

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



**Báo cáo BTL**  
**Thiết kế xây dựng phần mềm**

**Đề tài: Thiết kế xây dựng phần mềm**  
**Thương mại điện tử AIMS Project**

Giảng viên hướng dẫn:	TS. Nguyễn Thị Thu Trang	
Sinh viên thực hiện:	Phạm Thị Phương Nga	20200430
	Phạm Thị Kim Ngân	20204673
	Đỗ Minh Nghĩa	20204593
	Đoàn Đình Phúc	20200470

*Hà Nội, 1/2024*

## Contents

Đóng góp của các thành viên .....	4
1 Usecase Diagram.....	5
1.1 Usecase diagram.....	5
1.2 Quy trình nghiệp vụ.....	5
1.3 Đặc tả usecase.....	6
1.3.1 CRUD sản phẩm .....	6
1.3.2 Tìm kiếm sản phẩm .....	9
1.3.3 Lọc sản phẩm.....	12
1.3.4 Quản lý giỏ hàng.....	14
2 Usecase Analysis.....	18
2.1 Usecase CRUD sản phẩm.....	18
2.1.1 Sequence Diagram .....	18
2.1.2 Analysis class diagram .....	19
2.2 Usecase Tìm kiếm sản phẩm .....	20
2.2.1 Sequence Diagram .....	20
2.2.2 Analysis class diagram .....	21
2.3 Usecase Lọc sản phẩm.....	22
2.3.1 Sequence Diagram .....	22
2.3.2 Analysis class diagram .....	23
2.4 Usecase Quản lý giỏ hàng .....	24
2.4.1 Sequence Diagram .....	24
2.4.2 Analysis class diagram .....	25
3 Interface Design .....	25
3.1 User Interface Design.....	25
3.1.1. Media Manage Page .....	26
3.1.2. Home page.....	28
3.2 System Interface Design.....	29
4.2.1. Identify subsystems.....	29
4.2.2 Identify subsystem interface .....	29

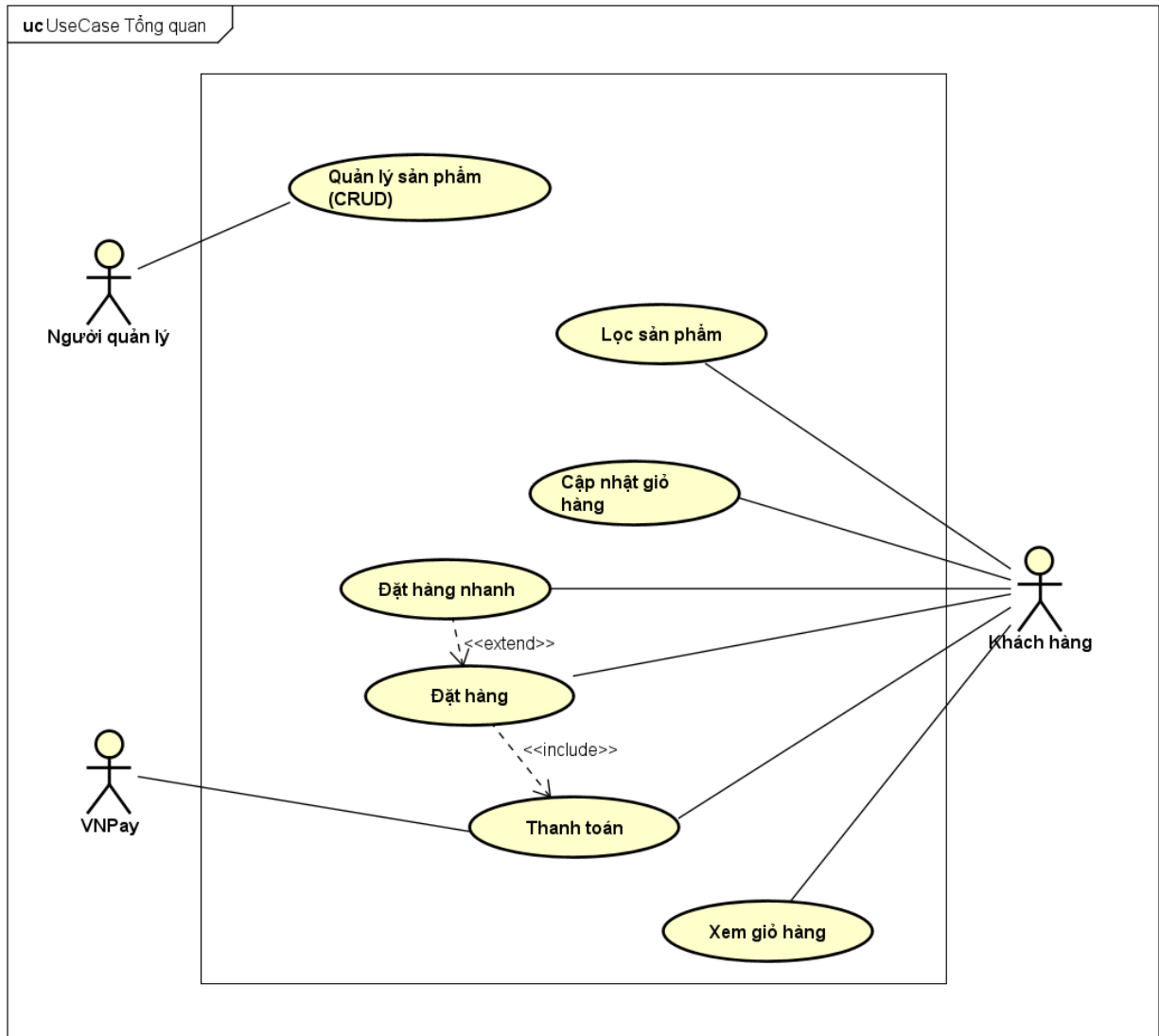
4.2.3	Subsystem design.....	30
4	Class Design.....	31
4.1	General class diagram.....	31
4.2	Detail class diagrams .....	31
4.2.1	Usecase Tìm kiếm, lọc sản phẩm.....	31
4.3	Relationship class diagram .....	33
5	Data Modeling.....	34
5.1	Conceptual data model .....	34
5.2	Database Design .....	35
5.2.1	Logical Data Model .....	35
5.2.2	Physical Data Model.....	35
6	Good Design .....	39
6.1	Cohesion .....	39
6.2	Coupling .....	41
6.3	SOLID .....	46
	Phụ lục hình ảnh.....	53

## Đóng góp của các thành viên

Họ và tên	MSSV	Đóng góp
Phạm Thị Phương Nga	20200430	<ul style="list-style-type: none"> <li>- Phân tích, thiết kế usecase CRUD sản phẩm (phân tích yêu cầu, phân tích usecase, thiết kế giao diện (mediamanage), thiết kế lớp, viết unit test)</li> <li>- Good design: phân tích, tìm các yếu tố cohesion, coupling, SOLID trong codebase</li> </ul>
Phạm Thị Kim Ngân	20204673	<ul style="list-style-type: none"> <li>- Phân tích, thiết kế usecase tìm sản phẩm theo tên, lọc sản phẩm (phân tích yêu cầu, phân tích usecase, thiết kế giao diện (homepage), thiết kế lớp, mô hình hóa dữ liệu, viết unit test)</li> <li>- Good design: phân tích, tìm các yếu tố cohesion, coupling, SOLID trong codebase</li> </ul>
Đỗ Minh Nghĩa	20204593	<ul style="list-style-type: none"> <li>- Phân tích, thiết kế cho usecase quản lý giỏ hàng</li> <li>- Good design: tìm các cohesion, coupling, solid</li> </ul>
Đoàn Đình Phúc	20200470	<ul style="list-style-type: none"> <li>- Good design: phân tích, tìm các yếu tố cohesion, coupling, SOLID trong codebase</li> </ul>

# 1 Usecase Diagram

## 1.1 Usecase diagram

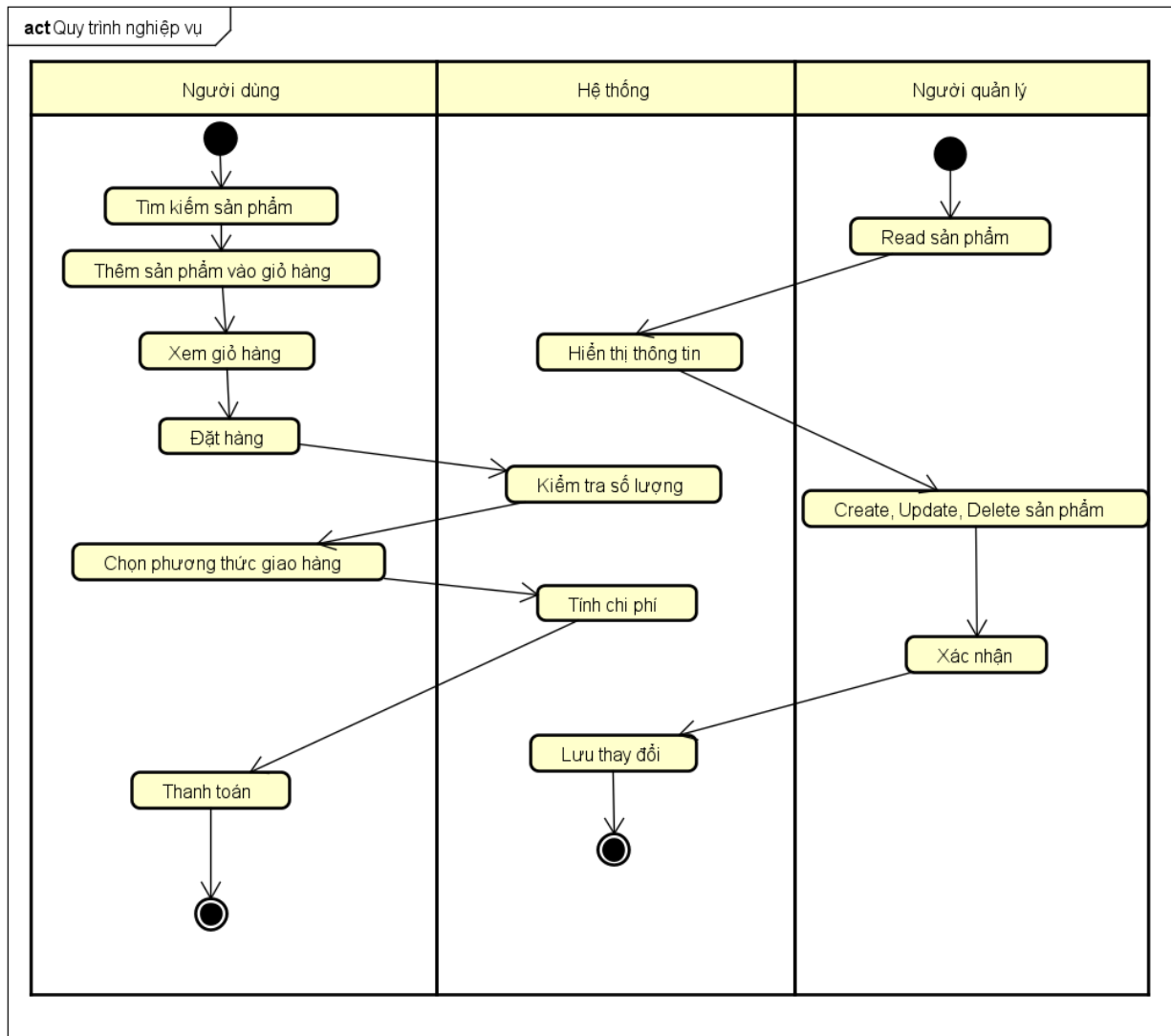


Hình 1: Biểu đồ Usecase tổng quan

## 1.2 Quy trình nghiệp vụ

- Quy trình nghiệp vụ mua hàng trên hệ thống của khách hàng: khách hàng mở ứng dụng tìm kiếm sản phẩm trên trang chủ, thêm sản phẩm mình muốn mua vào giỏ hàng, sau đó vào giỏ hàng để xem lại lựa chọn của mình. Sau khi ấn đặt hàng khách hàng cần điền thông tin giao hàng và chọn phương thức đặt hàng. Nếu khách hàng chọn giao hàng nhanh và đáp ứng được điều kiện thì hệ thống sẽ yêu cầu khách hàng điền thêm 1 form chỉ dẫn giao hàng. Sau đó khách hàng được điều hướng đến thanh toán qua VNPay, khi thành công thì kết thúc phiên giao dịch.

- Quy trình nghiệp vụ của người quản lý sản phẩm: người quản lý mở ứng dụng, truy cập màn hình quản lý sản phẩm để xem các phẩm hiện có, sau đó thực hiện thêm, sửa, xóa sản phẩm tùy theo nhu cầu. Người quản lý xác nhận, hệ thống thực hiện lưu thay đổi.



Hình 2: Biểu đồ quy trình nghiệp vụ

### 1.3 Đặc tả usecase

#### 1.3.1 CRUD sản phẩm

### Use Case “CRUD sản phẩm”

#### 1. Mã use case

UC001

#### 2. Giới thiệu

Use case mô tả sự tương tác giữa người quản lý sản phẩm và hệ thống khi người quản lý muốn xem thông tin, thêm, sửa, xóa sản phẩm.

### 3. Tác nhân

- Khách

### 4. Tiền điều kiện

Người quản lý đang ở giao diện trang chủ

### 5. Luồng sự kiện chính

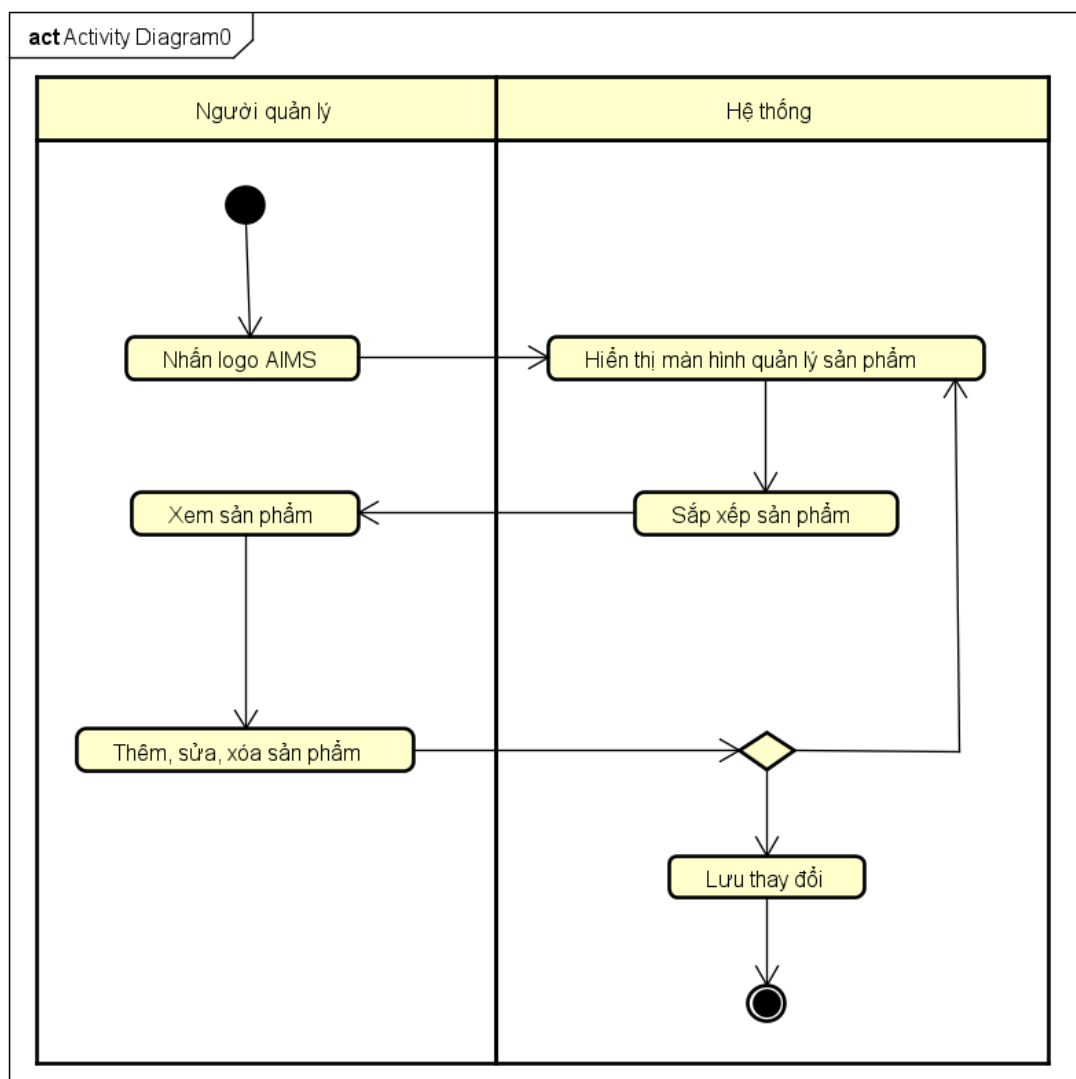
1. Người quản lý ấn logo AIMS.
2. Hệ thống hiển thị màn hình quản lý sản phẩm
3. Người quản lý ấn các trường thông tin sản phẩm
4. Hệ thống hiển thị các sản phẩm theo trường thông tin được chọn xếp theo alphabet.
5. Người quản lý ấn nút delete
6. Hệ thống xóa sản phẩm.
7. Người quản lý ấn nút sửa sản phẩm, điền các thông tin muốn thay đổi và xác nhận.
8. Hệ thống lưu thay đổi.
9. Người quản lý ấn nút thêm sản phẩm, chọn loại sản phẩm, điền các thông tin và xác nhận.
10. Hệ thống lưu thay đổi.

### 6. Luồng sự kiện thay thế

**Bảng 1-Luồng sự kiện thay thế của Use case "CRUD sản phẩm"**

STT	Vị trí	Điều kiện	Hành động	Vị trí tiếp tục
1.	Tại bước 7	Người quản lý ấn Cancel	▪ Hệ thống không lưu thay đổi	Bước 2
2.	Tại bước 9	Người quản lý ấn Cancel	▪ Hệ thống không lưu thay đổi	Bước 2

### 7. Biểu đồ hoạt động



Hình 3: Biểu đồ hoạt động Usecase CRUD sản phẩm

## 8. Dữ liệu đầu vào

Bảng 2-Dữ liệu đầu vào của thêm/sửa sản phẩm

STT	Trường dữ liệu	Mô tả	Bắt buộc?	Điều kiện hợp lệ	Ví dụ
1.	Title	Tên sản phẩm	Có		Doraemon
2.	Category	Loại hình	Có		DVD
3.	Value	Giá trị	Có	Number	12
4.	Price	Giá cả hiện tại	Có	Number	15
5.	Quantity	Số lượng sản phẩm	Có	Number	5
6.	Image	Link hình ảnh	Có		Doraemon.jpg



7.	Author	Tác giả	Không		Fujiko F. Fujio
8.	Cover type	Loại bìa (mềm, cứng)	Không		Mềm
9.	Publisher	Nhà xuất bản	Không		NXB Kim Đồng
10.	Publish date	Ngày xuất bản	Không	MM/DD/YYYY	01/01/2024
11.	Number of pages	Số trang	Không	Number	300
12.	Language	Ngôn ngữ	Không		Tiếng Nhật
13.	Book category	Thể loại (truyện tranh, truyện chữ, sách ảnh)	Không		Truyện tranh
14.	Disc type	Loại đĩa	Không		HD-DVD
15.	Director	Đạo diễn	Không		Tsutomu Shibayama
16.	Runtime	Thời lượng	Không	Number	120
17.	Studio	Hãng sản xuất	Không		TV Asahi
18.	Subtitle	Phụ đề	Không		Tiếng Việt
19.	Release date	Ngày phát hành	Không	MM/DD/YYYY	05/22/2023
20.	Film type	Thể loại phim (phim bộ, phim lẻ)	Không		Phim lẻ
21.	Artists	Nghệ sĩ	Không		Natsukawa Rimi
22.	Record label	Hãng ghi âm	Không		Koorogi 73
23.	Music type	Thể loại nhạc	Không		J-Pop

## 9. Dữ liệu đầu ra

No	Trường dữ liệu	Mô tả	Điều kiện hợp lệ	Example
1.				

## 10. Hậu điều kiện

Không

### 1.3.2 Tìm kiếm sản phẩm

#### Use Case “Tìm kiếm sản phẩm”

## 11. Mã use case

UC002

## 12. Giới thiệu

Use case mô tả sự tương tác giữa khách hàng và hệ thống khi khách hàng tìm kiếm sản phẩm

## 13. Tác nhân

- Khách

## 14. Tiền điều kiện

Khách hàng đang ở giao diện trang chủ

## 15. Luồng sự kiện chính (Thành công)

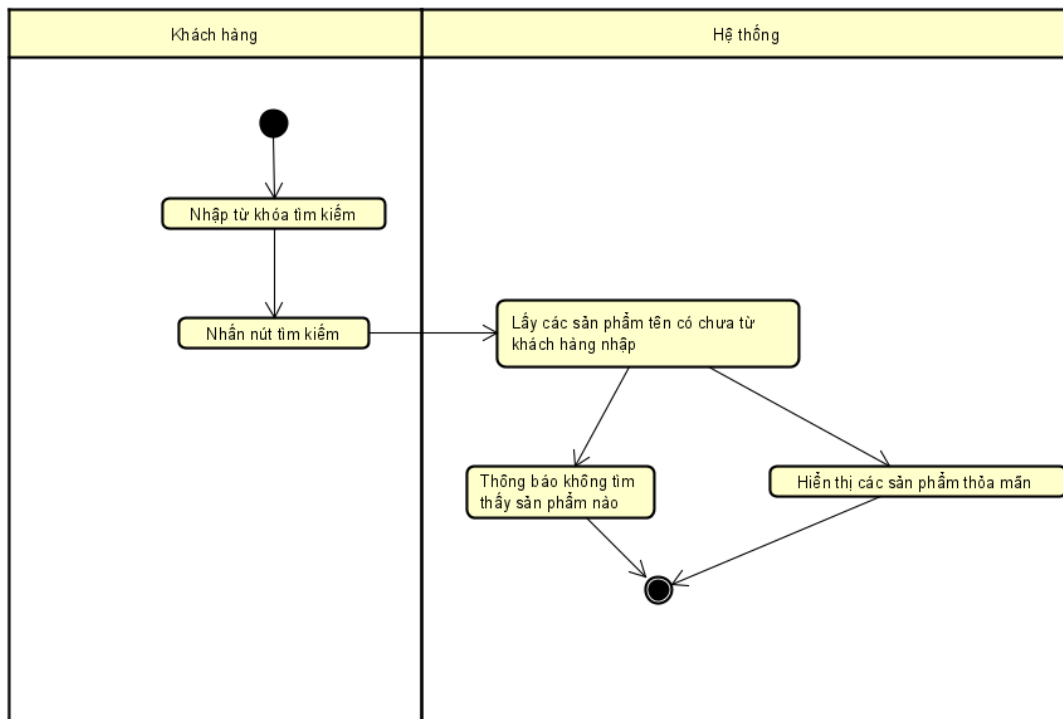
11. Khách hàng nhập từ khóa trên thanh tìm kiếm
12. Khách hàng ấn nút tìm kiếm
13. Hệ thống lấy ra các sản phẩm tên chứa nội dung khách hàng nhập
14. Hệ thống hiển thị các sản phẩm thỏa mãn

## 16. Luồng sự kiện thay thế

**Bảng 1-Luồng sự kiện thay thế của Use case "Tìm kiếm sản phẩm"**

STT	Vị trí	Điều kiện	Hành động	Vị trí tiếp tục
3.	Tại bước 4	Nếu không tìm thấy sản phẩm nào	▪ Hệ thống hiện thị thông báo không tìm thấy sản phẩm theo yêu cầu	Kết thúc usecase

## 17. Biểu đồ hoạt động



Hình 4: Biểu đồ hoạt động của Use case "Tìm kiếm sản phẩm"

## 18. Dữ liệu đầu vào

Bảng 2-Dữ liệu đầu vào của tìm kiếm sản phẩm

STT	Trường dữ liệu	Mô tả	Bắt buộc?	Điều kiện hợp lệ	Ví dụ
24.	Tên sản phẩm	Từ bất kì khách hàng nhập	Có	Không chỉ chứa mỗi khoảng trắng	book

## 19. Dữ liệu đầu ra

Danh sách sản phẩm thỏa mãn

No	Trường dữ liệu	Mô tả	Điều kiện hợp lệ	Example
2.	Tên sản phẩm			DVD Phim Vượt Ngục
3.	Số lượng 1 sản phẩm			2
4.	Giá một sản phẩm			50.000
5.	Ảnh sản phẩm			

## 20. Hậu điều kiện

Không

### 1.3.3 Lọc sản phẩm

## Use Case “Lọc sản phẩm”

### 1. Mã use case

UC003

### 2. Giới thiệu

Use case mô tả sự tương tác giữa khách hàng và hệ thống khi khách hàng lọc tìm sản phẩm

### 3. Tác nhân

- Khách

### 4. Tiền điều kiện

Khách hàng đang ở giao diện trang chủ

### 5. Luồng sự kiện chính (Thành công)

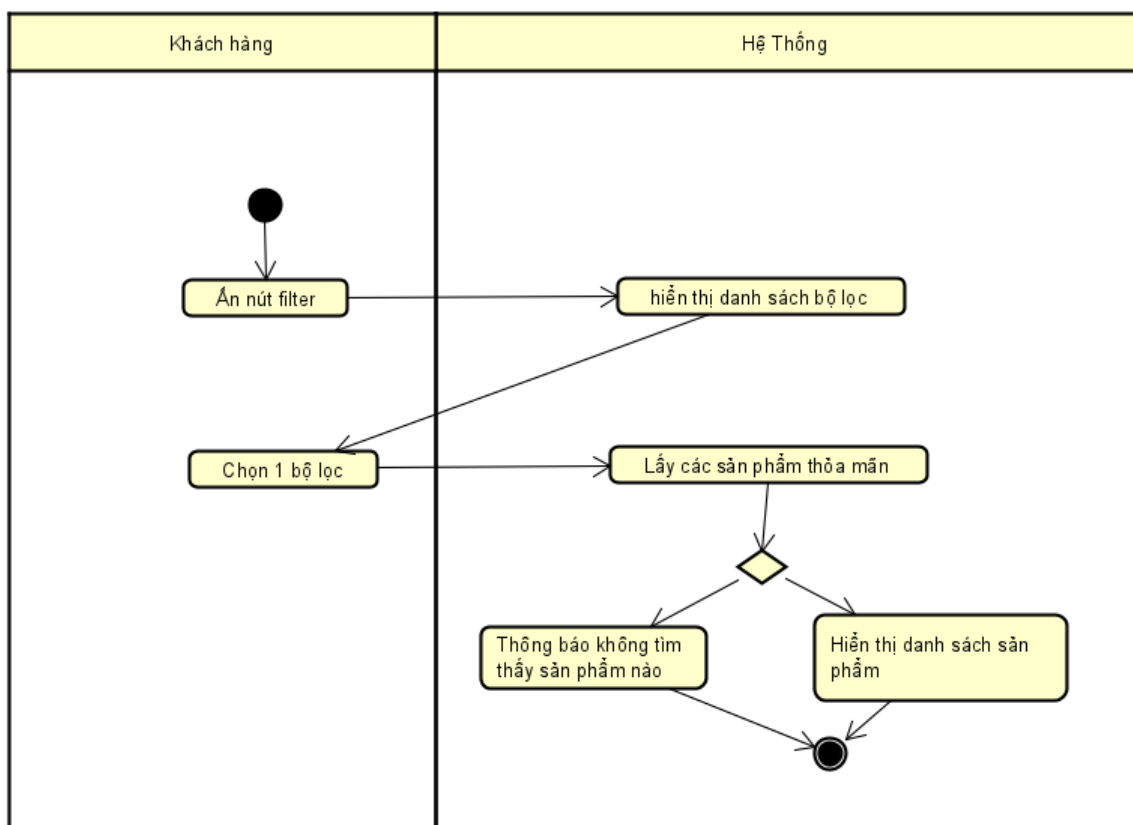
1. Khách hàng ấn vào nút filter
2. Hệ thống hiển thị danh sách các bộ lọc
3. Khách hàng chọn 1 bộ lọc mình muốn
4. Hệ thống lấy ra các sản phẩm thỏa mãn yêu cầu lọc
5. Hệ thống hiển thị danh sách sản phẩm

### 6. Luồng sự kiện thay thế

Bảng 1-Luồng sự kiện thay thế của Use case "Lọc sản phẩm"

STT	Vị trí	Điều kiện	Hành động	Vị trí tiếp tục
1.	Tại bước 4	Nếu không tìm thấy sản phẩm nào	▪ Hệ thống hiển thị thông báo không tìm thấy sản phẩm theo yêu cầu	Kết thúc usecase

### 7. Biểu đồ hoạt động



Hình 5: Biểu đồ hoạt động của Use case "Lọc sản phẩm"

## 8. Dữ liệu đầu vào

Bảng 2-Dữ liệu đầu vào của lọc sản phẩm

STT	Trường dữ liệu	Mô tả	Bắt buộc?	Điều kiện hợp lệ	Ví dụ
1.	Tên bộ lọc	Chọn từ danh sách	Có		dvd

## 9. Dữ liệu đầu ra

Danh sách sản phẩm thỏa mãn

No	Trường dữ liệu	Mô tả	Điều kiện hợp lệ	Example
1.	Tên sản phẩm			DVD Phim Vượt Ngục
2.	Số lượng 1 sản phẩm			2
3.	Giá một sản phẩm			50.000
4.	Ảnh sản phẩm			

## 10. Hậu điều kiện

Không

### 1.3.4 Quản lý giỏ hàng

## Use case “Quản lý giỏ hàng”

### 1. Mã use case

UC004

### 2. Giới thiệu

UseCase mô tả quá trình khách hàng thực hiện những thao tác để quản lý giỏ hàng của mình.

### 3. Tác nhân

Khách hàng

### 4. Tiền điều kiện

Không

### 5. Luồng sự kiện chính (Thành công)

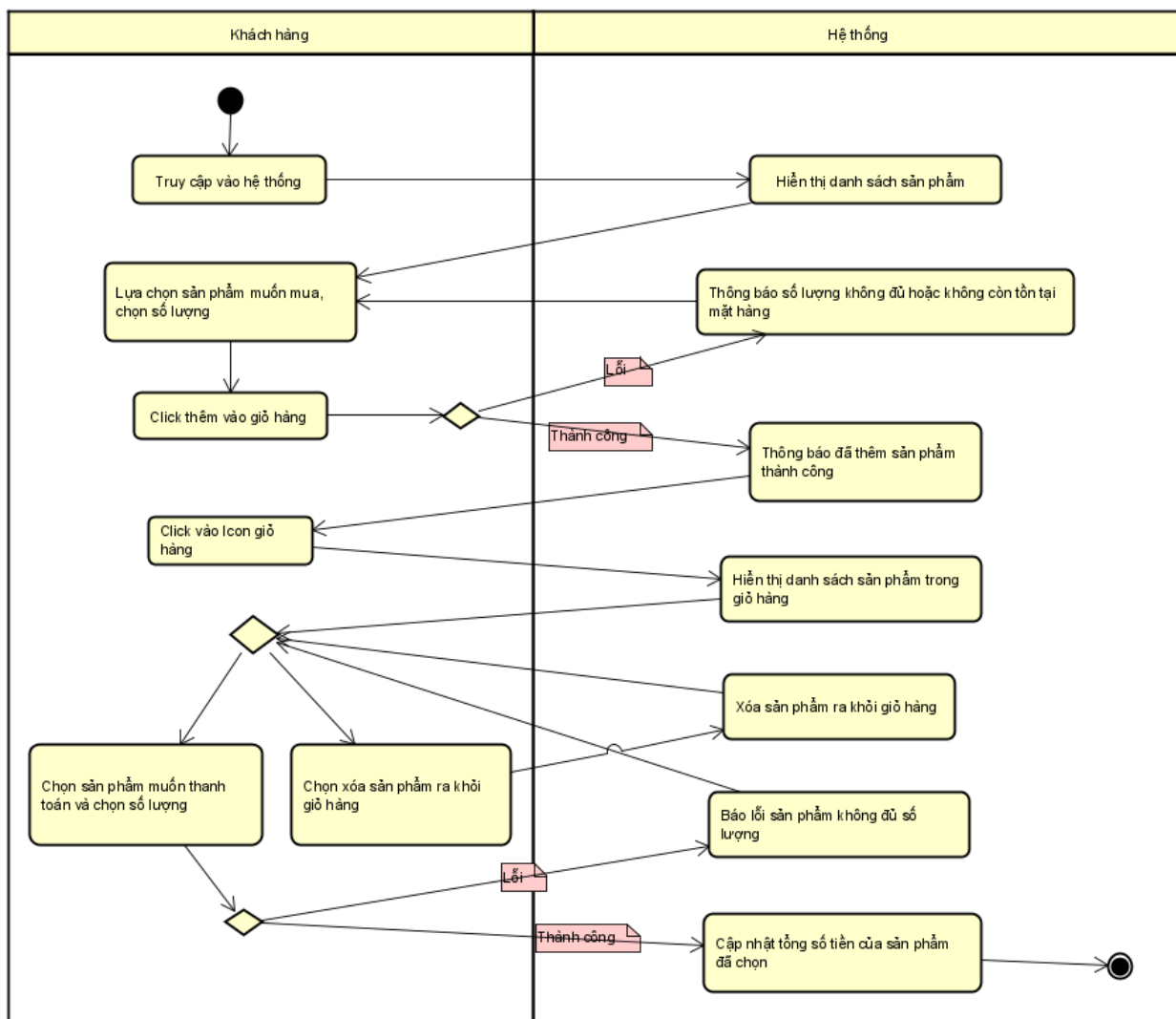
1. Khách hàng truy cập vào hệ thống.
2. Hệ thống hiển thị danh sách và thông tin sản phẩm.
3. Khách hàng chọn sản phẩm muốn mua, chọn số lượng sản phẩm cần mua.
4. Khách hàng click vào nút “Thêm vào giỏ hàng”.
5. Hệ thống hiển thị thông báo đã thêm sản phẩm thành công.

6. Khách hàng click vào giỏ hàng.
7. Hệ thống hiển thị danh sách sản phẩm có trong giỏ hàng.
8. Khách hàng lựa chọn sản phẩm muốn thanh toán.
9. Hệ thống hiển thị số tiền tổng sản phẩm đang chọn.

## 6. Luồng sự kiện thay thế

STT	Vị trí	Điều kiện	Hành động	Vị trí tiếp tục
1	4	Số lượng thêm vào giỏ hàng vượt quá số lượng trong kho	Hệ thống báo lỗi: Số lượng mặt hàng trong kho không đủ	Tiếp tục lại bước 3
2	8	Số lượng mua đang nhiều hơn số lượng trong kho	Hệ thống báo lỗi: Số lượng hàng hiện đang không đủ	Tiếp tục lại bước 8
3	8	Khách hàng chọn loại bỏ sản phẩm ra khỏi giỏ hàng.	Hệ thống loại sản phẩm ra khỏi giỏ hàng, nếu sản phẩm đang được chọn sẽ trừ số tiền của sản phẩm đó	Tiếp tục lại bước 8

## 7. Biểu đồ hoạt động



Hình 6: Biểu đồ hoạt động của Use case "Quản lý giỏ hàng"

## 8. Dữ liệu đầu vào

### Dữ liệu đầu vào của quản lý giỏ hàng

STT	Trường dữ liệu	Mô tả	Bắt buộc?	Điều kiện hợp lệ	Ví dụ
	Không				

## 9. Dữ liệu đầu ra

No	Trường dữ liệu	Mô tả	Điều kiện hợp lệ	Example
1	Tên sản phẩm			DVD Phim Vượt Ngục
2	Số lượng 1 sản phẩm			2



3	Tổng tiền sản phẩm			50.000
4	Thuế			10000
5	Thành tiền			110000

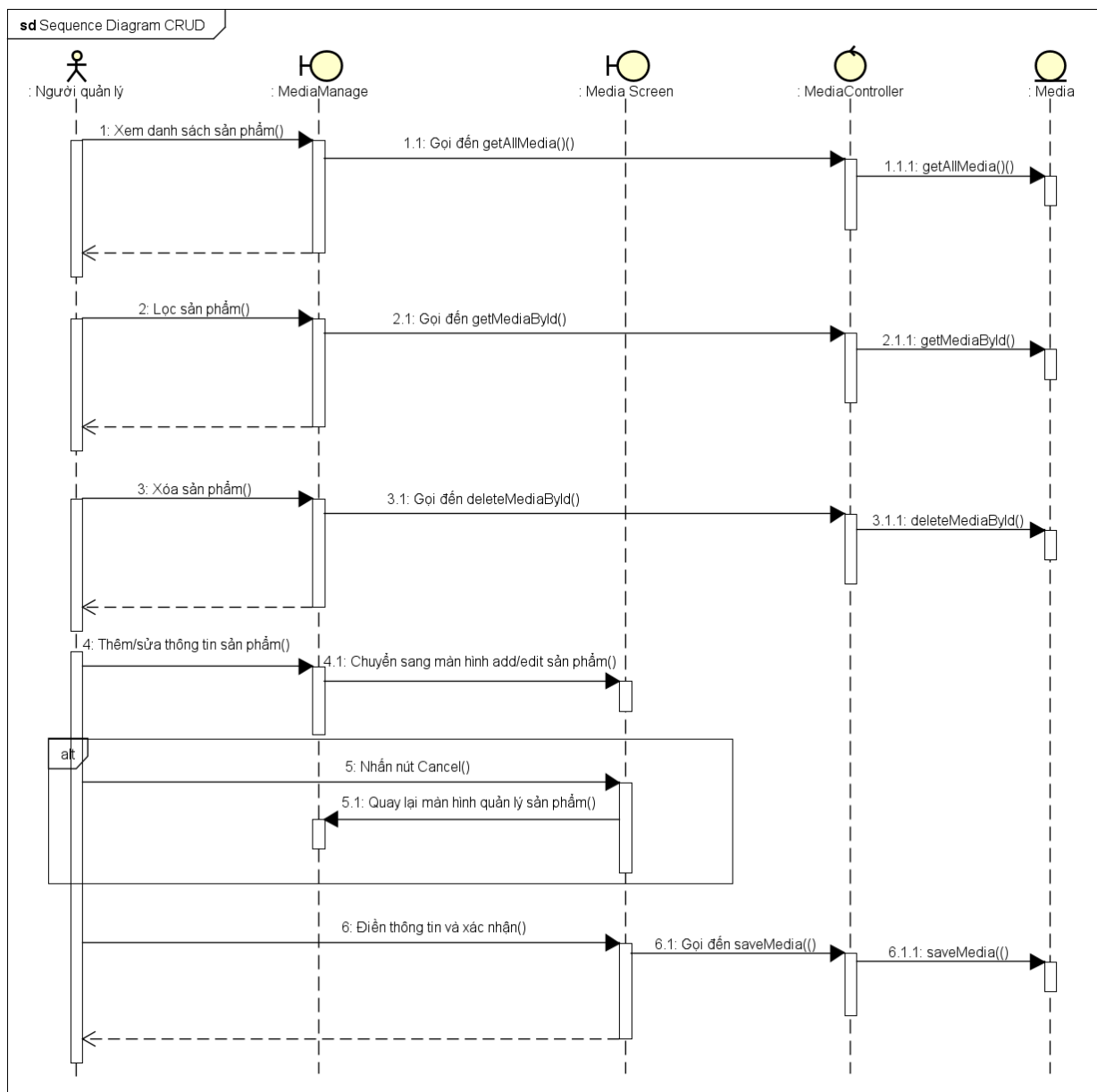
## 10. Hậu điều kiện

Không

## 2 Usecase Analysis

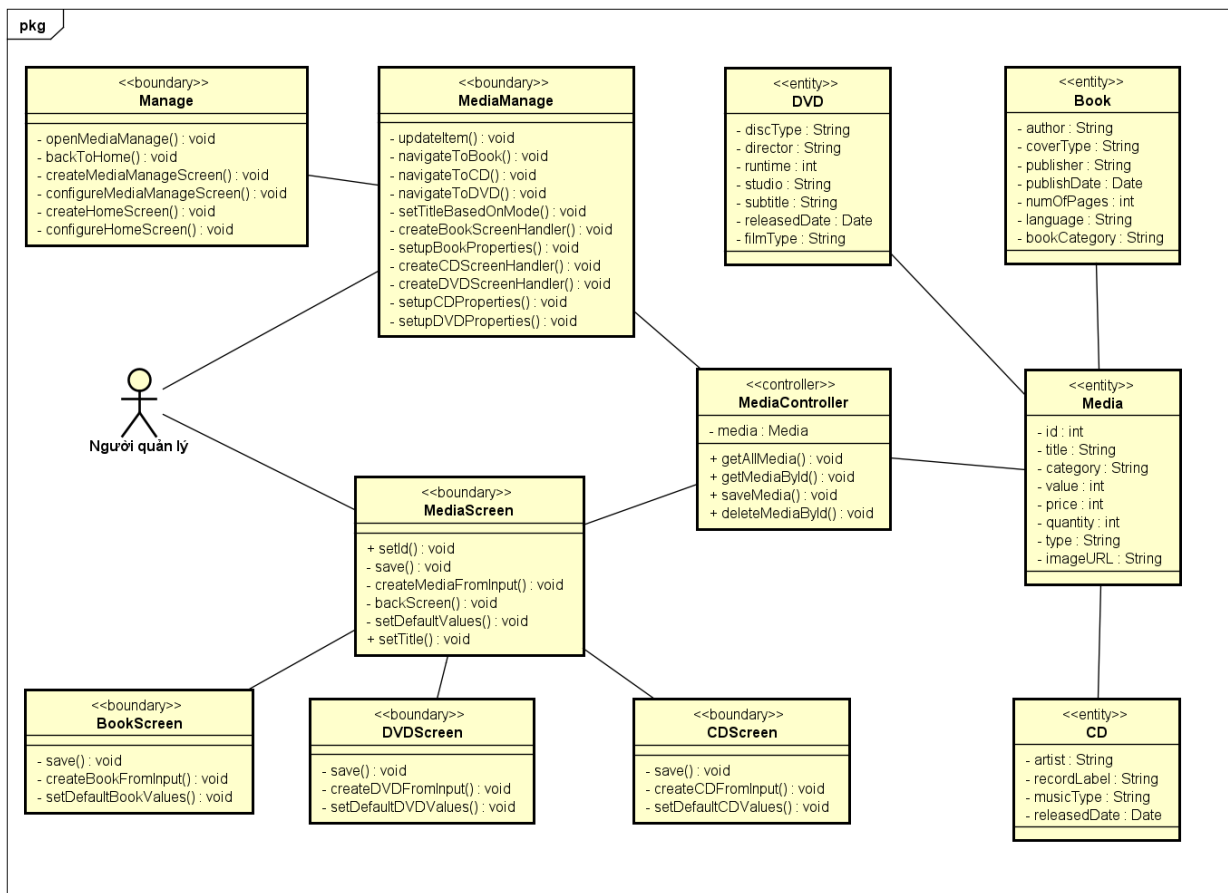
### 2.1 Usecase CRUD sản phẩm

#### 2.1.1 Sequence Diagram



Hình 7: Biểu đồ tuần tự CRUD sản phẩm

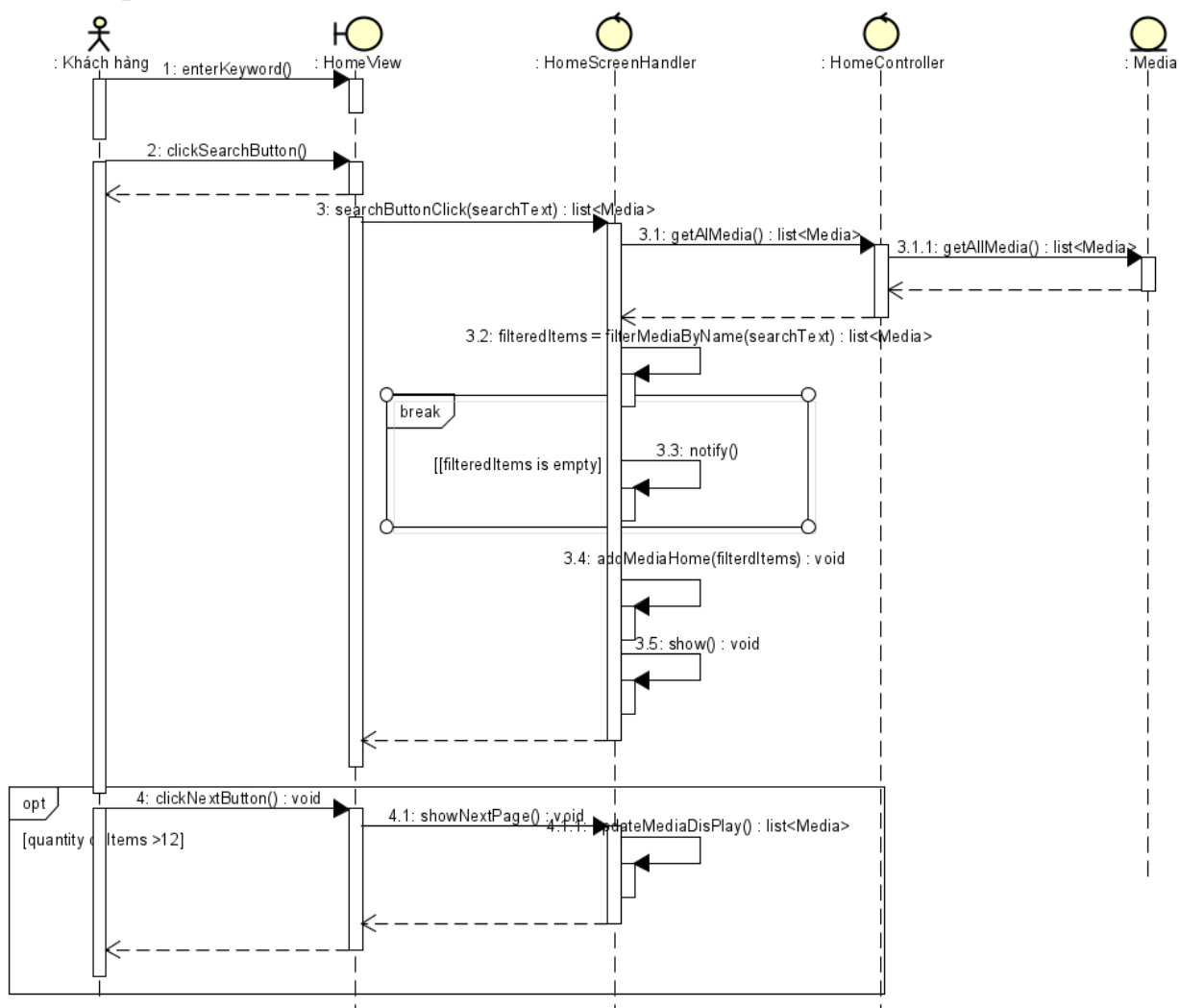
## 2.1.2 Analysis class diagram



Hình 8: Biểu đồ các lớp phân tích CRUD sản phẩm

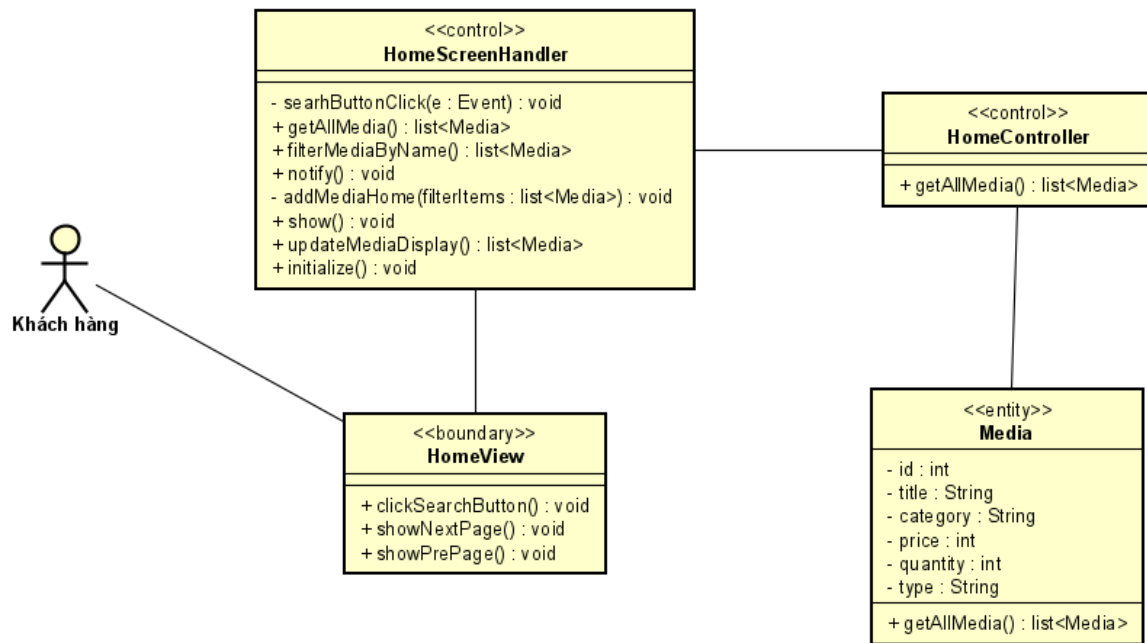
## 2.2 Usecase Tìm kiếm sản phẩm

### 2.2.1 Sequence Diagram



Hình 9: Biểu đồ tuần tự Tìm kiếm sản phẩm

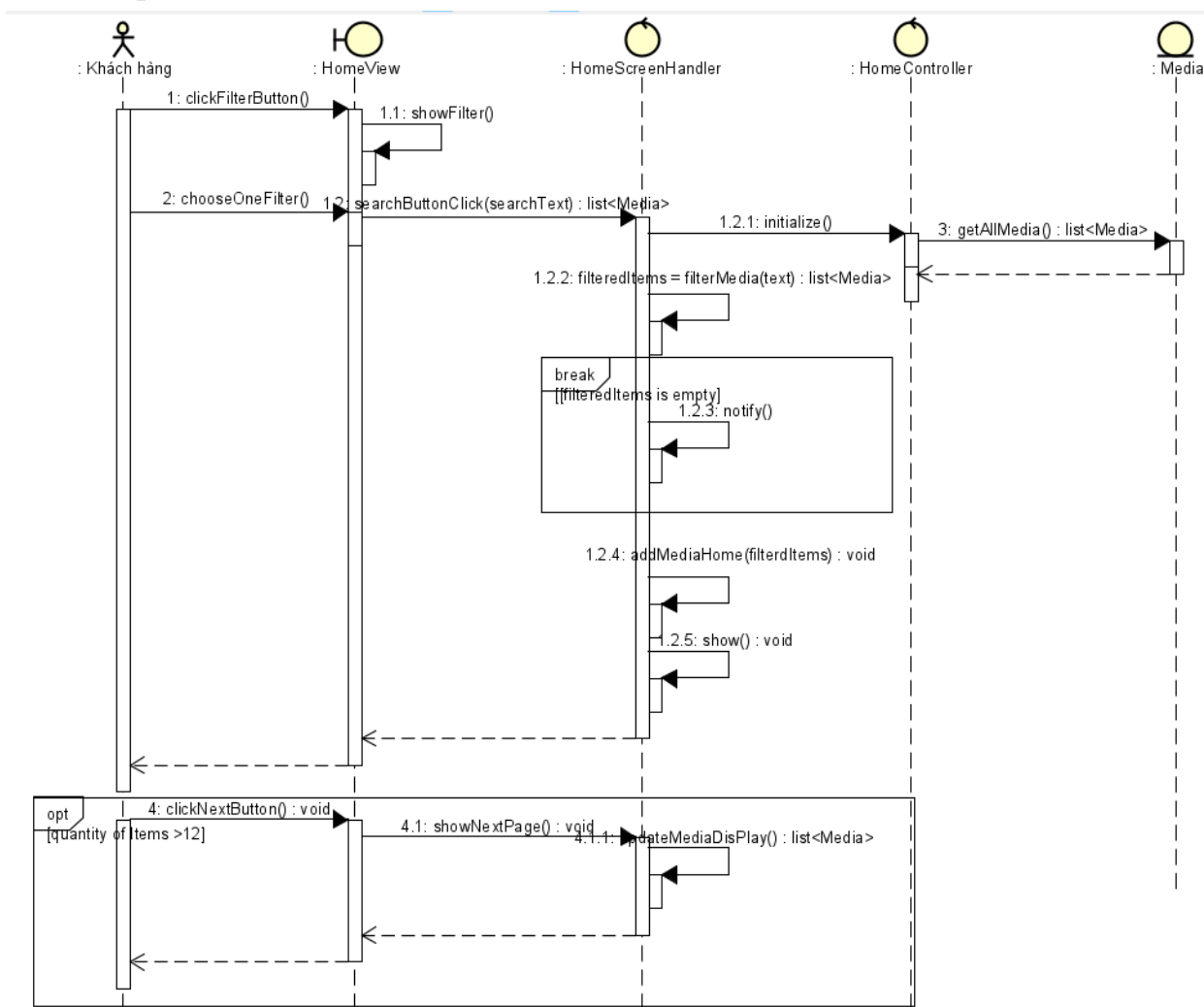
## 2.2.2 Analysis class diagram



Hình 10: Biểu đồ các lớp phân tích Tìm kiếm sản phẩm

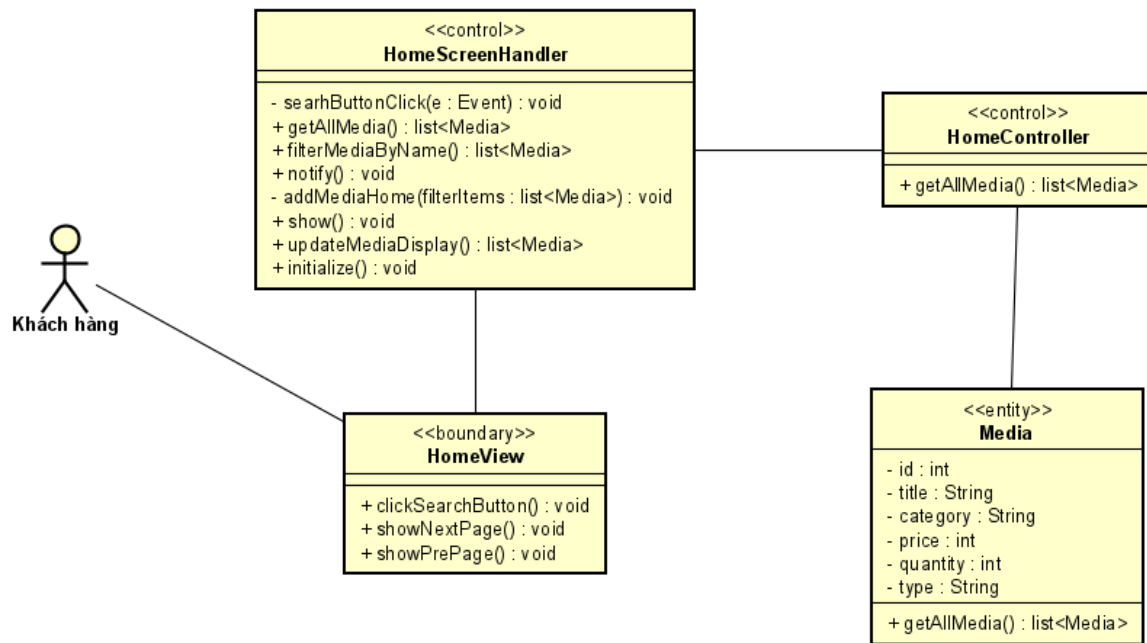
## 2.3 Usecase Lọc sản phẩm

### 2.3.1 Sequence Diagram



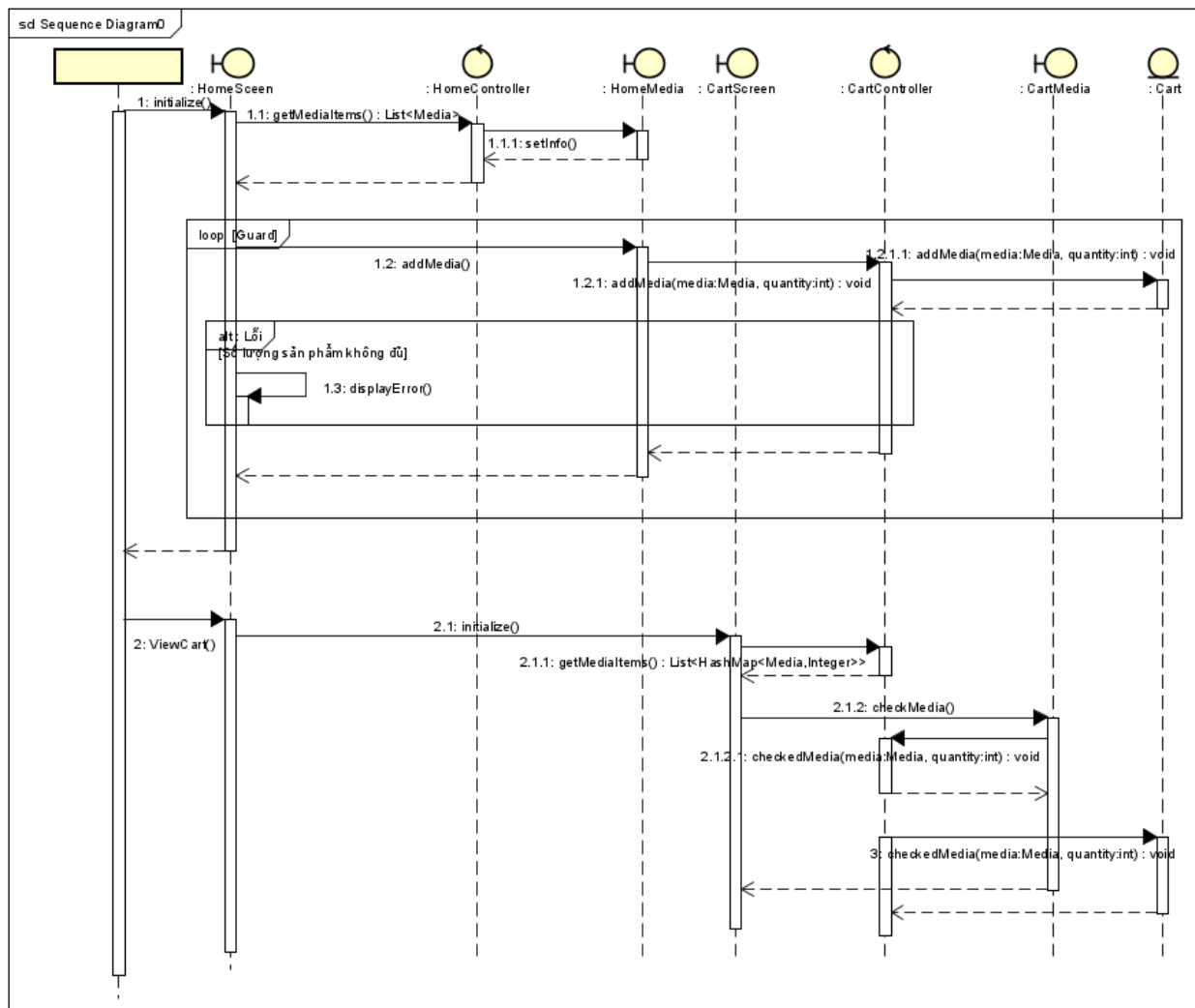
Hình 11: Biểu đồ tuần tự Lọc sản phẩm

### 2.3.2 Analysis class diagram



Hình 12: Biểu đồ các lớp phân tích Lọc sản phẩm

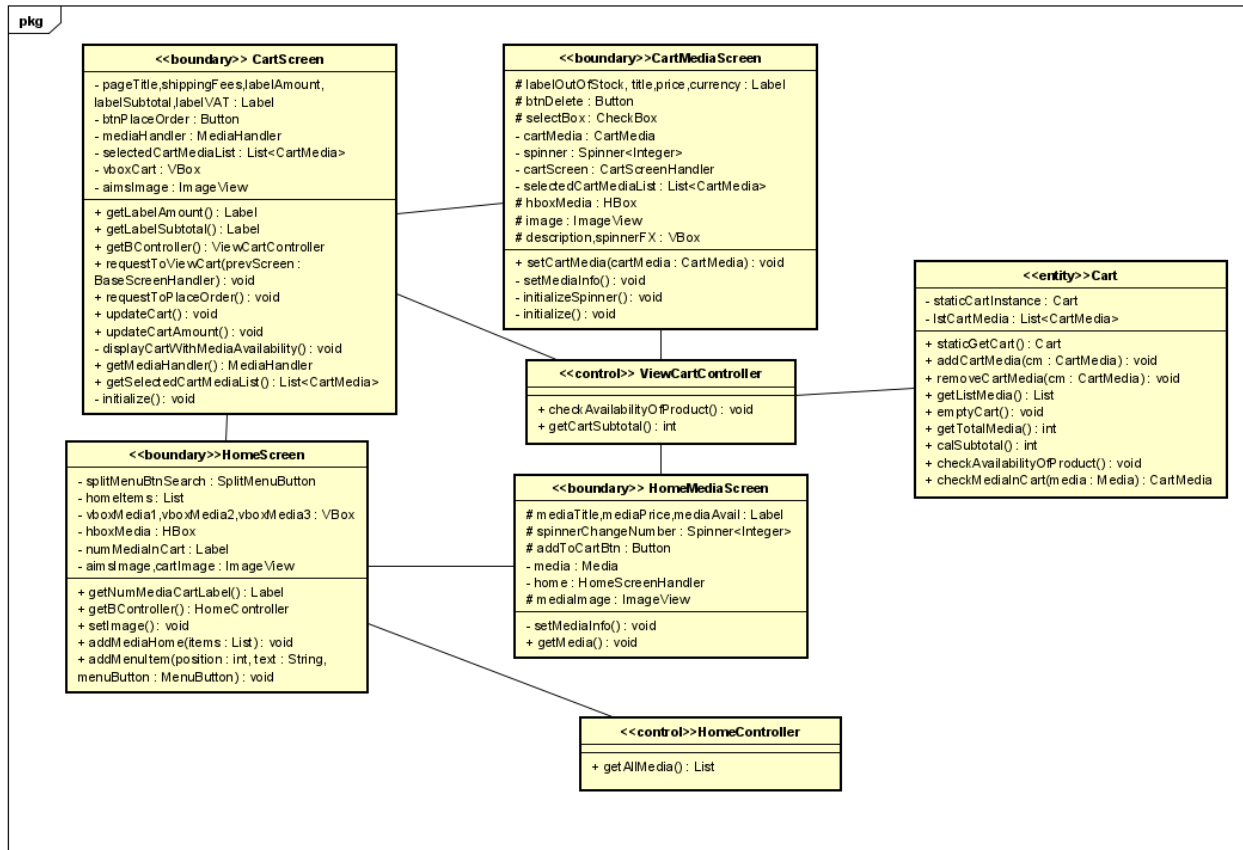
### 2.4.1 Sequence Diagram



Hình 13: Biểu đồ tuần tự Quản lý giỏ hàng



## 2.4.2 Analysis class diagram



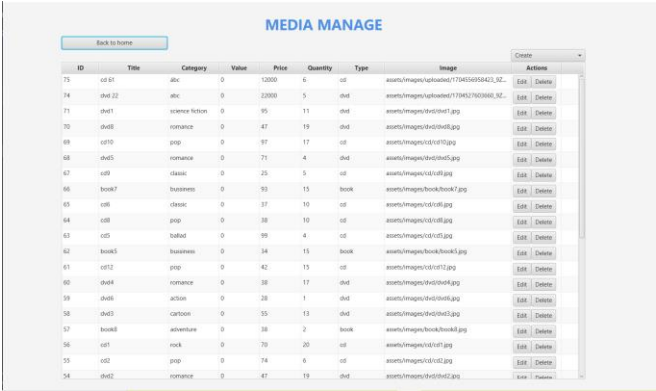
Hình 14: Biểu đồ các lớp phân tích Quản lý giỏ hàng

## 3 Interface Design

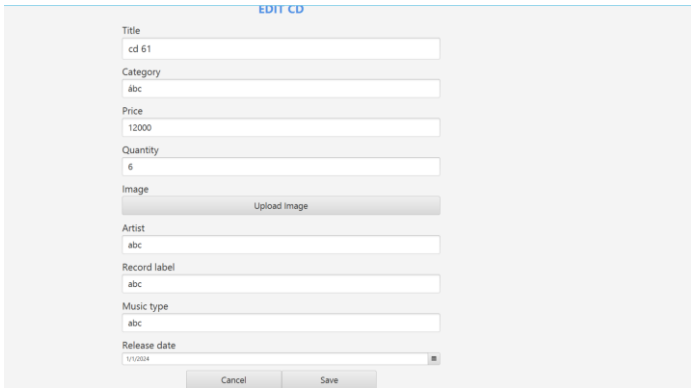
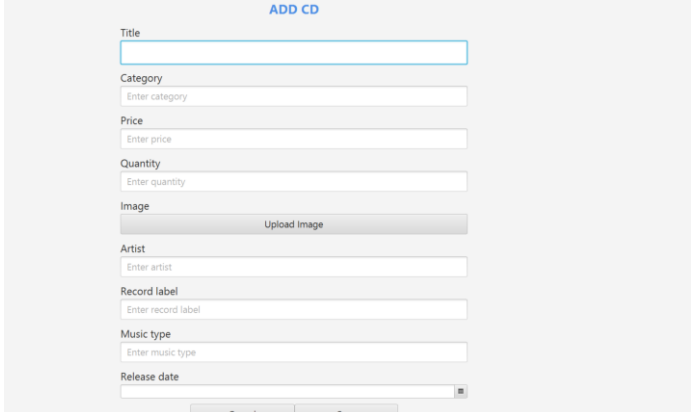
### 3.1 User Interface Design

### 3.1.1. Media Manage Page

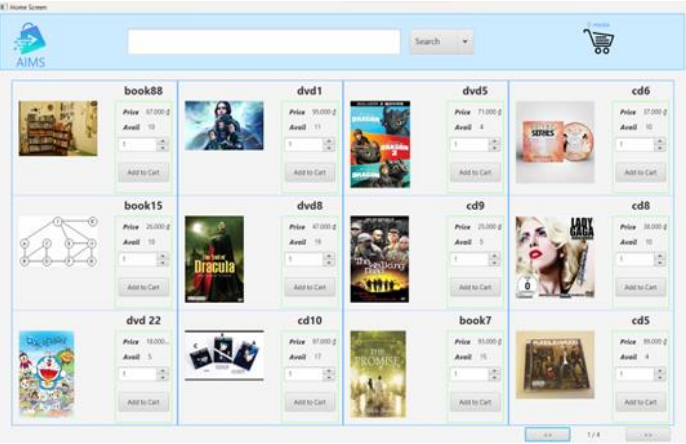
#### - MediaManage

AIMS Software		Date of creation	Approved by	Reviewed by	Person in charge
Screen specification	View MediaManage screen	5/1/2023			Phạm Thị Phương Nga
		Control	Operaton	Function	
		Area for display information item	Initial	Displays information of item	
		Back to home button	Click	Back to Home Page screen	
		Create button	Click	Choose category want to add	
		Delete button	Click	Delete item	
		Edit button	Click	Change information of item	
		ID, Title, Category, Price, Quantity, Type, Image	Click	Arrange follow alphabet	

- MediaScreen

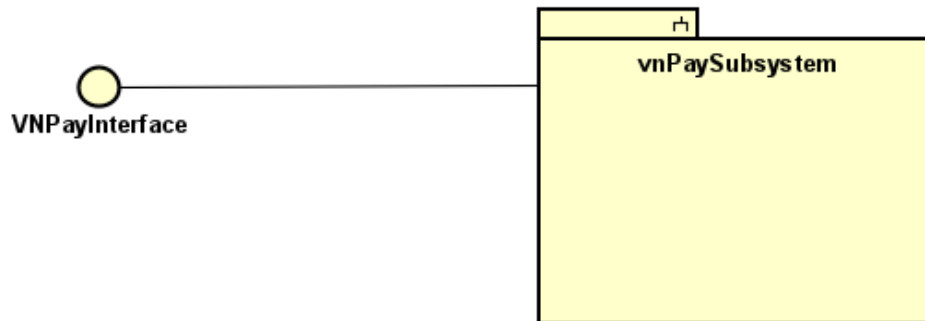
AIMS Software		Date of creation	Approved by	Reviewed by	Person in charge
Screen specification	View Edit/Add Media screen	5/1/2023			Phạm Thị Phương Nga
 		Control	Operation	Function	
		Header	Initial	Displays title with purpose	
		The white lines	Fill	Input of corresponding information	
		Areas Upload image	Click	Upload image in local.	
		Cancel button	Click	Cancel and back before screen	
		Save button	Click	Save the changes	

### 3.1.2. Home page

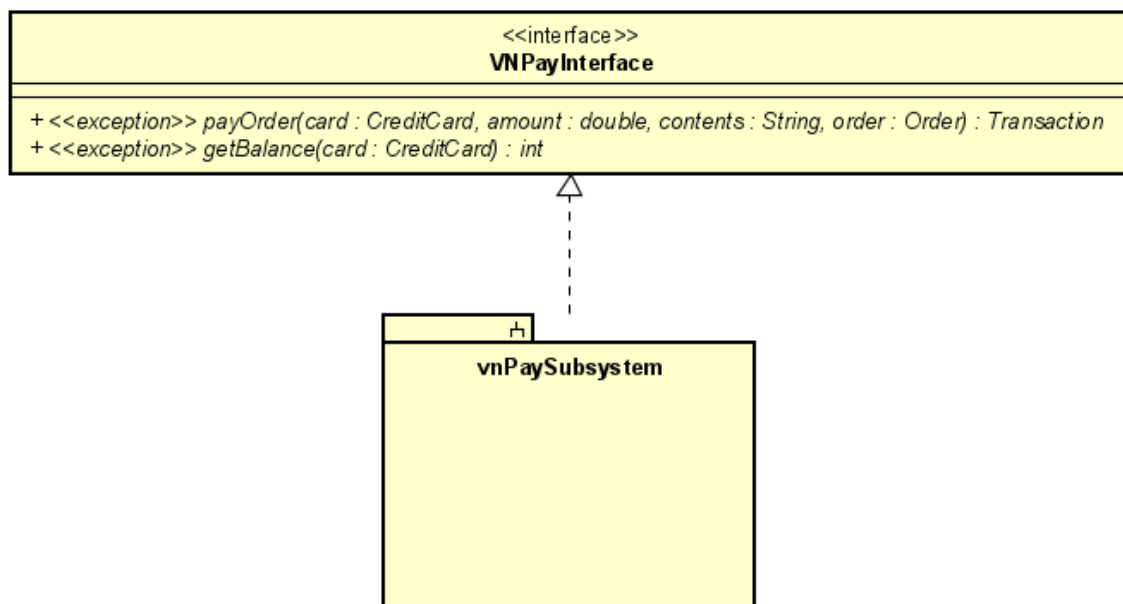
AIMS Software		Date of creation	Approved by	Reviewed by	Person in charge
Screen specification	View Home Page screen	5/1/2023			Phạm Thị Kim Ngân
		Control	Operation	Function	
		Area for display products	Initial	Displays media with the information	
		Search bar	Initial	Search product	
		Add to card button	Click	Add a product to the card	
		Cart button	Click	Display Card Page	
		AIMS logo	Click	Display items manage page	
		Pre/next button	Click	See others items	

## 3.2 System Interface Design

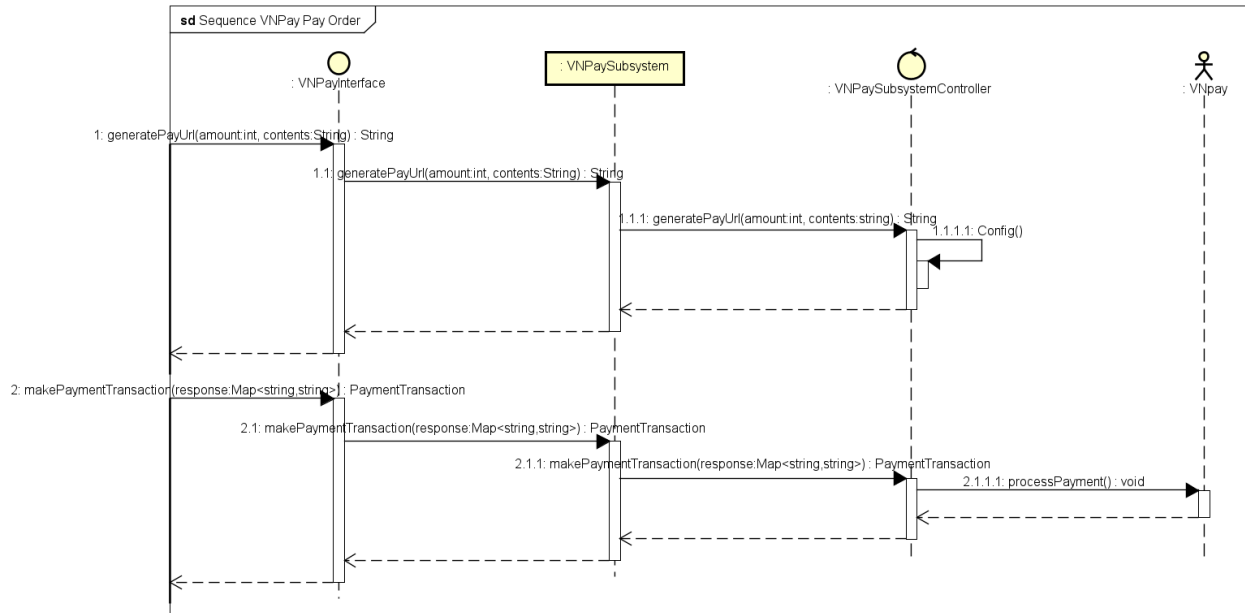
### 4.2.1. Identify subsystems



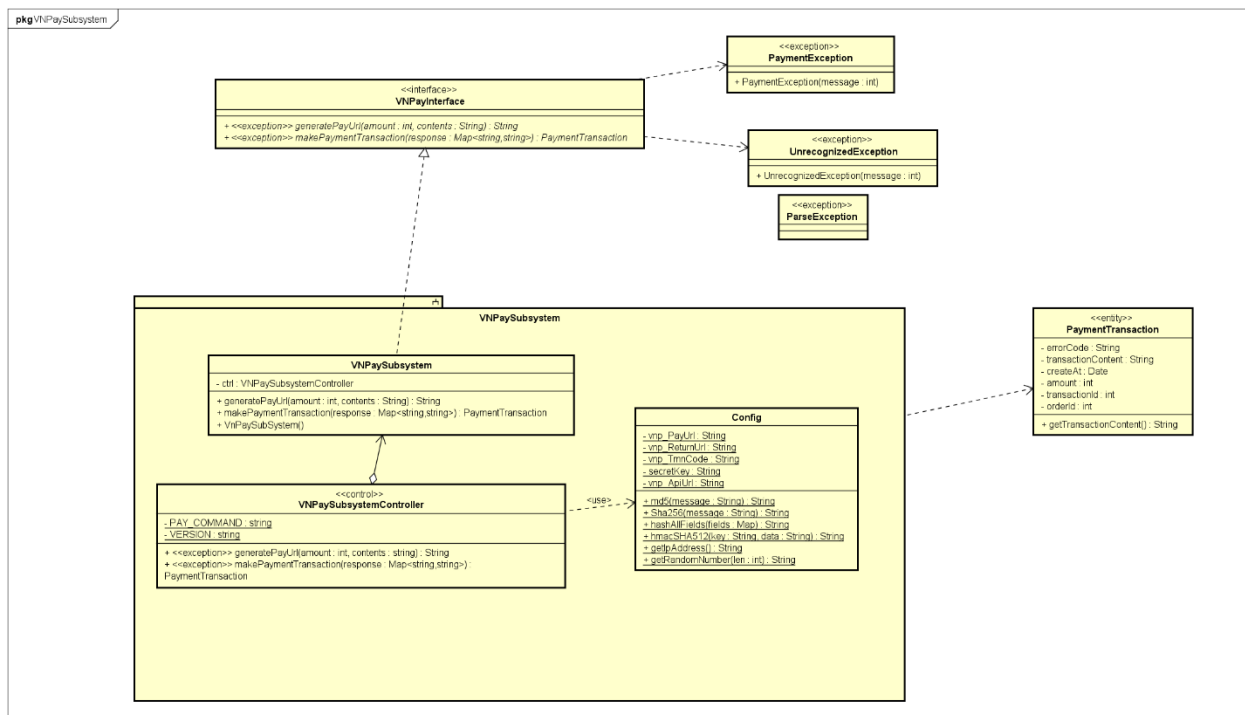
### 4.2.2 Identify subsystem interface



### 4.2.3 Subsystem design



Hình 15: Subsystem Interaction



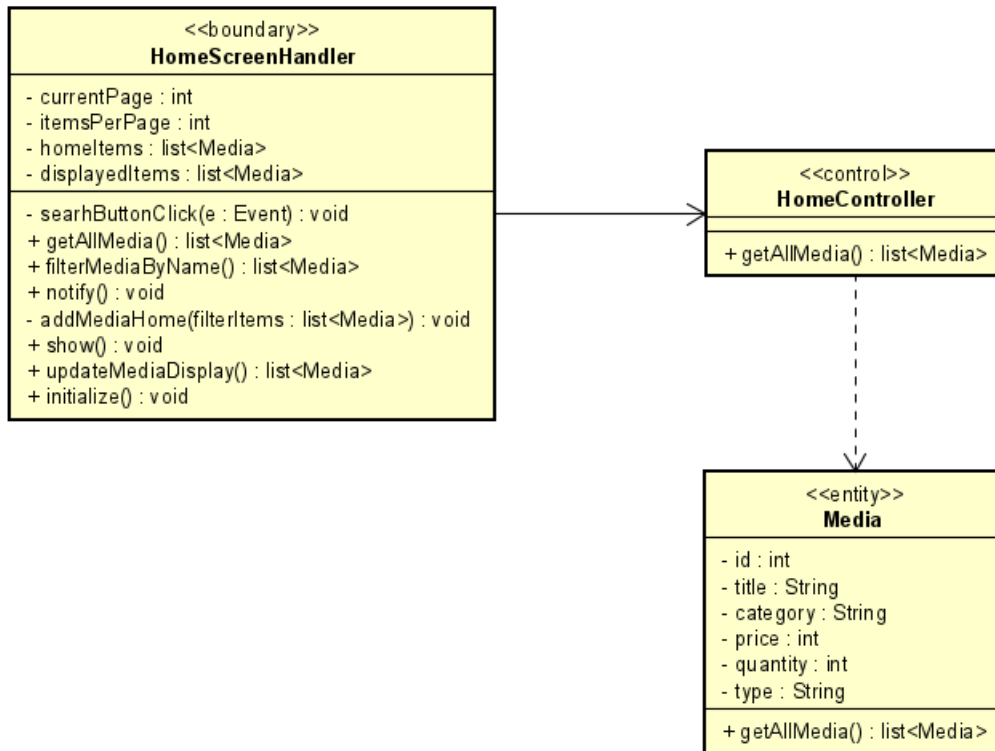
Hình 16: Subsystem class design

## 4 Class Design

### 4.1 General class diagram

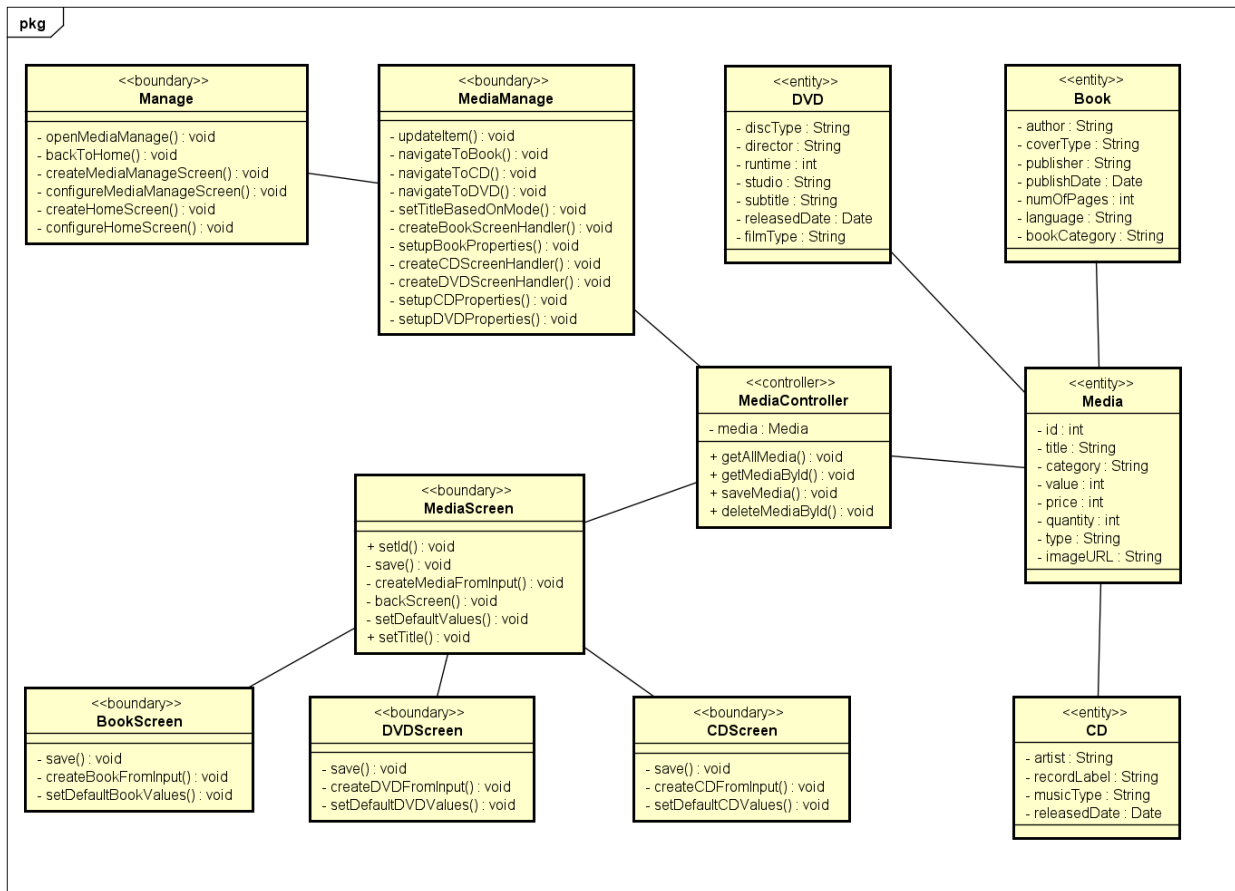
### 4.2 Detail class diagrams

#### 4.2.1. Usecase Tìm kiếm, lọc sản phẩm



Hình 16: Detail class diagrams Tìm kiếm, Lọc sản phẩm

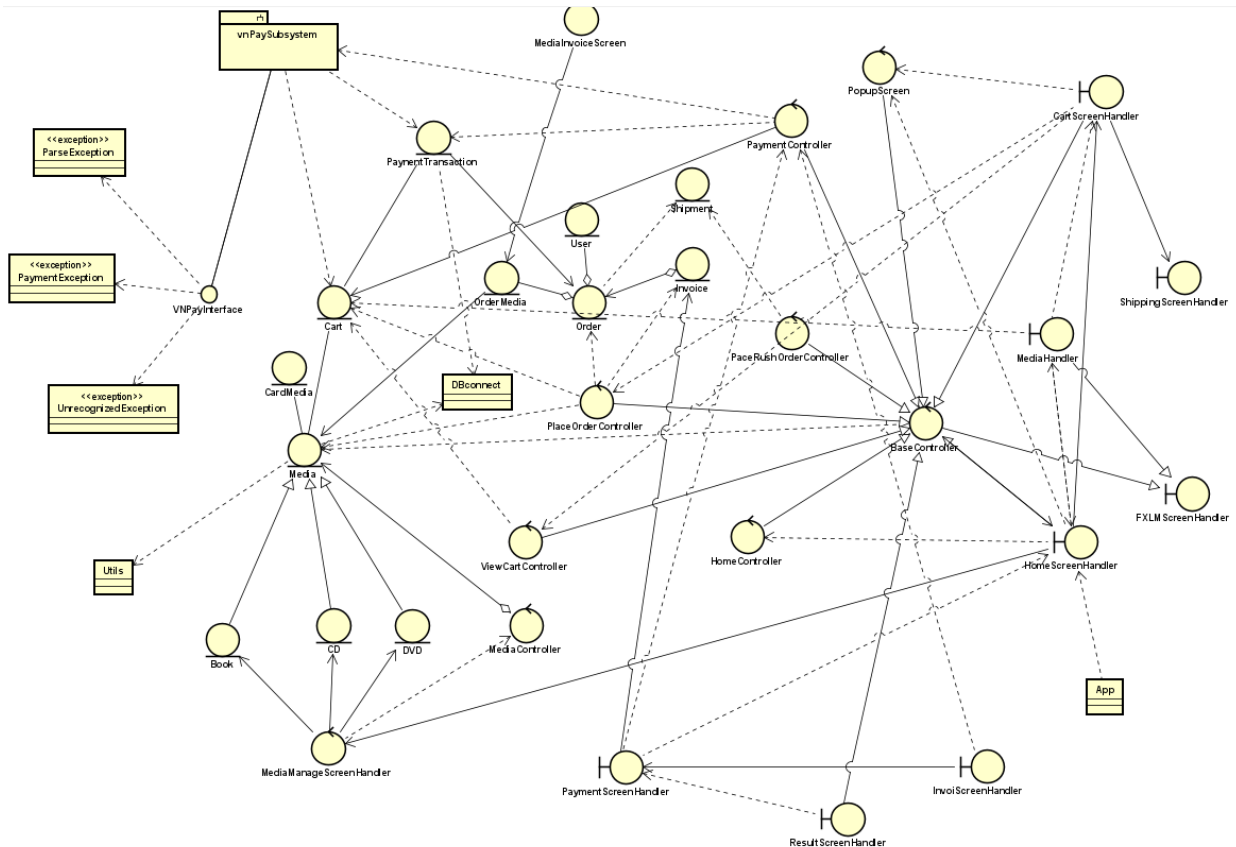
#### 4.2.2. Usecase CRUD sản phẩm



Hình 17: Detail class diagrams CRUD sản phẩm



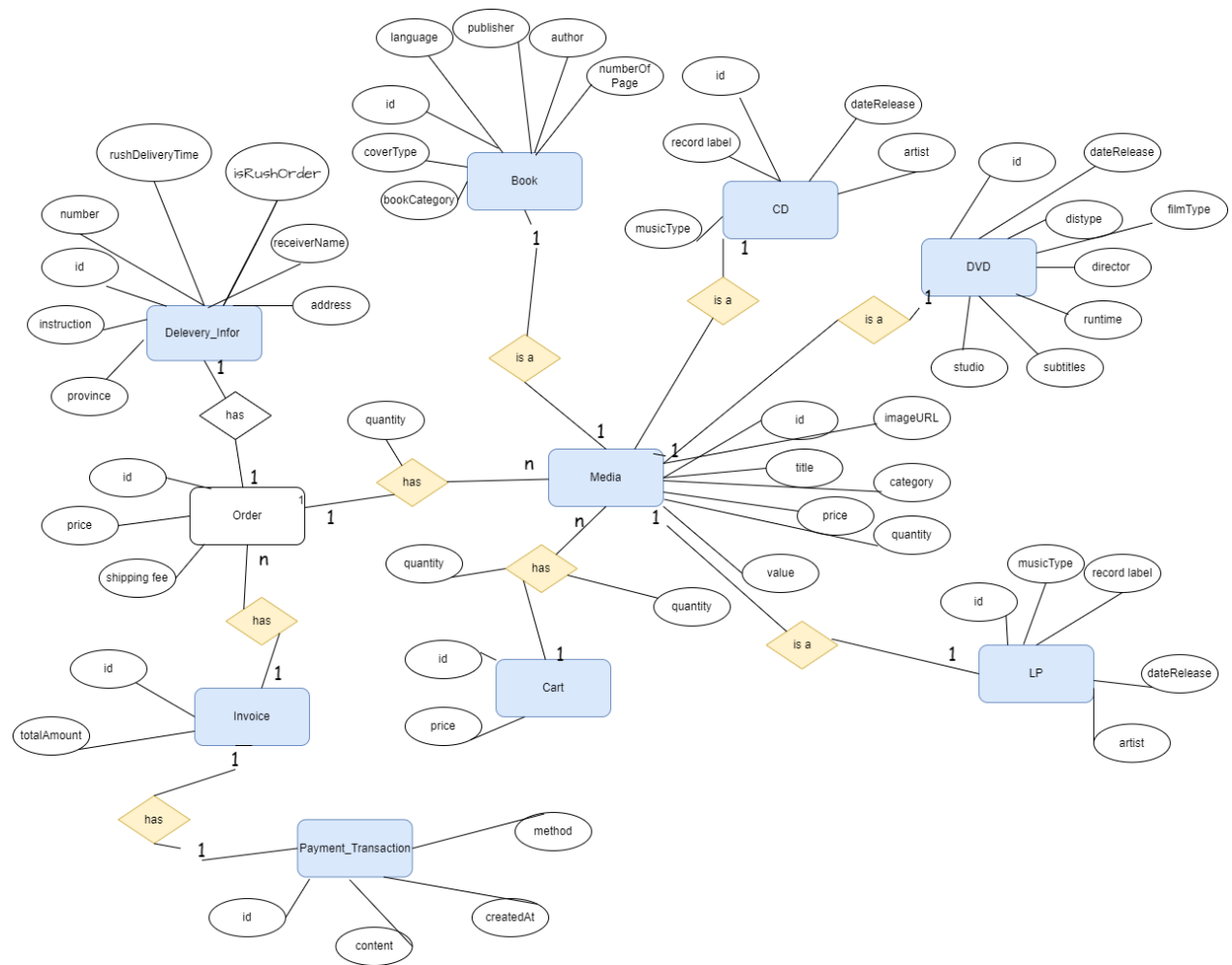
### 4.3 Relationship class diagram



Hình 18: Relationship class diagram

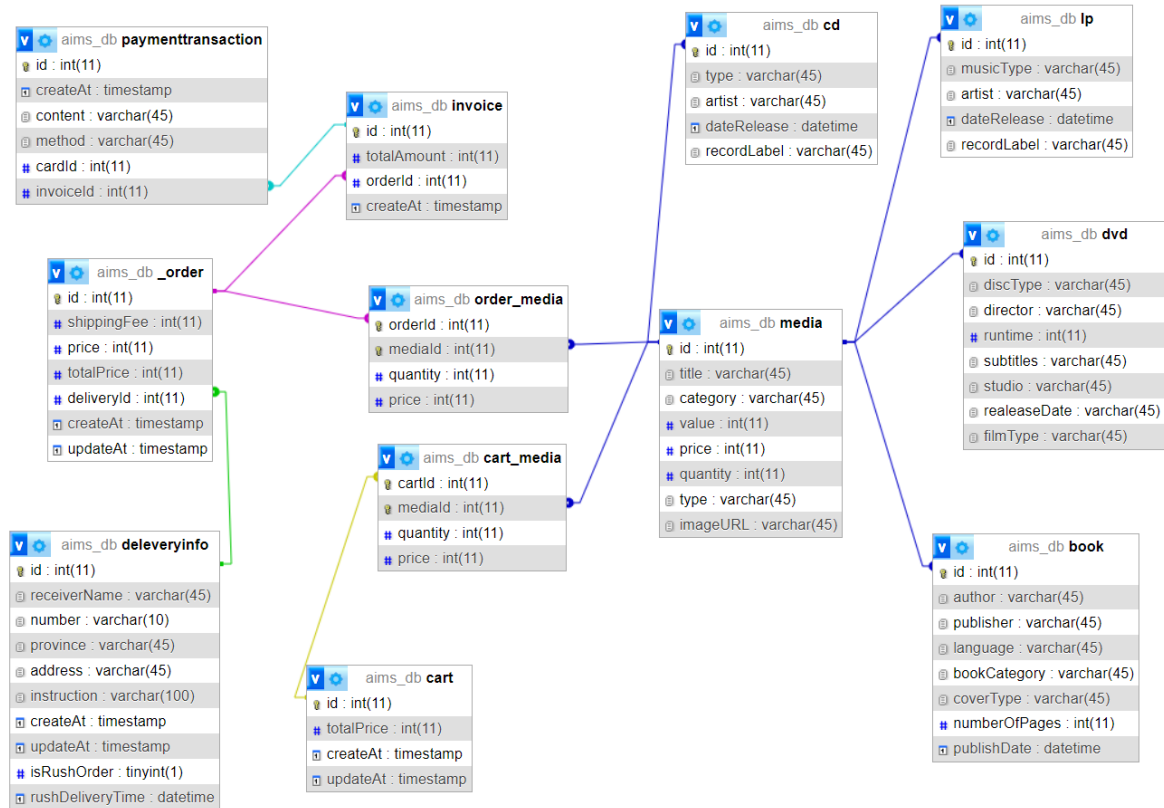
## 5 Data Modeling

### 5.1 Conceptual data model



## 5.2 Database Design

### 5.2.1 Logical Data Model



### 5.2.2 Physical Data Model

#### - Media

#	PK	FK	Column Name	Data type	Mandatory	Description
1.	x		id	int	yes	ID, auto increment
2.			title	Varchar(45)	yes	Product's name
3.			category	Varchar(45)	yes	Media type, eg., cd, DVD
4.			value	int	yes	Value of the product
5.			price	int	yes	Current price
6.			quantity	int	yes	Number of products
7.			type	Varchar(45)	yes	type
8.			imageUrl	Varchar(45)	yes	Product image path

#### - CD

#	PK	FK	Column Name	Data type	Mandatory	Description
1.		x	id	int	yes	ID, same as ID of Media of which type is CD

2.			musicType	Varchar(45)	yes	Music genres
3.			artist	Varchar(45)	yes	Artist's name
4.			releaseDate	datetime	No	Release date
5.			recordLabel	Varchar(45)	yes	Record label

- LD

#	PK	FK	Column Name	Data type	Mandatory	Description
1.		x	id	int	yes	ID, same as ID of Media of which type is CD
2.			musicType	Varchar(45)	yes	Music genres
3.			artist	Varchar(45)	yes	Artist's name
4.			releaseDate	datetime	No	Release date
5.			recordLabel	Varchar(45)	yes	Record label

- Book

#	PK	FK	Column Name	Data type	Mandatory	Description
1.		x	id	int	yes	ID, same as ID of Media of which type is Book
2.			author	Varchar(45)	yes	Authors of the book
3.			publisher	Varchar(45)	yes	Publishing house
4.			language	Varchar(45)	yes	Language
5.			bookCategory	Varchar(45)	yes	Book type
6.			coverType	Varchar(45)	yes	Book cover
7.			numOfPages	int	yes	Page number
8.			publishDate	datetime	yes	Date of publishing

- DVD

#	PK	FK	Column Name	Data type	Mandatory	Description
1.		x	id	int	yes	ID, same as ID of Media of which type is DVD
2.			discType	VARCHAR(45)	yes	Disc type
3.			director	VARCHAR(45)	Yes	Director
4.			runtime	int	Yes	Duration
5.			subtitles	VARCHAR(45)	Yes	Subtitles
6.			studio	VARCHAR(45)	yes	Manufacturer

7.			releaseDate	Datetime	Yes	Release date
8.			filmType	VARCHAR(45)	yes	Genres

- **Cart**

#	PK	FK	Column Name	Data type	Mandatory	Description
1.	x		id	int	yes	ID, auto increment
2.			totalPrice	int	yes	Total price
3.			createAt	timestamp	yes	The time the first product is added to the card
4.			updateAt	timestamp	yes	The last time product is update to the card

- **Cart\_Media**

#	PK	FK	Column Name	Data type	Mandatory	Description
5.		x	cardId	int	yes	Card ID
6.		x	mediaId	int	yes	same as ID of Media
7.			quantity	int	yes	Number of a product
8.			price	int	yes	Price of a product

- **DeliveryInfo**

#	PK	FK	Column Name	Data type	Mandatory	Description
1.	x		id	int	yes	ID, auto increment
2.			receiverName	Varchar(45)	yes	Receiver name
3.			number	Varchar(10)	yes	Receiver phone number
4.			province	Varchar(45)	yes	Provinces
5.			address	Varchar(45)	yes	Delivery address
6.			instruction	Varchar(100)	yes	Delivery instructions
7.			createAt	timestamp	yes	
8.			updateAt	timestamp	yes	
9.			isRushOrder	boolean	yes	rush delivery method
10.			rushDeliveryTime		no	

- **Order**

#	PK	FK	Column Name	Data type	Mandatory	Description
1.	x		id	int	yes	Id, auto increment

2.			shippingFee	int	yes	Shipping Fee
3.			price	int	yes	Selling price
4.			totalPrice	int	yes	Selling price + VAT
5.		x	deliveryId	int	yes	Delivery Info ID
6.			createAt	timestamp	yes	
7.			updateAt	timestamp	yes	

- **Order\_Media**

#	PK	FK	Column Name	Data type	Mandatory	Description
1.		x	orderId	int	yes	Order ID
2.		x	mediaId	int	yes	Media ID
3.			quantity	int	yes	Number
4.			price	int	yes	Selling price

- **Invoice**

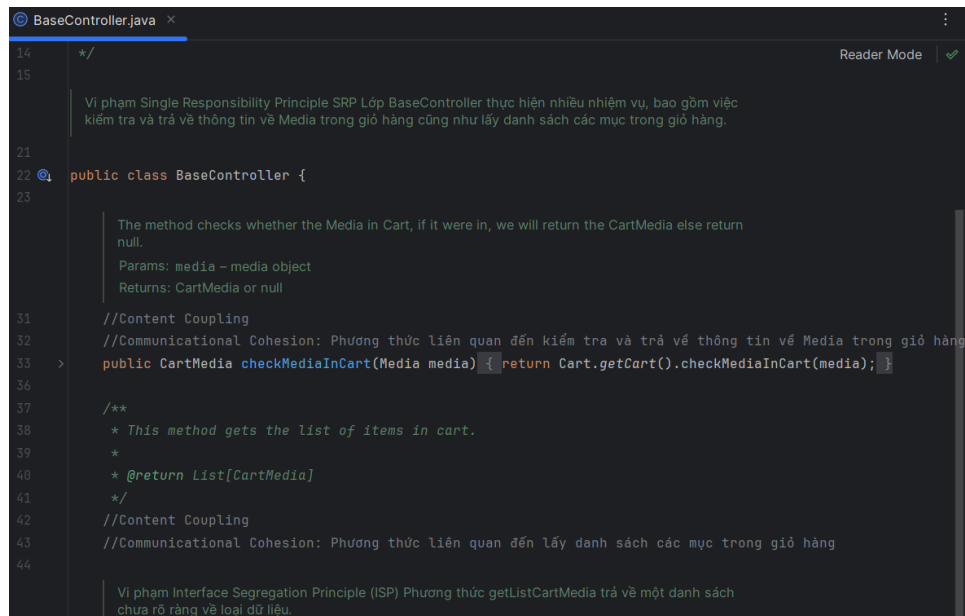
#	PK	FK	Column Name	Data type	Mandatory	Description
1.	x		id	int	yes	ID
2.			totalAmount	int	yes	Total
3.		x	orderId	int	yes	Order ID
4.			createAt	timestamp	yes	The moment create invoice

- **PaymentTransaction**

#	PK	FK	Column Name	Data type	Mandatory	Description
5.	x		id	int	yes	ID
6.			createAt	timestamp	yes	Date of creation
7.			content	Varchar(45)	yes	Transaction contents
8.			method	Varchar(45)	yes	Payment methods
9.		x	cardId	int	yes	ID of used card
10.		x	invoiceId	int	yes	Invoice ID

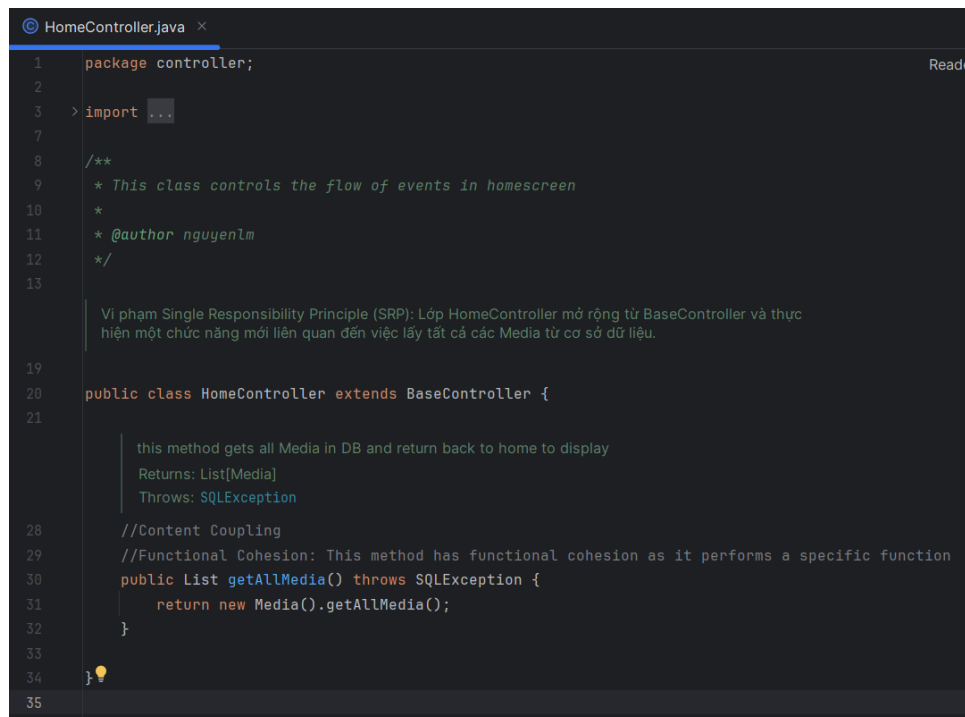
## 6 Good Design

### 6.1 Cohesion



```
14  */
15
16  Vi phạm Single Responsibility Principle SRP Lớp BaseController thực hiện nhiều nhiệm vụ, bao gồm việc
17  kiểm tra và trả về thông tin về Media trong giỏ hàng cũng như lấy danh sách các mục trong giỏ hàng.
18
19
20
21
22  @ public class BaseController {
23
24
25
26
27
28
29
30
31
32
33  > public CartMedia checkMediaInCart(Media media) { return Cart.getCart().checkMediaInCart(media); }
34
35
36
37
38  /**
39   * This method gets the list of items in cart.
40   *
41   * @return List[CartMedia]
42   */
43
44  //Content Coupling
45  //Communicational Cohesion: Phương thức liên quan đến kiểm tra và trả về thông tin về Media trong giỏ hàng
46
47  //Content Coupling
48  //Communicational Cohesion: Phương thức liên quan đến lấy danh sách các mục trong giỏ hàng
49
50
51
52  Vi phạm Interface Segregation Principle (ISP) Phương thức getListCartMedia trả về một danh sách
53  chưa rõ ràng về loại dữ liệu.
```

Hình 19: Cohesion at BaseController



```
1  package controller;
2
3  > import ...
4
5
6
7
8  /**
9   * This class controls the flow of events in homescreen
10   *
11   * @author nguyentlm
12   */
13
14
15  Vi phạm Single Responsibility Principle (SRP): Lớp HomeController mở rộng từ BaseController và thực
16  hiện một chức năng mới liên quan đến việc lấy tất cả các Media từ cơ sở dữ liệu.
17
18
19
20  public class HomeController extends BaseController {
21
22
23
24
25
26
27
28
29
30
31
32
33
34  }
35
```

Hình 20: Cohesion at HomeController

- Nga: Phân tích cohesion có trong file PlaceRushOrderController.java và ViewCartController.java

```

1  package controller;
2
3  > import ...
4
5
6
7
8  /**
9   * This class controls the flow of place rush order usecase in our AIMS project
10  */
11
12  Functional cohesion vì thực hiện một chức năng cụ thể là kiểm tra và xác nhận đơn hàng cần giao hàng
13  nhanh Communicational cohesion do có sự truyền đối tượng Shipment làm tham số trong phương thức
14  validatePlaceRushOrderData. Thành phần này cung cấp và sử dụng dữ liệu thông qua tham số.
15
16
17  public class PlaceRushOrderController extends BaseController {
18      Just for logging purpose
19
20
21      private static Logger LOGGER = utils.Utils.getLogger(PlaceRushOrderController.class.getName());
22

```

Hình 21: Cohesion at PlaceRushOrderController

```

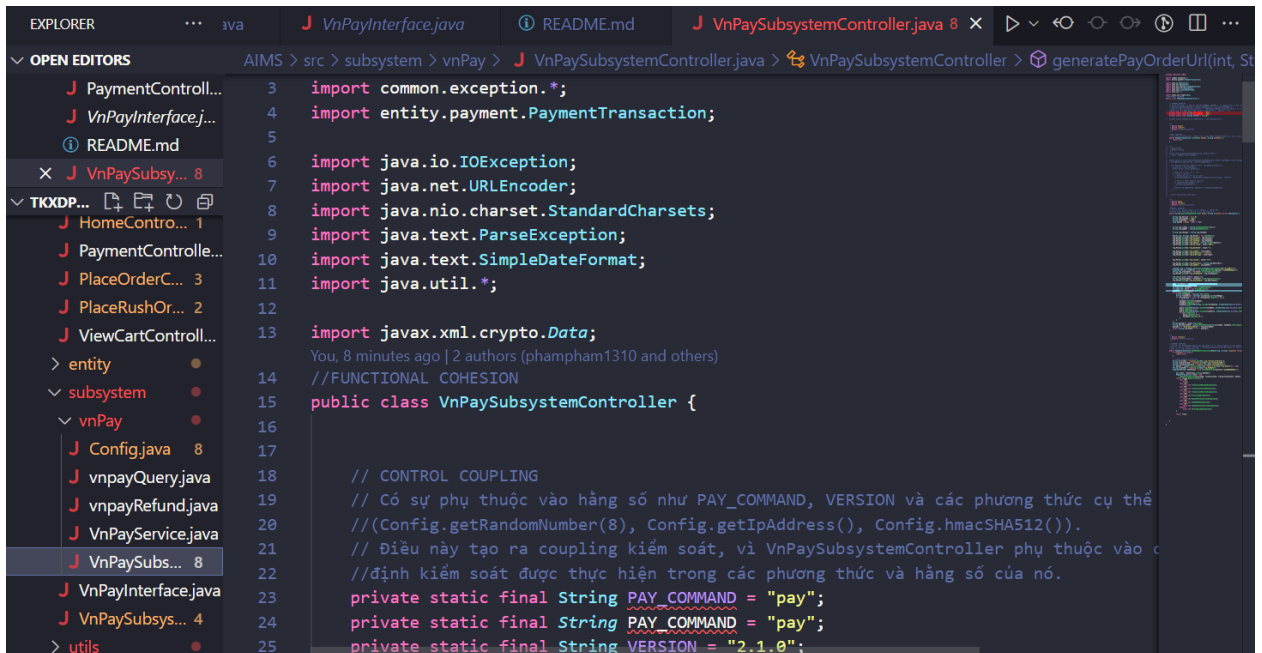
1  package controller;
2
3  > import ...
4
5
6
7  /**
8   * This class controls the flow of events when users view the Cart
9   *
10  */
11
12  Functional cohesion vì thực hiện một chức năng cụ thể liên quan đến xem giỏ hàng. Communicational
13  cohesion do có sự truyền đối tượng Cart làm tham số trong các phương thức.
14
15
16  public class ViewCartController extends BaseController {
17      /*
18       * SOLID: Không vi phạm nguyên tắc nào
19       */
20
21      /**
22       * This method checks the available products in Cart
23       *
24       * @throws SQLException

```

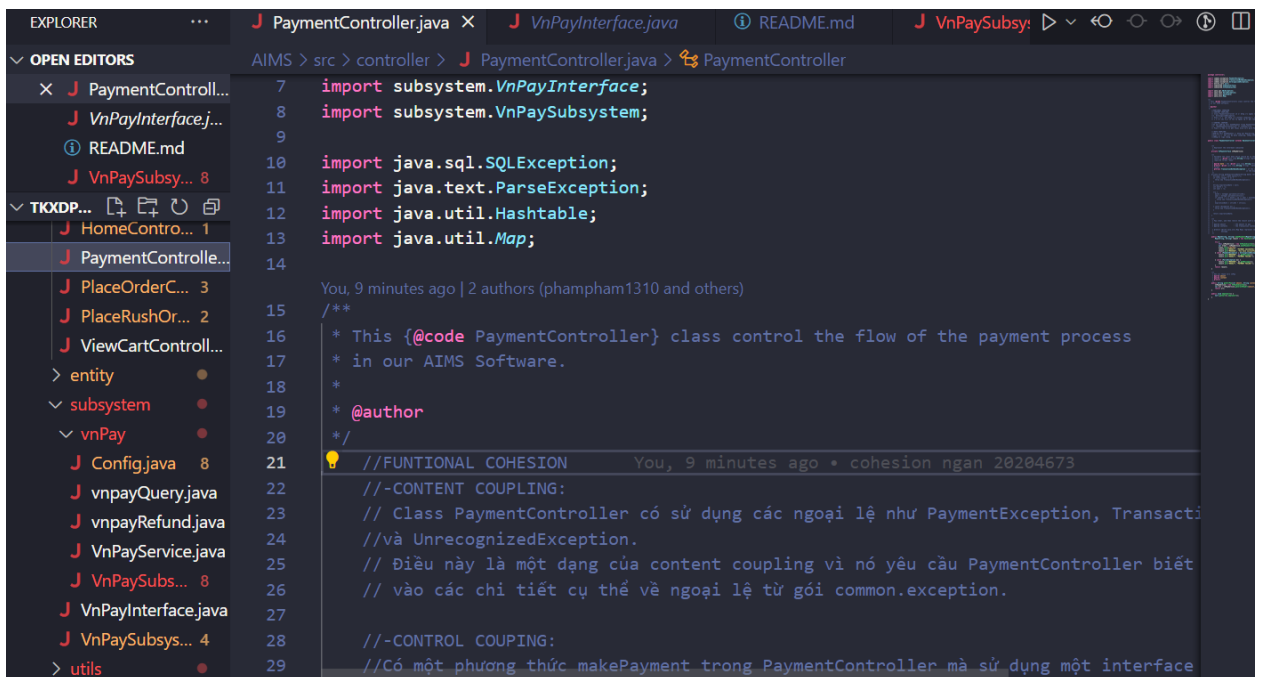
Hình 22: Cohesion at ViewCartController

- Ngân: Phân tích cohesion có trong file PaymentController.java, VnPaySubsystemController.java





Hình 23: Cohesion at VnPaySubsystemController.java

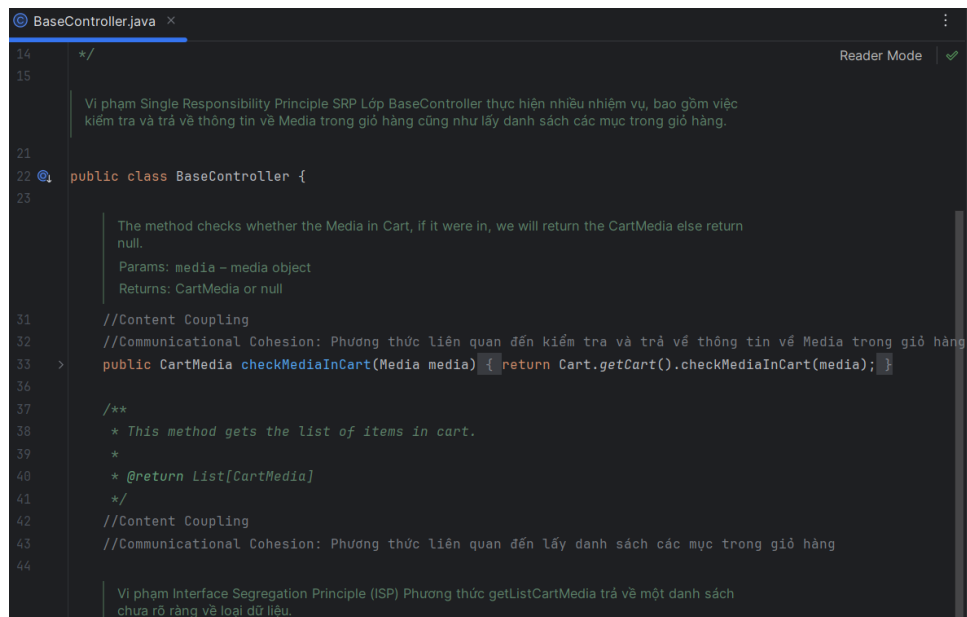


Hình 24: Cohesion at PaymentController.java

## 6.2 Coupling

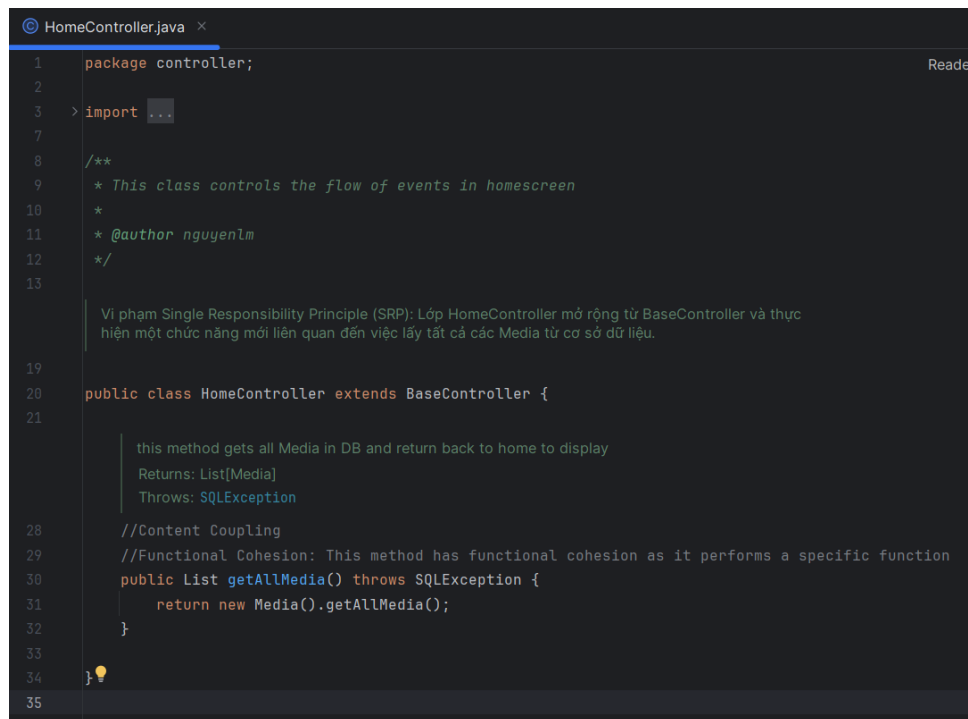
- Nga: Phát hiện coupling với file BaseController.java,

## HomeController.java, PlaceRushOrderController.java, ViewCartController.java



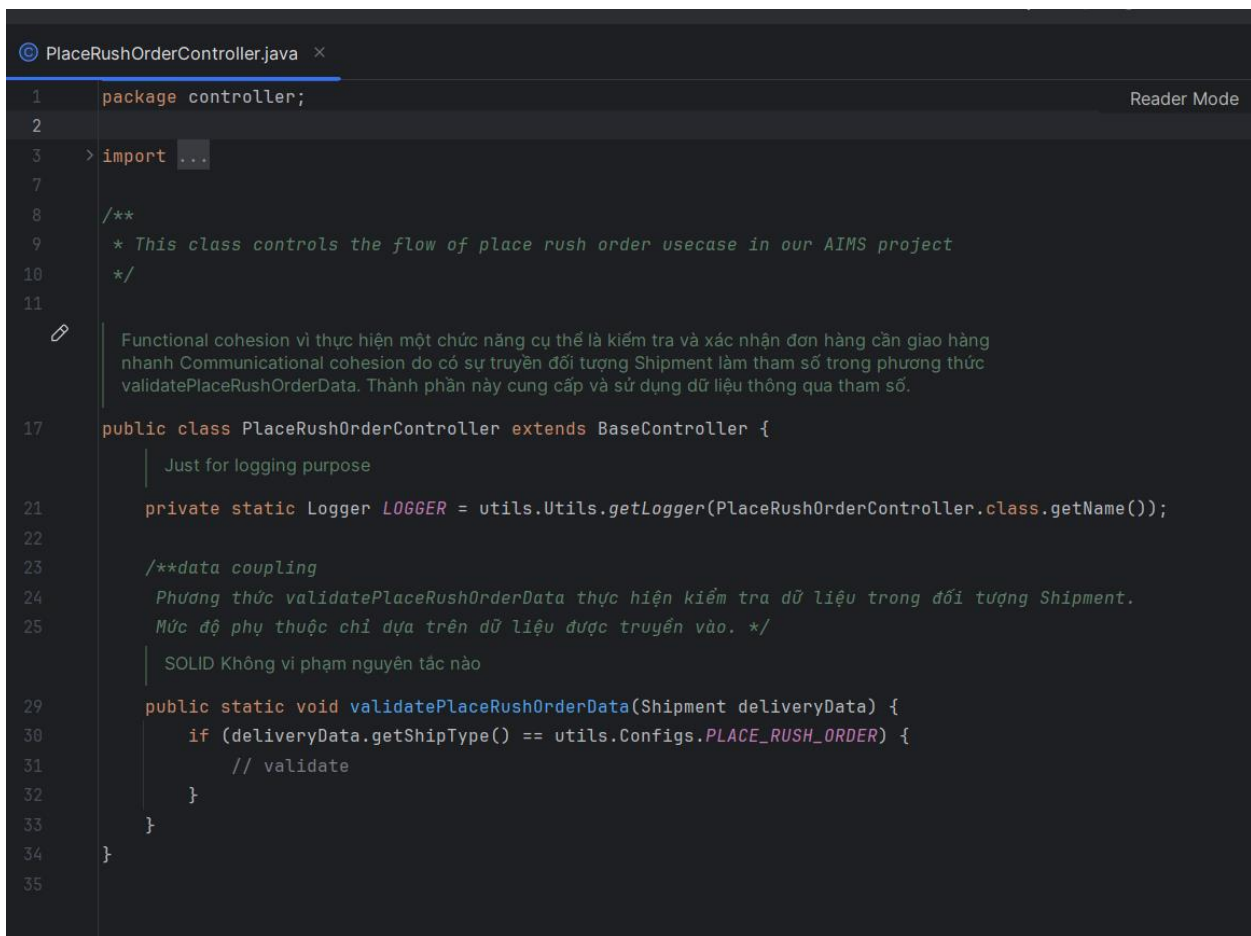
```
14  */
15
21
22  public class BaseController {
23
24      /**
25       * The method checks whether the Media in Cart, if it were in, we will return the CartMedia else return
26       * null.
27       * Params: media - media object
28       * Returns: CartMedia or null
29       */
30
31      //Content Coupling
32      //Communicational Cohesion: Phương thức liên quan đến kiểm tra và trả về thông tin về Media trong giỏ hàng
33      public CartMedia checkMediaInCart(Media media) { return Cart.getCart().checkMediaInCart(media); }
34
35      /**
36       * This method gets the list of items in cart.
37       *
38       * @return List[CartMedia]
39       */
40      //Content Coupling
41      //Communicational Cohesion: Phương thức liên quan đến lấy danh sách các mục trong giỏ hàng
42
43      /**
44       * Vi phạm Interface Segregation Principle (ISP) Phương thức getListCartMedia trả về một danh sách
45       * chưa rõ ràng về loại dữ liệu.
```

Hình 25: Coupling at BaseController



```
1  package controller;
2
3  import ...
4
5
6
7
8  /**
9   * This class controls the flow of events in homescreen
10   *
11   * @author nguyennlm
12   */
13
14  /**
15   * Vi phạm Single Responsibility Principle (SRP): Lớp HomeController mở rộng từ BaseController và thực
16   * hiện một chức năng mới liên quan đến việc lấy tất cả các Media từ cơ sở dữ liệu.
17   */
18
19
20  public class HomeController extends BaseController {
21
22      /**
23       * this method gets all Media in DB and return back to home to display
24       * Returns: List[Media]
25       * Throws: SQLException
26       */
27
28      //Content Coupling
29      //Functional Cohesion: This method has functional cohesion as it performs a specific function
30      public List getAllMedia() throws SQLException {
31          return new Media().getAllMedia();
32      }
33
34  }
35
```

Hình 26: Coupling at HomeController



```
1 package controller;
2
3 > import ...
4
5
6
7
8 /**
9  * This class controls the flow of place rush order usecase in our AIMS project
10  */
11
12
13
14
15
16
17 public class PlaceRushOrderController extends BaseController {
18     Just for logging purpose
19
20     private static Logger LOGGER = utils.Utils.getLogger(PlaceRushOrderController.class.getName());
21
22
23     /**data coupling
24      * Phương thức validatePlaceRushOrderData thực hiện kiểm tra dữ liệu trong đối tượng Shipment.
25      * Mức độ phụ thuộc chỉ dựa trên dữ liệu được truyền vào. */
26
27     SOLID Không vi phạm nguyên tắc nào
28
29     public static void validatePlaceRushOrderData(Shipment deliveryData) {
30         if (deliveryData.getShipType() == utils.Configs.PLACE_RUSH_ORDER) {
31             // validate
32         }
33     }
34 }
35
```

Functional cohesion vì thực hiện một chức năng cụ thể là kiểm tra và xác nhận đơn hàng cần giao hàng nhanh Communicational cohesion do có sự truyền đối tượng Shipment làm tham số trong phương thức validatePlaceRushOrderData. Thành phần này cung cấp và sử dụng dữ liệu thông qua tham số.

Hình 27: Coupling at PlaceRushOrderController

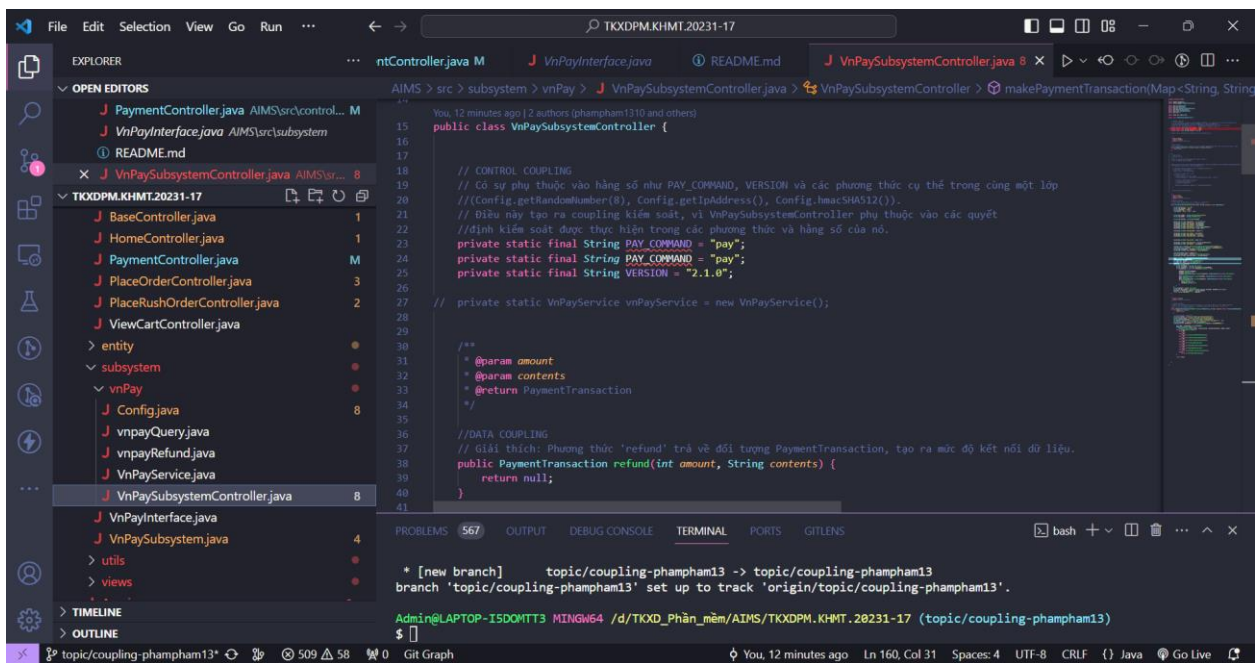
```

16 public class ViewCartController extends BaseController {
17     /*
18      * SOLID: Không vi phạm nguyên tắc nào
19      */
20
21     /**
22      * This method checks the available products in Cart
23      *
24      * @throws SQLException
25      */
26     /** control coupling
27      Phương thức checkAvailabilityOfProduct trong ViewCartController
28      kiểm soát luồng thực thi của phương thức checkAvailabilityOfProduct trong lớp Cart */
29     data coupling Phương thức này chỉ gọi các phương thức của lớp Cart thông qua việc truyền dữ liệu
30
31     public void checkAvailabilityOfProduct() throws SQLException {
32         Cart.getCart().checkAvailabilityOfProduct();
33     }
34
35     /**
36      * This method calculates the cart subtotal
37      *
38      * @return subtotal
39      */
40     data coupling Phương thức getCartSubtotal chỉ gọi các phương thức của lớp Cart thông qua việc
41     truyền dữ liệu
42
43     public int getCartSubtotal() {
44         int subtotal = Cart.getCart().calSubtotal();
45         return subtotal;
46     }

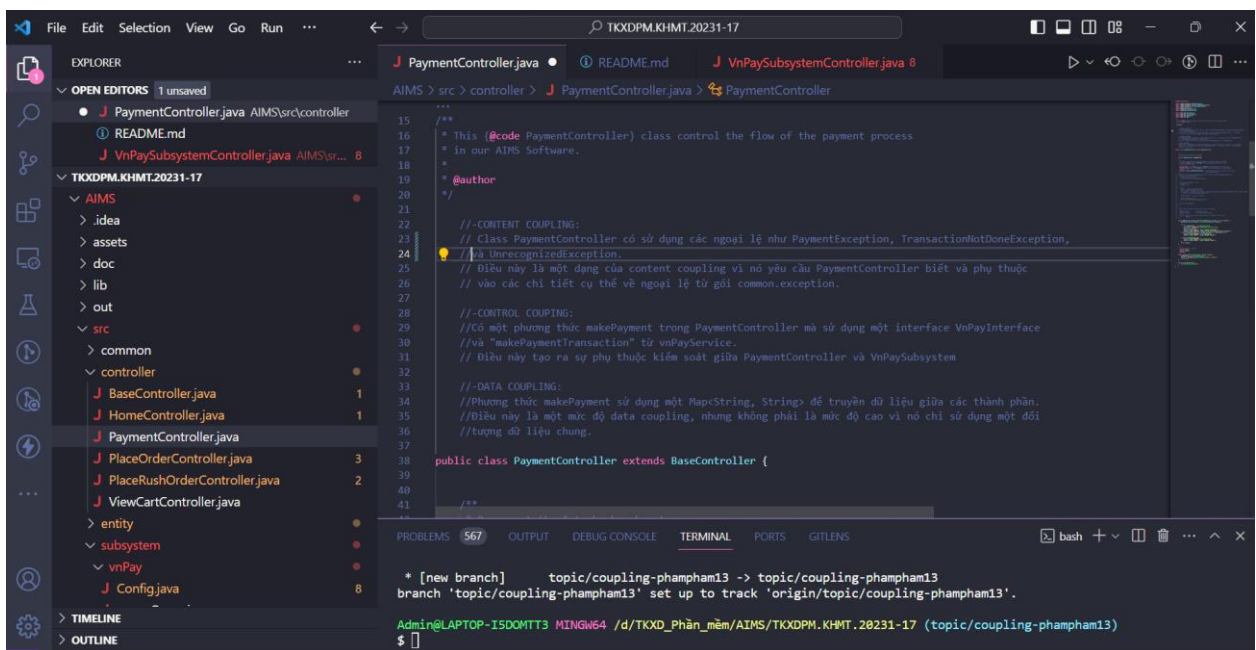
```

Hình 28: Coupling at ViewCartController

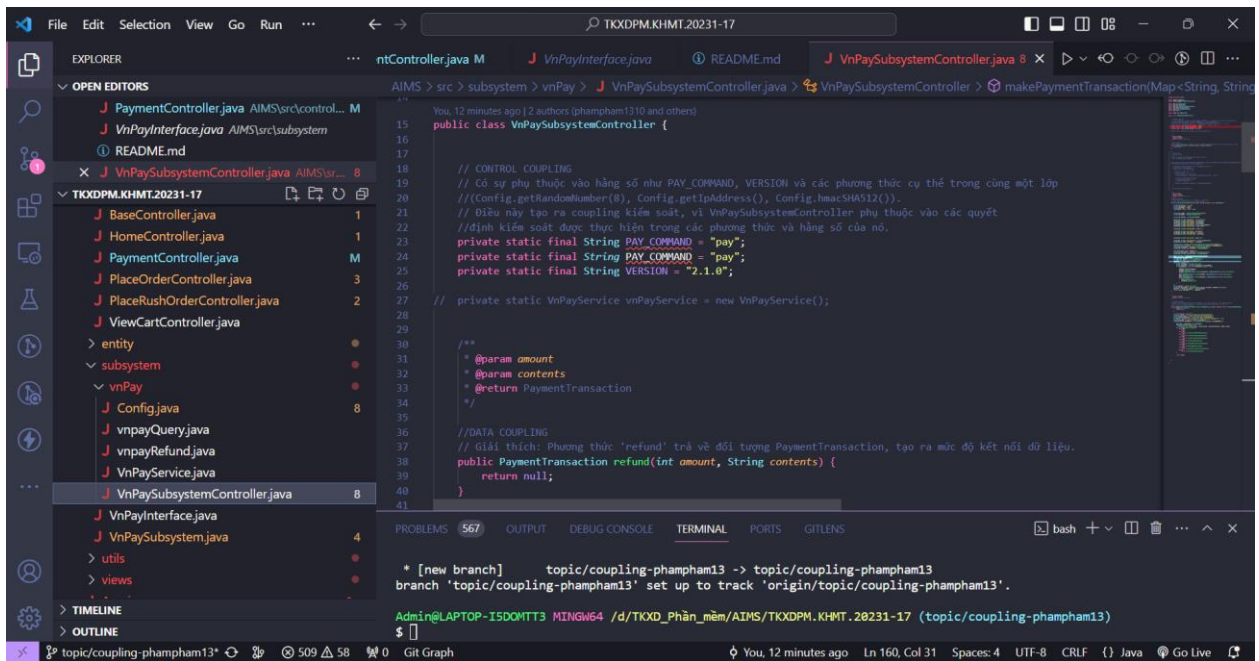
- Ngân: Phát hiện coupling với file PaymentController.java, VNPAYSubSystemController.java



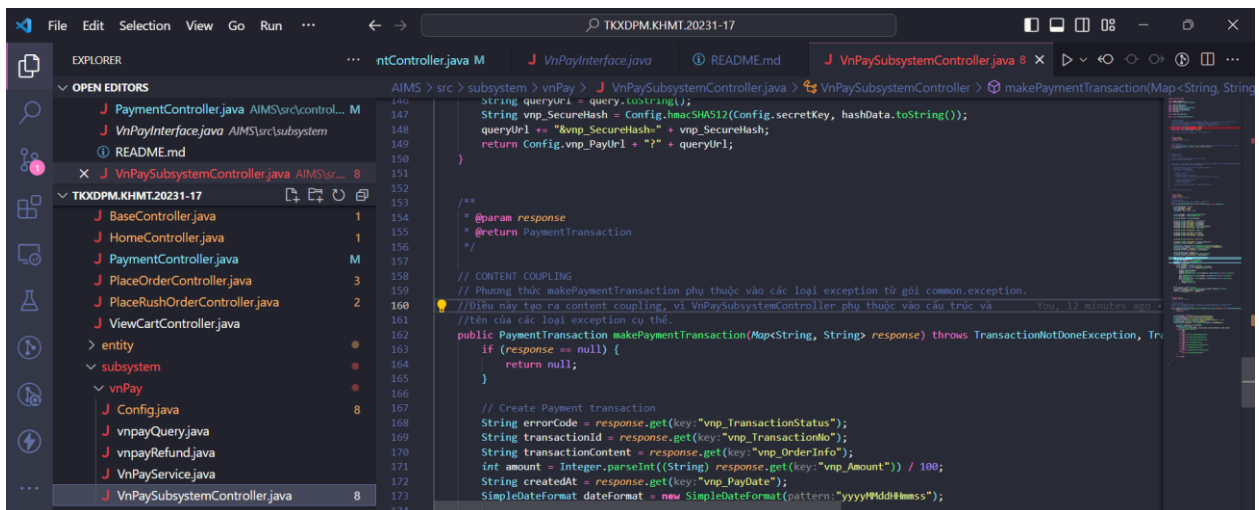
Hình 29: Coupling at PaymentController



Hình 30: Coupling at VnPaySubsystemController



Hình 31: Coupling at VnPaySubsystemController



Hình 32: Coupling at VnPaySubsystemController

## 6.3 SOLID



```

59 //Single Responsibility Principle
60 public int getTotalMedia() {
61     int total = 0;
62     for (Object obj : lstCartMedia) {
63         CartMedia cm = (CartMedia) obj;
64         total += cm.getQuantity();
65     }
66     return total;
67 }
68
69
70 Returns: int
71
72 //Single Responsibility Principle
73 public int calSubtotal() {
74     int total = 0;
75     for (Object obj : lstCartMedia) {
76         CartMedia cm = (CartMedia) obj;
77         total += cm.getPrice() * cm.getQuantity();
78     }
79     return total;
80 }
81
82
83
84 Throws: SQLException
85
86 //Dependency Inversion Principle
87 public void checkAvailabilityOfProduct() throws SQLException {
88     boolean allAvai = true;
89     for (Object object : lstCartMedia) {
90         CartMedia cartMedia = (CartMedia) object;
91         int requiredQuantity = cartMedia.getQuantity();

```

*Hình 33: SOLID at Cart*

```

Invoice.java x
1 package entity.invoice;
2
3 import entity.order.Order;
4
5 public class Invoice {
6
7     private Order order;
8     private int amount;
9
10    public Invoice() {
11
12    }
13    //Dependency Inversion Principle
14    > public Invoice(Order order) { this.order = order; }
17
18
19    | Returns: Order
22    > public Order getOrder() { return order; }
25
26    | Returns: int
29    > public int getAmount() { return amount; }
32
33    | Params: amount
36    > public void setAmount(int amount) { this.amount = amount; }
39    //Single Responsibility Principle
40    public void saveInvoice() {
41
42    }
43 }

```

*Hình 34: SOLID at Invoice*

- Nga: phát hiện vi phạm nguyên tắc SOLID trong các file BaseController.java, HomeController.java, PaymentController.java, PlaceOrderController.java



```
BaseController.java x
13  * @author nguyentlm
14  */
15
    Vi phạm Single Responsibility Principle SRP Lớp BaseController thực hiện nhiều nhiệm vụ, bao gồm việc
    kiểm tra và trả về thông tin về Media trong giỏ hàng cũng như lấy danh sách các mục trong giỏ hàng.

21
22  public class BaseController {
23
    The method checks whether the Media in Cart, if it were in, we will return the CartMedia else return
    null.
    Params: media – media object
    Returns: CartMedia or null

31    //Content Coupling
32    //Communicational Cohesion: Phương thức liên quan đến kiểm tra và trả về thông tin về Media trong giỏ hàng
33    public CartMedia checkMediaInCart(Media media) { return Cart.getCart().checkMediaInCart(media); }
36
37    /**
38     * This method gets the list of items in cart.
39     *
40     * @return List[CartMedia]
41     */
42    //Content Coupling
43    //Communicational Cohesion: Phương thức liên quan đến lấy danh sách các mục trong giỏ hàng
44
    Vi phạm Interface Segregation Principle (ISP) Phương thức getListCartMedia trả về một danh sách
    chưa rõ ràng về loại dữ liệu.

49    public List getListCartMedia() { return Cart.getCart().getListMedia(); }
52 }
```

Hình 35: SOLID at BaseController

```
BaseController.java HomeController.java x
1  package controller;
2
3  > import ...
7
8  /**
9   * This class controls the flow of events in homescreen
10   *
11   * @author nguyentlm
12   */
13
    Vi phạm Single Responsibility Principle (SRP): Lớp HomeController mở rộng từ BaseController và thực
    hiện một chức năng mới liên quan đến việc lấy tất cả các Media từ cơ sở dữ liệu.

19
20  public class HomeController extends BaseController {
21
```

Hình 36: SOLID at HomeController

```
PaymentController.java
35 //điều này là một mức độ data coupling, nhưng không phải là mức độ cao vì nó chỉ sử dụng một mức độ
36 //tương dữ liệu chung.
37
38 /**
39  * Vi phạm Single Responsibility Principle (SRP):
40  * Lớp PaymentController đang thực hiện nhiều nhiệm vụ,
41  * bao gồm việc xử lý thanh toán, tạo URL thanh toán, và xóa giỏ hàng.
42  */
43
44 Vi phạm Dependency Inversion Principle (DIP): Lớp PaymentController tạo mới một đối tượng
45 VnPaySubsystem bên trong phương thức, điều này làm cho lớp trở nên cứng nhắc và khó tái sử dụng.
46
49 public class PaymentController extends BaseController {
50
51     Represent the Interbank subsystem
52
55     private VnPayInterface vnPayService;
```

Hình 37: SOLID at PaymentController

```
PlaceOrderController.java
1 package controller;
2
3 > import ...
4
15
16
17 Vi phạm Dependency Inversion Principle (DIP): Lớp PlaceOrderController trực tiếp phụ thuộc vào Cart,
18 Order, và Media, thậm chí còn gọi đến các phương thức tĩnh của Media. Điều này làm cho lớp trở nên
19 cứng nhắc và khó kiểm thử.
20
21 public class PlaceOrderController extends BaseController {
22
```

Hình 38: SOLID at PlaceOrderController

- Ngân: phát hiện vi phạm nguyên tắc SOLID trong các file in ViewCartController.java, PlaceRushOrderController.java, VnPaySubsystem.java VnPaySubsystemController.

```
public class ViewCartController extends BaseController {
    /**
    * SOLID: Không vi phạm nguyên tắc nào
    */
}
```

Hình 39: SOLID at ViewCartController

```

public class PlaceRushOrderController extends BaseController {
    /**
     * Just for logging purpose
     */
    private static Logger LOGGER = utils.Utils.getLogger(PlaceRushOrderController.class.getName());

    /**data coupling
     * Phương thức validatePlaceRushOrderData thực hiện kiểm tra dữ liệu trong đối tượng Shipment.
     * Mức độ phụ thuộc chỉ dựa trên dữ liệu được truyền vào. */
    /**SOLID
     * Không vi phạm nguyên tắc nào
     */
    public static void validatePlaceRushOrderData(Shipment deliveryData) {
        if (deliveryData.getShipType() == utils.Configs.PLACE_RUSH_ORDER) {
            // validate
        }
    }
}

```

Hình 40: SOLID at PlaceRushOrderController

```

You, 3 minutes ago | 2 authors (phampham1310 and others)
public class VnPaySubsystemController {

    // CONTROL COUPLING
    // Có sự phụ thuộc vào hằng số như PAY_COMMAND, VERSION và các phương thức cụ thể trong cùng một lớp
    //((Config.getRandomNumber(8), Config.getIpAddress(), Config.hmacSHA512())).
    // Điều này tạo ra coupling kiểm soát, vì VnPaySubsystemController phụ thuộc vào các quyết
    //định kiểm soát được thực hiện trong các phương thức và hằng số của nó.
    private static final String PAY_COMMAND = "pay";
    private static final String PAY_COMMAND = "pay";
    private static final String VERSION = "2.1.0";

    //DATA COUPLING
    // Giải thích: Phương thức 'refund' trả về đối tượng PaymentTransaction, tạo ra mức độ kết nối dữ liệu.
    public PaymentTransaction refund(int amount, String contents) {
        return null;
    }

    /**
     * @param money
     * @param contents
     * @return PaymentTransaction
     */
    //CONTROL COUPLING
    // Giải thích: Sự phụ thuộc vào các hằng số và phương thức
    //cụ thể trong cùng một lớp tạo ra kết nối kiểm soát.
    /**
     * SOLID
     * SRP: Phương thức generatePayOrderUrl thực hiện nhiều công việc,
     * bao gồm cả việc tạo URL và tính toán hash.
     * Đề xuất giải pháp: Tách phương thức này thành các phương thức nhỏ,
     * mỗi phương thức có một trách nhiệm cụ thể
     */
}

```

Hình 41: SOLID at VnPaySubsystemController

-subsystem/ VnPaySubsystem.java: không có vi phạm nguyên tắc SOLID



## Phụ lục hình ảnh

Hình 1: Biểu đồ Usecase tổng quan .....	5
Hình 2: Biểu đồ quy trình nghiệp vụ .....	6
Hình 3: Biểu đồ hoạt động Usecase CRUD sản phẩm .....	8
Hình 4: Biểu đồ hoạt động của Use case "Tìm kiếm sản phẩm" .....	11
Hình 5: Biểu đồ hoạt động của Use case "Lọc sản phẩm" .....	13
Hình 6: Biểu đồ hoạt động của Use case "Quản lí giỏ hàng" .....	16
Hình 7: Biểu đồ tuần tự CRUD sản phẩm .....	18
Hình 8: Biểu đồ các lớp phân tích CRUD sản phẩm .....	19
Hình 9: Biểu đồ tuần tự Tìm kiếm sản phẩm .....	20
Hình 10: Biểu đồ các lớp phân tích Tìm kiếm sản phẩm .....	21
Hình 11: Biểu đồ tuần tự Lọc sản phẩm .....	22
Hình 12: Biểu đồ các lớp phân tích Lọc sản phẩm .....	23
Hình 13: Biểu đồ tuần tự Quản lí giỏ hàng .....	24
Hình 14: Biểu đồ các lớp phân tích Quản lí giỏ hàng .....	25
Hình 155: Susystem Interaction .....	30
Hình 16: Detail class diagrams Tìm kiếm, Lọc sản phẩm .....	31
Hình 17: Detail class diagrams CRUD sản phẩm .....	32
Hình 18: Relationship class diagram .....	33
Hình 19: Cohesion at BaseController .....	39
Hình 20: Conhesion at HomeController .....	39
Hình 21: Cohesion at PlaceRushOrderController .....	40
Hình 22: Cohesion at ViewCartController .....	40
Hình 23: Cohesion at VnPaySubsystemController.java .....	41
Hình 24: Cohesion at PaymentController.java .....	41
Hình 25: Coupling at BaseController .....	42
Hình 26: Coupling at HomeController .....	42
Hình 27: Coupling at PlaceRushOrderController .....	43
Hình 28: Coupling at ViewCartController .....	44
Hình 29: Coupling at PaymentController .....	45
Hình 30: Coupling at VnPaySubsystemController .....	45
Hình 31: Coupling at VnPaySubsystemController .....	46
Hình 32: Coupling at VnPaySubsystemController .....	46

Hình 33: SOLID at Cart .....	47
Hình 34: SOLID at Invoice .....	48
Hình 35: SOLID at BaseController .....	49
Hình 36: SOLID at HomeController .....	49
Hình 37: SOLID at PaymentController .....	50
Hình 38: SOLID at PlaceOrderController .....	50
Hình 39: SOLID at ViewCartController .....	50
Hình 40: SOLID at PlaceRushOrderController.....	51
Hình 41: SOLID at VnPaySubsystemController.....	51