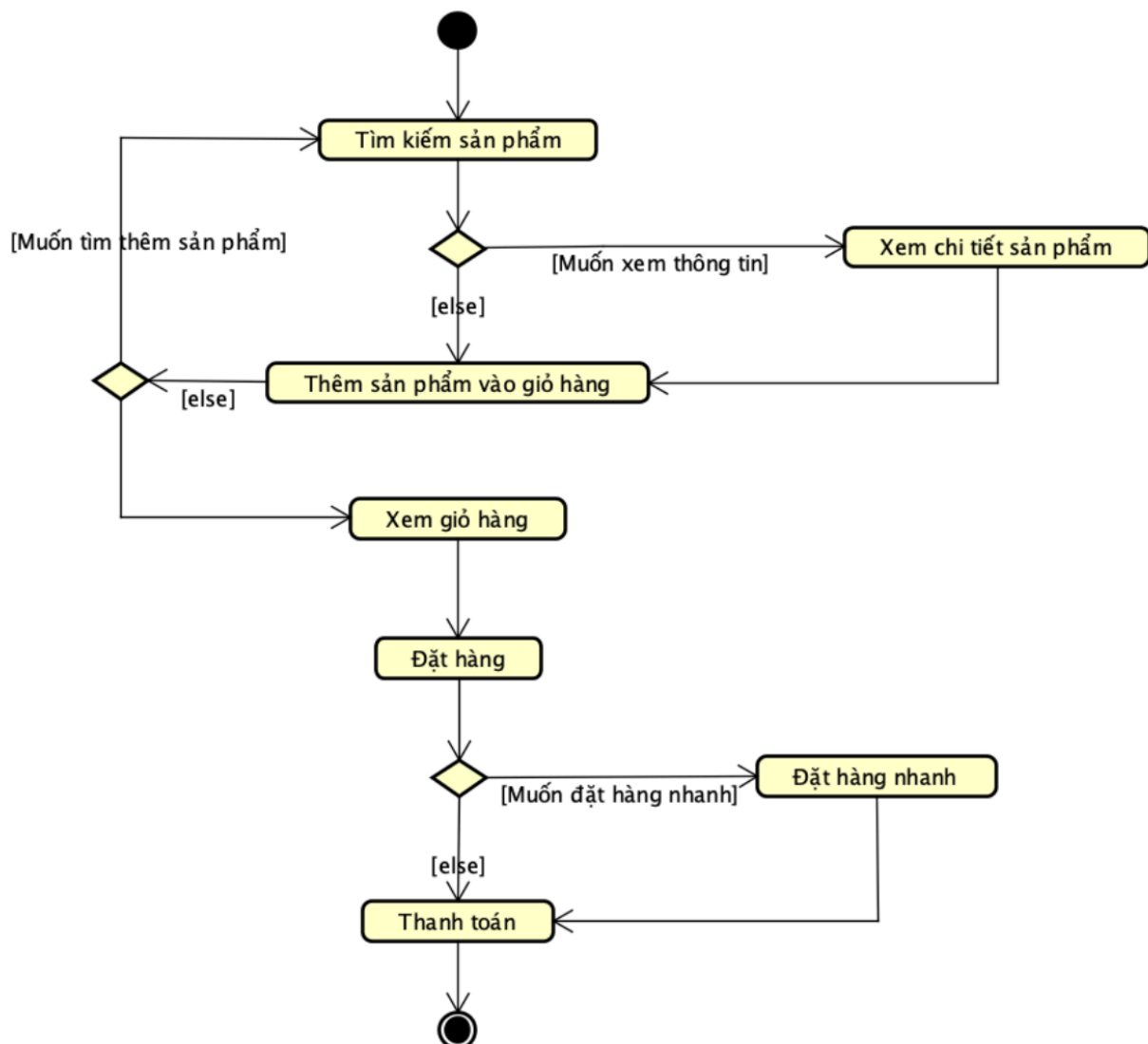




## 2.3. Quy trình nghiệp vụ

### 2.3.1. Quy trình đặt hàng và thanh toán

#### 2.3.1.1. Sơ đồ hoạt động



#### 2.3.1.2. Mô tả chi tiết

##### 1. Tìm kiếm và chọn sản phẩm:

- Khách hàng bắt đầu quy trình bằng cách tìm kiếm sản phẩm hoặc duyệt qua danh sách sản phẩm trên giao diện.
- Họ có thể xem chi tiết sản phẩm để đưa ra quyết định nên mua hay không.
- Họ chọn sản phẩm muốn mua và thêm vào giỏ hàng.

##### 2. Xem giỏ hàng và thực hiện thay đổi:

- Khách hàng kiểm tra giỏ hàng để xem tổng giá cả và thông tin sản phẩm.
- Họ có thể thay đổi số lượng sản phẩm, bỏ sản phẩm, hoặc tiếp tục mua sắm.

##### 3. Đặt hàng

- Khách hàng chọn đặt hàng (hoặc đặt hàng nhanh) và nhập thông tin giao hàng và thanh toán.
- Hệ thống kiểm tra số lượng tồn kho và thông báo nếu cần thiết.

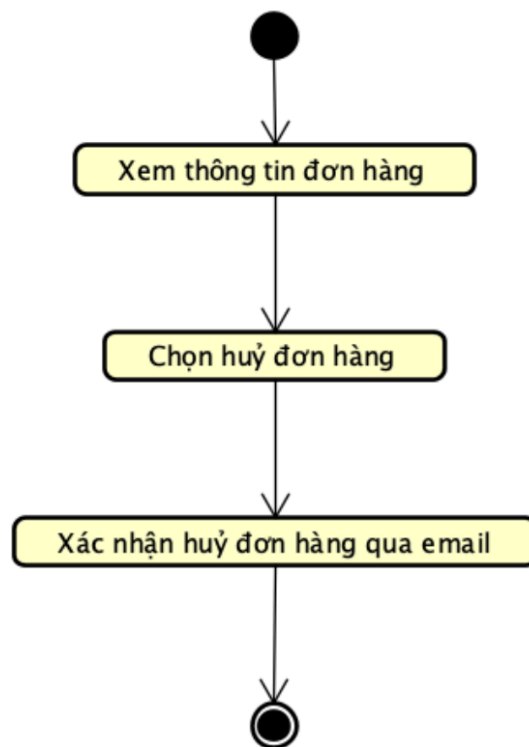
- Họ chọn phương thức thanh toán và xác nhận đặt hàng.

#### 4. Thanh toán

- Hệ thống chuyển khách hàng đến cổng thanh toán.
- Khách hàng cung cấp thông tin thanh toán và xác nhận giao dịch.
- Hệ thống xác nhận thanh toán và hiển thị thông tin đơn hàng.
- Thông báo kết quả và gửi email xác nhận đơn hàng đến khách hàng.

### 2.3.2. Quy trình huỷ đơn hàng

#### 2.3.2.1. Sơ đồ hoạt động



#### 2.3.2.2. Mô tả chi tiết

##### 1. Khách hàng yêu cầu huỷ đơn hàng

- Truy cập Email Thông Báo:
- Sau khi đặt hàng thành công, khách hàng nhận được email xác nhận đơn hàng từ hệ thống AIMS. Email này chứa các liên kết để xem thông tin đơn hàng và thực hiện huỷ đơn hàng.
- Nhấp vào liên kết huỷ đơn:
- Khách hàng chọn liên kết huỷ đơn hàng được cung cấp trong email để bắt đầu quy trình huỷ.

##### 2. Kiểm tra điều kiện huỷ đơn

- Trạng thái đơn hàng:

Hệ thống kiểm tra trạng thái của đơn hàng. Đơn hàng chỉ có thể được hủy nếu đang trong trạng thái "Pending Processing" (Chờ xử lý). Nếu đơn hàng đã được duyệt hoặc đang trong quá trình vận chuyển, việc hủy đơn sẽ không được phép.

### 3. Thực Hiện Hủy Đơn Hàng và Xử Lý Hoàn Tiền

- Cập Nhật Trạng Thái Đơn Hàng: Nếu đơn hàng thỏa mãn điều kiện hủy, hệ thống sẽ cập nhật trạng thái đơn hàng thành "Cancelled" (Đã Hủy).
- Xử Lý Hoàn Tiền: Hệ thống khởi tạo quy trình hoàn tiền thông qua Payment Gateway và VNPay.
- Gửi Yêu Cầu Hoàn Tiền: Hệ thống gửi yêu cầu hoàn tiền đến Payment Gateway, sau đó Payment Gateway giao tiếp với VNPay để thực hiện giao dịch hoàn tiền.
- Xác Nhận Hoàn Tiền: Sau khi VNPay xử lý, hệ thống nhận được xác nhận hoàn tiền và cập nhật trạng thái hoàn tiền cho đơn hàng.

### 4. Thông Báo Kết Quả Hủy Đơn Hàng

- Gửi Email Thông Báo: Hệ thống sử dụng Mail Sender và Gmail Sender để gửi email thông báo đến khách hàng về việc hủy đơn hàng thành công cùng với thông tin về hoàn tiền.
- Cập Nhật Thông Tin Trong Hệ Thống: Khách hàng có thể đăng nhập vào tài khoản (nếu đã đăng ký) để xem trạng thái đơn hàng đã được cập nhật thành "Đã Hủy" và thông tin hoàn tiền.

## 3. Phân tích yêu cầu

### 3.1. Đặc tả usecase UC001 “Đặt hàng”

#### 3.1.1. Mã usecase

UC001

#### 3.1.2. Giới thiệu

Usecase này trình bày cách khách hàng tương tác với hệ thống AIMS để tạo một đơn hàng, từ việc xác nhận sản phẩm trong giỏ hàng đến khi hoàn tất quá trình thanh toán.

#### 3.1.3. Tác nhân

Khách hàng

#### 3.1.4. Tiền điều kiện

- Hệ thống AIMS hoạt động ổn định.
- Thiết bị của khách hàng kết nối mạng Internet.
- Giỏ hàng của khách hàng đã chứa ít nhất một sản phẩm.

#### 3.1.5. Luồng sự kiện chính

1. Khách hàng mở giao diện giỏ hàng và yêu cầu đặt hàng.
2. Hệ thống kiểm tra tồn kho của các sản phẩm trong giỏ hàng.
3. Hệ thống hiển thị thông tin đơn hàng kèm form điền thông tin giao hàng.
4. Khách hàng cung cấp thông tin giao hàng (tên, địa chỉ, phương thức giao hàng).

5. Hệ thống tính phí giao hàng và tổng tiền cần thanh toán.
6. Hệ thống hiển thị tổng số tiền thanh toán.
7. Khách hàng xác nhận đơn hàng và chọn phương thức thanh toán.
8. Hệ thống chuyển sang Use Case “Thanh toán” để xử lý giao dịch.
9. Hệ thống xóa giỏ hàng, lưu thông tin đơn hàng và gửi email xác nhận đến khách hàng.
10. Hệ thống hiển thị thông báo đặt hàng thành công cùng thông tin chi tiết.

### 3.1.6. Luồng sự kiện thay thế


STT	Vị trí	Điều kiện	Hành động	Vị trí tiếp tục
1	Tại bước 3	Nếu lượng hàng trong kho không đủ để đáp ứng đơn đặt hàng	Hệ thống hiển thị số lượng hàng tồn kho với mỗi sản phẩm không đáp ứng và yêu cầu khách hàng cập nhật lại giỏ hàng	Kết thúc ca sử dụng
2	Tại bước 5	Nếu khách hàng bỏ trống trường bắt buộc hoặc nhập thông tin giao hàng không hợp lệ	Hệ thống thông báo lỗi: Không được bỏ trống các trường bắt buộc hoặc Thông tin giao hàng không hợp lệ	Tiếp tục tại bước 3
3	Tại bước 5	Nếu khách hàng chọn phương thức giao hàng nhanh	Hệ thống chèn UC Đặt hàng nhanh	Tiếp tục tại bước 5
4	Tại bước 9	Nếu thanh toán đơn hàng không thành công	Hệ thống thông báo rằng đặt hàng thất bại do thanh toán không thành công	Kết thúc ca sử dụng

Bảng 3.1: Luồng sự kiện thay thế của use case Đặt hàng

### 3.1.7. Dữ liệu đầu vào

STT	Trường dữ liệu	Mô tả	Bắt buộc	Điều kiện hợp lệ	Ví dụ
1	Tên người nhận	Tên người nhận hàng	Có		Tạ Tùng Dương
2	Email	Email của người đặt hàng	Có	Đúng định dạng email	hehehekth@gmail.com
2	Số điện thoại	Số điện thoại liên hệ	Có	10 chữ số	0819235369
3	Tỉnh/thành phố	Tên tỉnh/thành phố nhận hàng	Có	Là một tỉnh/thành phố được chọn từ danh sách được cung cấp	Hà Nội
4	Địa chỉ giao hàng	Địa chỉ cụ thể của người nhận hàng	Có		Số 12, hẻm 5/30/40 phố Tạ Quang Bửu, quận Hai Bà Trưng, Hà Nội
5	Phương thức giao hàng	Phương thức giao hàng mong muốn	Có		Giao hàng nhanh
6	Phương thức thanh toán	Phương thức thanh toán mong muốn	Có		VNPay

### 3.1.8. Dữ liệu đầu ra

STT	Trường dữ liệu	Mô tả	Định dạng hiển thị	Ví dụ
1	Tên sản phẩm	Tên sản phẩm khách hàng đặt		DVD phim Titanic
2	Số lượng	Số lượng của sản phẩm được đặt	Căn lề phải	2

3	Giá cả sản phẩm	Giá của mỗi sản phẩm	Căn lề phải Định dạng tiền tệ Việt Nam Dùng dấu phẩy để ngăn cách hàng nghìn	100,000 VNĐ
4	Tổng giá cả sản phẩm (chưa gồm VAT)	Tổng giá trước thuế của tất cả các sản phẩm được đặt hàng	Căn lề phải Định dạng tiền tệ Việt Nam Dùng dấu phẩy để ngăn cách hàng nghìn	500,000 VNĐ
5	VAT	Thuế giá trị gia tăng của các sản phẩm được đặt hàng	Căn lề phải Định dạng tiền tệ Việt Nam Dùng dấu phẩy để ngăn cách hàng nghìn	550,000 VNĐ
6	Phí giao hàng	Phí giao hàng	Căn lề phải Định dạng tiền tệ Việt Nam Dùng dấu phẩy để ngăn cách hàng nghìn	50,000 VNĐ
7	Tổng số tiền phải trả	Tổng số tiền cần thanh toán, bao gồm tổng giá sản phẩm, VAT và phí giao hàng	Căn lề phải Định dạng tiền tệ Việt Nam Dùng dấu phẩy để ngăn cách hàng nghìn	600,000 VNĐ

Bảng 3.3: Dữ liệu đầu ra về thông tin đơn hàng của use case Đặt hàng

STT	Trường dữ liệu	Mô tả	Định dạng hiển thị	Ví dụ
-----	----------------	-------	--------------------	-------

1	Mã giao dịch	Mã giao dịch thanh toán bằng thẻ (nếu thanh toán bằng thẻ trả trước)		193034129
2	Tên chủ thẻ	Tên chủ thẻ thực hiện thanh toán (nếu thanh toán bằng thẻ trả trước)		Phan Minh Anh Tuấn
3	Số tiền bị trừ	Số tiền bị trừ từ thẻ thanh toán (nếu thanh toán bằng thẻ trả trước)	Phải là số dương	100 000
4	Nội dung giao dịch	Mô tả chi tiết về nội dung của giao dịch (nếu thanh toán bằng thẻ trả trước)		
5	Ngày giờ giao dịch	Ngày và giờ thực hiện giao dịch (nếu thanh toán bằng thẻ trả trước)	Định dạng ngày và giờ hợp lệ	2024-09-23 16:33

Bảng 3.4: Dữ liệu đầu ra về thông tin thanh toán của use case Đặt hàng

3.1.9. Hậu điều kiện

Không có

3.2. Đặc tả usecase UC002 “Thanh toán”

3.2.1. Mã usecase

UC002

3.2.2. Giới thiệu

Use case này mô tả quy trình khách hàng thực hiện thanh toán thông qua hệ thống AIMS, bao gồm việc tương tác với các cổng thanh toán trực tuyến để hoàn tất giao dịch.

3.2.3. Tác nhân

- Khách hàng
- Hệ thống cổng thanh toán
- Hệ thống gửi thông báo

3.2.4. Tiền điều kiện

Hệ thống đã xác định tổng số tiền cần thanh toán của đơn hàng.

### 3.2.5. Luồng sự kiện chính

1. Tại giao diện đặt hàng, hệ thống hiển thị thông tin đơn hàng và tổng số tiền cần thanh toán.
2. Khách hàng chọn một trong các cổng thanh toán (VNPay hoặc Paypal).
3. Hệ thống chuyển hướng khách hàng đến giao diện của cổng thanh toán đã chọn.
4. Khách hàng điền thông tin cần thiết và xác nhận thanh toán trên cổng thanh toán.
5. Cổng thanh toán xử lý giao dịch và trả về thông tin giao dịch cho hệ thống AIMS.
6. Hệ thống lưu trữ chi tiết giao dịch.
7. Hệ thống gửi email thông báo thanh toán thành công đến khách hàng.
8. Hệ thống hiển thị thông tin chi tiết giao dịch và thông báo hoàn tất thanh toán.

### 3.2.6. Luồng sự kiện thay thế

STT	Vị trí	Điều kiện	Hành động	Vị trí tiếp tục
1	Tại bước 4	Nếu khách hàng huỷ thanh toán	Cổng thanh toán trả về thông tin giao dịch thất bại cho hệ thống AIMS	Kết thúc ca sử dụng

Bảng 3.5: Luồng sự kiện thay thế của use case Thanh toán

### 3.2.7. Dữ liệu đầu vào

Không có.

### 3.2.8. Dữ liệu đầu ra

STT	Trường dữ liệu	Mô tả	Định dạng hiển thị	Ví dụ
1	Mã đơn hàng	Mã đơn hàng của người dùng		95e2c048-ca7f-466b-8d0a-d3052a844690
2	Mã giao dịch	Mã số giao dịch		8RW57395J3839131A77F83800UE6464200
3	Phương thức thanh toán	Cổng thanh toán người dùng sử dụng	VNPay/Paypal	Paypal
4	Tổng tiền thanh toán	Số tiền mà người dùng đã thanh toán		30000 đồng



5	Thời gian thanh toán	Thời gian người dùng thanh toán	Định dạng ngày và giờ hợp lệ	1 tháng 1, 2025
---	----------------------	---------------------------------	------------------------------	-----------------

Bảng 3.6: Dữ liệu đầu ra về thông tin thanh toán của use case Thanh toán

### 3.2.9. Hậu điều kiện

Thông tin giao dịch đã được lưu

## 3.3. Đặc tả usecase UC003 “Đặt hàng nhanh”

### 3.3.1. Mã usecase

UC003

### 3.3.2. Giới thiệu

Use case này mô tả quá trình khách hàng sử dụng tính năng giao hàng nhanh khi đặt đơn hàng trên hệ thống AIMS, bao gồm việc xác minh điều kiện giao hàng và tính toán chi phí tương ứng.

### 3.3.3. Tác nhân

Khách hàng

### 3.3.4. Tiền điều kiện

Khách hàng đã nhập đầy đủ thông tin giao hàng trong đó có địa chỉ nhận hàng khi ở usecase đặt hàng.

### 3.3.5. Luồng sự kiện chính

1. Khách hàng chọn phương thức giao hàng là "Giao hàng nhanh".
2. Hệ thống xác minh địa chỉ nhận hàng có nằm trong khu vực hỗ trợ giao hàng nhanh không.
3. Hệ thống kiểm tra sản phẩm trong giỏ hàng có đáp ứng điều kiện giao hàng nhanh không.
4. Hệ thống tính phí giao hàng nhanh và hiển thị tổng chi phí cho đơn hàng.
5. Hệ thống phân nhóm sản phẩm theo phương thức giao hàng (giao hàng nhanh hoặc thông thường) và hiển thị trên giao diện.
6. Khách hàng bổ sung hoặc chỉnh sửa thông tin giao hàng nếu cần thiết.
7. Hệ thống xác minh lại thông tin đã nhập.
8. Hệ thống cập nhật chi phí và phương thức giao hàng tương ứng cho từng sản phẩm và hiển thị lại tổng tiền.

### 3.3.6. Luồng sự kiện thay thế

STT	Vị trí	Điều kiện	Hành động	Vị trí tiếp tục
-----	--------	-----------	-----------	-----------------

1	Tại bước 3	Nếu địa chỉ giao hàng không hỗ trợ giao hàng nhanh	Hệ thống thông báo địa chỉ giao hàng không hỗ trợ giao hàng nhanh và yêu cầu khách hàng cập nhật lại thông tin giao hàng hoặc phương thức giao hàng	Kết thúc ca sử dụng
2	Tại bước 4	Nếu không có sản phẩm nào trong giỏ hàng hỗ trợ giao hàng nhanh	Hệ thống thông báo không có sản phẩm nào hỗ trợ giao hàng nhanh và yêu cầu khách hàng cập nhật lại thông tin giao hàng hoặc phương thức giao hàng	Kết thúc ca sử dụng
3	Tại bước 8	Nếu khách hàng nhập thiếu trường bắt buộc	Hệ thống thông báo lỗi: Cần nhập các trường bắt buộc nhập	Tiếp tục tại bước 6
4	Tại bước 8	Nếu khách hàng nhập thông tin không hợp lệ	Hệ thống thông báo lỗi: Thông tin không hợp lệ	Tiếp tục tại bước 6

Bảng 3.7: Luồng sự kiện thay thế của use case Đặt hàng nhanh

### 3.3.7. Dữ liệu đầu vào

STT	Trường dữ liệu	Mô tả	Bắt buộc	Điều kiện hợp lệ	Ví dụ
1	Thời gian nhận hàng	Thời gian nhận hàng khi đặt giao hàng nhanh	Bắt buộc khi đặt giao hàng nhanh	Định dạng thời gian	2025-01-3 06:30 PM
2	Chỉ dẫn giao hàng	Hướng dẫn bổ sung cho việc giao hàng	Bắt buộc khi đặt giao hàng nhanh		Gần cổng Trần Đại Nghĩa của Đại học Bách Khoa Hà Nội

Bảng 3.8: Dữ liệu đầu vào use case Đặt hàng nhanh

### 3.3.8. Dữ liệu đầu ra

STT	Trường dữ liệu	Mô tả	Định dạng hiển thị	Ví dụ
1	Tên sản phẩm	Tên sản phẩm khách hàng đặt		DVD phim Titanic
2	Phương thức giao hàng	Phương thức giao hàng cho sản phẩm tương ứng (giao hàng nhanh hoặc giao hàng thường)		Giao hàng nhanh

Bảng 3.9: Dữ liệu đầu ra về thông tin đơn hàng của use case Đặt hàng nhanh

### 3.3.9. Hậu điều kiện

Không có

## 3.4. Đặc tả usecase UC004 “Huỷ đơn hàng”

### 3.4.1. Mã usecase

UC004

### 3.4.2. Giới thiệu

Use case này mô tả quy trình khách hàng hủy một đơn hàng đã đặt trước đó nhưng chưa được xác nhận hoặc xử lý bởi hệ thống.

### 3.4.3. Tác nhân

Khách hàng

### 3.4.4. Tiền điều kiện

Đơn hàng đã được tạo nhưng chưa được xác nhận hoặc xử lý bởi hệ thống.

### 3.4.5. Luồng sự kiện chính

1. Khách hàng truy cập mục “Theo dõi đơn hàng” trên giao diện hệ thống.
2. Hệ thống hiển thị danh sách các đơn hàng của khách hàng.
3. Khách hàng chọn một đơn hàng cần hủy và nhấn vào nút “Xem chi tiết đơn hàng”.
4. Hệ thống hiển thị thông tin chi tiết của đơn hàng, bao gồm trạng thái và tùy chọn “Huỷ đơn hàng”.
5. Khách hàng nhấn vào nút “Huỷ đơn hàng”.
6. Hệ thống gửi email yêu cầu khách hàng xác nhận việc hủy đơn hàng.
7. Khách hàng xác nhận qua email.
8. Hệ thống hủy đơn hàng, cập nhật trạng thái đơn hàng trong hệ thống và gửi thông báo thành công qua email.

9. Hệ thống thực hiện hoàn tiền (nếu có) thông qua phương thức thanh toán ban đầu.

#### 3.4.6. Luồng sự kiện thay thế

STT	Vị trí	Điều kiện	Hành động	Vị trí tiếp tục
1	Tại bước 4	Nếu đơn hàng đã được xác nhận	Hệ thống không cung cấp lựa chọn huỷ đơn hàng	Kết thúc ca sử dụng

Bảng 3.10: Luồng sự kiện thay thế của use case Huỷ đơn hàng

#### 3.4.7. Dữ liệu đầu vào

STT	Trường dữ liệu	Mô tả	Bắt buộc	Điều kiện hợp lệ	Ví dụ
1	Mã đơn hàng	Mã của đơn hàng mà khách hàng muốn huỷ	Có		aef21477-dce2-42af-abc0-785e4ceaa65c

Bảng 3.11: Dữ liệu đầu vào của use case Huỷ đơn hàng

#### 3.4.8. Dữ liệu đầu ra

Không có

#### 3.4.9. Hậu điều kiện

- Đơn hàng bị huỷ không còn xuất hiện trong danh sách đơn hàng của khách hàng.
- Cập nhật số lượng sản phẩm trong kho.

### 3.5. Đặc tả usecase UC005 “Xem sản phẩm”

#### 3.5.1. Mã usecase

UC005

#### 3.5.2. Giới thiệu

Use case này mô tả sự tương tác giữa **Khách hàng** và hệ thống **AIMS** khi Khách hàng muốn xem thông tin chi tiết về một sản phẩm cụ thể trong cửa hàng trực tuyến. Quá trình này bao gồm việc lựa chọn sản phẩm từ danh sách, xem thông tin chi tiết và đánh giá các thông số liên quan đến sản phẩm đó.

### 3.5.3. Tác nhân

Khách hàng (Customer)

### 3.5.4. Tiền điều kiện

Khách hàng đã truy cập vào phần mềm AIMS.

Hệ thống đã có danh sách sản phẩm được hiển thị cho khách hàng.

Sản phẩm cần xem thông tin đã được thêm vào hệ thống.

### 3.5.5. Luồng sự kiện chính

#### Chọn Sản Phẩm:

- Khách hàng duyệt qua danh sách sản phẩm và chọn một sản phẩm cụ thể để xem chi tiết.

#### Yêu Cầu Xem Thông Tin Sản Phẩm:

- Khách hàng nhấp vào sản phẩm đã chọn để yêu cầu xem thông tin chi tiết.

#### Hệ Thống Hiển Thị Thông Tin Sản Phẩm:

- AIMS** truy xuất dữ liệu sản phẩm từ cơ sở dữ liệu và hiển thị các thông tin chi tiết như tiêu đề, danh mục, giá cả, mô tả, thông số kỹ thuật tùy thuộc vào loại sản phẩm (sách, CD, DVD, v.v.).

#### Khách Hàng Đánh Giá và Tương Tác:

- Khách hàng có thể xem các thông tin bổ sung như đánh giá sản phẩm, hình ảnh, và các sản phẩm liên quan.

Luồng sự kiện thay thế

Bảng Luồng Sự Kiện Thay Thế cho UC005 “Xem sản phẩm”

No	Location	Condition	Action	Resume Location
1	Step 1	Nếu sản phẩm không còn trong kho	Hiển thị thông báo “Sản phẩm hiện không có sẵn”	Use case ends
2	Step 3	Nếu khách hàng không muốn thêm vào giỏ	Quay lại danh sách sản phẩm	Use case ends
3	Step 2	Nếu thông tin sản phẩm bị lỗi hoặc không tải được	Hiển thị thông báo lỗi và yêu cầu thử lại sau	Use case ends

## Dữ liệu đầu vào

No	Data fields	Description	Mandatory	Valid condition	Example
1.	Product ID	Mã định danh duy nhất của sản phẩm	Yes	Phải tồn tại trong cơ sở dữ liệu	PROD12345
2.	Customer Action	Hành động của khách hàng (chọn xem chi tiết)	Yes	Phải là hành động hợp lệ	Click vào sản phẩm "Book A"

## Dữ liệu đầu ra

No	Data fields	Description	Display format	Example
1.	Product Details	Thông tin chi tiết của sản phẩm	Trang chi tiết sản phẩm	Tiêu đề, hình ảnh, mô tả, giá, thông tin bổ sung
2.	Availability Status	Tình trạng còn hàng của sản phẩm	Text Format	"Còn 10 sản phẩm trong kho"
3.	Action Options	Các tùy chọn hành động sau khi xem sản phẩm	Buttons/Links	"Thêm vào giỏ hàng", "Quay lại"

## Hậu điều kiện

Khách hàng đã xem thành công thông tin chi tiết của sản phẩm.

Hệ thống cung cấp các tùy chọn tiếp theo cho khách hàng như thêm vào giỏ hàng hoặc quay lại danh sách sản phẩm.

## 3.6. Đặc tả usecase UC006 "Tìm kiếm sản phẩm"

### 3.6.1. Mã usecase

UC006

### 3.6.2. Giới thiệu

Use case này mô tả quá trình tương tác giữa **Khách hàng** và **AIMS** khi Khách hàng muốn tìm kiếm sản phẩm dựa trên các thuộc tính như tên sản phẩm, danh mục, tác giả, nghệ sĩ, v.v. Quá trình này bao gồm việc nhập từ khóa tìm kiếm, hệ thống xử lý và hiển thị kết quả tìm kiếm phù hợp.

### 3.6.3. Tác nhân

Khách hàng (Customer)

### 3.6.4. Tiền điều kiện

Khách hàng đã truy cập vào phần mềm AIMS.  
Có ít nhất một sản phẩm có sẵn trong cửa hàng.

### 3.6.5. Luồng sự kiện chính

#### Nhập Từ Khóa Tìm Kiếm:

- Khách hàng nhập từ khóa hoặc các thuộc tính cần tìm kiếm vào thanh tìm kiếm.

#### Gửi Yêu Cầu Tìm Kiếm:

- Khách hàng nhấn nút “Tìm kiếm” để gửi yêu cầu.

#### Xử Lý Tìm Kiếm:

- AIMS** xử lý yêu cầu tìm kiếm dựa trên từ khóa và các thuộc tính đã nhập.

#### Hiển Thị Kết Quả Tìm Kiếm:

- AIMS** hiển thị danh sách các sản phẩm phù hợp với tiêu chí tìm kiếm, mỗi trang hiển thị 20 sản phẩm ngẫu nhiên hoặc theo thứ tự phù hợp.

#### Tùy Chọn Hành Động:

- Khách hàng có thể chọn xem chi tiết sản phẩm, thêm vào giỏ hàng, hoặc tiếp tục tìm kiếm thêm sản phẩm khác.
- 

### 3.6.6. Luồng sự kiện thay thế

Bảng Luồng Sự Kiện Thay Thế cho UC006 “Tìm kiếm sản phẩm”

No	Location	Condition	Action	Resume Location
1	Step 1	Nếu không nhập từ khóa tìm kiếm	Hiển thị thông báo yêu cầu nhập từ khóa tìm kiếm	Use case ends
2	Step 4	Nếu không có sản phẩm nào phù hợp	Hiển thị thông báo “Không tìm thấy sản phẩm phù hợp”	Use case ends
3	Step 3	Nếu hệ thống gặp lỗi trong quá trình xử lý	Hiển thị thông báo lỗi và yêu cầu thử lại sau	Use case ends

### 3.6.7. Dữ liệu đầu vào

#### Input data of UC006 “Tìm kiếm sản phẩm”

No	Data fields	Description	Mandator y	Valid condition	Example
1.	Search Keyword	Từ khóa hoặc thuộc tính để tìm kiếm sản phẩm	Yes	Phải là chuỗi ký tự hợp lệ	"Book A", "Music", "Hanoi"
2.	Filter Options	Các tùy chọn lọc bổ sung (nếu có)	No	Nếu có, phải tương ứng với thuộc tính sản phẩm	Danh mục: Sách, Giá: Dưới 200,000 VND

### 3.6.8. Dữ liệu đầu ra

#### Output data of UC006 “Tìm kiếm sản phẩm”

No	Data fields	Description	Display format	Example
1.	Search Results	Danh sách các sản phẩm phù hợp với tìm kiếm	Table/List Format	Danh sách 20 sản phẩm phù hợp
2.	No Results Message	Thông báo khi không tìm thấy sản phẩm phù hợp	Text Format	"Không tìm thấy sản phẩm phù hợp với yêu cầu của bạn."
3.	Search Filters Applied	Các bộ lọc đã được áp dụng trong tìm kiếm	Text/Icons	"Danh mục: Sách, Giá: Dưới 200,000 VND"

### 3.6.9. Hậu điều kiện

- Khách hàng đã nhận được danh sách các sản phẩm phù hợp với yêu cầu tìm kiếm.
- Nếu có kết quả, khách hàng có thể tiếp tục tương tác với các sản phẩm được tìm thấy.
- Nếu không có kết quả, khách hàng có thể điều chỉnh lại tiêu chí tìm kiếm.

## 3.7. Đặc tả usecase UC007 “Sắp xếp sản phẩm”

### Mã usecase

UC007



## Giới thiệu

Use case này mô tả quá trình tương tác giữa **Khách hàng** và **AIMS** khi Khách hàng muốn sắp xếp danh sách sản phẩm dựa trên các tiêu chí như giá cả, đánh giá, tên sản phẩm, v.v. Quá trình này giúp khách hàng dễ dàng tìm thấy sản phẩm mong muốn theo thứ tự ưu tiên cá nhân.

## Tác nhân

- **Khách hàng (Customer)**

## Tiền điều kiện

- Khách hàng đã thực hiện tìm kiếm sản phẩm hoặc đang xem danh sách sản phẩm.
- Có ít nhất một sản phẩm trong danh sách để sắp xếp.

## Luồng sự kiện chính

1. **Chọn Tiêu Chí Sắp Xếp:**
  - Khách hàng chọn tiêu chí sắp xếp từ menu hoặc các tùy chọn sẵn có (ví dụ: Giá từ thấp đến cao, Giá từ cao đến thấp, Tên sản phẩm A-Z, Tên sản phẩm Z-A).
2. **Gửi Yêu Cầu Sắp Xếp:**
  - Khách hàng nhấn nút “Sắp xếp” để áp dụng tiêu chí đã chọn.
3. **Xử Lý Sắp Xếp:**
  - **AIMS** xử lý danh sách sản phẩm dựa trên tiêu chí sắp xếp đã chọn.
4. **Hiển Thị Danh Sách Sắp Xếp:**
  - **AIMS** cập nhật và hiển thị danh sách sản phẩm theo thứ tự mới đã sắp xếp.
5. **Tùy Chọn Hành Động:**
  - Khách hàng có thể tiếp tục tương tác với danh sách sản phẩm đã sắp xếp, như xem chi tiết sản phẩm, thêm vào giỏ hàng, hoặc thực hiện sắp xếp lại.

## Luồng sự kiện thay thế

Bảng N - Luồng Sự Kiện Thay Thế cho UC007 “Sắp xếp sản phẩm”

No	Location	Condition	Action	Resume Location
1	Step 1	Nếu không chọn tiêu chí sắp xếp	Hệ thống giữ nguyên thứ tự hiện tại của danh sách	Use case ends
2	Step 3	Nếu hệ thống gặp lỗi trong quá trình sắp xếp	Hiển thị thông báo lỗi và yêu cầu thử lại sau	Use case ends
3	Step 4	Nếu không có sản phẩm để sắp xếp	Hiển thị thông báo “Không có sản phẩm để sắp xếp”	Use case ends

## Dữ liệu đầu vào

Bảng A - Dữ Liệu Đầu Vào của UC007 “Sắp xếp sản phẩm”

No	Data Fields	Description	Mandatory	Valid Condition	Example
1	Sort Criteria	Tiêu chí sắp xếp được chọn bởi khách hàng	Yes	Phải là một trong các tiêu chí sắp xếp hợp lệ	"Giá từ thấp đến cao", "Tên A-Z"
2	Current Product List	Danh sách sản phẩm hiện tại cần sắp xếp	Yes	Phải có ít nhất một sản phẩm trong danh sách	Danh sách 20 sản phẩm tìm kiếm

## Dữ liệu đầu ra

Bảng B - Dữ Liệu Đầu Ra của UC003 “Sắp xếp sản phẩm”

No	Data Fields	Description	Display Format	Example
1	Sorted Product List	Danh sách sản phẩm đã được sắp xếp theo tiêu chí	Table/List Format	Danh sách sản phẩm theo thứ tự giá từ thấp đến cao
2	Sort Confirmation	Thông báo xác nhận việc sắp xếp thành công	Text Format	"Danh sách sản phẩm đã được sắp xếp theo giá từ thấp đến cao."
3	Updated Sort Options	Các tùy chọn sắp xếp hiện tại	Dropdown/Menu	"Giá từ cao đến thấp", "Tên Z-A"

## Hậu điều kiện

- Danh sách sản phẩm được sắp xếp theo tiêu chí mà khách hàng đã chọn.
- Khách hàng nhận được thông báo xác nhận việc sắp xếp thành công.
- Khách hàng có thể tiếp tục tương tác với danh sách sản phẩm đã sắp xếp theo ý muốn.

## 3.8. Đặc tả usecase UC008 “Xem lịch sử mua hàng”

### 3.8.1. Mã usecase

UC008

### 3.8.2. Giới thiệu

Use case mô tả cách khách hàng truy cập và xem danh sách các đơn hàng đã thực hiện trong quá khứ thông qua AIMS.

### 3.8.3. Tác nhân

Khách hàng

### 3.8.4. Tiền điều kiện

Khách hàng đã đăng nhập thành công vào hệ thống.

### 3.8.5. Luồng sự kiện chính

1. Khách hàng truy cập mục “Lịch sử mua hàng” từ menu điều hướng chính.
2. Hệ thống hiển thị danh sách các đơn hàng của khách hàng, bao gồm:
  - Mã đơn hàng
  - Ngày đặt hàng
  - Trạng thái đơn hàng (đang xử lý, đã giao, đã hủy)
  - Tổng số tiền của từng đơn hàng
3. Khách hàng chọn một đơn hàng để xem chi tiết.
4. Hệ thống hiển thị thông tin chi tiết của đơn hàng, bao gồm:
  - Danh sách sản phẩm đã mua
  - Số lượng và giá của từng sản phẩm
  - Thông tin giao hàng (địa chỉ, phương thức, trạng thái giao hàng)
  - Tổng tiền đã thanh toán (bao gồm VAT và phí giao hàng)

### 3.8.6. Luồng sự kiện thay thế

### 3.8.7. Dữ liệu đầu vào

Không có

### 3.8.8. Dữ liệu đầu ra

### 3.8.9. Hậu điều kiện

## 4. Kiến trúc hệ thống và thiết kế kiến trúc

### 4.1. Kiến trúc clean architecture

Clean Architecture chia hệ thống thành các tầng chính, giúp ứng dụng dễ bảo trì, mở rộng, và tách biệt các thành phần.

-Các tầng trong Clean Architecture:

**+Entities (Tầng thực thể cốt lõi):**

- Chứa logic nghiệp vụ cốt lõi và các đối tượng chính của hệ thống.
- Độc lập với framework hoặc công nghệ.

**+Use Cases (Tầng nghiệp vụ cụ thể):**

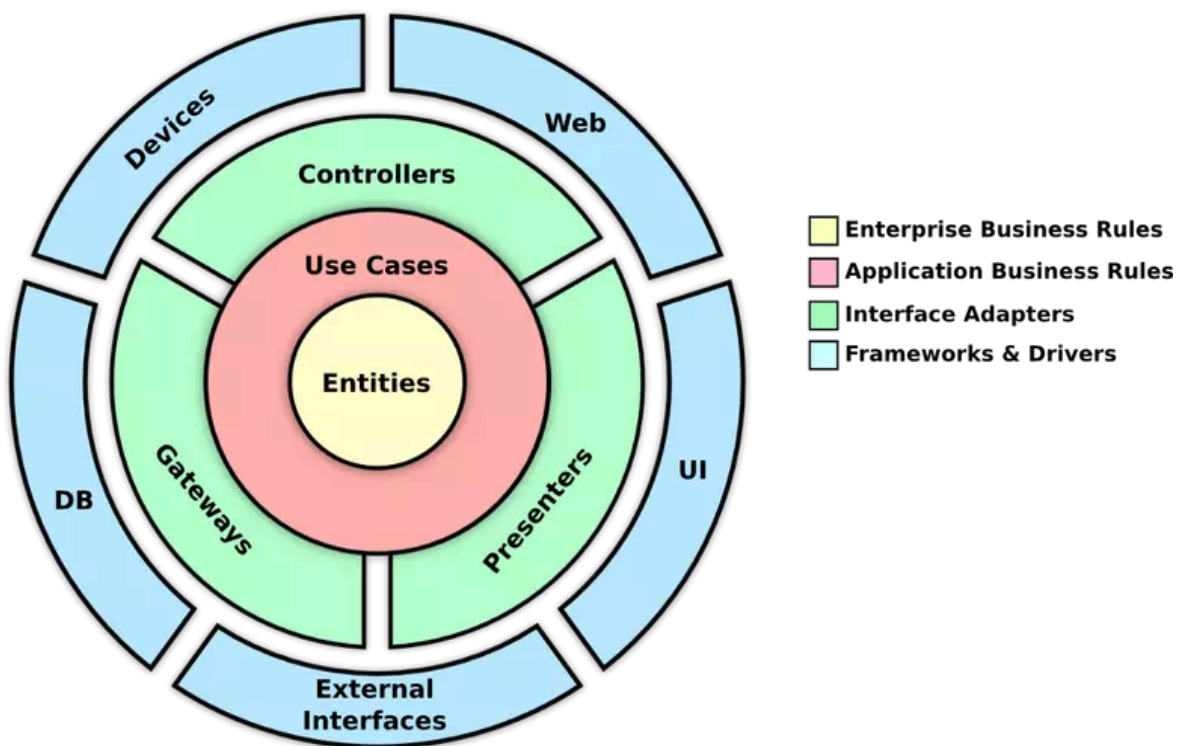
- Định nghĩa logic nghiệp vụ và luồng xử lý cụ thể của ứng dụng.
- Sử dụng các thực thể từ tầng Entities và tương tác với tầng bên dưới.

**+Interface Adapters (Tầng giao tiếp):**

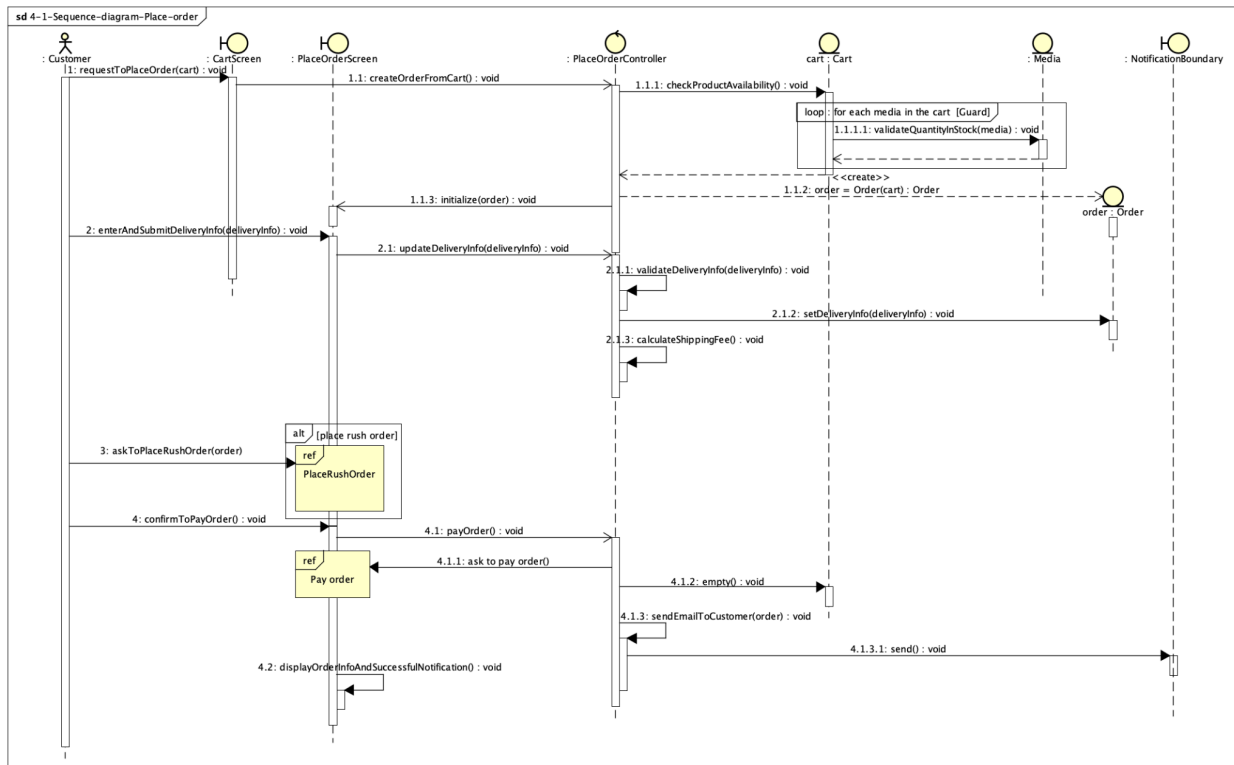
- Chuyển đổi dữ liệu giữa logic nghiệp vụ và giao diện người dùng hoặc các dịch vụ bên ngoài.
- Bao gồm API Controller và các công cụ chuyển đổi định dạng dữ liệu.

**+Frameworks and Drivers (Tầng phụ thuộc bên ngoài):**

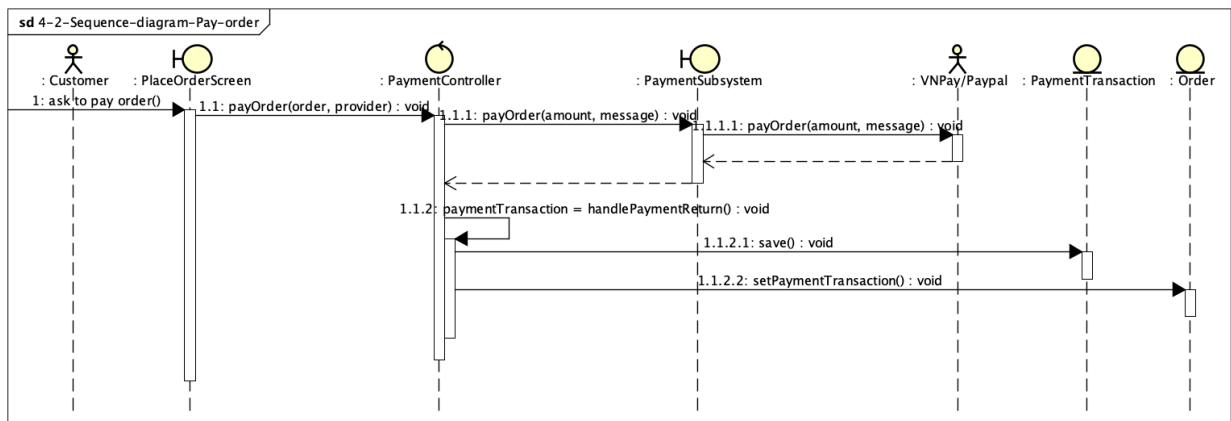
- Chứa các framework, công cụ, và công nghệ như cơ sở dữ liệu, API bên thứ ba (VNPay), hoặc các công cụ UI.
- Phụ thuộc vào các tầng bên trên nhưng không ảnh hưởng đến chúng.



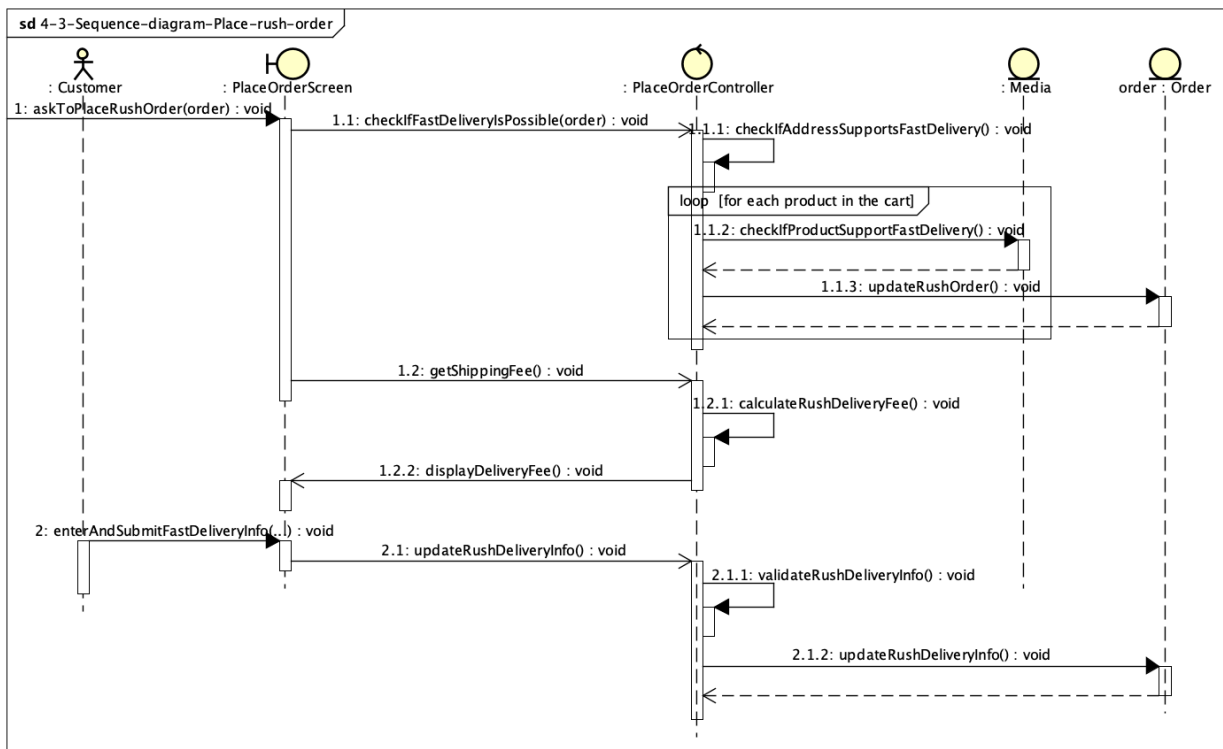
#### 4.2.1. Usecase UC01 “Đặt hàng”



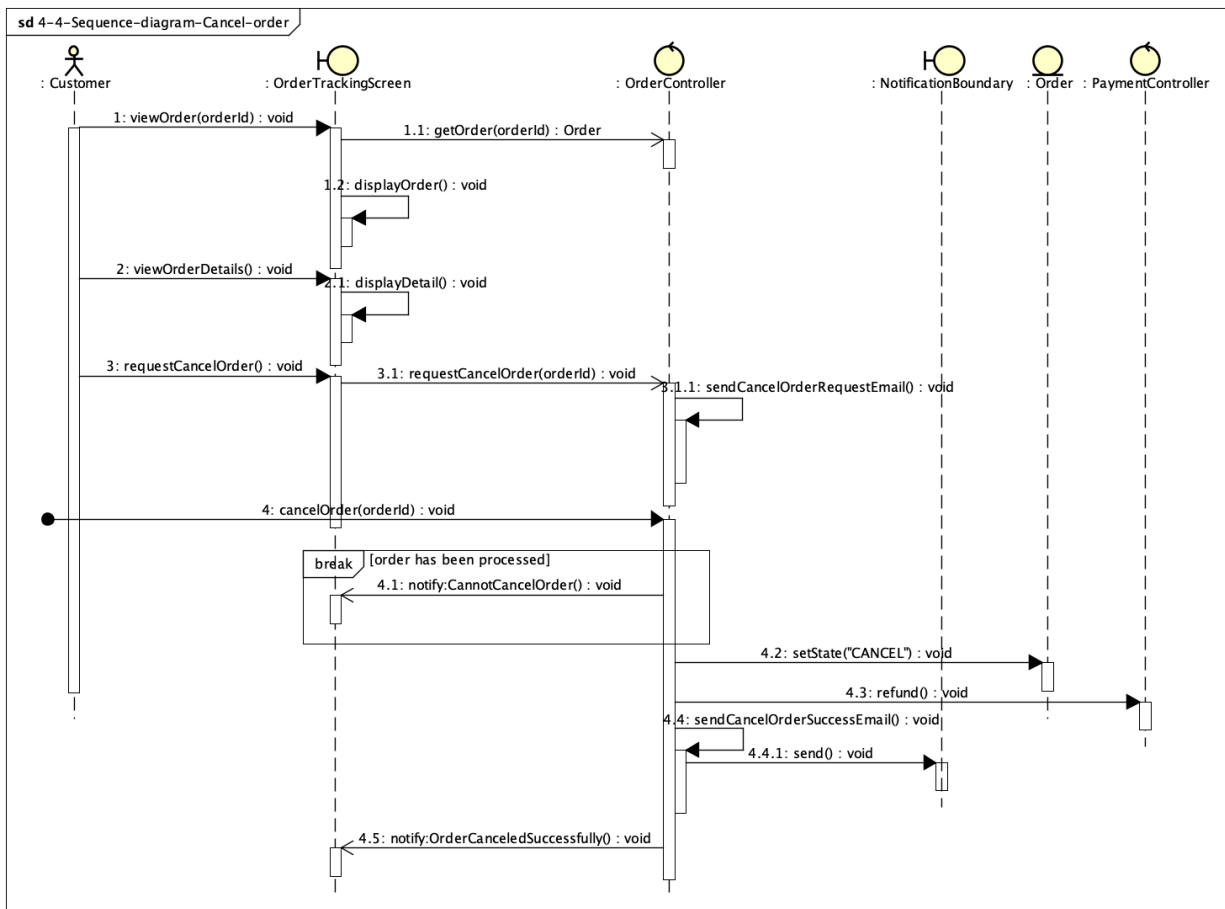
#### 4.2.2. Usecase UC02 “Thanh toán”



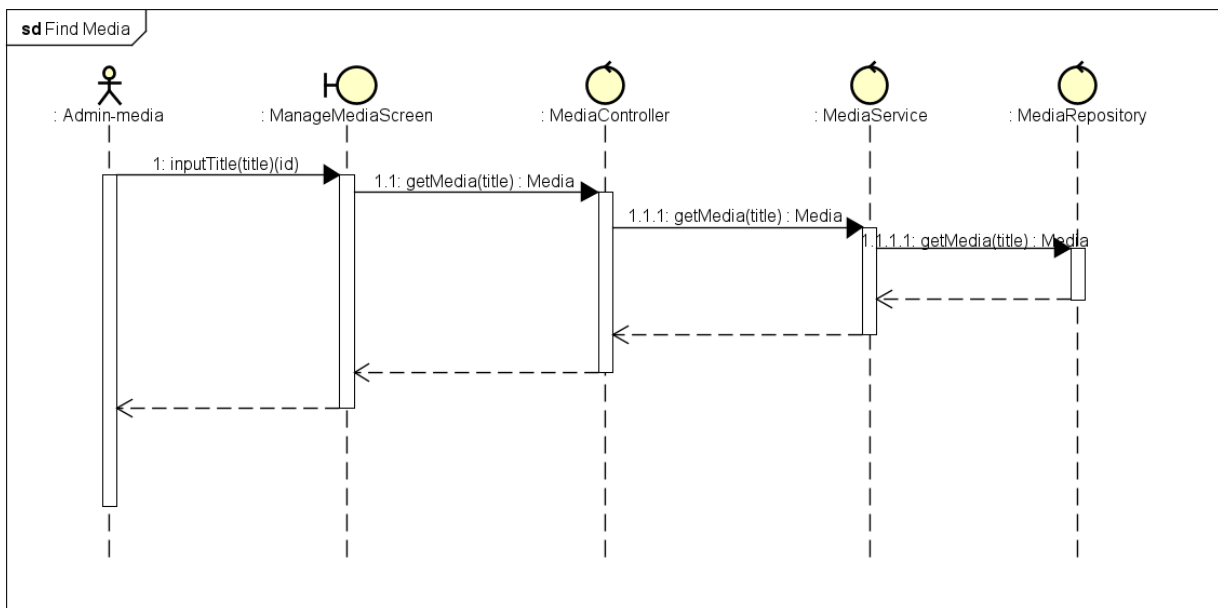
### 4.2.3. Usecase UC03 “Đặt hàng nhanh”



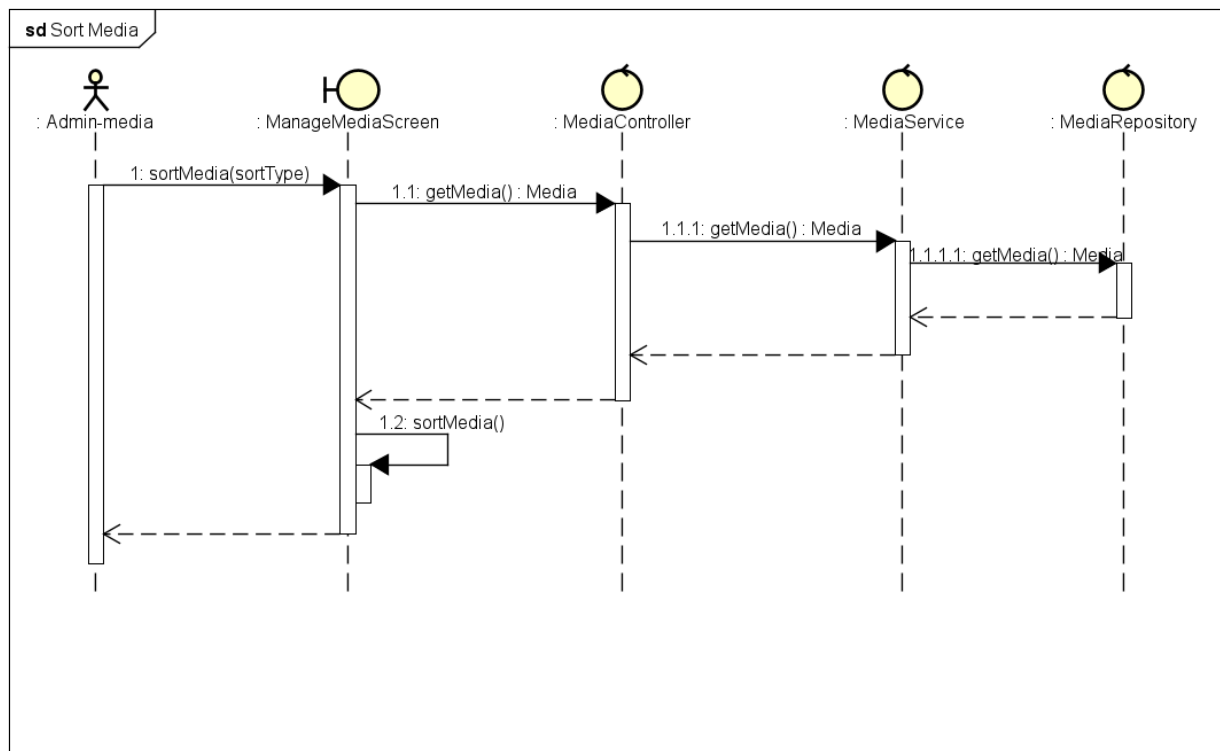
#### 4.2.4. Usecase UC04 “Huỷ đơn hàng”



#### 4.2.5. Usecase UC06 “Tìm kiếm sản phẩm”



#### 4.2.6. Usecase UC07 “Sắp xếp sản phẩm”



#### 4.3. Biểu đồ phân tích lớp

Usecase UC01 “Đặt hàng”

Usecase UC01 “Thanh toán”

Usecase UC01 “Đặt hàng nhanh”

Usecase UC01 “Huỷ đơn hàng”



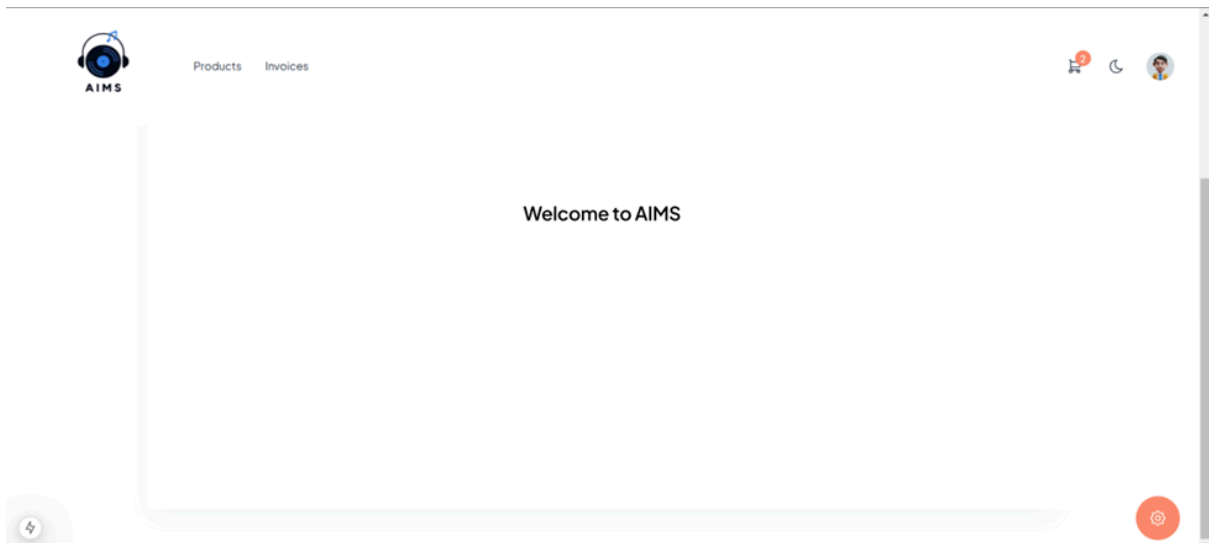
## 5. Thiết kế chi tiết

### 5.1. Thiết kế giao diện người dùng

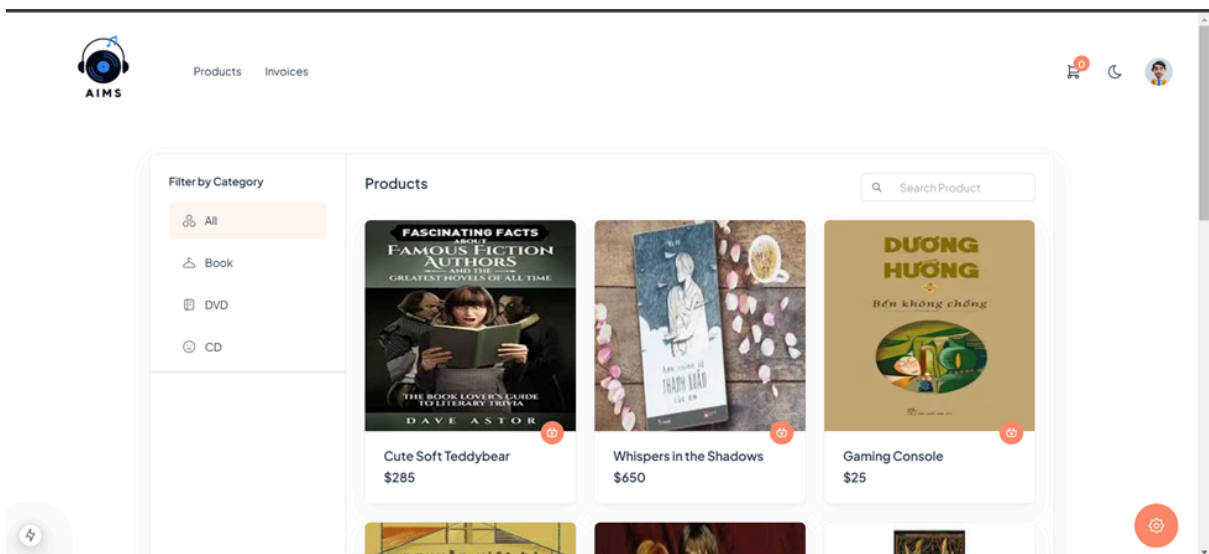
#### 5.1.1. Biểu đồ chuyển màn hình

#### 5.1.2. Đặc tả màn hình

##### 5.1.2.1. Home screen

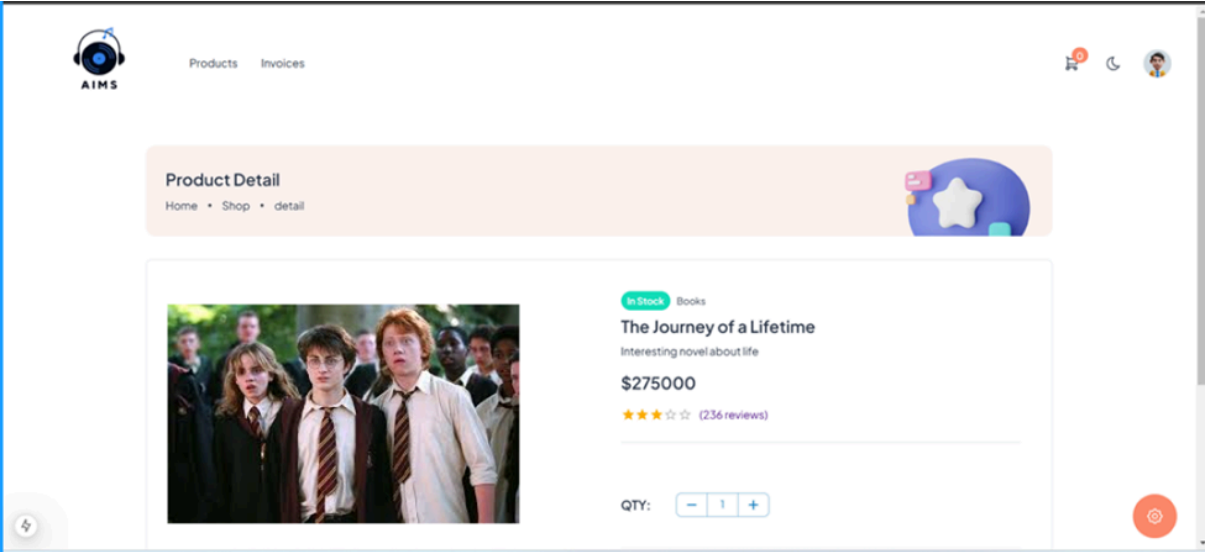


##### 5.1.2.2. Store screen



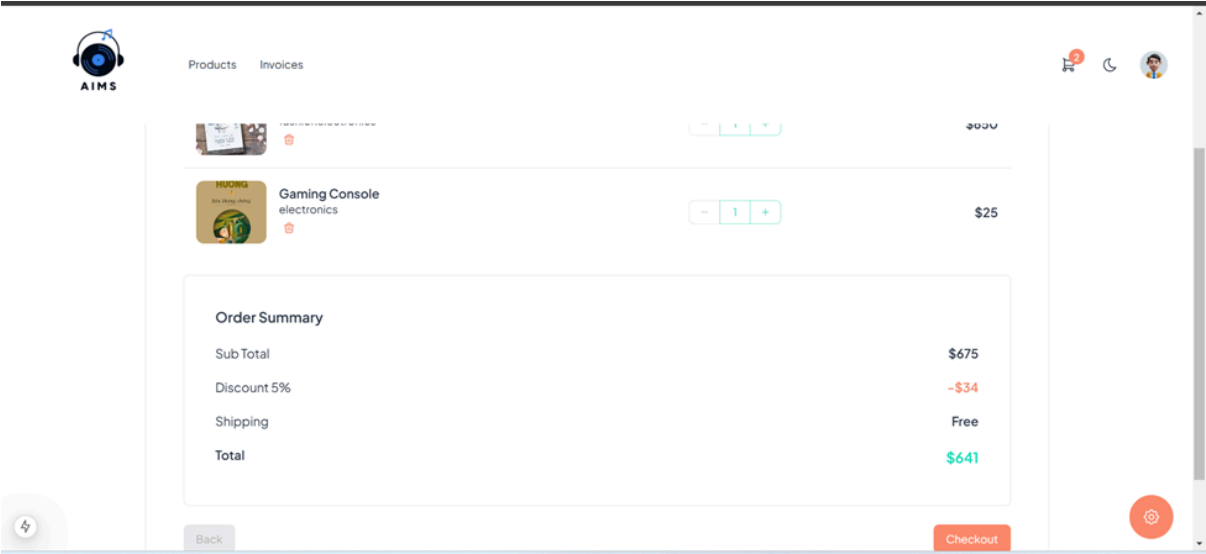
Store Screen			
Date of creation	Approved by	Reviewed by	Person in charge
01/01/2025			
Control	Operation	Function	
Thanh điều hướng	Click	Hiển thị và cho phép người dùng truy cập các màn hình khác nhau.	
Khu vực hiển thị các sản phẩm	Khởi tạo	Hiển thị danh sách các sản phẩm hiện có	
Đơn hàng	Khởi tạo	Hiển thị các thông tin của đơn hàng: mã đơn hàng, ngày đặt hàng, tổng tiền, phương thức thanh toán, danh sách sản phẩm đã đặt	
Nút "Xem chi tiết"	Click	Mở trang xem chi tiết đơn hàng	
Khu vực các lựa chọn tìm kiếm	Khởi tạo	Hiển thị các trường lọc	
Thanh "Tìm kiếm"	Nhập dữ liệu	Cho phép người dùng nhập tiêu đề của media để tìm kiếm media có tiêu đề giống	
Radio Button "Book", "CD", "DVD",	Click	Ngoài tìm kiếm bằng tiêu đề phương tiện, người dùng có thể lọc loại phương tiện	
Button "Giỏ hàng"	Click	Thêm sản phẩm vào giỏ	
Nút "Prev"	Click	Hiển thị trang sản phẩm trước trang hiện tại	
Nút "Next"	Click	Hiển thị trang sản phẩm sau trang hiện tại	

5.1.2.3. Product details

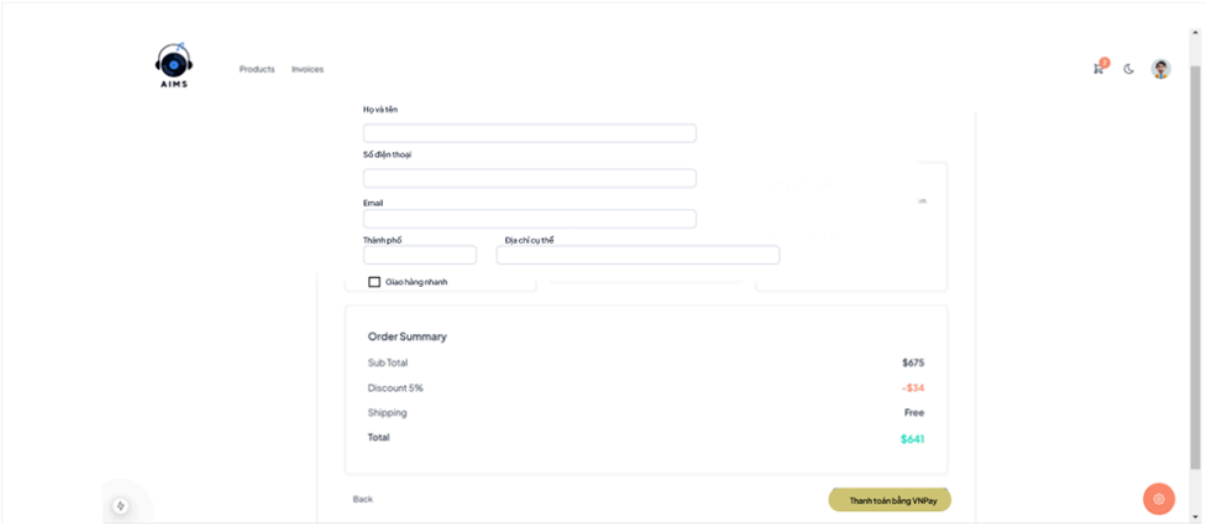


Product Details			
Date of creation	Approved by	Reviewed by	Person in charge
01/01/2025			
Control	Operation	Function	
Khu vực hiển thị các thông tin của sản phẩm	Khởi tạo	Người dùng có thể xem chi tiết thông tin cả sản phẩm	
Button “Add to Cart”	Click	thêm sản phẩm vào giỏ hàng	
Các button “-” hoặc “+”	Click	Điều chỉnh số lượng media muốn thêm vào giỏ hàng	

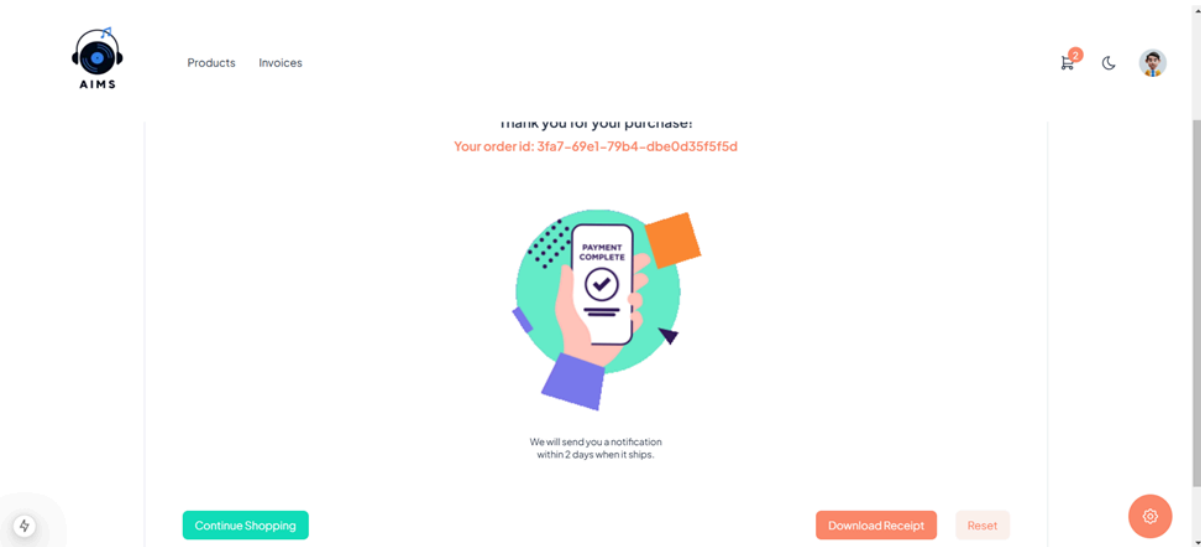
5.1.2.4. Cart screen



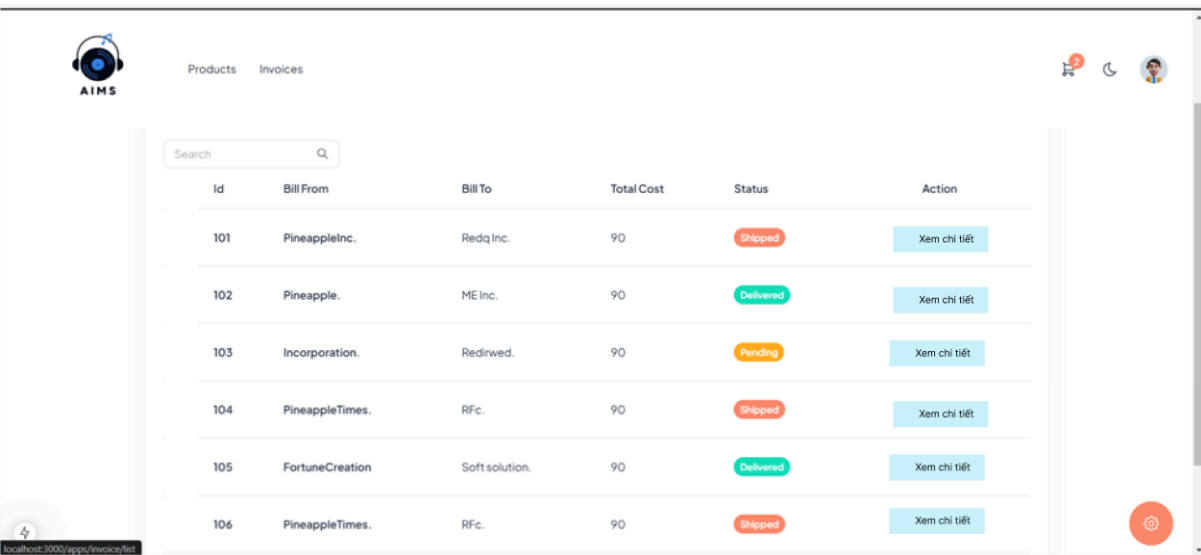
5.1.2.5. Checkout screen



5.1.2.6. Place order result screen



5.1.2.7. Order tracking screen (Màn hình theo dõi đơn hàng)



Order tracking screen			
Date of creation	Approved by	Reviewed by	Person in charge
01/01/2025			
Control	Operation	Function	
Thanh tìm kiếm	Nhập dữ liệu	Cho phép khách hàng nhập mã đơn hàng cần tìm kiếm cần tìm	
Đơn hàng	Khởi tạo	Hiển thị các thông tin của đơn hàng: mã đơn hàng, ngày đặt hàng, tổng tiền, phương thức thanh toán, danh sách sản phẩm đã đặt	

Nút “Xem chi tiết”	Click	Mở trang xem chi tiết đơn hàng
--------------------	-------	--------------------------------

#### 5.1.2.8. Order details (pop-up Xem chi tiết đơn hàng)

AIMS Products Invoices

Thông tin giao hàng: Nguyễn Văn A  
Sdt: 01246789  
Email: abcd@gmail.com  
Địa chỉ: Hai Bà Trưng, Hà Nội

Thông tin đơn hàng: Mã đơn hàng: 1  
Thời gian đặt hàng: 28/12/2014  
Tình trạng: Đang chờ

Item Name	Unit Price	Unit	Total Cost
Course	10	9	90

Sub Total: 90  
Vat: 9  
Grand Total: 99

Hủy đơn hàng

Order details			
Date of creation	Approved by	Reviewed by	Person in charge
01/01/2025			
Control	Operation	Function	
Khu vực hiển thị thông tin đơn hàng	Khởi tạo	Hiển thị các thông tin của đơn hàng gồm: mã đơn hàng, thông tin vận chuyển, ngày đặt hàng, tổng đơn, VAT, phí giao hàng, tổng tiền thanh toán	
Nút “Huỷ đơn hàng”	Click	Cho phép khách hàng yêu cầu huỷ đơn hàng đã đặt nếu đơn hàng chưa được người quản lý sản phẩm xử lý, ngược lại nút này sẽ không xuất hiện	

#### 5.1.2.9. Biểu đồ lớp

#### 5.1.2.10. Biểu đồ trình tự

Notification subsystem

#### 5.1.2.11. Biểu đồ lớp

#### 5.1.2.12. Biểu đồ trình tự

### 5.2. Mô hình dữ liệu

Mô hình hoá dữ liệu mức khái niệm



### 5.2.1.3. Thiết kế chi tiết cơ sở dữ liệu

Bảng Media

STT	Tên Cột	PK	FK	Kiểu dữ liệu	Bắt buộc	Mô tả
1	id	✓		int	✓	Khóa chính, mã sản phẩm
2	title			varchar(100)	✓	Tên sản phẩm
3	category			varchar(100)	✓	Danh mục sản phẩm
4	value			double(10,2)	✓	Giá trị sản phẩm
5	price			double(10,2)	✓	Giá bán (chưa VAT)
6	quantity_in_stock			int	✓	Số lượng tồn kho
7	is_able_to_rush_delivery			bool		Có thể giao nhanh không
8	weight			double(10,2)	✓	Trọng lượng sản phẩm
9	image_url			text		URL hình ảnh
10	barcode_url			text		URL mã vạch
11	description			text		Mô tả sản phẩm

Bảng Book

STT	Tên Cột	PK	FK	Kiểu dữ liệu	Bắt buộc	Mô tả
1	media_id		✓	int	✓	Khóa ngoại, tham chiếu đến bảng Media
2	authors			text	✓	Tên tác giả
3	cover_type			enum	✓	Loại bìa
4	publisher			varchar(100)	✓	Nhà xuất bản
5	publication_date			timestamp	✓	Ngày xuất bản
6	pages			int	✓	Số trang

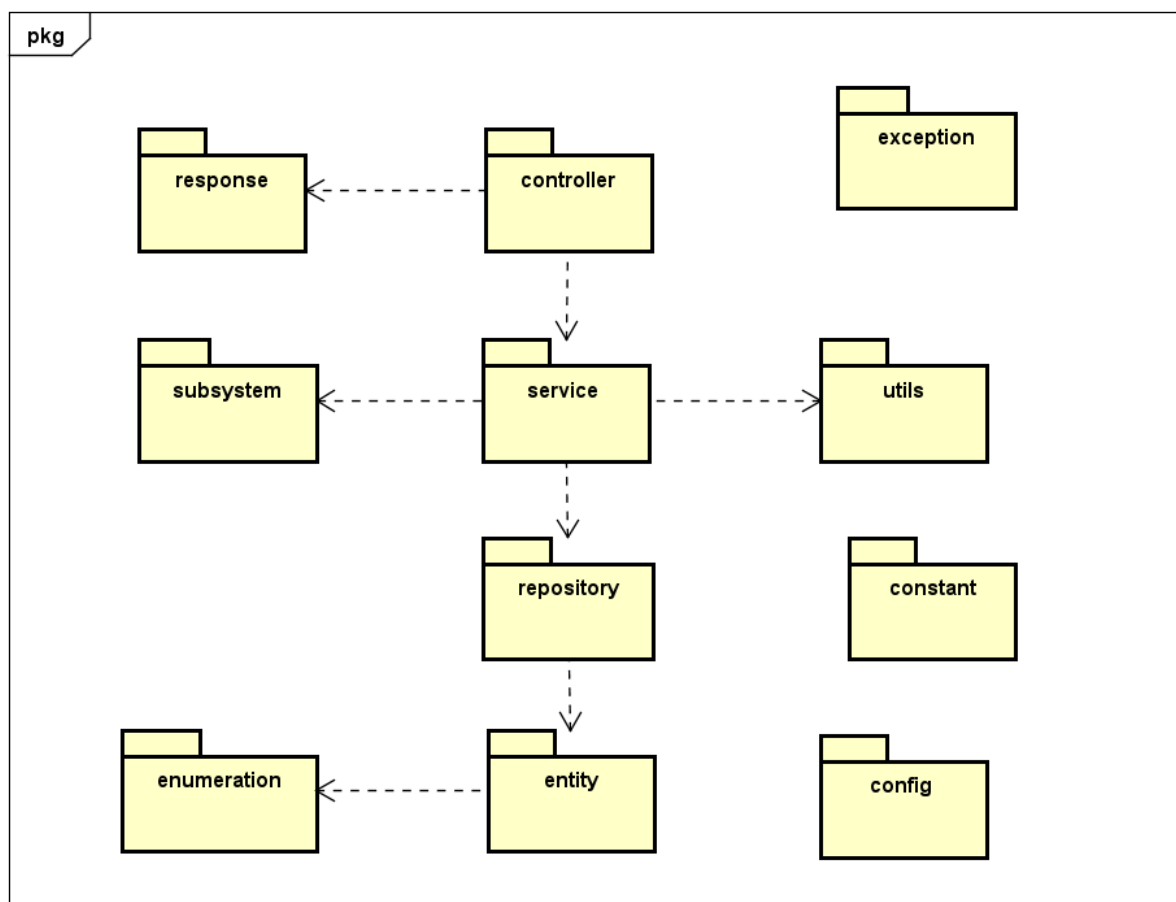
Bảng DVD



STT		Tên Cột	PK	FK	Kiểu dữ liệu	Bắt buộc	Mô tả
1		media_id		✓	int	✓	Khóa ngoại, tham chiếu đến bảng Media
2		authors			text	✓	Tên tác giả
3		cover_type			enum	✓	Loại bìa
4		publisher			varchar(100)	✓	Nhà xuất bản
5		publication_date			timestamp	✓	Ngày xuất bản
6		pages			int	✓	Số trang

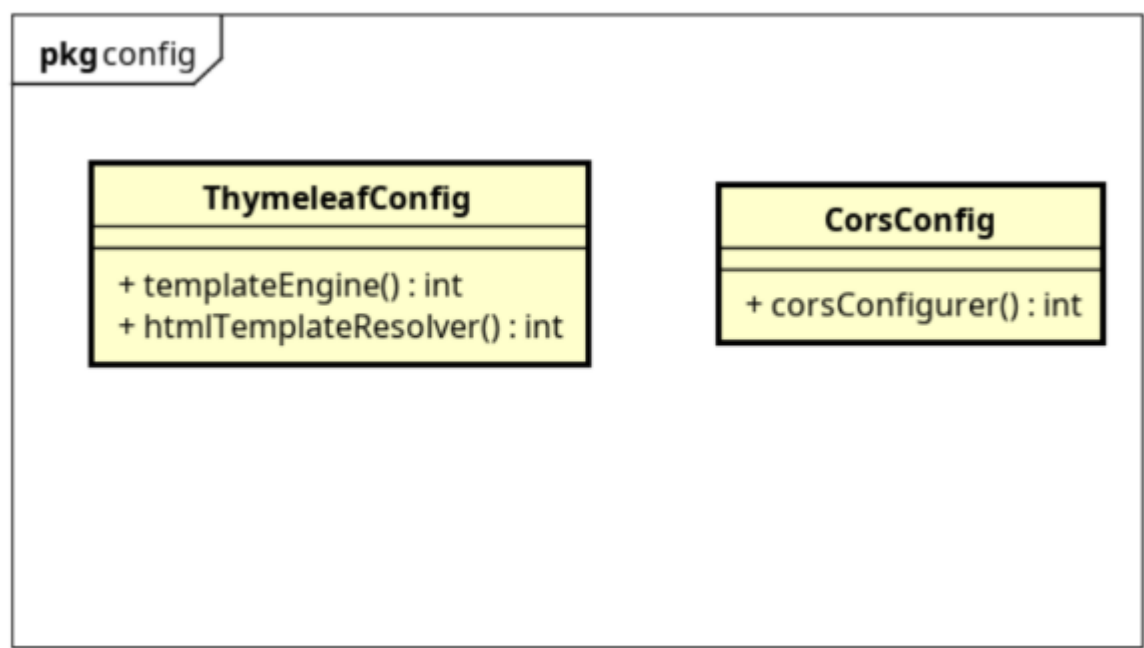
## 5.3. Thiết kế lớp

### 5.3.1. Tổng quan

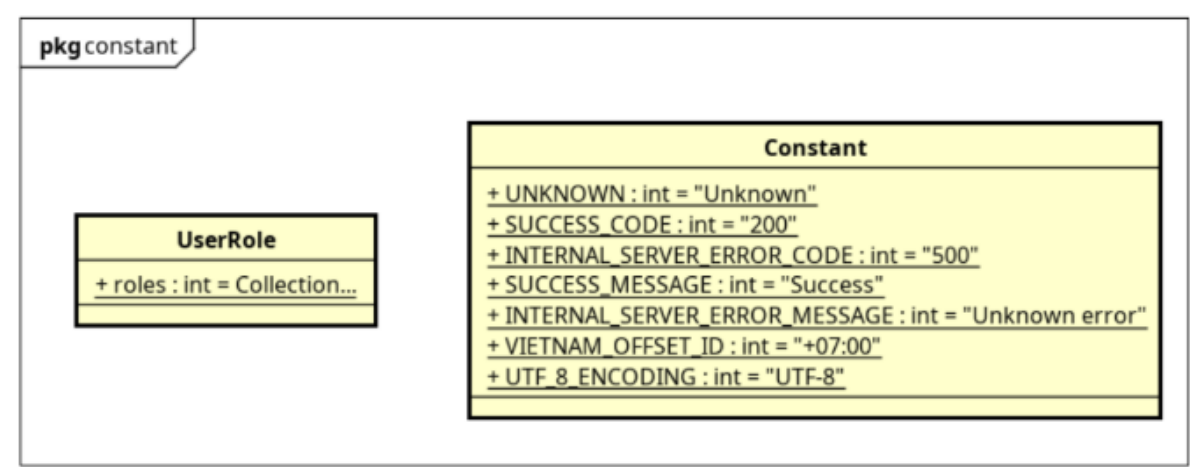


5.3.2. Chi tiết

5.3.2.1. Package config



5.3.2.2. Package constant



5.3.2.3. Package controller

5.3.2.4.

5.3.2.5.

## 6. Xem xét thiết kế (Design Consideration)

Tại mục này, nhóm sẽ chỉ ra một số vấn đề liên quan đến các nguyên tắc thiết kế của Project AIMS [<https://github.com/trangntt-for-student/AIMS>], và triển khai giải pháp trong mã nguồn mới tại [<https://github.com/DI-huyentm/TKXDPM.VP.20241.05.git>]

### 6.1. Goals and Guidelines

Mục tiêu trọng tâm trong thiết kế hệ thống là đảm bảo khả năng mở rộng và dễ dàng bảo trì. Các nguyên tắc và định hướng thiết kế được đề ra bao gồm:

- **Tính linh hoạt:** Hệ thống được xây dựng để dễ bảo trì và nâng cấp, đồng thời có khả năng thích nghi nhanh chóng với các yêu cầu thay đổi trong tương lai.
- **Tuân thủ SOLID Principles:** Các nguyên tắc SOLID được áp dụng nhằm tăng tính linh hoạt, cải thiện khả năng mở rộng và giảm thiểu sự phụ thuộc giữa các thành phần của hệ thống.
- **Chất lượng mã nguồn:** Đảm bảo mã nguồn đạt chất lượng cao thông qua việc tuân thủ các tiêu chuẩn lập trình, kết hợp với việc kiểm thử toàn diện nhằm đảm bảo hệ thống hoạt động ổn định và đáng tin cậy.

### 6.2. Architectural Strategies

- **Frontend:** Sử dụng React để xây dựng giao diện người dùng dưới dạng ứng dụng web, nhằm nâng cao tính tương tác và cải thiện trải nghiệm của người dùng.

- **Backend:** Tích hợp Spring Boot trong phát triển backend, đảm bảo hệ thống có hiệu suất cao và khả năng mở rộng đáp ứng nhu cầu sử dụng.
- **Database:** MySQL được lựa chọn làm hệ quản trị cơ sở dữ liệu, tập trung vào việc duy trì tính nhất quán và hỗ trợ khả năng mở rộng hiệu quả.

## 6.3. Coupling và Cohesion

### 6.3.1. Coupling

#### 6.3.1.1. Content Coupling

Related modules	Description	Improvement Direction
Order.setDeliveryInfo(HashMap deliveryInfo)	Bất kỳ module bên ngoài nào cũng có thể thay đổi trực tiếp thuộc tính `deliveryInfo`	Thay vì sử dụng HashMap có thể bị thay đổi trực tiếp, tạo một lớp bất biến cho DeliveryInfo.

#### Mã nguồn thực thi giải pháp:

```
public class DeliveryInfo extends TimedEntity {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;

    @Column(name = "customer_name")
    private String customerName;

    @Column(name = "email")
    private String email;

    @Column(name = "phone_number")
    private String phoneNumber;

    @Column(name = "province")
    private String province;

    @Column(name = "address")
    private String address;

}
```

#### 6.3.1.2. Common Coupling

#### 6.3.1.3. Control Coupling

Related modules	Description	Improvement Direction
calculateShippingFee(order Order)	Tính toán phí vận chuyển dựa trên điều kiện của Đơn hàng.	Tạo một Service riêng biệt cho logic tính phí vận chuyển

### Mã nguồn thực thi giải pháp:

```
public interface CalculationService {  
  
    // Calculate subtotal for cart media  
    Double calculateCartMediaSubtotal(List<CartMedia> cartMediaList);  
  
    // Calculate subtotal for order media  
    Double calculateOrderMediaSubtotal(List<OrderMedia> orderMediaList);  
  
    // Calculate VAT based on subtotal  
    Double calculateVAT(Double subtotal);  
  
    // Calculate delivery fee for an order  
    Double calculateDeliveryFee(Order order);  
  
    // Calculate total cost with VAT and delivery fee  
    Double calculateTotal(Double subtotal, Double VAT);  
  
    // Calculate total cost with VAT, delivery fee, and rush delivery fee  
    Double calculateTotal(Double subtotal, Double VAT, Double deliveryFee);  
}
```

#### 6.3.1.4. Stamp Coupling

Related modules	Description	Improvement Direction
Cart.checkMediaInCart(Media media)	Chỉ cần ID của đối tượng Media để kiểm tra, nhưng lại phải truyền toàn bộ đối tượng Media vào phương thức checkMediaInCart	Chỉ truyền thông tin cần thiết, trong trường hợp này là ID của Media, thay vì truyền toàn bộ đối tượng Media

### Mã nguồn thực thi giải pháp:

```
public interface CartService {  
    // Add media to cart  
    AddMediaToCartOutput addMediaToCart(AddMediaToCartInput input);  
}  
  
public class AddMediaToCartInput {  
    private String cartId;  
    private Long mediaId;  
    private Integer quantity;  
}
```

#### 6.3.1.5. Data Coupling

Related modules	Description	Improvement Direction
Cart.addCartMedia(CartMedia cm)	Chỉ sử dụng CartMedia như một đối số cần thiết	
Cart.removeCartMedia(CartMedia cm)	Chỉ sử dụng CartMedia như một đối số cần thiết	
CartMedia(Media media, Cart cart, int quantity, int price)	Chỉ sử dụng tham số cần thiết để khởi tạo CartMedia	

### 6.3.2. Cohesion

#### 6.3.2.1. Coincidental Cohesion

Related modules	Description	Improvement Direction

#### 6.3.2.2. Logical Cohesion

Related modules	Description	Improvement Direction
MyMap	Tất cả các phương pháp đều liên quan về mặt logic đến việc xử lý dữ liệu JSON, nhưng thực hiện các nhiệm vụ khác nhau như chuyển đổi, phân tích cú pháp và ánh xạ.	Sử dụng một lớp riêng cho mỗi phương pháp

#### 6.3.2.3. Temporal Cohesion

Related modules	Description	Improvement Direction

#### 6.3.2.4. Procedural Cohesion

Related modules	Description	Improvement Direction

validateName(string Name), validatePhoneNumber(string phoneNumber), validateAddress(string address) trong lớp PlaceOrderController	Tất cả các phương pháp đều là các bước để xác thực thông tin giao hàng	
payOrder(int amount, String orderInfo), onTransactionCompleted(Payme ntTransaction transactionResult), emptyCart() trong lớp PaymentController	Tất cả các phương pháp đều là các bước cho quy trình thanh toán	

#### 6.3.2.5. Communicational Cohesion

Related modules	Description	Improvement Direction
createInvoice(Order order), calculateShippingFee(Order order) trong lớp PlaceOrderController	Tất cả các phương pháp chia sẻ cùng một đầu vào	
Md5(String message), Sha256(String message) trong lớp VnPayConfig	Mã hóa một chuỗi, làm việc trên cùng một đầu vào	
addOrderMedia(OrderMedia om), removeOrderMedia(OrderM edia om) trong lớp Order	Quản lý OrderMedia, làm việc trên cùng một đầu vào	

#### 6.3.2.6. Sequential Cohesion

Related modules	Description	Improvement Direction
createOrder(), createInvoice(Order order) trong lớp PlaceOrderController	createOrder trả về một đơn hàng, trở thành input cho createInvoice	

#### 6.3.2.7. Functional Cohesion

Related modules	Description	Improvement Direction
CartMedia	Tất cả các phương pháp đều giúp xác định đối tượng CartMedia và trích xuất các đặc điểm của đối tượng đó	
Cart	Tất cả các phương pháp đều giúp quản lý giỏ hàng và các mặt hàng trong đó.	

## 6.4. Design Principles: SOLID

Nhóm đưa ra giải pháp đảm bảo nguyên tắc SOLID trong mã nguồn

### 6.4.1. Single Responsibility Principle (SRP)

Related modules	Description	Improvement Direction
Lớp BaseController	Lớp BaseController không hoàn toàn là "base" theo nghĩa thuần túy của một lớp cha dùng chung. Nó chứa logic cụ thể liên quan đến giỏ hàng, không phù hợp nếu có các controller khác không liên quan đến giỏ hàng.	Tổ chức các thành phần vào các gói như Controller, Service, Entity, và Repository để đảm bảo mỗi lớp chỉ thực hiện một nhiệm vụ cụ thể, giúp duy trì trách nhiệm duy nhất.
Lớp PaymentController	1. Xử lý quá nhiều trách nhiệm: Điều phối luồng thanh toán (phương thức payOrder). Lưu kết quả giao dịch vào cơ sở dữ liệu (onTransactionCompleted). Làm trống giỏ hàng (emptyCart). 2. Controller tự tạo đối tượng VnPaySubsystemControllert rách nhiệm này không cần thiết, sẽ làm giảm khả năng mở rộng.	
Lớp Cart	Lớp vừa quản lý instance, vừa quản lý các item Các phương thức quản lý cart và instance không liên quan đến nhau.	



Lớp PlaceOrderController	Lớp đảm nhận quá nhiều trách nhiệm, vi phạm nguyên lý SRP: Kiểm tra tính khả dụng của sản phẩm. Tạo đơn hàng (createOrder) và hóa đơn (createInvoice). Xử lý và xác thực thông tin giao hàng. Tính phí vận chuyển.	
--------------------------	--	--

#### 6.4.2. Open-Closed Principle (OCP)

Related modules	Description	Improvement Direction
Lớp Media	Phương thức khởi tạo Media trong getMediaById và getAllMedia: Nếu muốn thêm loại media mới với các thuộc tính riêng, bạn cần sửa đổi lớp Media hoặc viết các logic xử lý mới liên quan đến loại media đó. Điều này phá vỡ OCP vì việc thêm tính năng mới (loại media) dẫn đến thay đổi mã hiện có.	Sử dụng Factory Pattern để thiết kế các class sao cho có thể mở rộng mà không cần sửa đổi các phần hoạt động, ví dụ như class Media không cần phải sửa đổi khi thêm loại media mới.

#### Mã nguồn thực thi giải pháp:

```

@Target(ElementType.TYPE)
@Retention(RetentionPolicy.RUNTIME)
public @interface MediaFactoryProvider {
    String value();
}

public interface MediaFactoryInterface {
    Media build(String mediaMap);
}

@MediaFactoryProvider(value = "DVD")
public class DVDFactory implements MediaFactoryInterface {
    @Override
    public Media build(String jsonPayload) {
        return JsonMapper.convertJsonToObject(jsonPayload, DVD.class);
    }
}

```

### 6.4.3. Liskov Substitution Principle (LSP)

Related modules	Description	Improvement Direction
Lớp PaymentTransaction	Trong trường hợp mở rộng module PaymentTransaction trong tương lai (VD: thêm các loại tính năng khác như refund), cần đảm bảo rằng các lớp con giữ nguyên hành vi của các phương thức như isSuccess.	Sử dụng abstraction để định nghĩa các hành vi chung

#### Mã nguồn thực thi giải pháp:

```
public interface PaymentService {  
  
    PayOrderOutput payOrder(PayOrderInput input);  
  
    void refund(RefundInput input);  
}
```

### 6.4.4. Interface Segregation Principle (ISP)

Related modules	Description	Improvement Direction
IPaymentProcessor	Trong tương lai, chúng ta có thể sẽ phải tích hợp nhiều phương thức thanh toán khác nhau, mà không cần sửa mã nguồn của các lớp sâu.	Triển khai Factory Pattern cho việc lựa chọn cổng thanh toán.

#### Mã nguồn thực thi giải pháp:

```
public class PaymentSubsystemFactory {  
  
    private static final Map<String, Class<? extends PaymentSubsystem>> paymentSubsystemMap =  
        new HashMap<>();  
  
    static {  
        registerPaymentSubsystemClasses();  
    }  
}
```

```

private static void registerPaymentSubsystemClasses() {
    Reflections reflections = new Reflections(new ConfigurationBuilder()
        .forPackages("vn.hust.aims.subsystem") // Package for paymentsubsystem
        .addScanners(new SubTypesScanner()));

    Set<Class<? extends PaymentSubsystem>> productClasses = reflections.getSubTypesOf(
        PaymentSubsystem.class);

    for (Class<? extends PaymentSubsystem> paymentSubsystemClass : productClasses) {
        Provider factoryAnnotation = paymentSubsystemClass.getAnnotation(Provider.class);
        if (factoryAnnotation != null) {
            String key = factoryAnnotation.value();
            paymentSubsystemMap.put(key, paymentSubsystemClass);
        }
    }
}

public static PaymentSubsystem get(ProviderType type) {
    try {
        Class<?> subsystemClass = paymentSubsystemMap.get(type.name());
        return (PaymentSubsystem) subsystemClass.getDeclaredConstructor().newInstance();
    } catch (Exception e) {
        // Handle exceptions appropriately
        throw new PaymentNotSupportedException();
    }
}

```

#### 6.4.5. Dependency Inversion Principle (DIP)

Related modules	Description	Improvement Direction
CD, DVD, Book, Media	Phụ thuộc trực tiếp vào lớp triển khai. Khi cần thêm một loại media mới (ví dụ, LP), phải sửa đổi mã của MediaController, phá vỡ nguyên tắc DIP.	Thiết kế các class để phụ thuộc vào lớp abstraction, không phải phụ thuộc vào implementations cụ thể, sử dụng interface để giảm sự phụ thuộc trực tiếp vào các class cụ thể. Khi cần tạo đối tượng Media thuộc 1 trong 4 loại Book, CD, DVD, LP thì chỉ cần gọi MediaFactoryInterface.

#### Mã nguồn thực thi giải pháp:

```

public interface MediaFactoryInterface {
    Media build(String mediaMap);
}

```

```

@MediaFactoryProvider(value = "DVD")

```

```

public class DVDFactory implements MediaFactoryInterface {
    @Override
    public Media build(String jsonPayload) {
        return JsonMapper.convertJsonToObject(jsonPayload, DVD.class);
    }
}

```

## 6.5. Design Pattern

### 6.5.1. Factory Pattern

Sử dụng Factory Pattern để quản lý và tạo đối tượng theo yêu cầu, giúp khởi tạo đối tượng một cách linh hoạt và dễ mở rộng. Áp dụng trong việc tạo subsystem và các đối tượng khác như PaymentSubsystem. Nhờ đó, khi thêm 1 phương thức thanh toán mới (Ví dụ: PayPal), không cần chỉnh sửa mã nguồn trước đó.

#### Mã nguồn thực thi giải pháp:

```

public class PaymentSubsystemFactory {
    public static PaymentSubsystem get(ProviderType type) {
        try {
            Class<?> subsystemClass = paymentSubsystemMap.get(type.name());
            return (PaymentSubsystem) subsystemClass.getDeclaredConstructor().newInstance();
        } catch (Exception e) {
            throw new PaymentNotSupportedException();
        }
    }
}

```

## 7. Đánh giá công việc