

HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

School of Information and Communications Technology

## Final Report

Version 1.1

AIMS.

Môn: Thiết kế và xây dựng phần mềm

Giảng viên: Nguyễn Thị Thu Trang

Nhóm: 13

Nông Hoàng Lâm – 20204575

Nguyễn Ngọc Linh – 20204576

Vũ Văn Lợi - 20204577

*Hanoi, <10, 2023r>*

## Contents

1	Phân công công việc.....	3
2	Introduction .....	4
2.1	Objective.....	4
2.2	Scope .....	4
2.3	Glossary .....	4
2.4	References .....	4
3	Overall requirements .....	5
3.1	Actors.....	5
3.2	General use case diagram .....	5
3.3	Business processes.....	6
4	Detail requirements .....	7
4.1	Specification of Use case UC001 – Đặt hàng.....	7
4.2	Specification of Use case UC002 – Thanh toán” .....	10
4.3	Specification of Use case UC003 – Đặt hàng nhanh” .....	11
4.4	Specification of Use case UC004 – Hủy đơn hàng” .....	14
4.5	Specification of Use case UC005 – “Tìm kiếm sản phẩm” .....	17
4.6	Specification of Use case UC004 – “Quản lý giỏ hàng” .....	19
5	Supplementary specification .....	22
5.1	Functionality .....	22
5.2	Usability.....	22
5.3	Reliability .....	22
5.4	Performance.....	22
5.5	Maintainability.....	22
5.6	Design Constraints.....	22
6	INTERACTION DIAGRAMS.....	23
6.1	Sequence diagram UseCase “Hủy đơn hàng” .....	23

6.2	Sequence diagram UseCase “Tìm kiếm sản phẩm” .....	23
6.3	Communication diagram UseCase “Quản lý giỏ hàng” .....	24
6.4	Communication diagram UseCase “Tìm kiếm sản phẩm” .....	25
7	ANALYSIS CLASS DIAGRAM.....	26
7.1	ANALYSIS CLASS DIAGRAM Tổng quan.....	26
7.2	ANALYSIS CLASS DIAGRAM UseCase “Hủy đơn hàng” .....	26
7.3	ANALYSIS CLASS DIAGRAM “Tìm kiếm sản phẩm” .....	27
7.4	ANALYSIS CLASS DIAGRAM UseCase “Quản lý giỏ hàng” .....	27
8	INTERFACE DESIGN .....	28
9	DETAIL CLASS DIAGRAM.....	36
9.1	Lớp thiết kế tổng quan. ....	36
9.2	Lớp thiết kế cho hủy đơn hàng. ....	37
9.3	CONCEPTUAL DATA MODEL .....	38
10	DATABASE DESIGN.....	39
10.1.1	Logical Data Model .....	39
10.1.2	Physical Data Model.....	39
10.1.3	SQL code .....	42

## 1 Phân công công việc

Thành viên	Đóng góp	Phần trăm
Nông Hoàng Lâm	<ul style="list-style-type: none"><li>- UseCase Hủy đơn hàng: Đặc tả UseCase, Activity Diagram, Sequence Diagram, Analysis Class Diagram, TestPlan</li><li>- Interface Design, Data Modeling</li><li>- Good Design: Coupling, Cohesion, SOLID Folder Model + Subsystem</li><li>- Video Demo</li></ul>	33%
Nguyễn Ngọc Linh	<ul style="list-style-type: none"><li>- UseCase Quản lý giỏ hàng: Đặc tả UseCase, Activity Diagram, Sequence Diagram, Analysis Class Diagram, TestPlan</li><li>- Interface Design, Data Modeling</li><li>- Good Design: Coupling, Cohesion, SOLID Folder Controller + Utils</li></ul>	33%
Vũ Văn Lợi	<ul style="list-style-type: none"><li>- UseCase Tìm kiếm sản phẩm: Đặc tả UseCase, Activity Diagram, Sequence Diagram, Analysis Class Diagram, Test Plan</li><li>- Interface Design, Data Modeling</li><li>- Good Design: Coupling, Cohesion, SOLID Folder Views</li></ul>	33%

## **2 Introduction**

### **2.1 *Objective***

Tài liệu này trình bày miêu tả về chức năng của phần mềm thương mại điện tử AIMS. Tài liệu này dành cho các bên liên quan và phía lập trình viên.

### **2.2 *Scope***

Bao gồm các chức năng chính của phần mềm

### **2.3 *Glossary***

### **2.4 *References***

### 3 Overall requirements

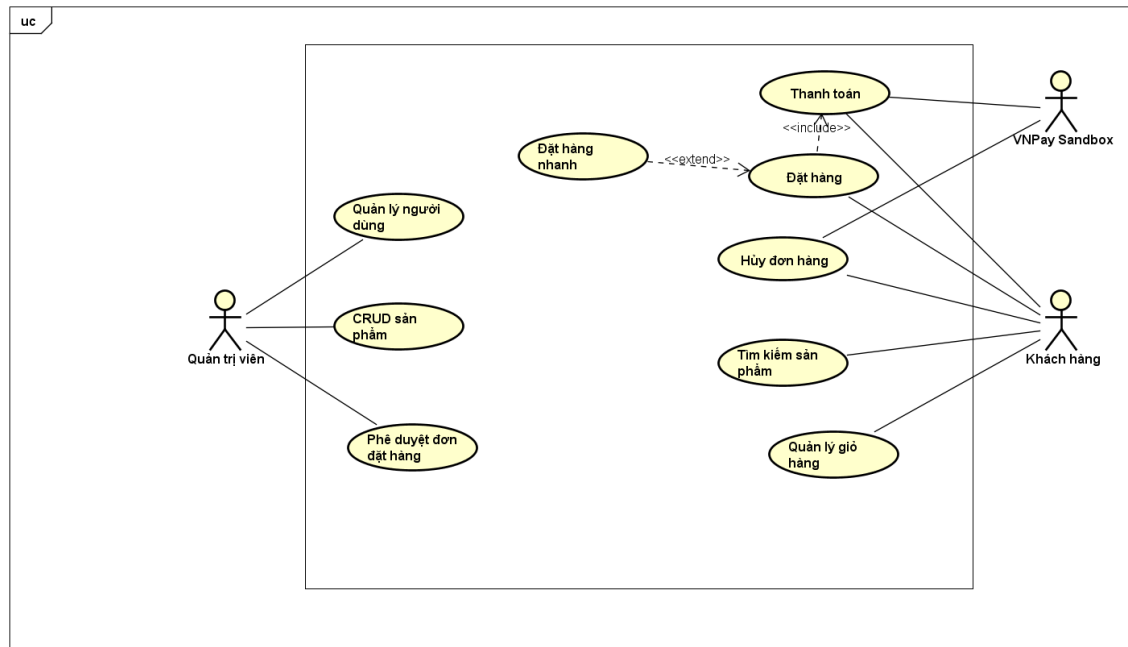
#### 3.1 Actors

Khách hàng

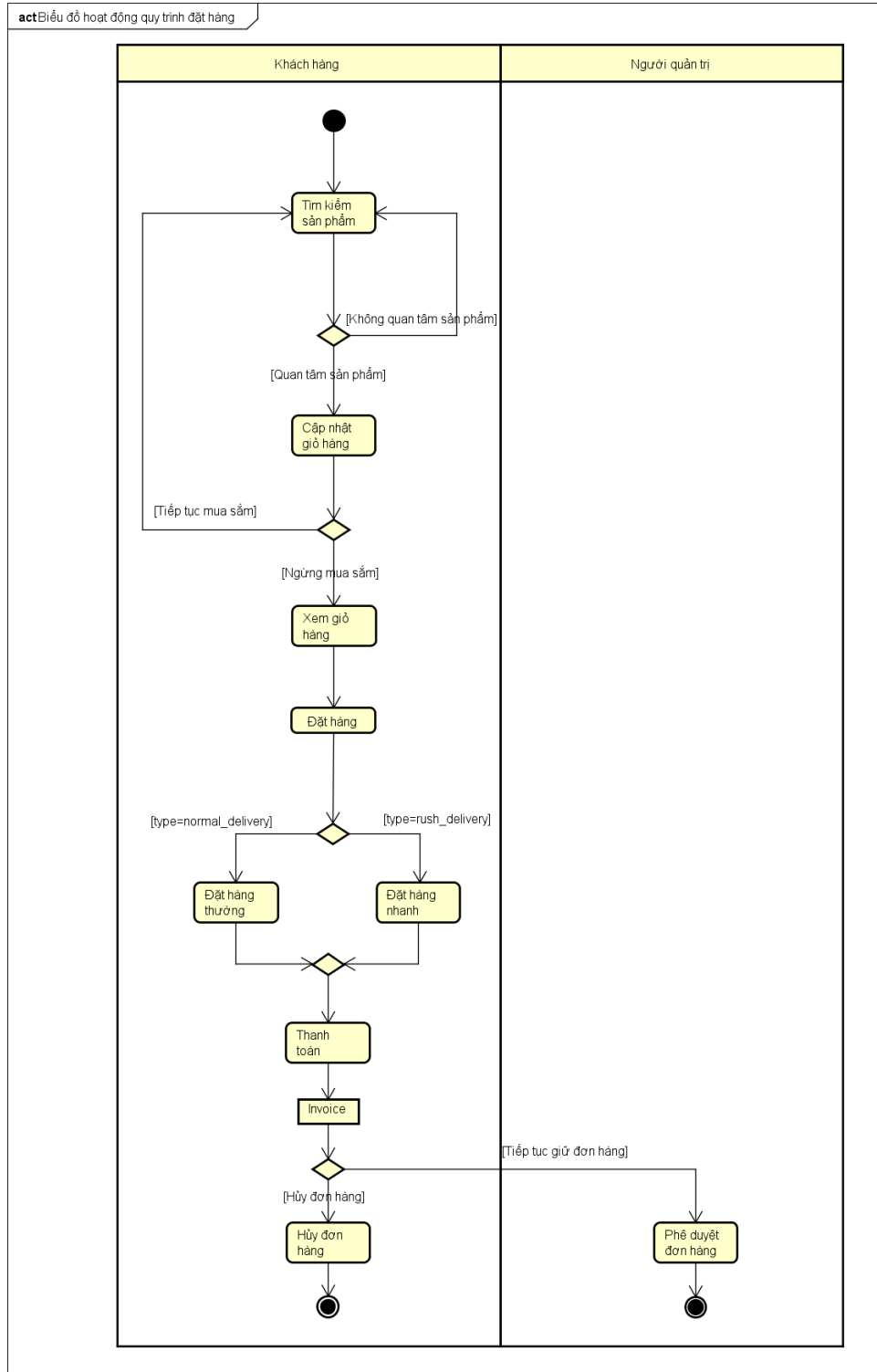
VNPay sandbox

#### 3.2 General use case diagram

Phần mềm AIMS giúp khách hàng có thể đặt sản phẩm và thanh toán.



### 3.3 Business processes



## 4 Detail requirements

### 4.1 Specification of Use case UC001 – Đặt hàng

#### 1. Use case code

UC001

#### 2. Brief Description

Usecase này giúp khách hàng có thể đặt sản phẩm và thanh toán nội vùng.

#### 3. Actors

Khách hàng.

#### 4. Preconditions.

Không có.

#### 5. Basic Flow of Events

1. Khách hàng bấm vào xem giỏ hàng.
2. Hệ thống tính phí tổng giá sản phẩm.
3. Hệ thống kiểm tra các sản phẩm trong giỏ còn có tồn kho.
4. Hệ thống sẽ hiện ra danh sách các món hàng khách hàng muốn đặt (tên sản phẩm, số lượng, và giá cả).
5. Khách hàng bấm vào nút “Đặt hàng”.
6. Hệ thống kiểm tra các sản phẩm trong giỏ còn có tồn kho.
7. Hệ thống hiển thị phiếu điền thông tin giao hàng, yêu cầu khách hàng cập nhật thông tin giao hàng.
8. Khách hàng điền các thông tin cần thiết và không chọn “giao hàng nhanh”. Bấm “Cập nhật”.
9. Hệ thống kiểm tra thông tin đầu vào.
10. Hệ thống tính phí giao hàng.
11. Hệ thống sẽ hiển thị thông tin đơn hàng tạm thời.
12. Gọi tiếp đến Usecase “Thanh toán”.
13. Hệ thống làm trống giỏ hàng.

#### Alternative flows



No	Location	Condition	Action	Resume location
1.	Tại bước 3	Nếu hệ thống kiểm tra trong kho không đủ các sản phẩm	▪ Hệ thống sẽ thông báo tới khách hàng phải sản phẩm không đủ trong kho và yêu cầu khách hàng cập nhật lại giỏ hàng các sản phẩm thiếu	Tiếp tục ở bước 4.
2.	Tại bước 8	nếu khách hàng bỏ trống các trường thông tin bắt buộc hoặc ghi sai định dạng	▪ hệ thống sẽ yêu cầu khách hàng khách hàng nhập đầy đủ thông tin	Tiếp tục bước 8
3.	Tại bước 8	nếu khách hàng chọn “Giao hàng nhanh”	▪ Chèn vào usecase “Đặt hàng nhanh”	Tiếp tục usecase “Đặt hàng nhanh”
4.	Tại bước 1	nếu trong giỏ hàng không có sản phẩm	▪ hệ thống sẽ thông báo trong giỏ hàng không có sản phẩm	Tiếp tục bước 1

## 6. Input data

**Table A-Input data of “phiếu thông tin giao hàng”.**

No	Data fields	Description	Mandatory	Valid condition	Example
1.	Địa chỉ giao hàng		Có	50 từ tối đa	Số 1, Đường Tạ Quang Bửu, quận Hai Bà Trưng.
2.	Tên người nhận		Có	30 ký tự tối đa	Nông Hoàng Lâm
3.	Số điện thoại		Có	Từ 9-11 chữ số.	09123456789
4.	Tỉnh	Chọn từ danh sách 64 tỉnh	Có	Chọn 1 tỉnh	Hà Nội
5.	Số thẻ		Có	16 ký tự	1234 5678 1234 5678
6.	Tên chủ thẻ		Có	30 ký tự tối đa, viết hoa, không dấu	NONG HOANG LAM

## 7. Output data

**Table B-Output data of “Bảng thông tin đơn đặt hàng tạm thời”**

No	Data fields	Description	Display format	Example
1.	Tiêu đề	Tên của sản phẩm		Đĩa nhạc Pop
2.	Giá	Giá trên một đơn vị sản phẩm	Số nguyên dương >0 Có dấu , tách giữa phần nghìn Đơn vị nghìn đồng	100,000 đồng
3.	Số lượng	Số lượng của sản phẩm	Số nguyên dương > 0	3
4.	Tổng số tiền của sản phẩm	Tổng số tiền của loại sản phẩm	Số nguyên dương >0	300,000 đồng
5.	Tổng tiền phải trả khi chưa tính VAT	Tổng số tiền của tất cả sản phẩm trong giỏ trước khi tính VAT	Có dấu , tách giữa phần nghìn Đơn vị nghìn đồng	500,000 đồng
6.	Tổng tiền phải trả khi đã tính VAT	Tổng số tiền của tất cả sản phẩm trong giỏ sau khi tính VAT	Số nguyên dương >0 Có dấu , tách giữa phần nghìn Đơn vị nghìn đồng Hiện %VAT đã tính và thành tiền của VAT	550,000 đồng (10%VAT = 50,000 đồng)

## 4.2 Specification of Use case UC002 – Thanh toán”

### 1. Use case code

UC002

### 2. Brief Description

Khi khách hàng muốn thanh toán đơn hàng của mình với AIMS thông qua cổng thanh toán ngân hàng.

### 3. Actors

Khách hàng, ngân hàng.

### 4. Preconditions

### 5. Basic Flow of Events

1. Khách hàng chọn phương thức thanh toán sử dụng cổng thanh toán VNPay.
2. Hệ thống hiển thị phiếu thông tin thanh toán.
3. Khách hàng điền thông tin giao dịch và nội dung giao dịch và bấm xác nhận.
4. Hệ thống sẽ hiển thị thông tin giao dịch.
5. Hệ thống gửi thông tin giao dịch đến hòm thư điện tử của khách hàng.
6. Hệ thống hiển thị thông báo “Giao dịch thành công”.

### 6. Alternative flows

**Table N-Alternative flows of events for UC Place order**

No	Location	Condition	Action	Resume location
5.	Tại bước 3	nếu khách hàng đưa thông tin số thẻ sai	▪ hệ thống sẽ hiển thị thông báo “Không tìm thấy số tài khoản, vui lòng nhập lại thông tin”.	Tiếp tục bước 3.
6.	Tại bước 3	nếu tài khoản của khách hàng không đủ số dư	▪ hệ thống sẽ hiển thị thông báo “Số dư không đủ, vui lòng thanh toán lại.”	Tiếp tục bước 3.

### 7. Input data

**Table A-Input data of form điền thông tin bên ngân hàng.**

No	Data fields	Description	Mandatory	Valid condition	Example
1.	Số thẻ		Có	16 ký tự	1234 5678 1234 5678
2.	Tên chủ thẻ		Có	20 ký tự tối đa	NONG HOANG LAM
3.	Tên ngân hàng		Có	Chọn từ danh sách các ngân hàng	Ngân hàng BIDV

## 8. Output data

**Table B-Output data of “Bảng thông tin giao dịch”**

No	Data fields	Description	Display format	Example
1.	mã giao dịch		Được format theo template sau: “AIMS” + lần giao dịch	AIMS191103
2.	tên chủ thẻ		Viết hoa, không dấu	NONG HOANG LAM
3.	số tiền bị trừ		Số nguyên âm, có đơn vị tiền tệ vnd đứng sau	-108,000 đồng
4.	nội dung giao dịch		Được ghi cùng với mã giao dịch	Giao dịch thanh toán từ AIMS, mã giao dịch: AIMS19103
5.	số dư		Số nguyên dương >0 Có dấu , tách giữa phần nghìn Đơn vị nghìn đồng	999,000 đồng
6.	ngày giờ giao dịch		Được định dạng theo “DD/MM/YYYY:h:m:s”	19/11/2023:19:23:56

### 4.3 Specification of Use case UC003 – Đặt hàng nhanh”

#### 1. Use case code

UC003

#### 2. Brief Description

Khi khách hàng muốn đặt đơn hàng của mình nhanh.

#### 3. Actors

Khách hàng

#### 4. Preconditions

Usecase mở rộng của usecase đặt hàng, khi người dùng ấn “Đặt hàng nhanh”

#### 5. Basic Flow of Events

1. Hệ thống thêm các trường thông tin đặt hàng nhanh vào phiếu thông tin giao hàng.
2. Người dùng nhập các trường thông tin đặt hàng nhanh.
3. Hệ thống kiểm tra địa điểm giao hàng nhanh có hỗ trợ.
4. Hệ thống kiểm tra các sản phẩm có hỗ trợ giao hàng nhanh.
5. Hệ thống cập nhật phí giao hàng nhanh.
6. Hệ thống hiển thị phiếu thông tin “đặt hàng nhanh”, bao gồm chi phí đặt hàng nhanh, các sản phẩm có thể giao hàng nhanh và các sản phẩm không hỗ trợ .
7. Người dùng bấm nút xác nhận.
8. Hệ thống cập nhật chi phí giao hàng nhanh, các sản phẩm giao hàng nhanh, các sản phẩm giao hàng thường tại phiếu thông tin “giao hàng”.

#### 6. Alternative flows

**Table 2-Alternative flows of events for UC Place order**

No	Location	Condition	Action	Resume location
1.	Tại bước 4	nếu các sản phẩm đều không hỗ trợ giao hàng nhanh.	▪ hệ thống sẽ hiển thị thông báo yêu cầu người dùng cập nhật lại đơn hàng.	Tiếp tục tại bước 4
2.	Tại bước 3	nếu địa chỉ không hỗ trợ giao hàng nhanh	▪ hệ thống sẽ hiển thị thông báo yêu cầu người dùng phải nhập lại thông tin.	Tiếp tục tại bước 3
3.	Tại bước 2	Nếu có trường thông tin bị để trống hoặc ghi sai định dạng	▪ Hệ thống sẽ hiển thị thông báo yêu cầu người dùng sửa lại thông tin	Tiếp tục tại bước 3.

## 7. Input data

**Table A-Input data of “phiếu thông tin giao hàng nhanh”.**

No	Data fields	Description	Mandatory	Valid condition	Example
1.	Địa chỉ giao hàng		Có	50 từ tối đa	Số 1, Đường Tạ Quang Bửu, quận Hai Bà Trưng.
2.	Địa chỉ giao hàng nhanh		Có	50 từ tối đa	Số 1, Đường Tạ Quang Bửu, quận Hai Bà Trưng
3.	Chỉ dẫn giao hàng		Không	50 từ tối đa	Gửi đến văn phòng vào 11 giờ trưa
4.	Thời gian nhận hàng		Có	Định dạng theo hh:mm	11:30
5.	Số điện thoại		Có	Từ 9-11 chữ số.	09123456789
6.	Tỉnh	Chọn từ danh sách 64 tỉnh	Có	Chọn 1 tỉnh	Thành phố Hà Nội

## 8. Output data

**Table B-Output data of “Bảng thông tin đơn đặt hàng nhanh”**

No	Data fields	Description	Display format	Example
7.	Tiêu đề	Tên của sản phẩm		Đĩa nhạc Pop
8.	Giá	Giá trên một đơn vị sản phẩm	Số nguyên dương >0 Có dấu , tách giữa phần nghìn Đơn vị nghìn đồng	100,000 đồng
9.	Loại đặt hàng	Sản phẩm là đặt hàng nhanh hay đặt hàng thường	Sẽ có biểu tượng đặt hàng nhanh bên cạnh ảnh sản phẩm.	
10.	Số lượng	Số lượng của sản phẩm	Số nguyên dương > 0	3
11.	Đặt hàng nhanh	Có phải sản phẩm đặt hàng nhanh	Một ô tích, không cho phép người dùng thay đổi	v
12.	Giá thành đặt hàng nhanh		Số nguyên dương >0 Có dấu , tách giữa phần nghìn Đơn vị nghìn đồng	50,000 đồng
13.	Tổng số tiền	Tổng số tiền của loại sản	Số nguyên dương >0	300,000 đồng

	của sản phẩm	phẩm	Có dấu , tách giữa phần nghìn	
14.	Tổng tiền phải trả khi chưa tính VAT	Tổng số tiền của tất cả sản phẩm trong giỏ trước khi tính VAT	Đơn vị nghìn đồng	500,000 đồng
15.	Tổng tiền phải trả khi đã tính VAT	Tổng số tiền của tất cả sản phẩm trong giỏ sau khi tính VAT	Số nguyên dương >0 Có dấu , tách giữa phần nghìn Đơn vị nghìn đồng Hiện %VAT đã tính và thành tiền của VAT	550,000 đồng (10%VAT = 50,000 đồng)

#### **4.4 Specification of Use case UC004 – Hủy đơn hàng”**

##### **1. Use case code**

UC004

##### **2. Brief Description**

Khi khách hàng muốn hủy đơn hàng mà mình đã thanh toán và đơn hàng chưa được kiểm duyệt.

##### **3. Actors**

Khách hàng

##### **4. Preconditions**

Đang chạy phần mềm và đang ở tại Screen Home và người dung đang có kết nối với mạng internet.

##### **5. Basic Flow of Events**

1. Người dung bấm vào tìm kiếm đơn hàng.
2. Hệ thống hiển thị trang tìm kiếm đơn hàng.
3. Người dung nhập số điện thoại của đơn hàng đã đặt để tìm kiếm đơn hàng.
4. Hệ thống hiển thị tất cả các đơn hàng có số điện thoại mà người dùng yêu cầu.
5. Người dung bấm nút hủy đơn hàng của đơn hàng mình muốn hủy.
6. Hệ thống xử lý hoàn tiền.
7. Hệ thống hiển thị thông báo hủy đơn hàng thành công.”.

##### **6. Alternative flows**

**Table 2-Alternative flows of events for UC Place order**

No	Location	Condition	Action	Resume location
1.	Tại bước 2	Nếu không tìm thấy bất cứ sản phẩm nào	▪ hệ thống sẽ hiển thị thông báo không tồn tại sản phẩm.	Tiếp tục tại bước 2
2.	Tại bước 2	nếu số điện thoại không phù hợp.	▪ hệ thống sẽ hiển thị thông báo số điện thoại không hợp lệ.	Tiếp tục tại bước 2

## 7. Input data

**Table A-Input data of “phiếu thông tin giao hàng nhanh”.**

No	Data fields	Description	Mandatory	Valid condition	Example
1.	Số điện thoại	Số điện thoại để tìm kiếm đơn hàng	Có	Từ 9-11 chữ số.	09123456789

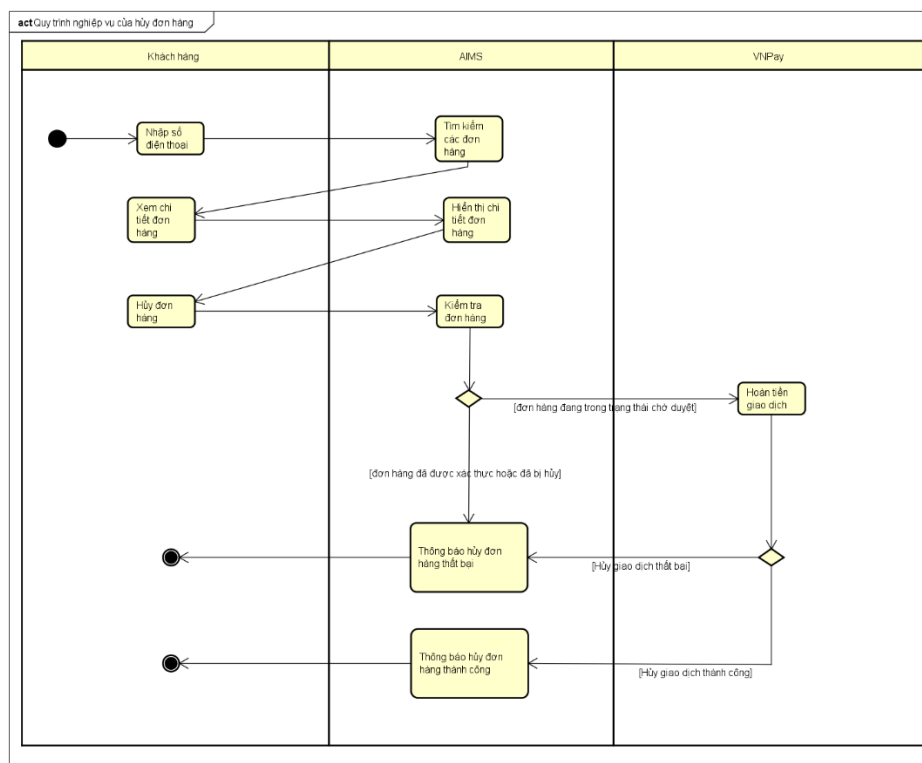
## 8. Output data

**Table B-Output data of “Bảng thông tin hủy đơn hàng”**

No	Data fields	Description	Display format	Example
1.	Mã đơn hàng	Mã của đơn hàng đã đặt		1
2.	Tổng số tiền của đơn hàng		Số nguyên dương >0 Có dấu , tách giữa phần nghìn Đơn vị nghìn đồng	100,000 đồng
3.	Địa chỉ		Chữ	3 Tạ Quang Bửu
4.	Tỉnh thành		Chữ	Hà Nội
5.	Email		Định dạng email	nhl@gmail.com
6.	Số điện thoại		9-11 chữ số Có dấu , tách giữa phần nghìn Đơn vị nghìn đồng	082123456
7.	Mã giao dịch		Chữ số	47112345

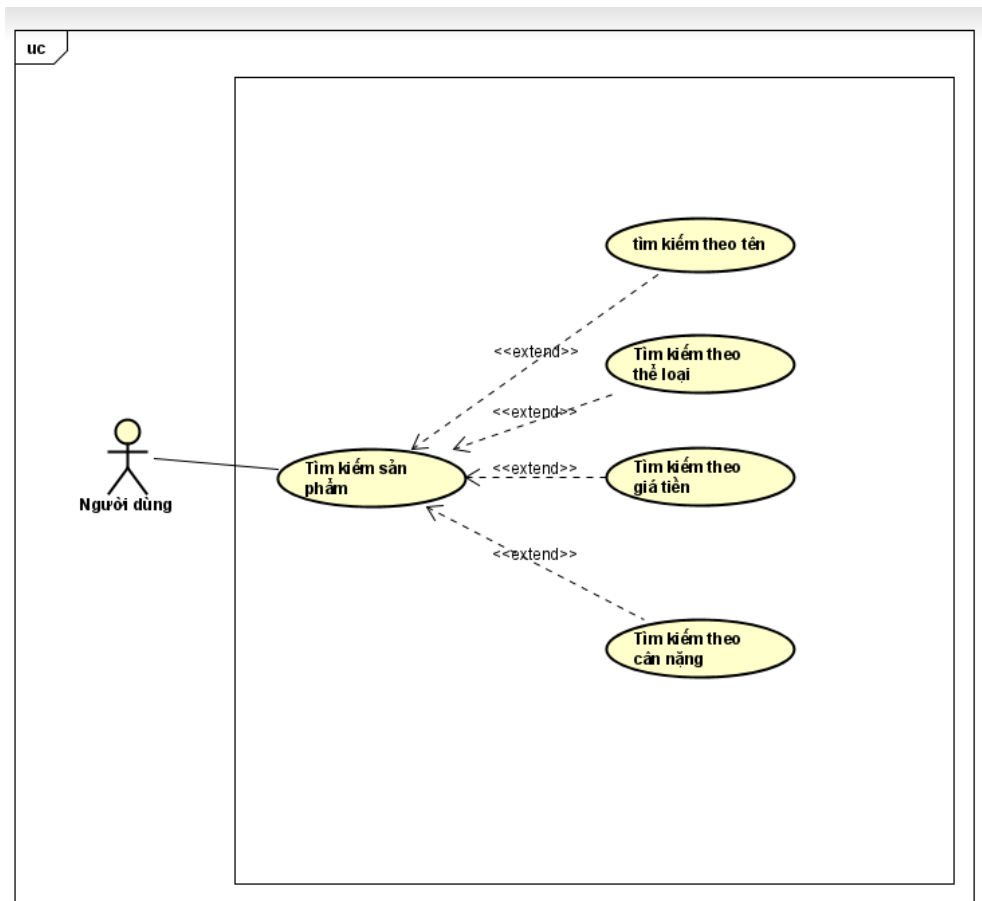


8.	Trạng thái đơn hàng	<p>Là một trong các trạng thái sau đây:</p> <ul style="list-style-type: none"> <li>Pending: Đang chờ thanh toán</li> <li>Canceled: Đã hủy đơn hàng do chưa thanh toán.</li> <li>Refunded: Đã hoàn tiền đơn hàng.</li> <li>Success: Đơn hàng đã thanh toán và được duyệt thành công.</li> </ul>	refunded
----	---------------------	--	----------



*Biểu đồ hoạt động “Hủy đơn hàng”*

#### 4.5 Specification of Use case UC005 – “Tìm kiếm sản phẩm”



**1. Mã use case**

UC003

**2. Mô tả ngắn gọn**

Use case này mô tả tương tác giữa phần mềm AIMS với khách hàng khi khách hàng muốn tìm kiếm sản phẩm trong hệ thống

**3. Tác nhân**

a. Khách hàng

**4. Tiền điều kiện**

Không có

**5. Luồng sự kiện chính**

1. Phần mềm AIMS hiển thị màn trang chủ với danh sách tất cả các sản phẩm đang được bán.
2. Người dùng chọn hình thức tìm kiếm
3. Người dùng điền thông tin sản phẩm rồi ấn nút search hoặc nút enter.
4. Phần mềm AIMS sẽ tìm kiếm trong DataBase những sản phẩm thỏa mãn từ khóa tìm kiếm. Nếu không có sản phẩm thỏa mãn sẽ hiển thị không tìm thấy sản phẩm.

## 6. Luồng sự kiện thay thế

Không có

## 7. Dữ liệu vào

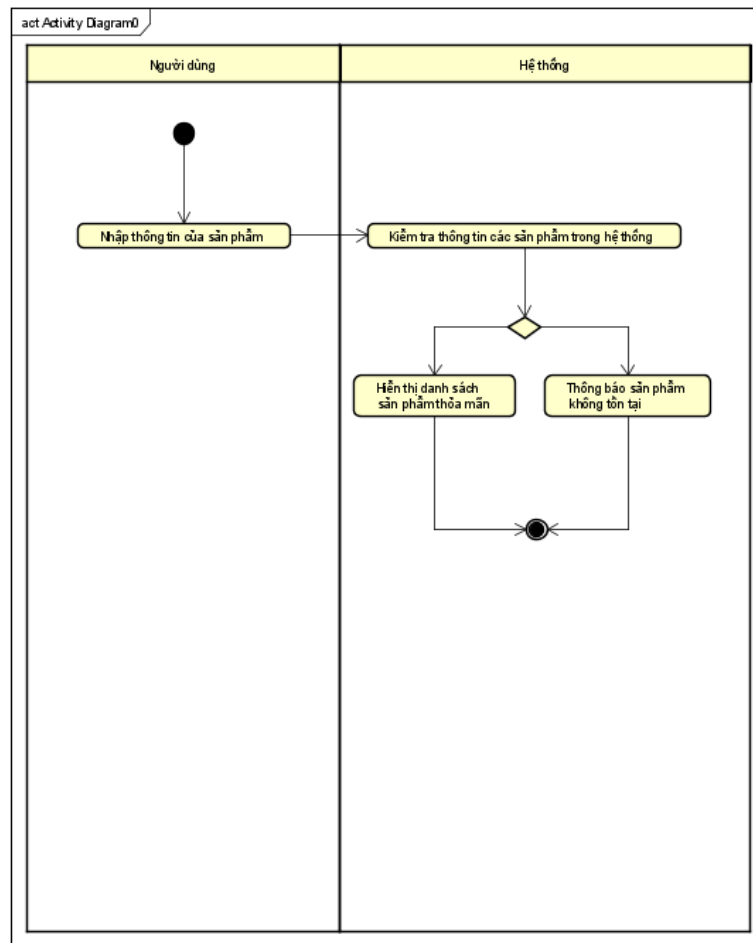
STT	Trường dữ liệu	Mô tả	Bắt buộc	Điều kiện hợp lệ	Ví Dụ
1.	Thông tin tìm kiếm	Thông tin vật phẩm cần tìm	Có	Tất cả	cÁ heo DVD Harry porter

## 8. Dữ liệu ra

STT	Trường dữ liệu	Mô tả	Bắt buộc	Điều kiện hợp lệ	Ví Dụ
1.	Tên vật phẩm	Tên vật phẩm tìm thấy	Có	Tất cả	cá heo DVD Harry porter
2	Giá	Giá của vật phẩm	Có	Gồm các chữ số từ 0 đến 9, mỗi 3 chữ số từ cuối sẽ ngăn cách bởi dấu phẩy	400,000

## 9. Hậu điều kiện

Không có



*Biểu đồ hoạt động UseCase “Tìm kiếm sản phẩm”*

## 4.6 Specification of Use case UC004 – “Quản lý giỏ hàng”

### 1. Use case code

UC003

### 2. Brief Description

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

### 3. Actors

Khách hàng

### 4. Preconditions

Không

## 5. Basic Flow of Events (Success – Chọn sản phẩm mua)

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. Alternative flows

**Table N-Alternative flows of events for UC Place order**

No	Location	Condition	Action	Resume location
7.	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
8.	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
9.	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. Input data

**Table A-Input data of ...**

No	Data fields	Description	Mandatory	Valid condition	Example
7.					

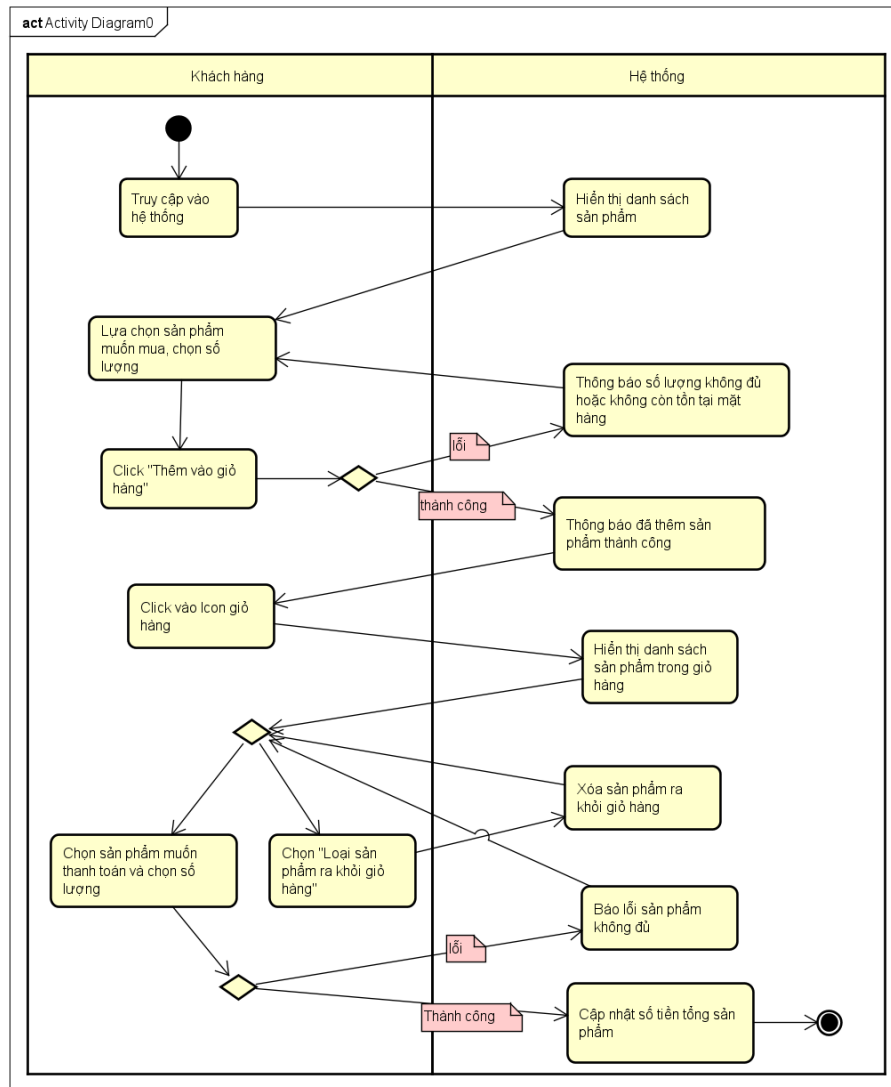
## 8. Output data

**Table B-Output data of ...**

No	Data fields	Description	Display format	Example
16.				

## 9. Postconditions

Không



*Biểu đồ hoạt động UseCase “Quản lý giỏ hàng”*

## **5 Supplementary specification**

### **5.1 Functionality**

Đặt hàng, đặt hàng nhanh, hủy đơn hàng, thanh toán, quản lý người dùng, phê duyệt đơn hàng, cập nhật giỏ hàng, tìm kiếm sản phẩm, CRUD sản phẩm.

### **5.2 Usability**

Hệ thống hoạt động 24/7

### **5.3 Reliability**

Hệ thống có thể hoạt động trở lại bình thường trong vòng 1 giờ sau khi xảy ra lỗi.

### **5.4 Performance**

Có thể cho phép phục vụ 1000 khách hàng cùng lúc mà hiệu suất không bị giảm đáng kể, đồng thời có thể hoạt động 300 giờ liên tục không hỏng hóc, Thời gian đáp ứng của hệ thống tối đa là 2 giây khi bình thường hoặc 3 giây lúc cao điểm.

### **5.5 Maintainability**

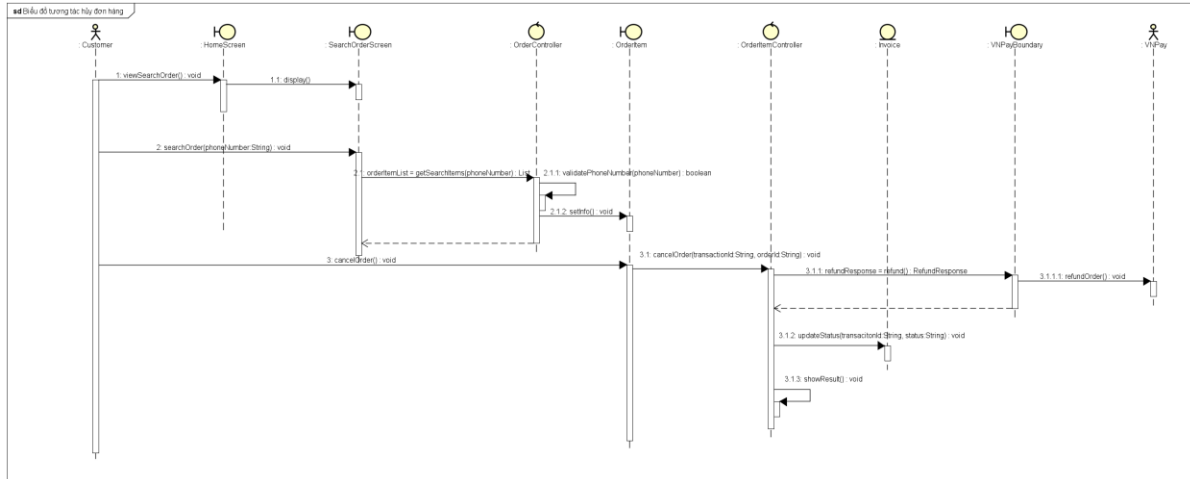
Hệ thống có thể hoạt động trở lại bình thường trong vòng 1 giờ sau khi xảy ra lỗi

### **5.6 Design Constraints**

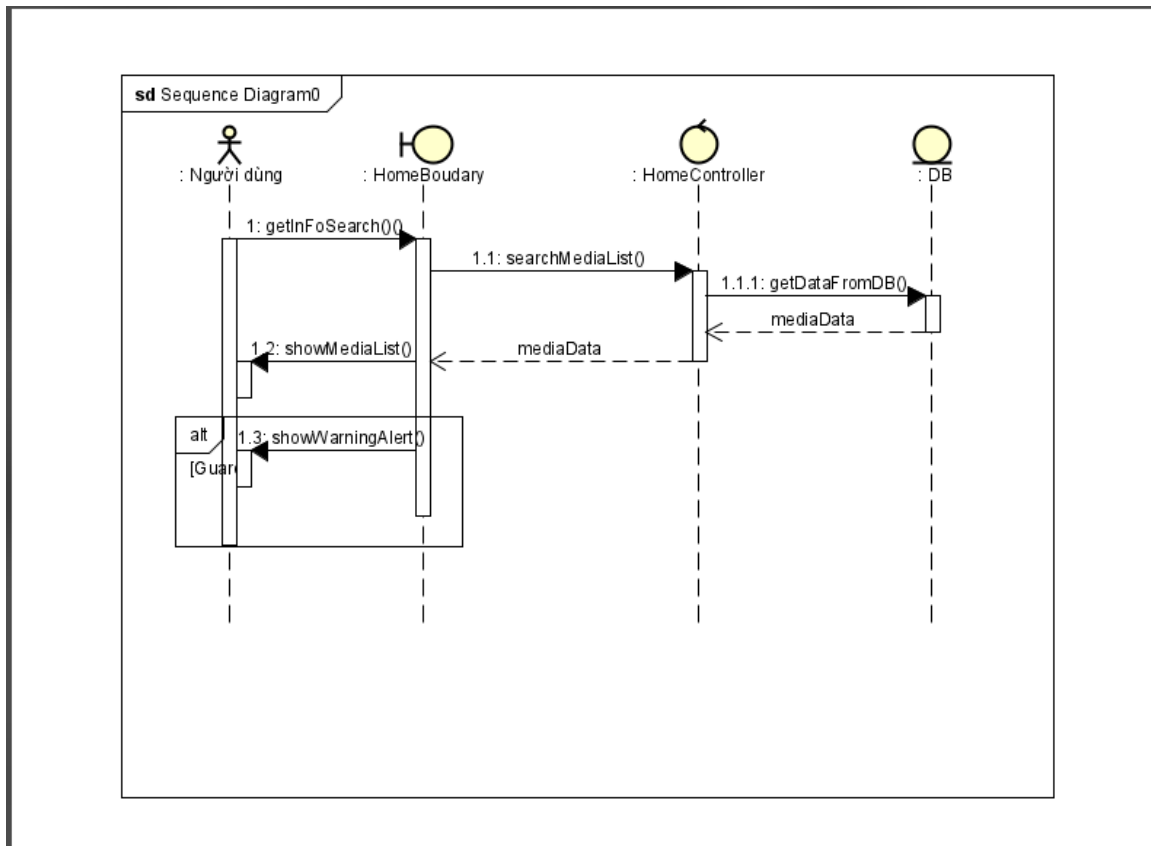
Không có

## 6 INTERACTION DIAGRAMS

### 6.1 Sequence diagram UseCase “Hủy đơn hàng”

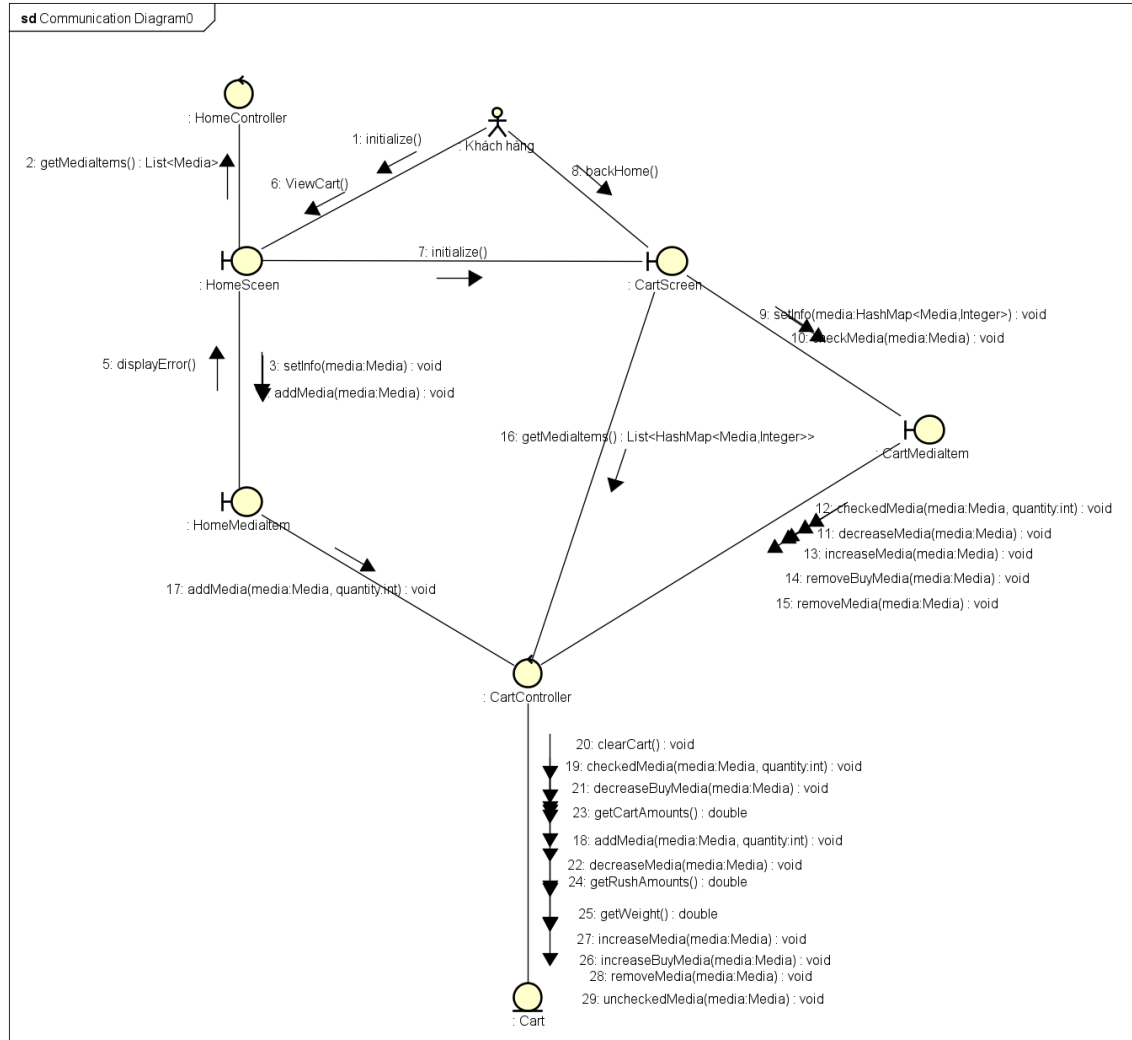


### 6.2 Sequence diagram UseCase “Tìm kiếm sản phẩm”

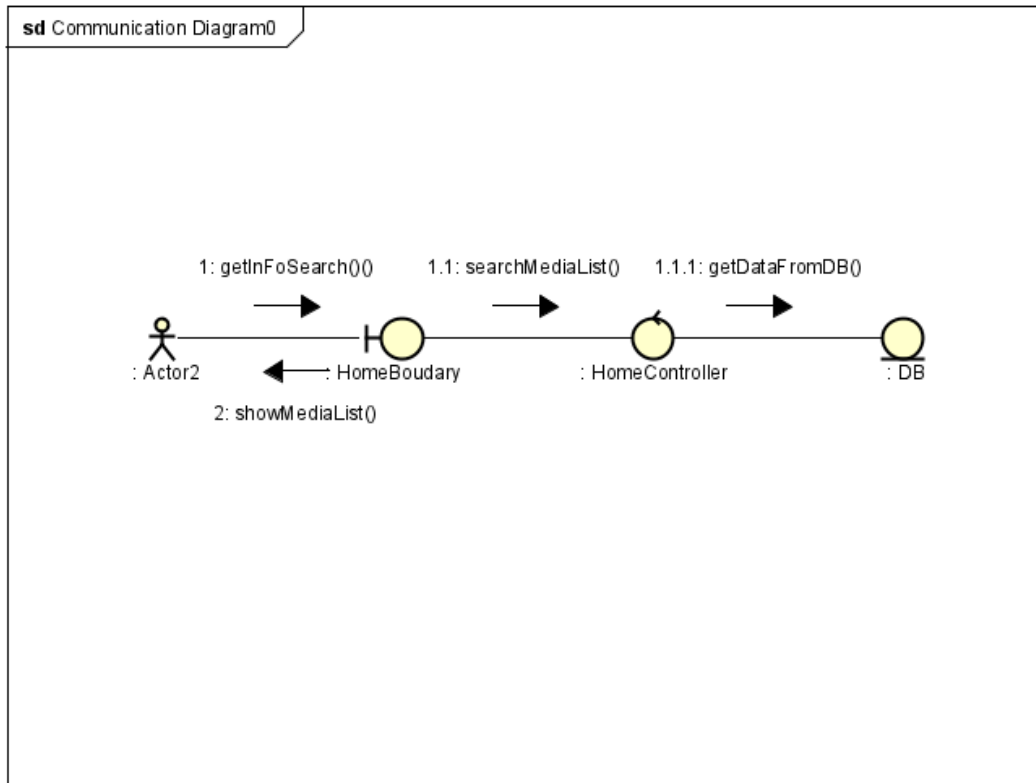




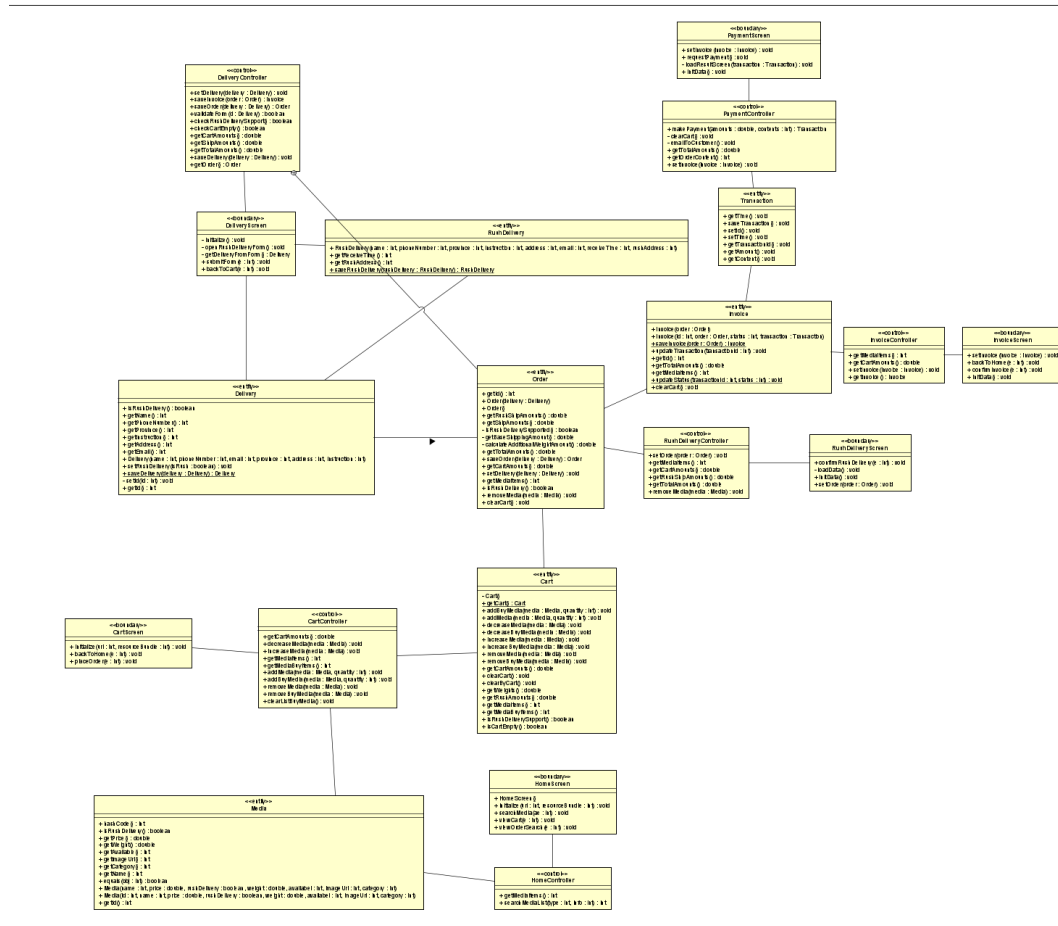
### 6.3 Communication diagram UseCase “Quản lý giỏ hàng”



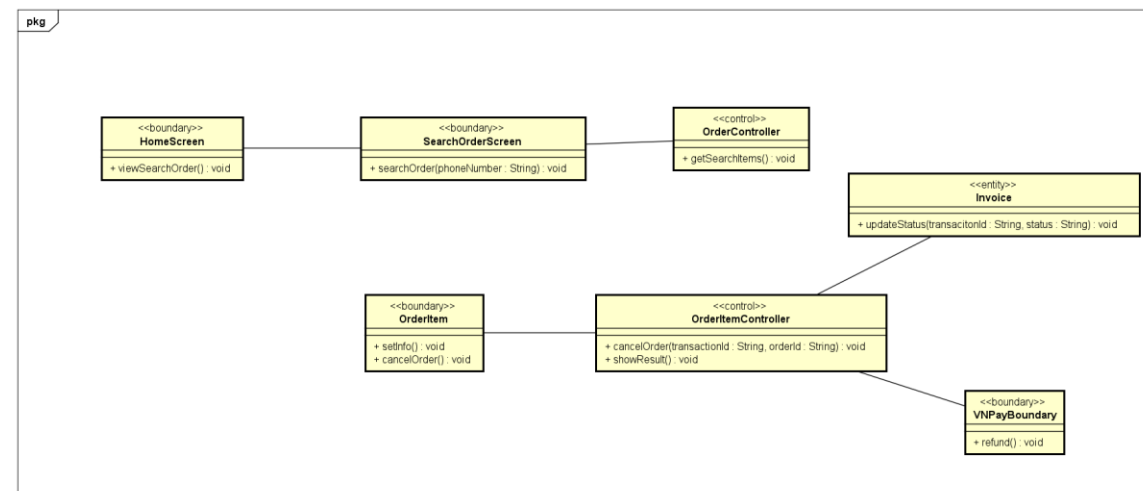
## 6.4 Communication diagram UseCase “Tìm kiếm sản phẩm”



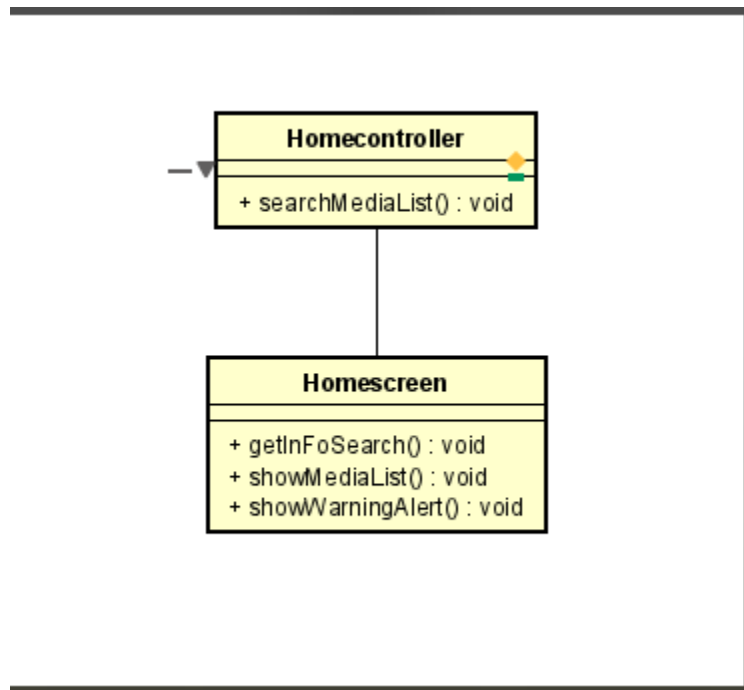
## 7.1 ANALYSIS CLASS DIAGRAM Tổng quan



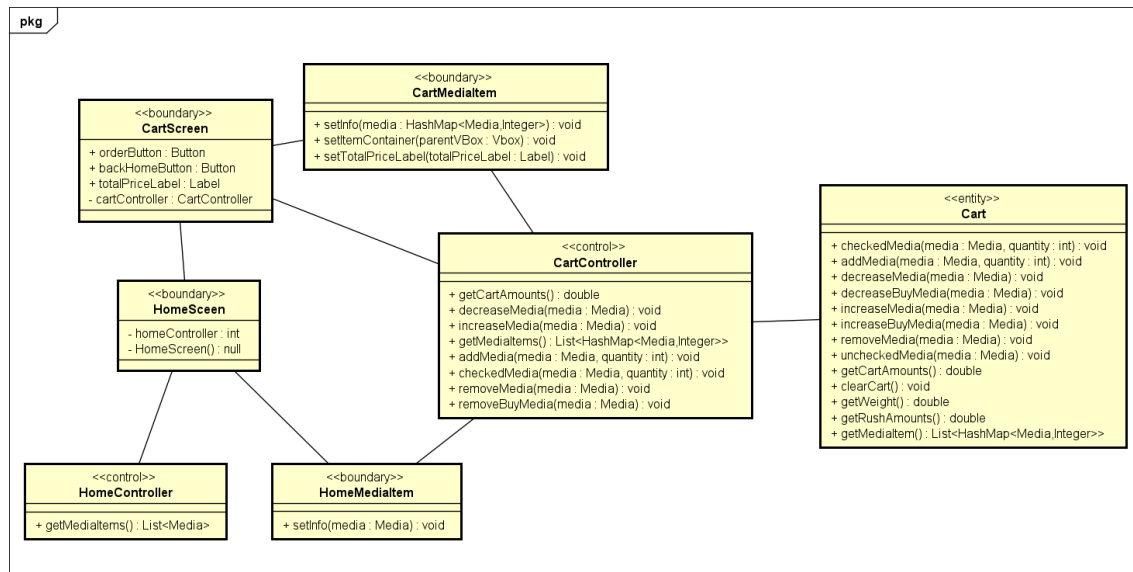
## 7.2 ANALYSIS CLASS DIAGRAM UseCase “Hủy đơn hàng”



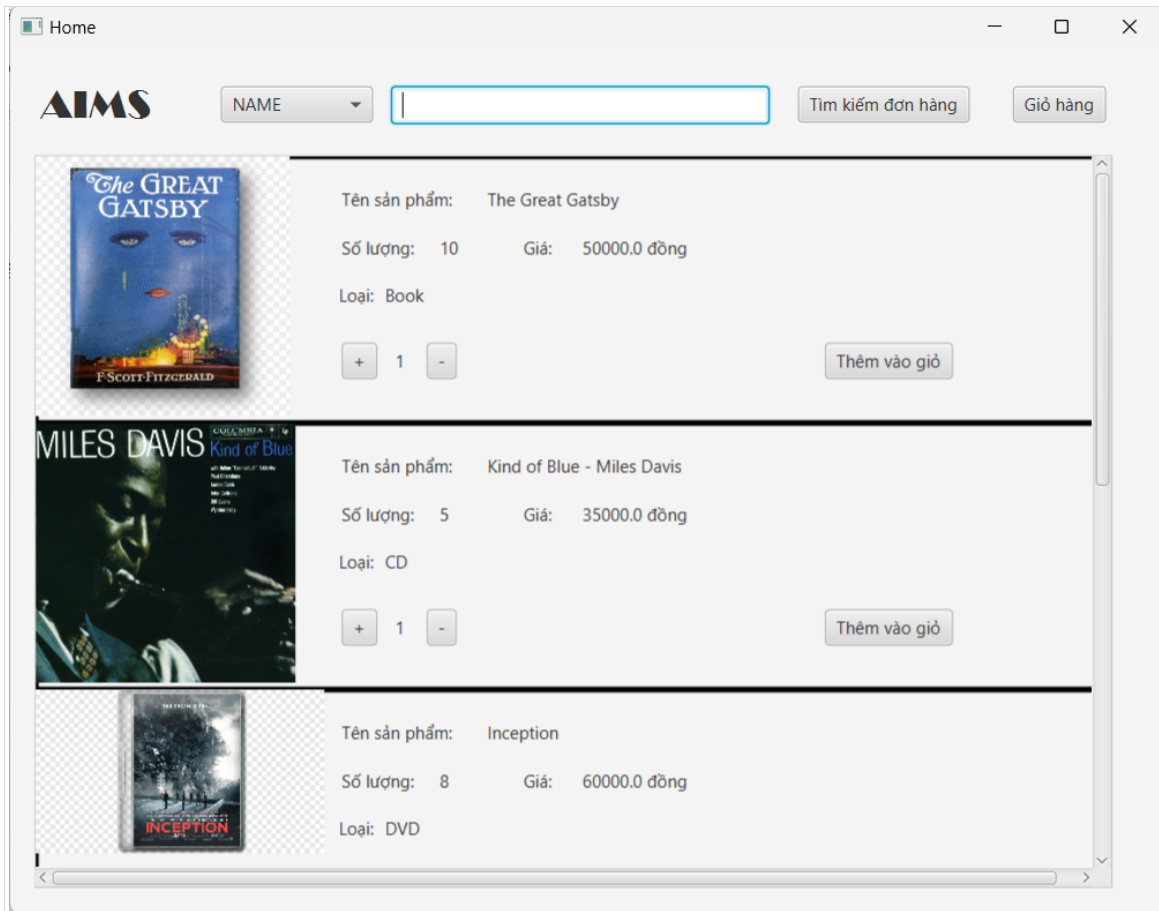
### 7.3 ANALYSIS CLASS DIAGRAM “Tìm kiếm sản phẩm”



### 7.4 ANALYSIS CLASS DIAGRAM UseCase “Quản lý giỏ hàng”



## 8 INTERFACE DESIGN



### Home Screen Specification

Screen Element	Control	Operation	Function
Product List	Interactive List	Scrolling/Clicking	Display products available for purchase
Quantity Selector	Input Field/Button	Writing/Clicking	Select the quantity of the product
Add to Cart Button	Button	Clicking	Add the selected quantity of product to the cart

Back

Order search

Phone number

09123456789

Search

Order ID: 65

Total amounts: 100000.0

Status: pending

Order ID:

Receiver: Nông Thanh Ngọc

Phone number: 09123456789

Email: 123@gmail.com

Province: Hà Nội

Address: 123

Cancel Order

Order Search Screen Specification

Screen Element	Control	Operation	Function
Phone Number Field	Text Field	Writing	Enter a phone number to search for orders
Search Button	Button	Clicking	Initiate search for orders associated with the phone number



Họ và tên:

Số điện thoại:

Email

Tỉnh

Hà Nội

Địa chỉ:

Chỉ dẫn giao hàng: (Không bắt buộc)

Thành tiền sản phẩm: 57000.0 đồng

Tiền giao hàng: 37000.0 đồng

Tổng tiền: 94000.0 đồng

☐ Giao hàng nhanh

Thời gian nhận:

Địa chỉ nhận hàng nhanh:

Hủy

Xác nhận

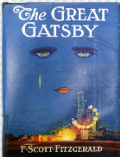
### Delivery Screen Specification

Screen Element	Control	Operation	Function
Recipient Name Field	Text Field	Writing	Enter the name of the recipient for delivery
Address Field	Text Field	Writing	Enter the delivery address
Delivery Options	Checkbox/Dropdown	Selecting/Toggling	Choose delivery options and services



Home

# Invoice Screen




Tên sản phẩm: The Great Gatsby

Số lượng: 10

Loại: Book

Giá: 50000.0 đồng

**Số lượng đặt hàng: 3**



Tên sản phẩm: Kind of Blue - Miles Davis

Số lượng: 5

Loại: CD

Giá: 35000.0 đồng

**Số lượng đặt hàng: 2**

Tổng tiền:

220000.0 đồng

Xác nhận

Hủy

## Invoice Screen Specification

Screen Element	Control	Operation	Function
Itemized List	Interactive List	Display	Show a list of purchased items with prices
Total Amount	Label	Display	Display the total amount due
Confirm Payment Button	Button	Clicking	Confirm the purchase and proceed to payment

## Payment Screen

Content: Paying for order 1 for 94000

Total Price: 94000.0

Thanh toán

### Payment Screen Specification

Screen Element	Control	Operation	Function
Total Price	Label	Display	Show the total price to be paid
Payment Button	Button	Clicking	Initiate the payment process

## Payment Result

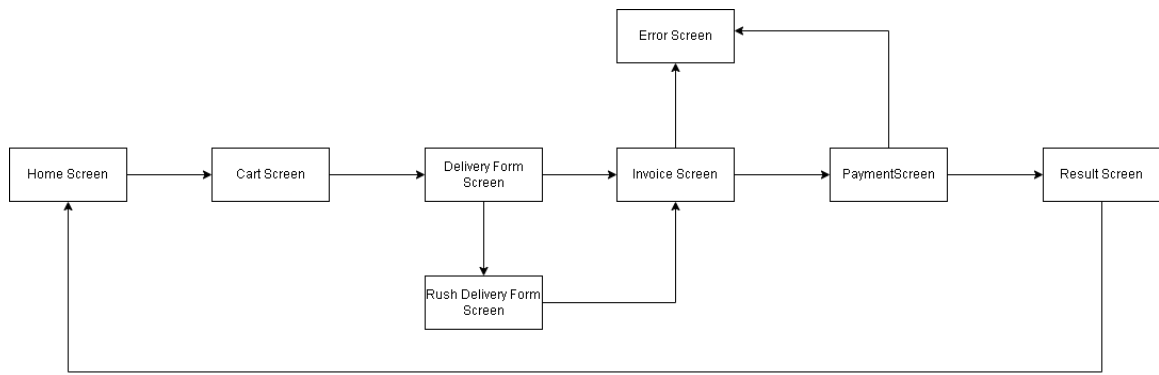
**Giao dịch thất bại**

Payment was canceled

Quay lại màn hình chính

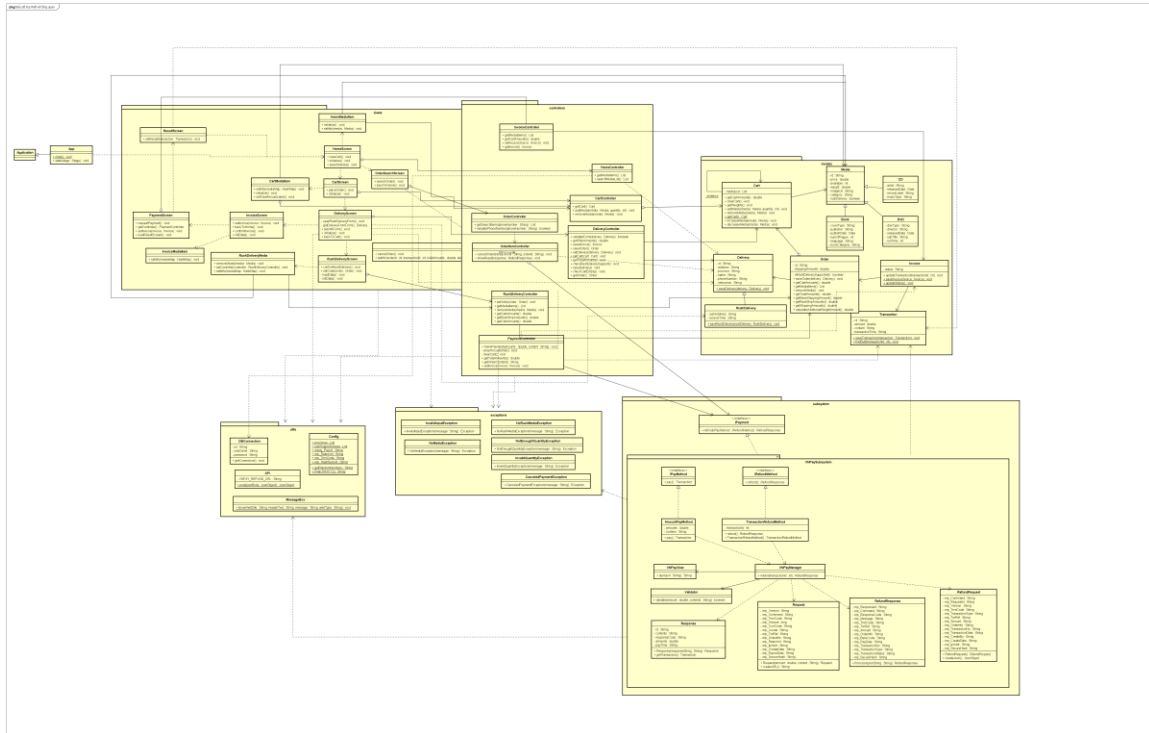
### Payment Result Screen Specification

Screen Element	Control	Operation	Function
Failure Message	Label	Display	Show that the payment process was canceled
Return Home Button	Button	Clicking	Navigate back to the main screen

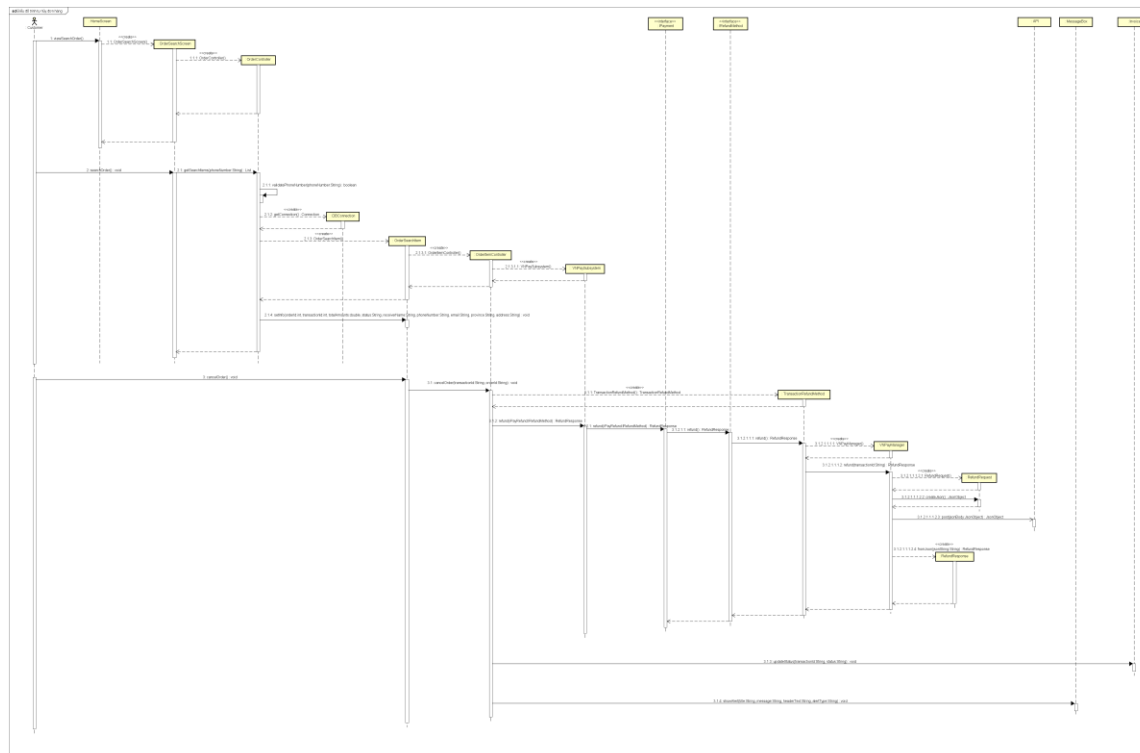
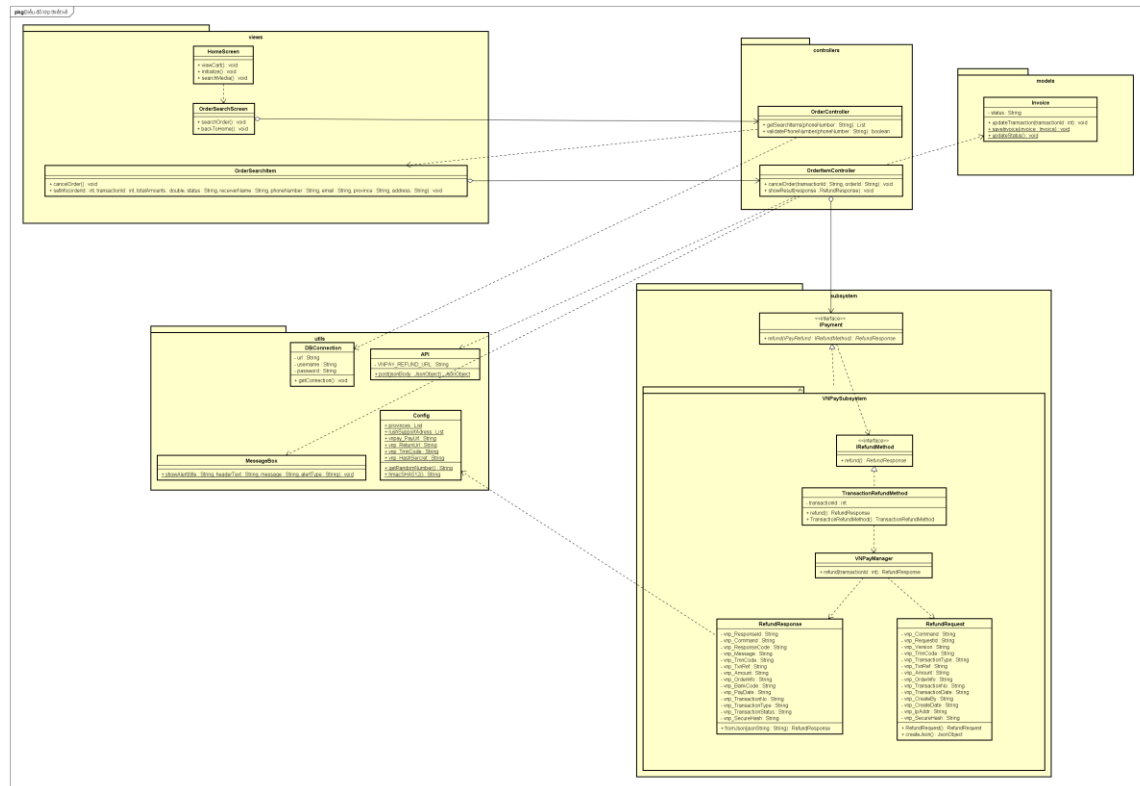


## 9 DETAIL CLASS DIAGRAM

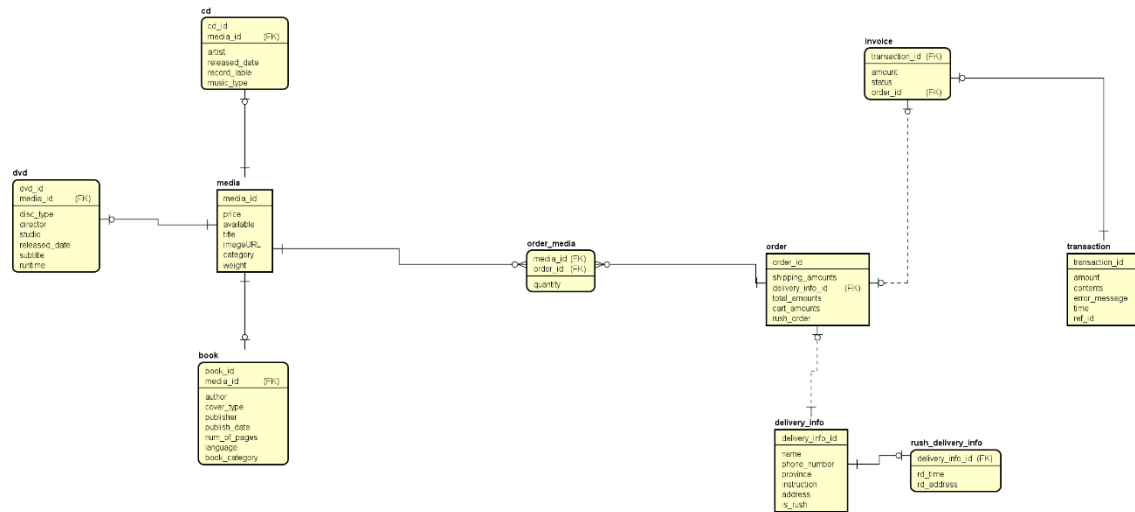
### 9.1 Lớp thiết kế tổng quan.



## 9.2 Lớp thiết kế cho hủy đơn hàng.

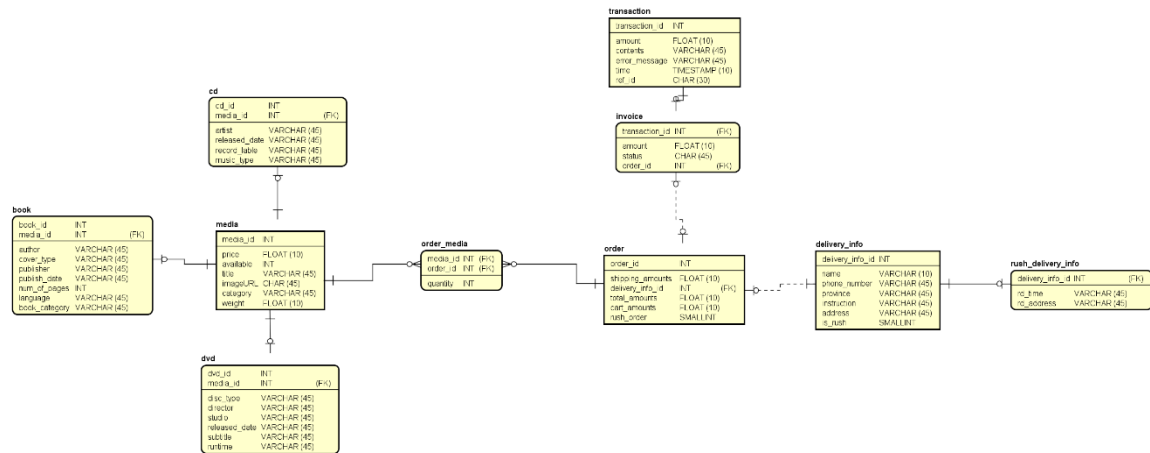


## 9.3 CONCEPTUAL DATA MODEL



# 10 DATABASE DESIGN

## 10.1.1 Logical Data Model



## 10.1.2 Physical Data Model

### - Media

#	Khóa chính	Khóa ngoài	Thuộc tính	Kiểu dữ liệu	Bắt buộc	Mô tả
1	x		Media_id	int	có	
2			category	Nvarchar(50)	có	
3			Price	double	có	
4			available	int	có	
5			Title	Nvarchar(50)	có	
6			weight	double	có	
7			imageUrl	Nvarchar(200)	có	

### - CD/Đĩa than

#	Khóa chính	Khóa ngoài	Thuộc tính	Kiểu dữ liệu	Bắt buộc	Mô tả
1	x		id	int	có	
2		x	Media_id	Nvarchar(200)	có	
3			artists	Nvarchar(50)	có	



<b>4</b>			<b>recordLabel</b>	<b>Nvarchar(50)</b>	<b>có</b>	
<b>5</b>			<b>musicType</b>	<b>Nvarchar(50)</b>	<b>có</b>	
<b>6</b>			<b>releaseDate</b>	<b>datetime</b>	<b>có</b>	

- **Book**

<b>#</b>	<b>Khóa chính</b>	<b>Khóa ngoài</b>	<b>Thuộc tính</b>	<b>Kiểu dữ liệu</b>	<b>Bắt buộc</b>	<b>Mô tả</b>
<b>1</b>	<b>x</b>		<b>Book_id</b>	<b>int</b>	<b>có</b>	
<b>2</b>		<b>x</b>	<b>Media_id</b>	<b>Int</b>	<b>có</b>	
<b>3</b>			<b>author</b>	<b>Nvarchar(50)</b>	<b>có</b>	
<b>4</b>			<b>coverType</b>	<b>Nvarchar(50)</b>	<b>có</b>	
<b>5</b>			<b>publisher</b>	<b>Nvarchar(50)</b>	<b>có</b>	
<b>6</b>			<b>publishDate</b>	<b>datetime</b>	<b>có</b>	
<b>7</b>			<b>numberOfPages</b>	<b>int</b>	<b>có</b>	
<b>8</b>			<b>language</b>	<b>Nvarchar(50)</b>	<b>có</b>	
<b>9</b>			<b>bookCategory</b>	<b>Nvarchar(50)</b>	<b>có</b>	

- **DVD**

<b>#</b>	<b>Khóa chính</b>	<b>Khóa ngoài</b>	<b>Thuộc tính</b>	<b>Kiểu dữ liệu</b>	<b>Bắt buộc</b>	<b>Mô tả</b>
<b>1</b>	<b>x</b>		<b>id</b>	<b>int</b>	<b>có</b>	
<b>2</b>		<b>x</b>	<b>Media_id</b>	<b>int</b>	<b>có</b>	
<b>3</b>			<b>discType</b>	<b>Nvarchar(50)</b>	<b>có</b>	
<b>4</b>			<b>director</b>	<b>Nvarchar(50)</b>	<b>có</b>	
<b>5</b>			<b>runtime</b>	<b>datetime</b>	<b>có</b>	
<b>6</b>			<b>studio</b>	<b>Nvarchar(50)</b>	<b>có</b>	
<b>7</b>			<b>subtitle</b>	<b>Nvarchar(50)</b>	<b>có</b>	
<b>8</b>			<b>releasedDate</b>	<b>datetime</b>	<b>có</b>	

- **Delivery**

#	Khóa chính	Khóa ngoài	Thuộc tính	Kiểu dữ liệu	Bắt buộc	Mô tả
1	x		id	int	có	
2			name	Nvarchar(50)	có	
3			address	Nvarchar(100)	có	
4			province	Nvarchar(50)	có	
5			phoneNumber	Nvarchar(10)	có	
6			deliveryIntruction	Nvarchar(50)	có	
7			isRush	boolean	có	

- **Order**

#	Khóa chính	Khóa ngoài	Thuộc tính	Kiểu dữ liệu	Bắt buộc	Mô tả
1	x		id	int	có	
2			shippingFees	int	có	
3		x	deliveryInfoId	int	có	
4			totalAmounts	Float	có	
5			cartAmounts	Float	có	
6			rushOrder	boolean	có	

- **Invoice**

#	Khóa chính	Khóa ngoài	Thuộc tính	Kiểu dữ liệu	Bắt buộc	Mô tả
1	x		id	int	có	
2			totalAmount	int	có	
3		x	orderId	int	có	

4		X	transactionId	Int	có	
---	--	---	---------------	-----	----	--

- Transaction

#	Khóa chính	Khóa ngoài	Thuộc tính	Kiểu dữ liệu	Bắt buộc	Mô tả
1	x		id	int	có	
2			createTime	datetime	có	
3			content	Nvarchar(50)	có	
4			errorMessage	Nvarchar(25)	có	
5			amounts	float	có	
6			refId	int	có	

### 10.1.3 SQL code

```
-- phpMyAdmin SQL Dump
-- version 5.2.1
-- https://www.phpmyadmin.net/
--
-- Host: 127.0.0.1
-- Generation Time: Dec 31, 2023 at 06:30 PM
-- Server version: 10.4.32-MariaDB
-- PHP Version: 8.0.30

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
START TRANSACTION;
SET time_zone = "+00:00";

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

--
-- Database: `aims`
--

--
-- Table structure for table `book`
--

CREATE TABLE `book` (
  `book_id` int(11) NOT NULL,
```

```

`media_id` int(11) DEFAULT NULL,
`author` varchar(45) DEFAULT NULL,
`cover_type` varchar(45) DEFAULT NULL,
`publisher` varchar(45) DEFAULT NULL,
`publish_date` varchar(45) DEFAULT NULL,
`num_of_pages` int(11) DEFAULT NULL,
`language` varchar(45) DEFAULT NULL,
`book_category` varchar(45) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

--
-- Dumping data for table `book`
--

INSERT INTO `book` (`book_id`, `media_id`, `author`, `cover_type`, `publisher`,
`publish_date`, `num_of_pages`, `language`, `book_category`) VALUES
(1, 1, 'John Doe', 'Hardcover', 'Example Publisher', '2022-01-01', 300, 'English',
'Fiction'),
(2, 2, 'Jane Smith', 'Paperback', 'Another Publisher', '2021-12-15', 250,
'Spanish', 'Non-Fiction'),
(3, 3, 'Bob Johnson', 'Hardcover', 'Great Books Publishing', '2022-03-10', 400,
'French', 'Mystery');

-- -----

--
-- Table structure for table `cd`
--

CREATE TABLE `cd` (
  `cd_id` int(11) NOT NULL,
  `media_id` int(11) DEFAULT NULL,
  `artist` varchar(45) DEFAULT NULL,
  `released_date` varchar(45) DEFAULT NULL,
  `record_label` varchar(45) DEFAULT NULL,
  `music_type` varchar(45) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

-- -----

--
-- Table structure for table `delivery_info`
--

CREATE TABLE `delivery_info` (
  `delivery_info_id` int(11) NOT NULL,
  `name` varchar(50) DEFAULT NULL,
  `phone_number` varchar(45) DEFAULT NULL,
  `province` varchar(45) DEFAULT NULL,
  `instruction` varchar(45) DEFAULT NULL,
  `address` varchar(45) DEFAULT NULL,
  `email` varchar(45) DEFAULT NULL,
  `is_rush` smallint(6) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

--
-- Dumping data for table `delivery_info`
--

```

```

-- -----
--
-- Table structure for table `dvd`
--

CREATE TABLE `dvd` (
  `dvd_id` int(11) NOT NULL,
  `media_id` int(11) DEFAULT NULL,
  `disc_type` varchar(45) DEFAULT NULL,
  `director` varchar(45) DEFAULT NULL,
  `studio` varchar(45) DEFAULT NULL,
  `released_date` varchar(45) DEFAULT NULL,
  `subtitle` varchar(45) DEFAULT NULL,
  `runtime` varchar(45) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

-- -----
--
-- Table structure for table `invoice`
--

CREATE TABLE `invoice` (
  `invoice_id` int(11) NOT NULL,
  `transaction_id` int(11) DEFAULT NULL,
  `order_id` int(11) DEFAULT NULL,
  `amount` float DEFAULT NULL,
  `status` char(45) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

-- -----
--
-- Table structure for table `media`
--

CREATE TABLE `media` (
  `media_id` int(11) NOT NULL,
  `price` float DEFAULT NULL,
  `available` int(11) DEFAULT NULL,
  `name` varchar(45) DEFAULT NULL,
  `imageURL` char(45) DEFAULT NULL,
  `category` varchar(45) DEFAULT NULL,
  `weight` float DEFAULT NULL,
  `support_rush_delivery` smallint(6) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

--
-- Dumping data for table `media`
--

INSERT INTO `media` (`media_id`, `price`, `available`, `name`, `imageURL`,
`category`, `weight`, `support_rush_delivery`) VALUES
(1, 19000, 50, 'Book Title 1', 'book_image_url_1.jpg', 'Book', 1.5, 1),
(2, 15000, 30, 'Book Title 2', 'book_image_url_2.jpg', 'Book', 1.2, 0),

```

```
(3, 23000, 25, 'Book Title 3', 'book_image_url_3.jpg', 'Book', 1.8, 1);
```

```
-- -----
```

```
--  
-- Table structure for table `order`  
--
```

```
CREATE TABLE `order` (  
  `order_id` int(11) NOT NULL,  
  `delivery_info_id` int(11) DEFAULT NULL,  
  `shipping_amounts` float DEFAULT NULL,  
  `total_amounts` float DEFAULT NULL,  
  `cart_amounts` float DEFAULT NULL,  
  `rush_order` smallint(6) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

```
--  
-- Dumping data for table `order`  
--
```

```
-- -----
```

```
--  
-- Table structure for table `order_media`  
--
```

```
CREATE TABLE `order_media` (  
  `media_id` int(11) NOT NULL,  
  `order_id` int(11) NOT NULL,  
  `delivery_info_id` int(11) NOT NULL,  
  `quantity` int(11) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

```
CREATE TABLE `rush_delivery_info` (  
  `delivery_info_id` int(11) NOT NULL,  
  `rd_time` varchar(45) DEFAULT NULL,  
  `rd_address` varchar(45) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

```
-- -----
```

```
--  
-- Table structure for table `transaction`  
--
```

```
CREATE TABLE `transaction` (  
  `transaction_id` int(11) NOT NULL,  
  `amount` float DEFAULT NULL,  
  `contents` varchar(45) DEFAULT NULL,  
  `error_message` varchar(45) DEFAULT NULL,  
  `time` timestamp(6) NOT NULL DEFAULT current_timestamp(6) ON UPDATE  
current_timestamp(6),  
  `ref_id` int(10) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

```

--
-- Indexes for dumped tables
--

--
-- Indexes for table `book`
--
ALTER TABLE `book`
  ADD PRIMARY KEY (`book_id`),
  ADD KEY `media_id` (`media_id`);

--
-- Indexes for table `cd`
--
ALTER TABLE `cd`
  ADD PRIMARY KEY (`cd_id`),
  ADD KEY `media_id` (`media_id`);

--
-- Indexes for table `delivery_info`
--
ALTER TABLE `delivery_info`
  ADD PRIMARY KEY (`delivery_info_id`);

--
-- Indexes for table `dvd`
--
ALTER TABLE `dvd`
  ADD PRIMARY KEY (`dvd_id`),
  ADD KEY `media_id` (`media_id`);

--
-- Indexes for table `invoice`
--
ALTER TABLE `invoice`
  ADD PRIMARY KEY (`invoice_id`),
  ADD UNIQUE KEY `transaction_id` (`transaction_id`),
  ADD KEY `order_id` (`order_id`);

--
-- Indexes for table `media`
--
ALTER TABLE `media`
  ADD PRIMARY KEY (`media_id`);

--
-- Indexes for table `order`
--
ALTER TABLE `order`
  ADD PRIMARY KEY (`order_id`),
  ADD KEY `delivery_info_id` (`delivery_info_id`);

--
-- Indexes for table `order_media`
--
ALTER TABLE `order_media`
  ADD PRIMARY KEY (`media_id`, `order_id`, `delivery_info_id`),

```

```

ADD KEY `order_id` (`order_id`);

--
-- Indexes for table `rush_delivery_info`
--
ALTER TABLE `rush_delivery_info`
  ADD PRIMARY KEY (`delivery_info_id`);

--
-- Indexes for table `transaction`
--
ALTER TABLE `transaction`
  ADD PRIMARY KEY (`transaction_id`);

--
-- AUTO_INCREMENT for dumped tables
--

--
-- AUTO_INCREMENT for table `book`
--
ALTER TABLE `book`
  MODIFY `book_id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=4;

--
-- AUTO_INCREMENT for table `cd`
--
ALTER TABLE `cd`
  MODIFY `cd_id` int(11) NOT NULL AUTO_INCREMENT;

--
-- AUTO_INCREMENT for table `delivery_info`
--
ALTER TABLE `delivery_info`
  MODIFY `delivery_info_id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=53;

--
-- AUTO_INCREMENT for table `dvd`
--
ALTER TABLE `dvd`
  MODIFY `dvd_id` int(11) NOT NULL AUTO_INCREMENT;

--
-- AUTO_INCREMENT for table `invoice`
--
ALTER TABLE `invoice`
  MODIFY `invoice_id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=14;

--
-- AUTO_INCREMENT for table `media`
--
ALTER TABLE `media`
  MODIFY `media_id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=4;

--
-- AUTO_INCREMENT for table `order`
--
ALTER TABLE `order`

```



```

MODIFY `order_id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=51;

--
-- AUTO_INCREMENT for table `transaction`
--
ALTER TABLE `transaction`
  MODIFY `transaction_id` int(11) NOT NULL AUTO_INCREMENT,
  AUTO_INCREMENT=14266178;

--
-- Constraints for dumped tables
--

--
-- Constraints for table `book`
--
ALTER TABLE `book`
  ADD CONSTRAINT `book_ibfk_1` FOREIGN KEY (`media_id`) REFERENCES `media`
  (`media_id`);

--
-- Constraints for table `cd`
--
ALTER TABLE `cd`
  ADD CONSTRAINT `cd_ibfk_1` FOREIGN KEY (`media_id`) REFERENCES `media`
  (`media_id`);

--
-- Constraints for table `dvd`
--
ALTER TABLE `dvd`
  ADD CONSTRAINT `dvd_ibfk_1` FOREIGN KEY (`media_id`) REFERENCES `media`
  (`media_id`);

--
-- Constraints for table `invoice`
--
ALTER TABLE `invoice`
  ADD CONSTRAINT `invoice_ibfk_1` FOREIGN KEY (`transaction_id`) REFERENCES
  `transaction` (`transaction_id`) ON UPDATE CASCADE,
  ADD CONSTRAINT `invoice_ibfk_2` FOREIGN KEY (`order_id`) REFERENCES `order`
  (`order_id`);

--
-- Constraints for table `order`
--
ALTER TABLE `order`
  ADD CONSTRAINT `order_ibfk_1` FOREIGN KEY (`delivery_info_id`) REFERENCES
  `delivery_info` (`delivery_info_id`);

--
-- Constraints for table `order_media`
--
ALTER TABLE `order_media`
  ADD CONSTRAINT `order_media_ibfk_1` FOREIGN KEY (`media_id`) REFERENCES `media`
  (`media_id`),
  ADD CONSTRAINT `order_media_ibfk_2` FOREIGN KEY (`order_id`) REFERENCES `order`
  (`order_id`);

```

```

--
-- Constraints for table `rush_delivery_info`
--
ALTER TABLE `rush_delivery_info`
  ADD CONSTRAINT `rush_delivery_info_ibfk_1` FOREIGN KEY (`delivery_info_id`)
REFERENCES `delivery_info` (`delivery_info_id`);
COMMIT;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;

```