HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY
School of Information and Communications Technology

Final Report
Version 1.1

AIMS: An Internet Media Store Software Design and Construction

Group 02
Trần Xuân Bách 20204634
Nguyễn Ngọc Quỳnh Anh 20204631
Nguyễn Vũ Thục Anh 20204632
Vũ Đức Anh 20193985

Hanoi, 1/2024

Contents

1. Team's members contribution

Name	Role	Contribution
Trần Xuân Bách	Team Leader	View list invoice and refund function
Nguyễn Ngọc Quỳnh Anh	Member	Search, filter and pagination products
Nguyễn Vũ Thục Anh	Member	View detail product
Vũ Đức Anh	Member	View Invoice

2. Use Case Diagrams

2.1. Use Case Diagram

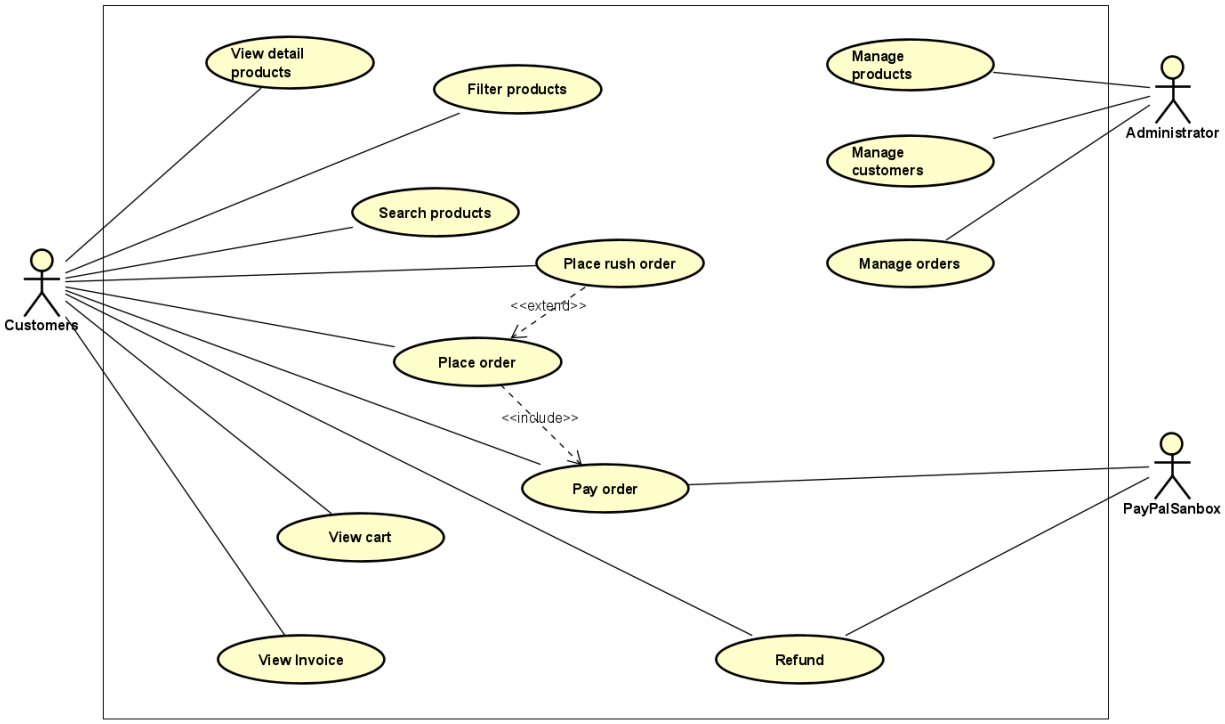


Figure 1: General Use Case Diagram

2.2. Businesss processes

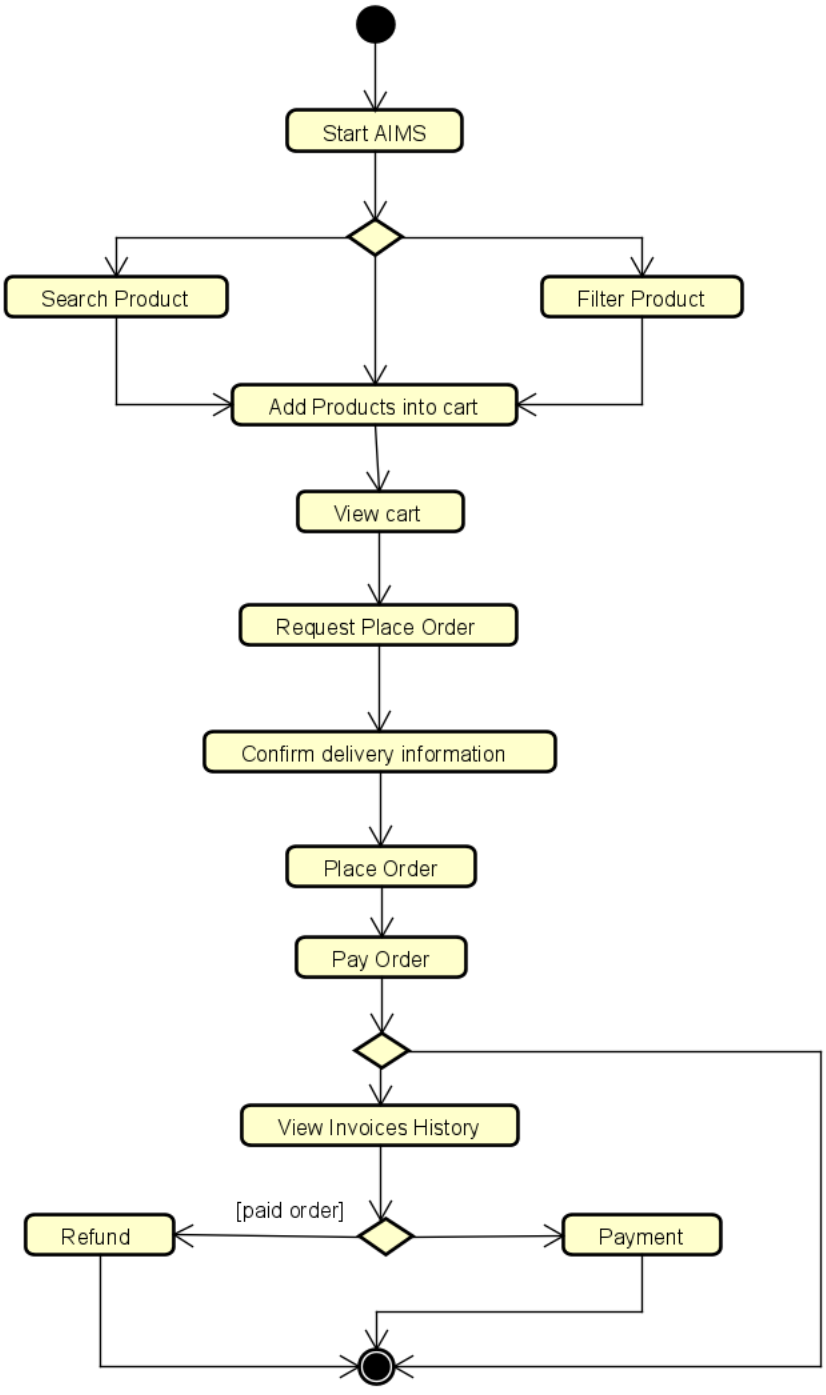


Figure 2: Activity Diagram of customer's usage process

2.3. Case Specification

2.3.1. UC001: View Invoice

2.3.2. UC002: Refund

Use Case "Refund"

1. Use Case code

UC002

2. Brief Description

This use case describes the interaction between Customer and AIMS system when Customer perform order refund.

3. Actors

Customer

4. Preconditions

The customer has viewed invoice

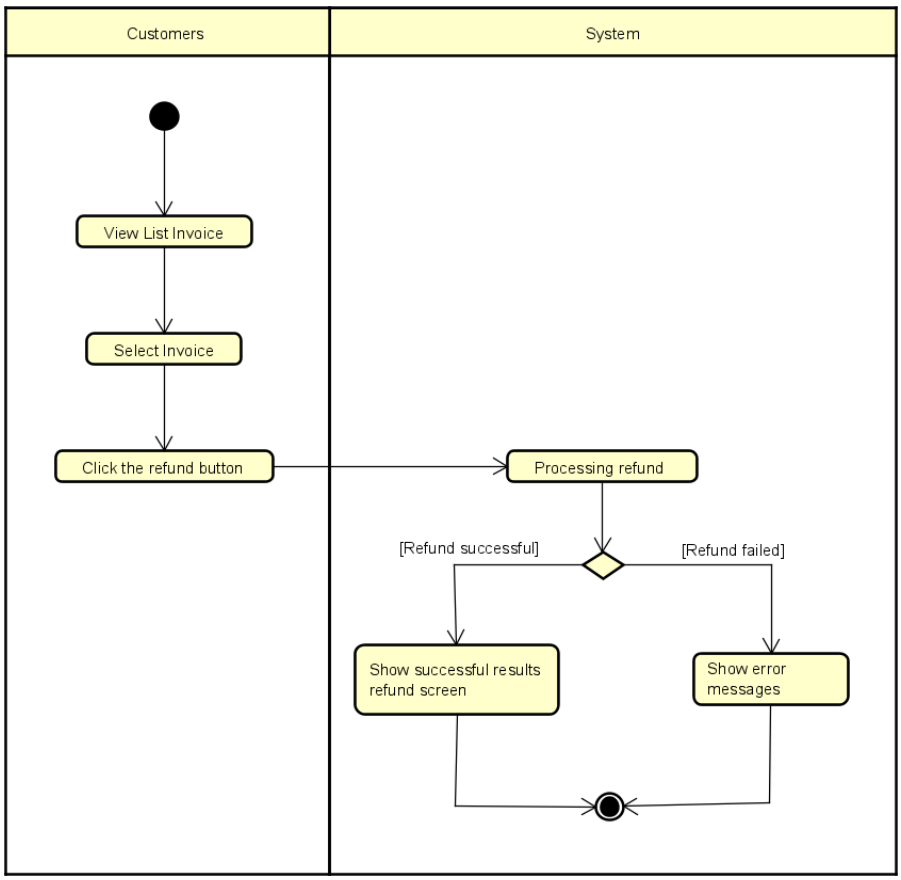
5. Basic Flow of Events

1. The system displays list invoice information (UC001)
2. The customer selects the invoice that needs a refund
3. The system processing the refund
4. The system displays successful refund results

6. Alternative Flows

No	Location	Condition	Action	Resume Location
1.	4	If the invoice has expired for a refund	▪ The system displays an error message screen	
2.		If refund processing fails	▪ The system displays the message: Refund failed	

7. Activity Diagram



2.3.3. UC003: Search Products

Use Case "Search Products"

1. Use Case code

UC003

2. Brief Description

This use case describes the interaction between Customer and AIMS system when Customer wishes to search products

3. Actors

Customer

4. Preconditions

The customer has viewed the product list

5. Basic Flow of Events

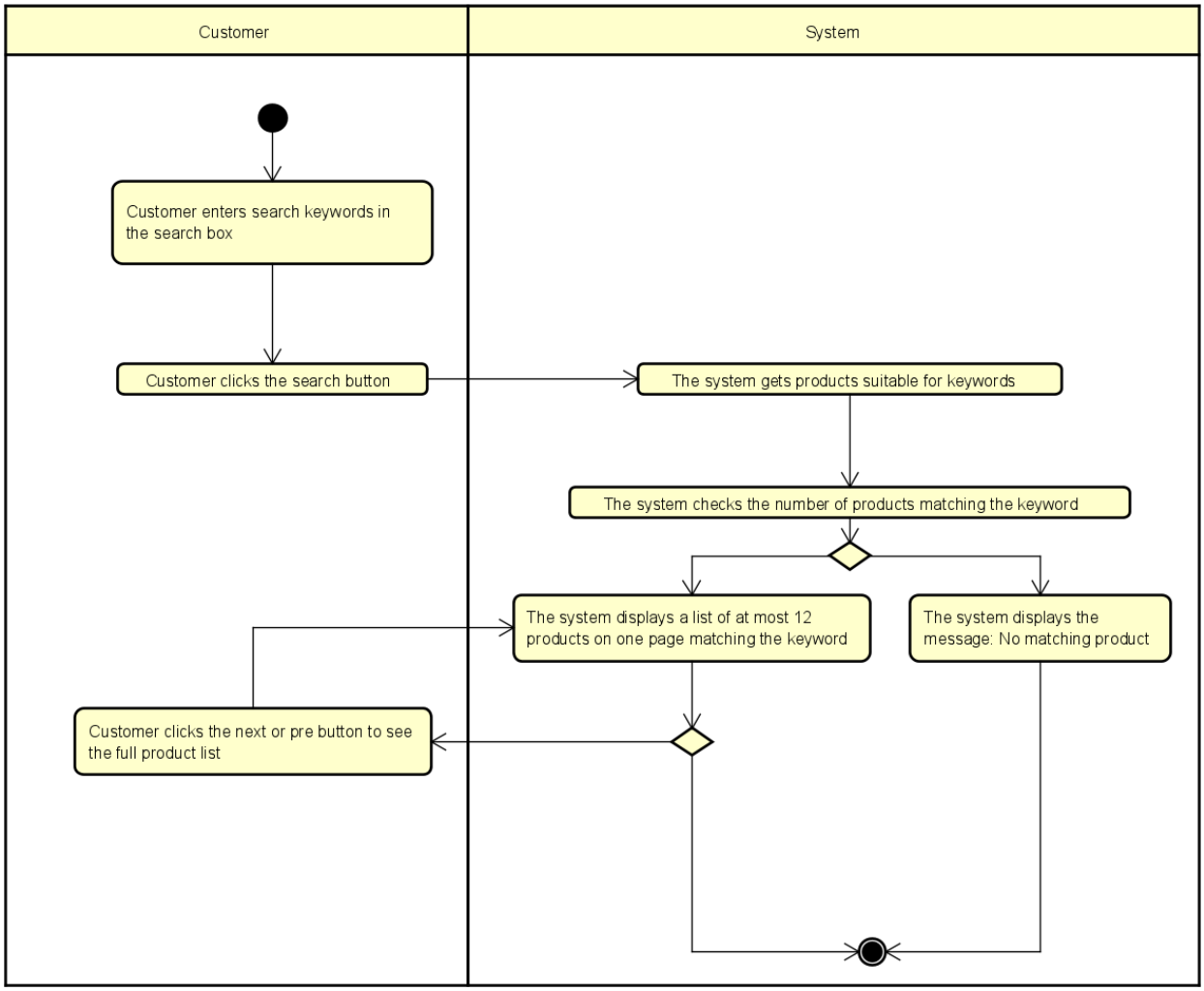
1. Customer enters search keywords in the search box
2. Customer clicks the search button
3. The system gets products suitable for keywords
4. The system checks the number of products matching the keyword
5. The system displays a list of at most 12 products on one page matching the keyword

6. Alternative Flows

Table 4 -Alternative Flow of Use case "Search Products"

No	Location	Condition	Action	Resume Location
1.	5	If there are no products matching the keyword	▪ The system displays the message: There are no matching products	
2.		If there are more than 12 products matching the keyword	▪ Customer clicks the next or pre button to see the full product list	5

7. Activity Diagram



2.3.4. UC004 Filter Products

Use Case “Filter Products”

1. Use Case code

UC004

2. Brief Description

This use case describes the interaction between Customer and AIMS system when Customer wishes to filter products

3. Actors

Customer

4. Preconditions

The customer has viewed the product list

5. Basic Flow of Events

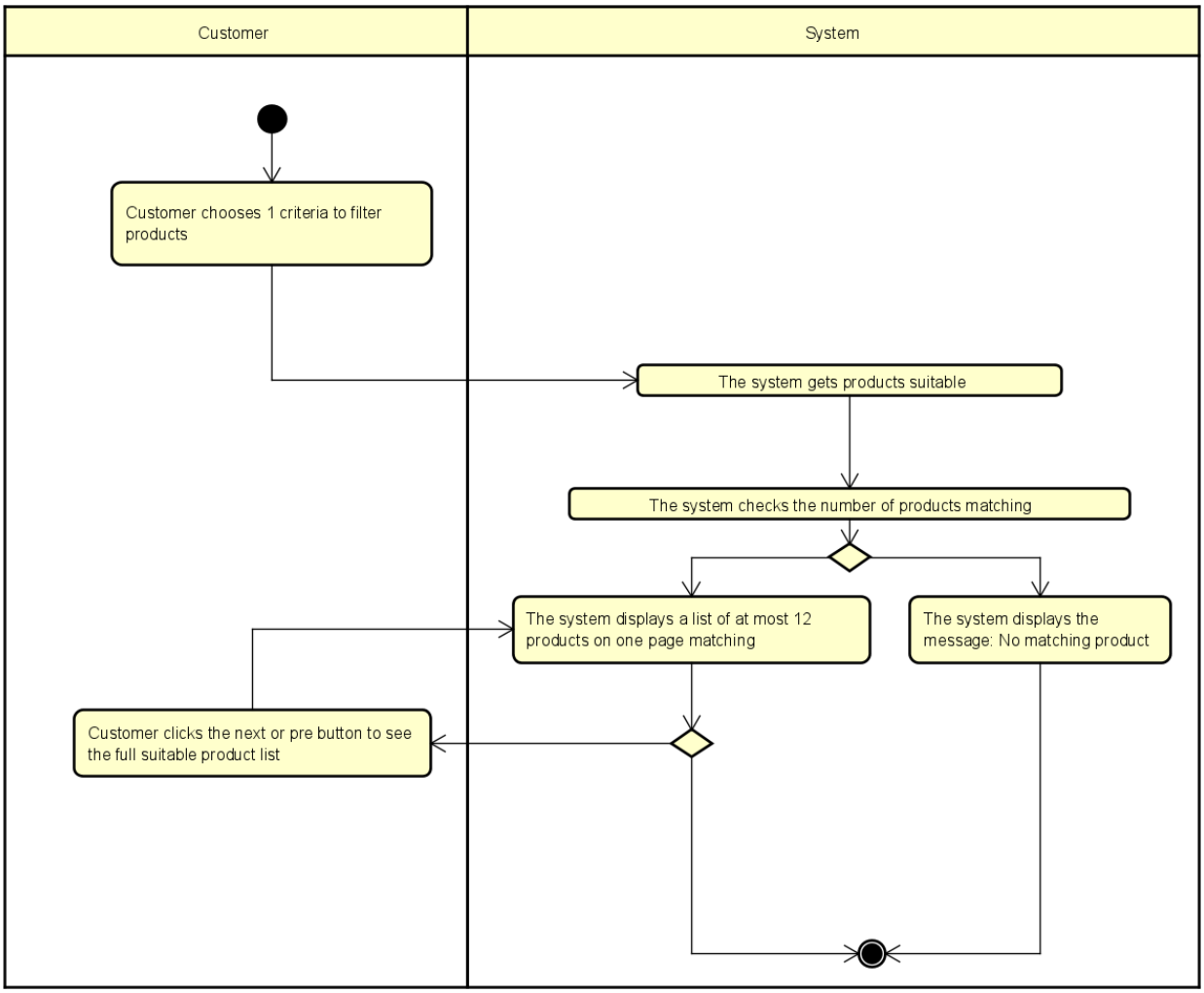
1. Customer chooses 1 criteria to filter products
2. The system gets products suitable
3. The system checks the number of products matching
4. The system displays a list of at most 12 products on one page matching

6. Alternative Flows

Table 4 -Alternative Flow of Use case "Filter Products"

No	Location	Condition	Action	Resume Location
1.	5	If there are no products matching	▪ The system displays the message: There are no matching products	
2.		If there are more than 12 products matching	▪ Customer clicks the next or pre button to see the full product list	4

7. Activity Diagram



2.3.5. UC005 Detail Products

Use Case "Detail Products"

1. Use Case code

UC005

2. Brief Description

This use case describes viewing product details

3. Actors

Customer

4. Preconditions

The customer has viewed detail product

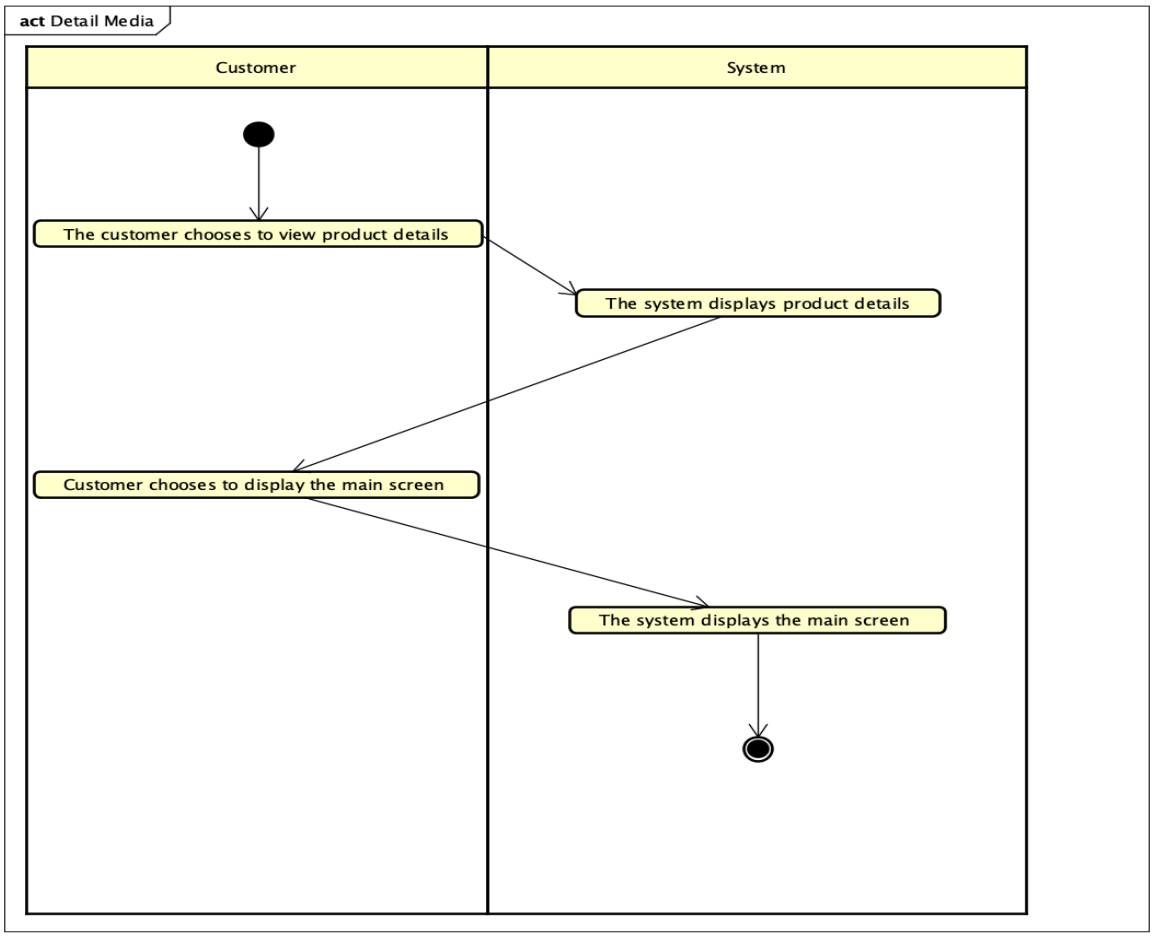
5. Basic Flow of Events

1. The customer chooses to view product details
2. The system displays product details

3. Customer chooses to display the main screen
4. The system displays the main screen

6. Alternative Flows

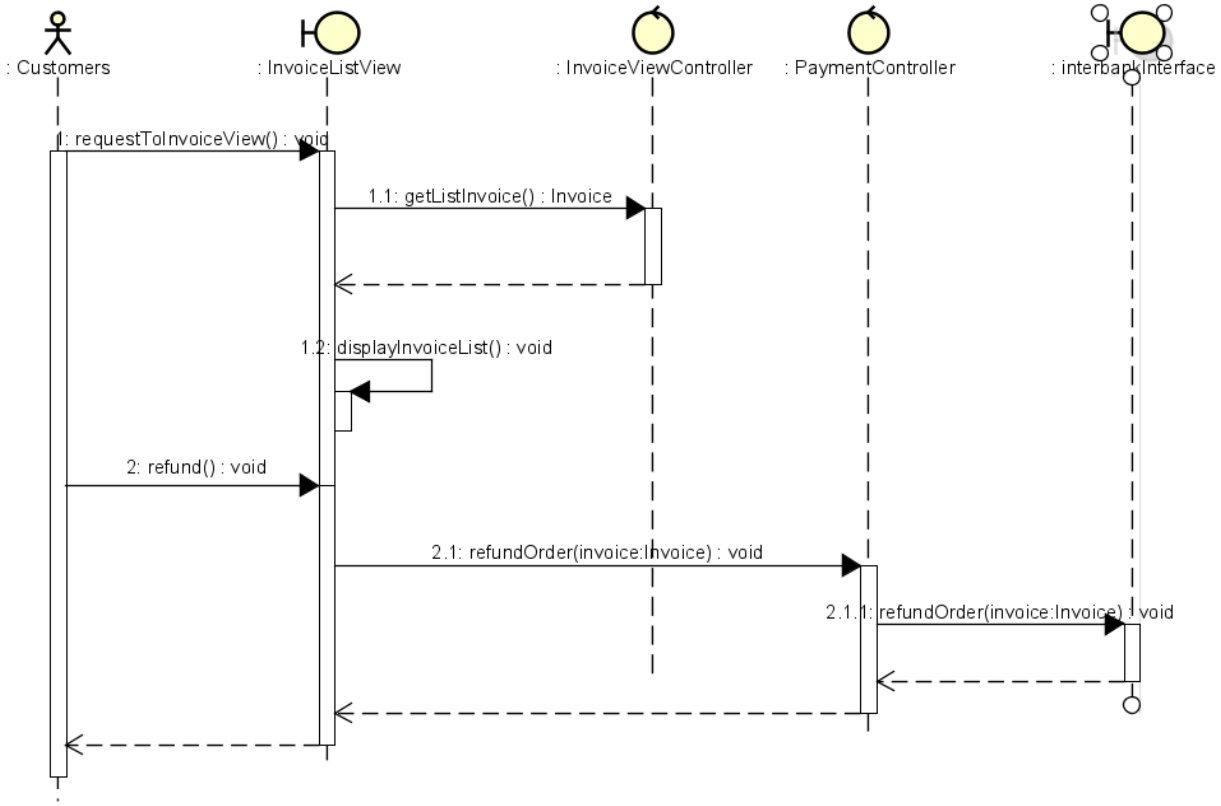
7. Activity Diagram



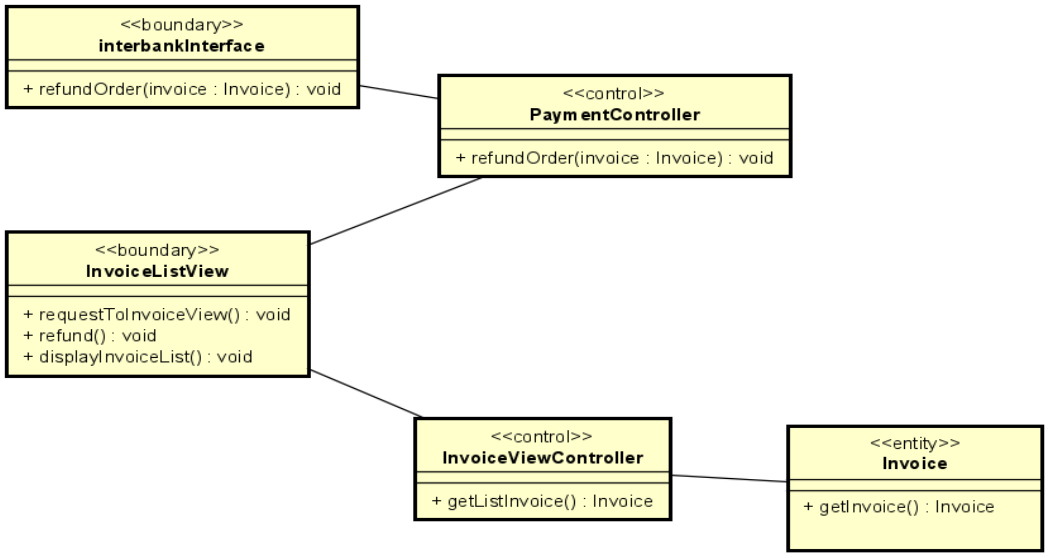
3. Use Case Analysis

3.1. Use Case “Refund”

3.1.1. Sequence Diagram for UC “Refund”

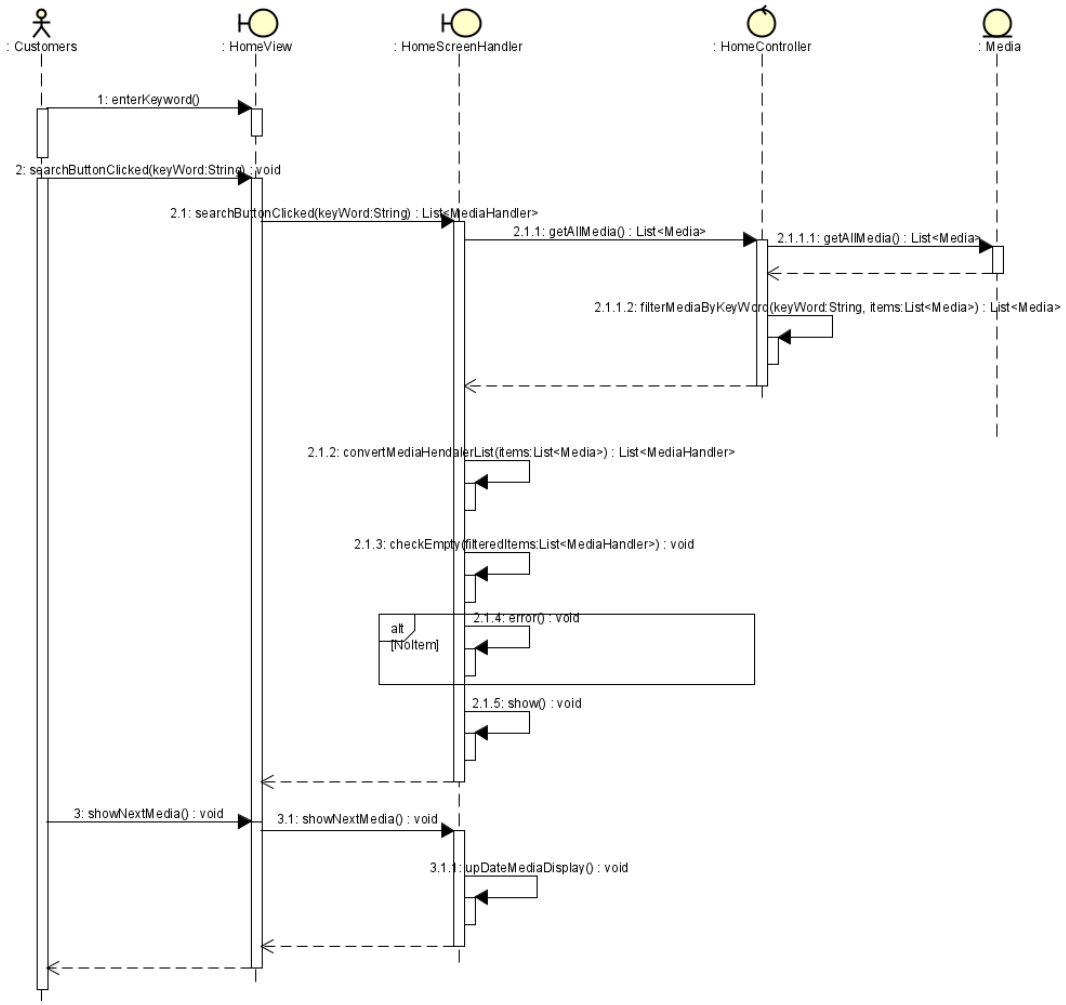


3.1.2. Analysis Class Diagram for UC “Refund”

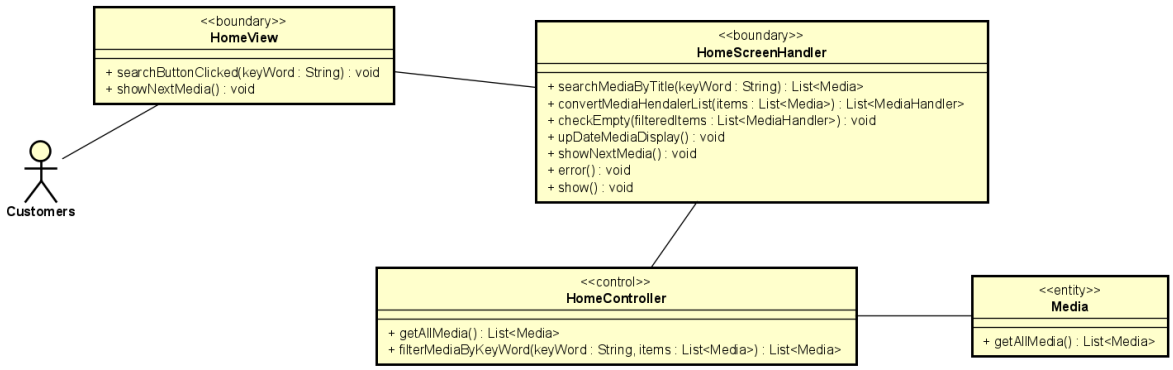


3.2. Use Case “Search Products”

3.2.1. Sequence Diagram for UC “Search Products”

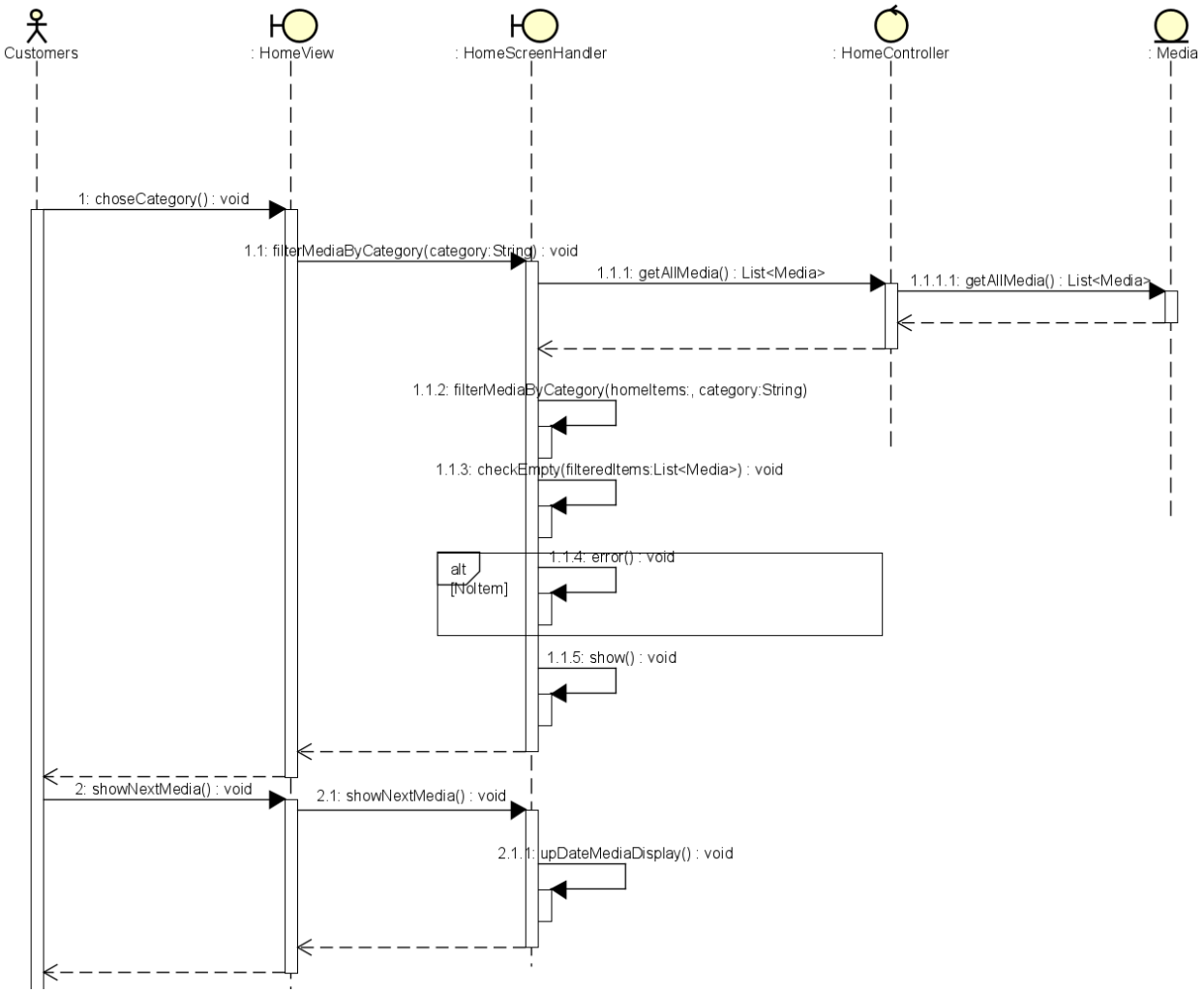


3.2.2. Analysis Class Diagram for UC "Search Products”

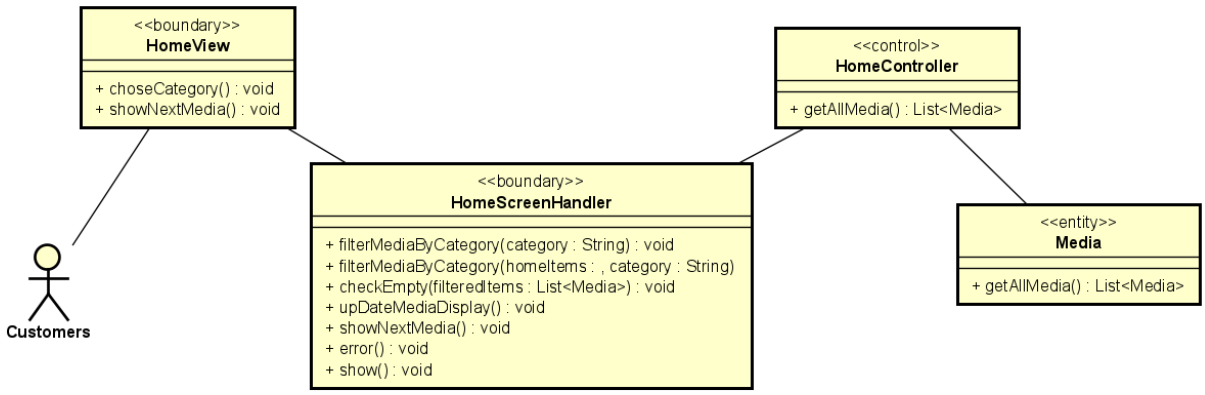


3.3. Use Case “Filter Products”

3.3.1. Sequence Diagram for UC “Filter Products”

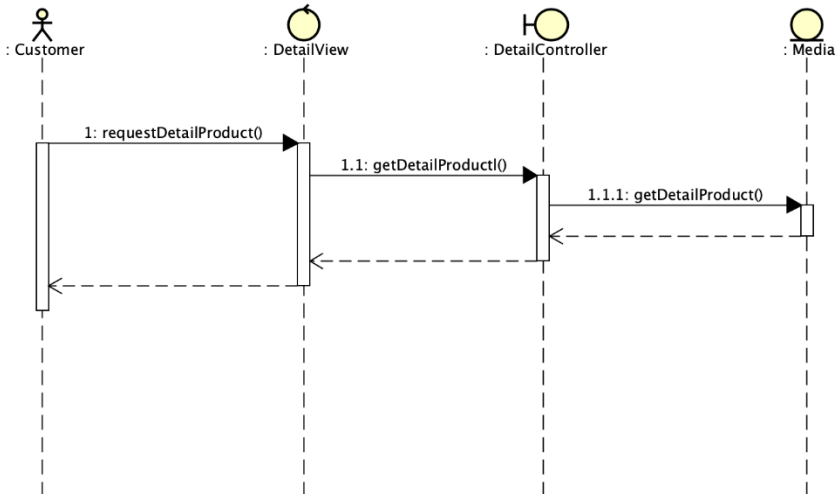


3.3.2. Analysis Class Diagram for UC "Filter Products”

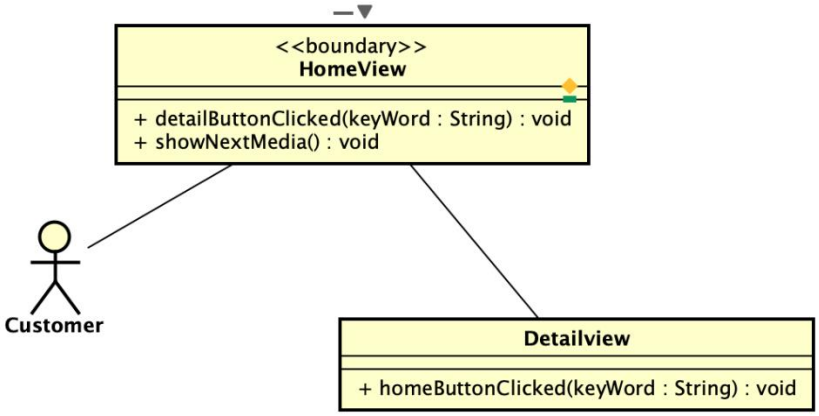


3.4. Use Case “Detail Products”

3.4.1. Sequence Diagram for UC “Detail Products”



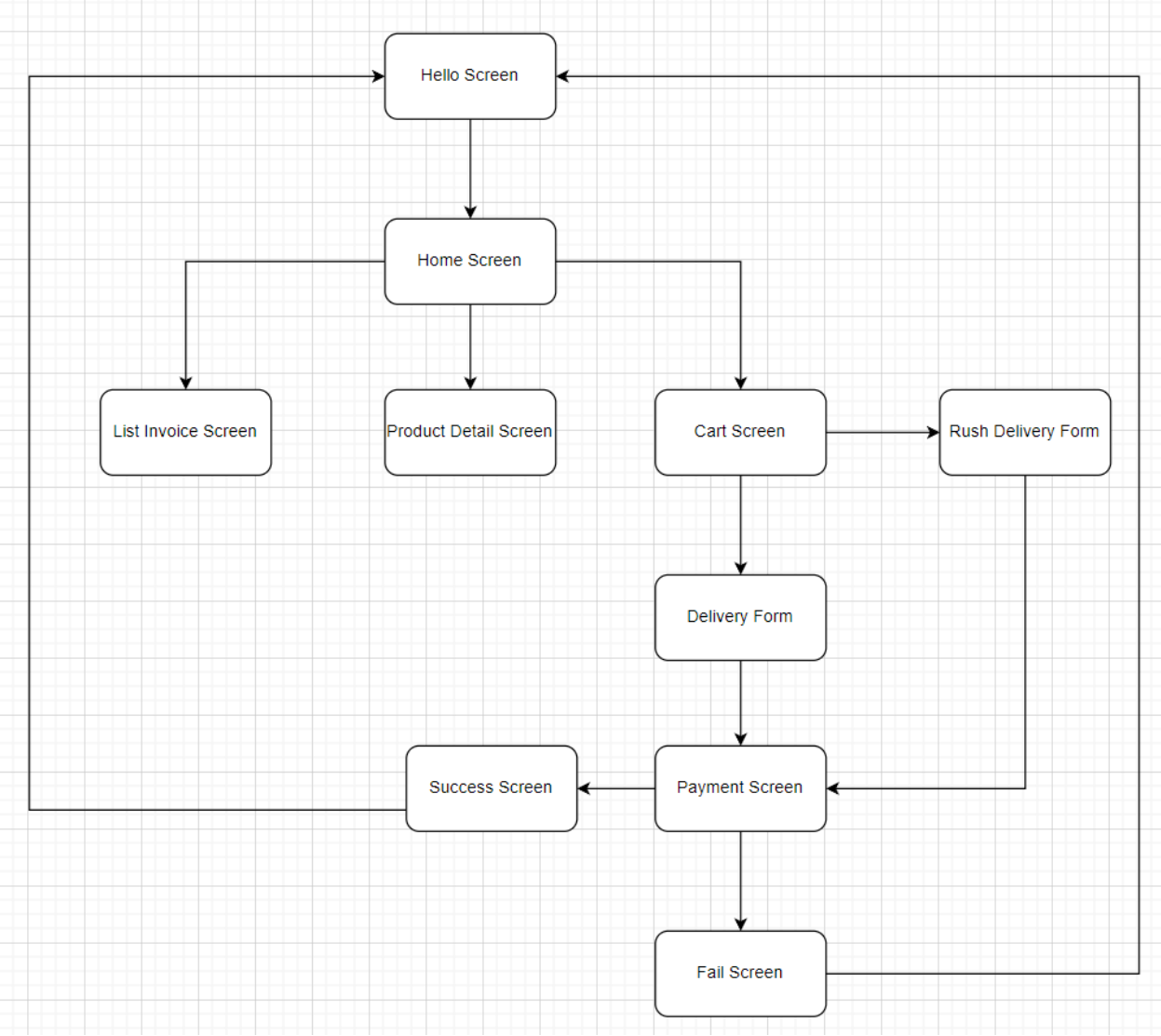
3.4.2. Analysis Class Diagram for UC "Detail Products”



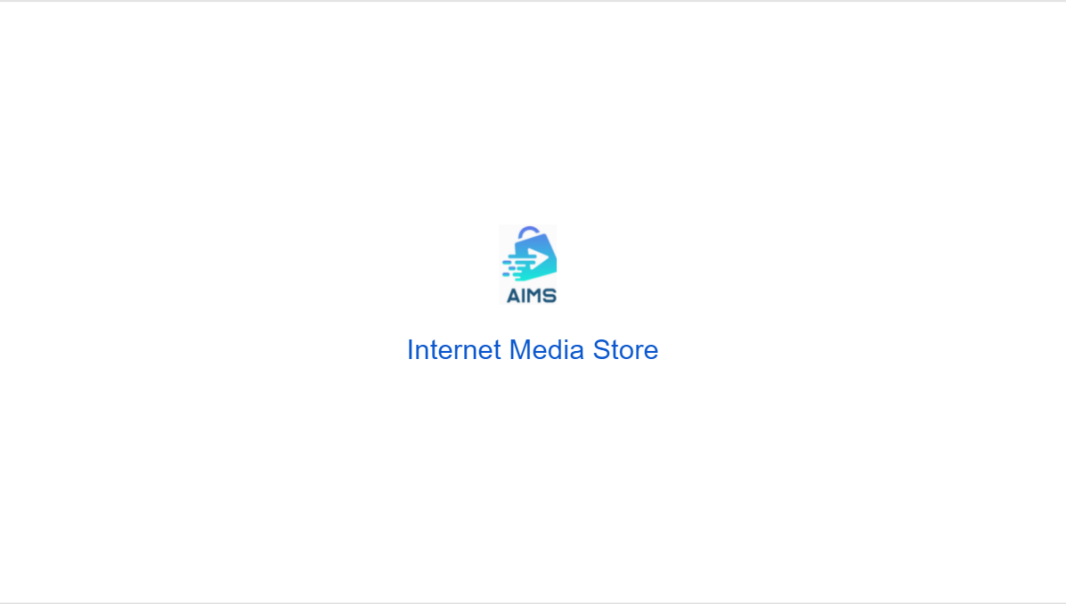
4. Interface Design

4.1. User Interface Design

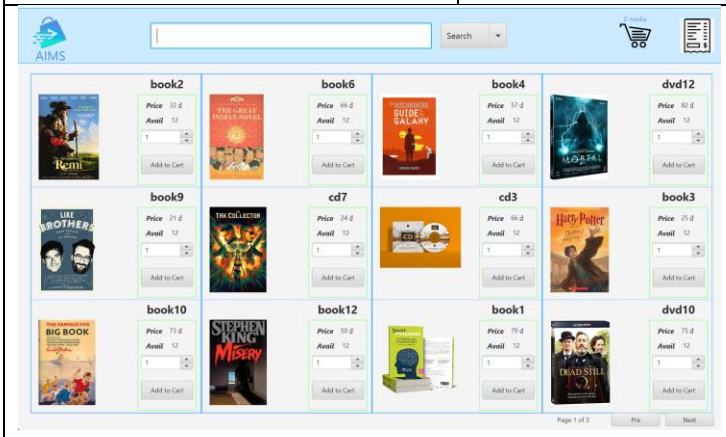
Screen transition flow



Hello Page

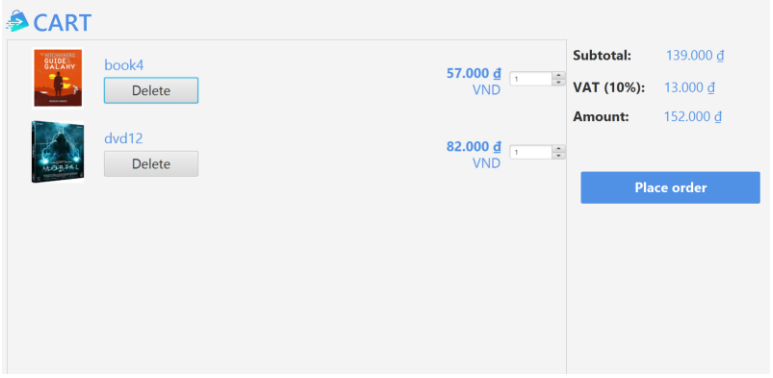


Home Page

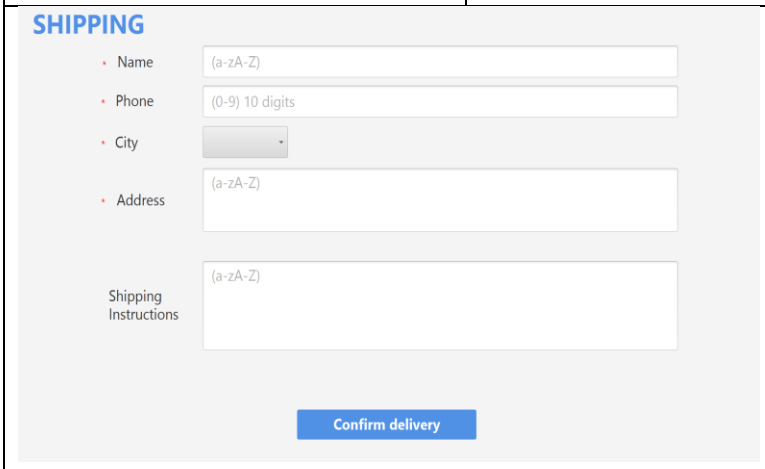
AIMS Software		Date of creation	Person in charge	
Screen specification	Home page screen	27/10/2023	Nguyễn Ngọc Quỳnh Anh	
		Control	Operator	Function
		Search Input Area	Type	Input Search Keyword
		Search Button	Click	Search Media with input in search area
		Cart Button	Click	Display Cart Page
		Area for displaying Banner	Initial	Display Banner
		Area for displaying Media	Initial	Display Media
		Next Button	Click	Display next page
		Pre Button	Click	Display previous page

Cart Page

AIMS Software	Date of creation	Person in charge

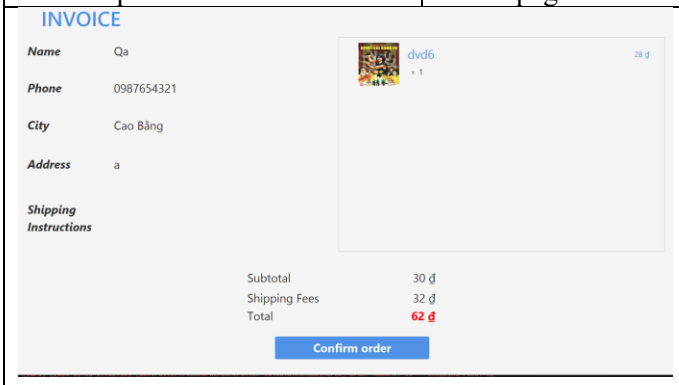
Screen specification	Home page screen	27/10/2023	Nguyễn Ngọc Quỳnh Anh	
		Control	Operator	Function
		Area for displaying Price	Initial	Display price
		Area for displaying Media	Initial	Display Media
		Place order Button	Click	Display Delivery Form
		Delete Button	Click	Remove product from cart

Delivery Information Form Page

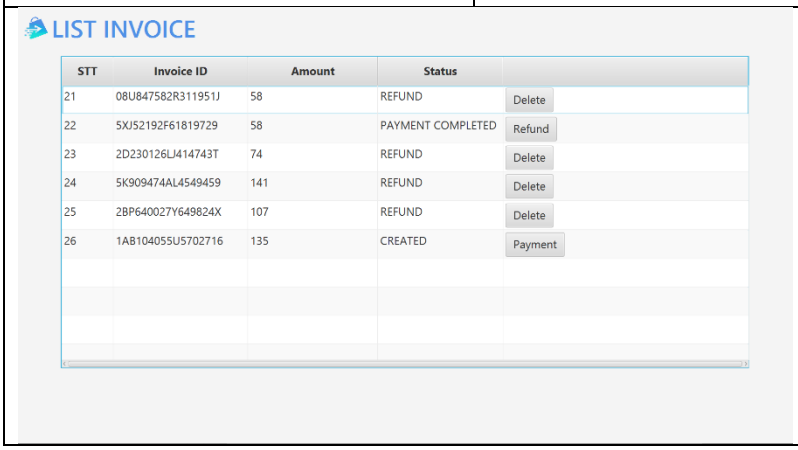
AIMS Software		Date of creation	Person in charge	
Screen specification	Home page screen	27/10/2023	Nguyễn Ngọc Quỳnh Anh	
		Control	Operator	Function
		Name Input area	Type	Set name reciever
		Phone input area	Type	Set phone reciever
		Province Input area	Choose	Set province reciever
		Address Input area	Type	Set address reciever
		Instruction Input area	Type	Set instruction reciever
		Submit button	Click	Send form and display Rush Order Form Page or Invoice Page

Invoice Page

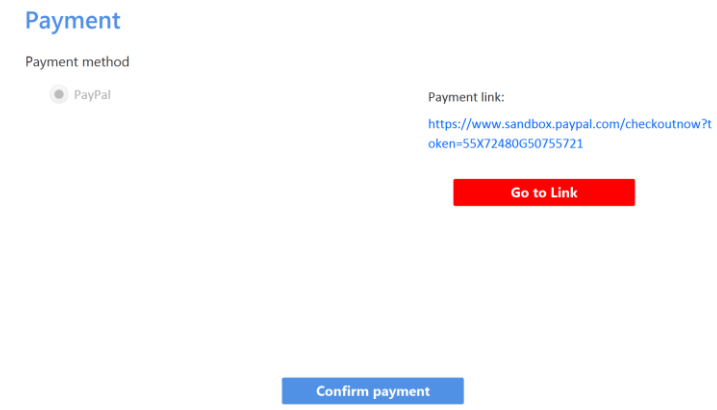
AIMS Software	Date of creation	Person in charge

Screen specification	Home page screen	27/10/2023	Nguyễn Ngọc Quỳnh Anh	
		Control	Operator	Function
		Area for displaying Price	Initial	Display price
		Area for displaying Delivery Information	Initial	Display Delivery Information
		Area for displaying Media	Initial	Display Media
		Confirm order Button	Click	Display Payment Page

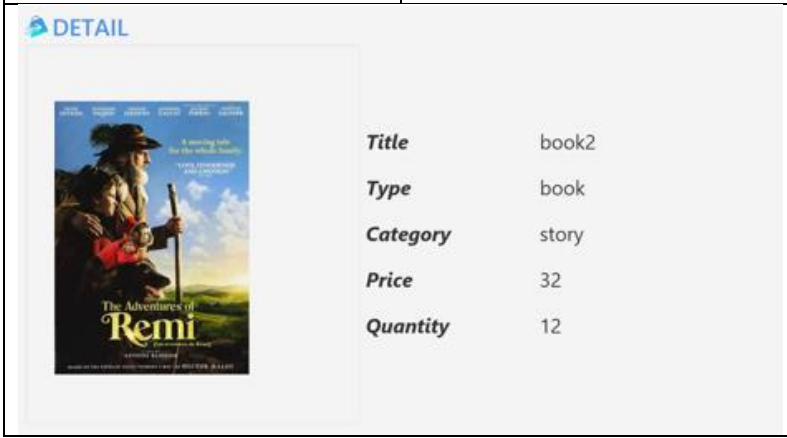
List Invoice Page

AIMS Software		Date of creation	Person in charge	
Screen specification	Home page screen	29/11/2023	Trần Xuân Bách	
		Control	Operator	Function
		Delete Button	Click	DeleteInvoice
		Refund Button	Click	RefundOrder
		Payment Buton	Click	PayOrder

Pay Order Page

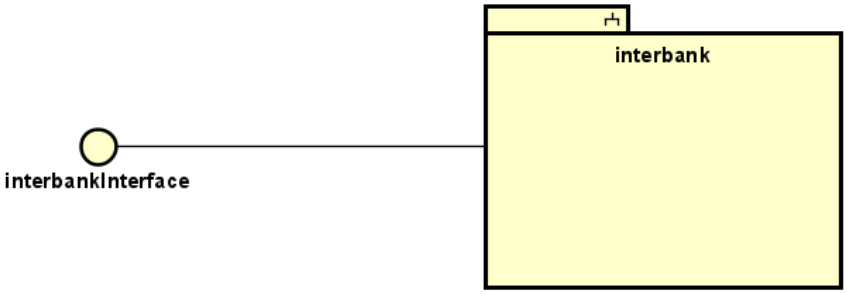
<table>
<tr><td colspan="2">AIMS Software</td><td>Date of creation</td><td colspan="2">Person in charge</td></tr>
<tr><td>Screen specification</td><td>Pay Order page screen</td><td>29/11/2023</td><td colspan="2">Trần Xuân Bách</td></tr>
<tr><td colspan="2" rowspan="4"> 
</td><td>Control</td><td>Operator</td><td>Function</td></tr>
<tr><td>Payment Link</td><td>Inital</td><td>Display paypal sanbox link to pay order</td></tr>
<tr><td>Go to link button</td><td>Click</td><td>Go to link to pay order by Paypal</td></tr>
<tr><td>Confirm Payment Buton</td><td>Click</td><td>Confirm payment</td></tr>
</table>

Detail Product Page

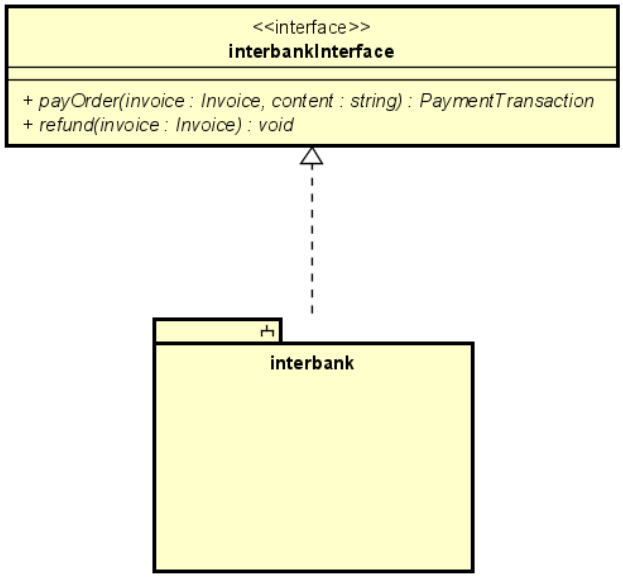
<table>
<tr><td colspan="2">AIMS Software</td><td>Date of creation</td><td colspan="2">Person in charge</td></tr>
<tr><td>Screen specification</td><td>Detail product page screen</td><td>29/11/2023</td><td colspan="2">Nguyễn Vũ Thục Anh</td></tr>
<tr><td colspan="2" rowspan="3"> 
</td><td>Control</td><td>Operator</td><td>Function</td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td></tr>
</table>

4.2. System Interface Design

3.2.1. Identify subsystems

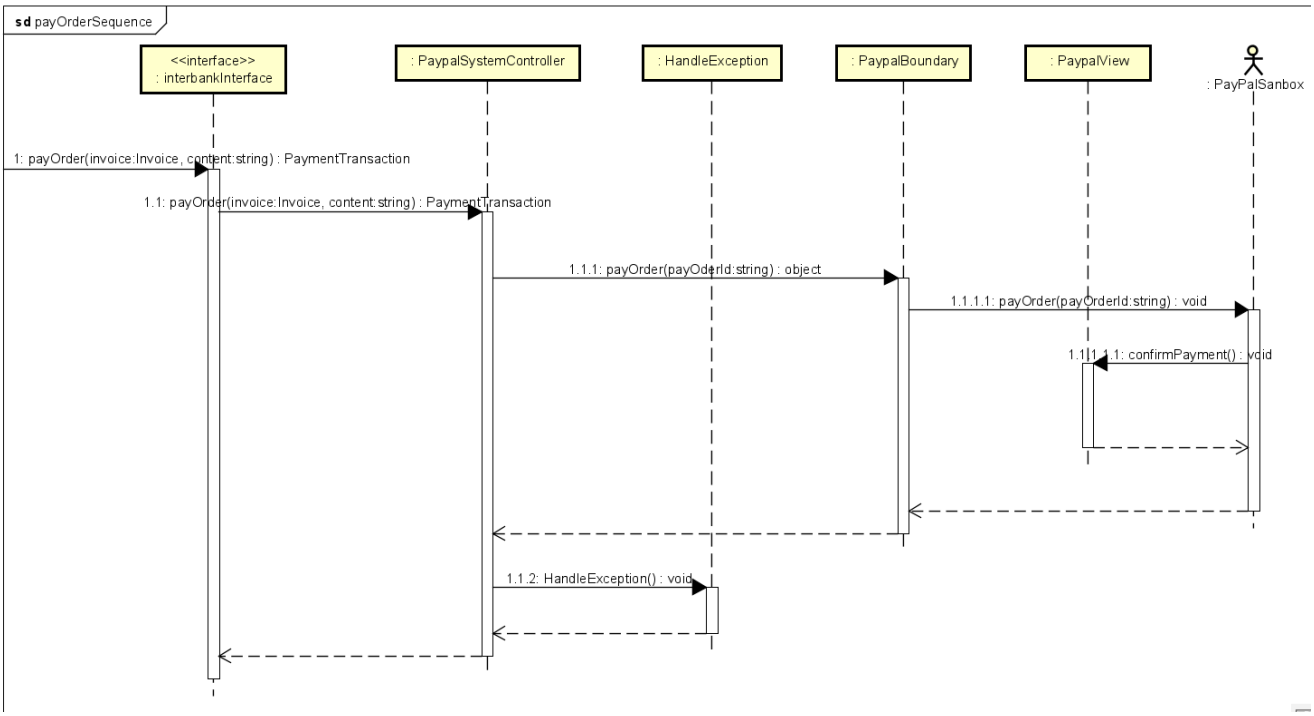


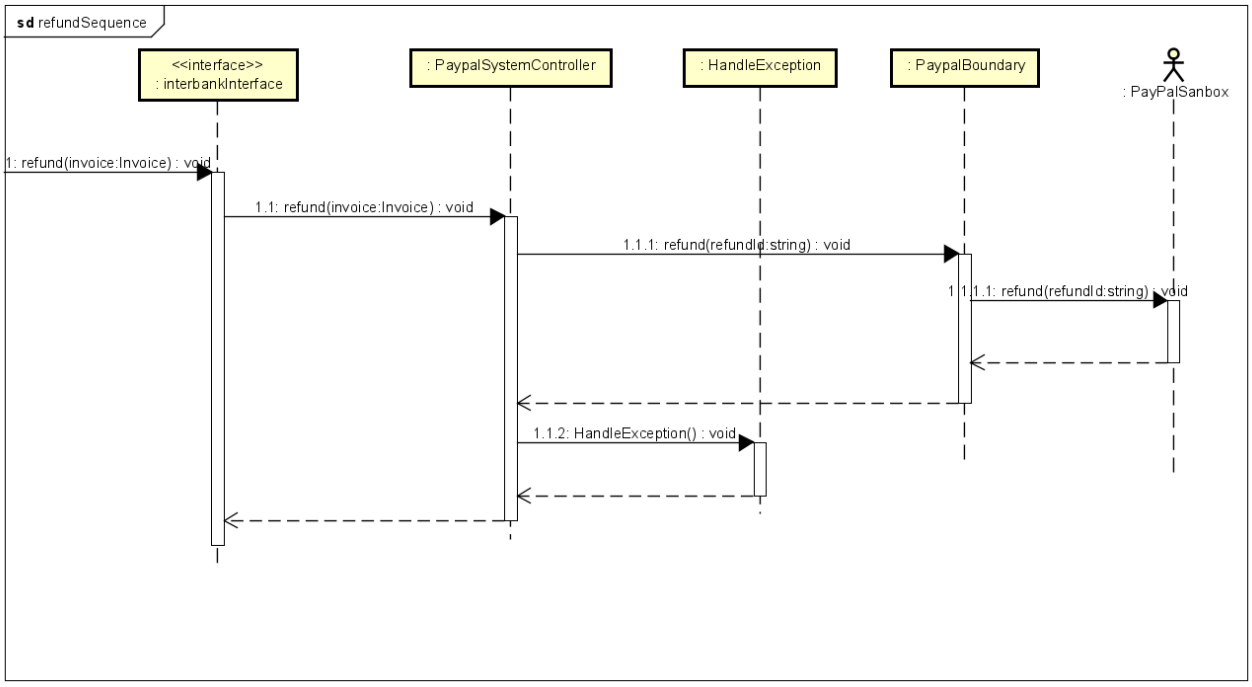
3.2.2. Identify subsystem interface



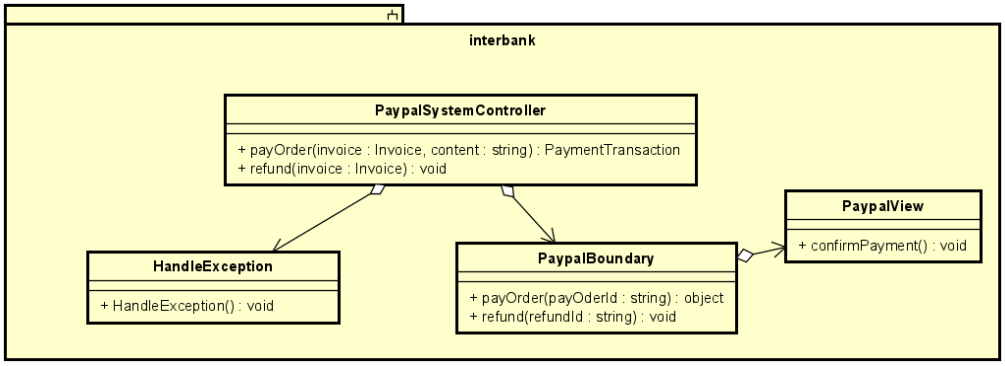
3.2.3. Subsystem design

Distribute subsystem behavior to subsystem elements

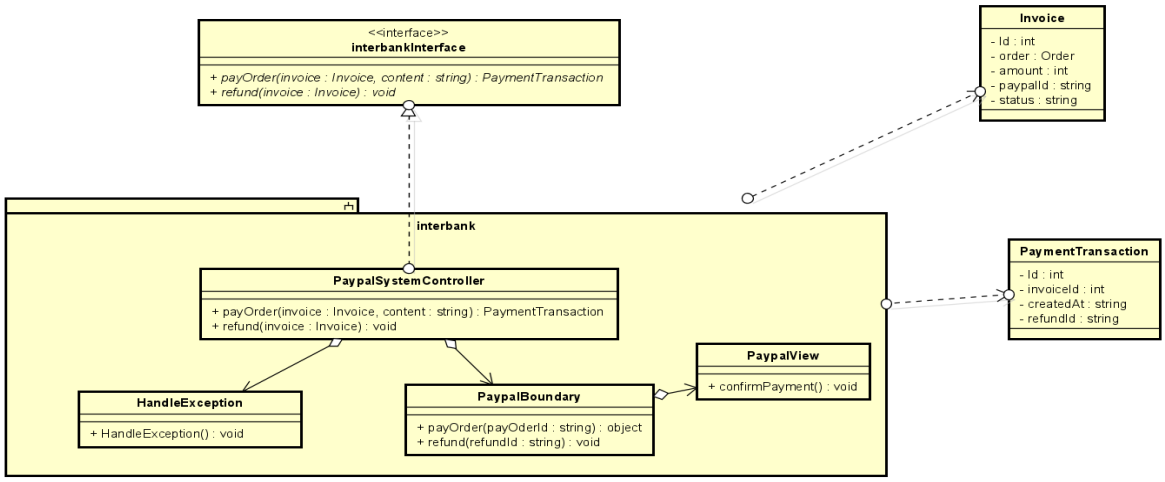




Document subsystem elements

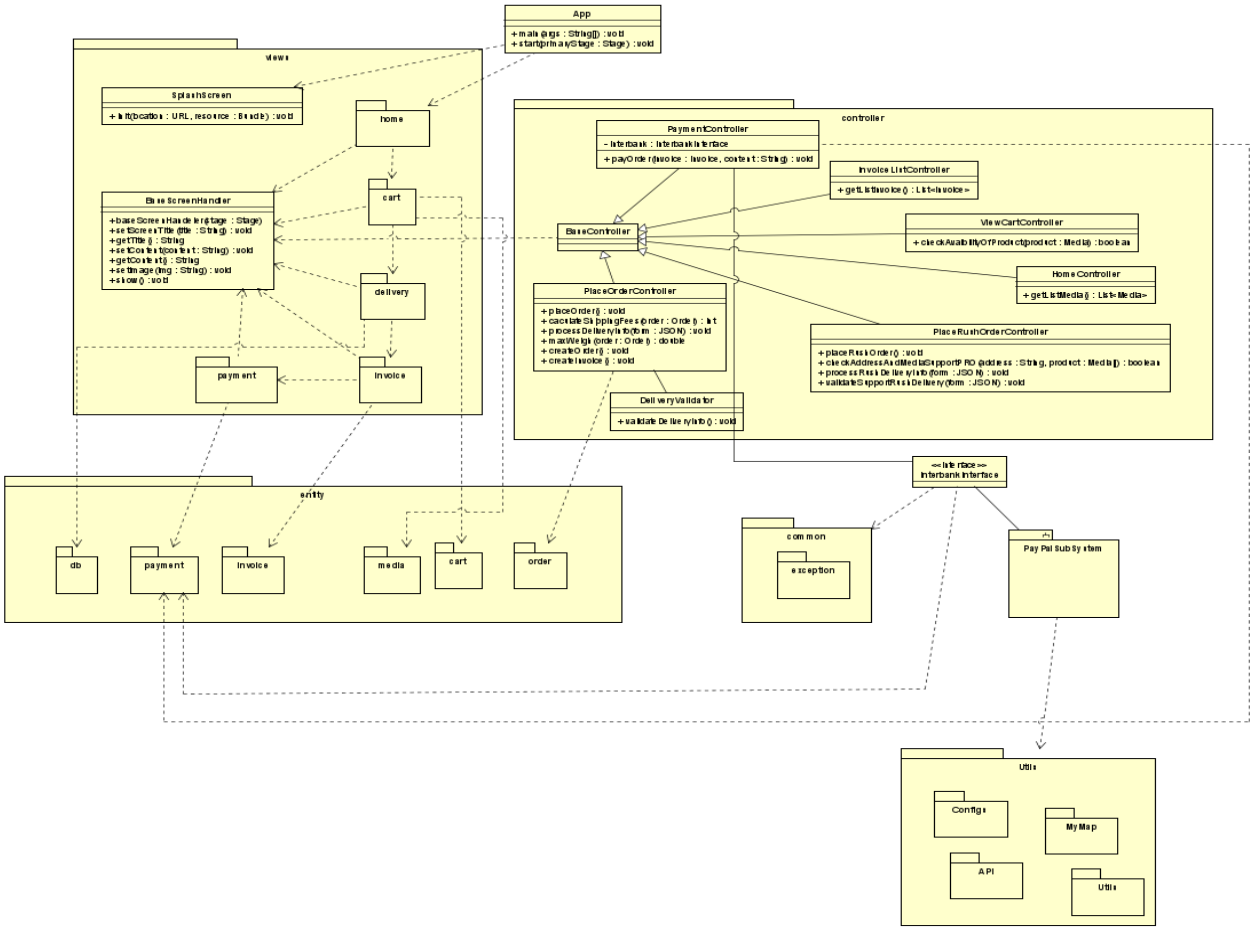


Checkpoints

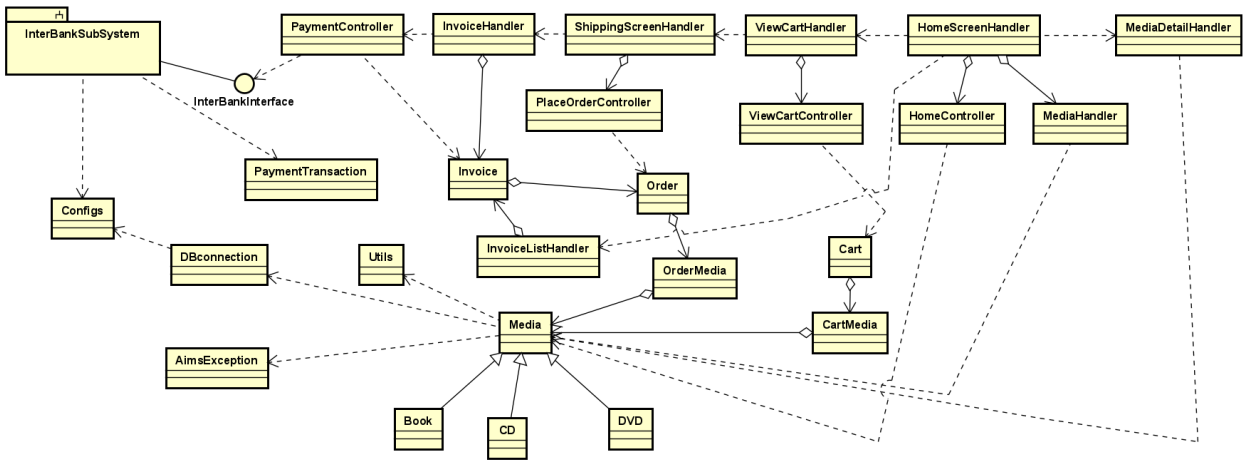


5. Class Design

5.1. General Class Diagram

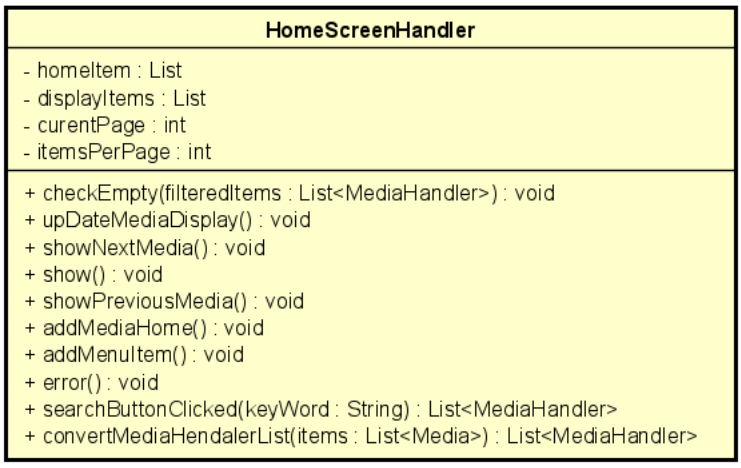


5.2. Relationship Class Diagram



5.3. Class Design

5.3.1. HomeScreenHandler



Attribute

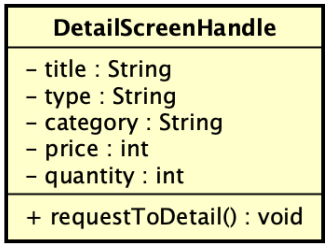
#	Name	Data type	Default value	Description
1	homeItems	List	NULL	Holds the media items fetched from the controller.
2	displayedItems	List	NULL	Holds the currently displayed media items
3	curentPage	int	0	Tracks the current page number
3	itemsPerPage	int	12	Represents the number of items to display per page

Operation

#	Name	Return type	Description (purpose)
1	convertMediaHandlerList	List<MediaHandler>	Convert List<Media> to List<MediaHanlder>
2	checkEmpty	Void	Checks if the filtered media items list is empty and handles displaying a message accordingly
3	updateMediaDisplay	Void	Update and manage the display of media items on the home screen based on the current page

4	showNextMedia	Void	Displays the next set of media items on the screen based on pagination
5	showPreviousMedia	Void	Displays the previous set of media items on the screen based on pagination
6	addMediaHome	Void	Populates the home screen with media items passed as a list.
7	addMenuItems	Void	Adds menu items based on specified text and position to the given menu button

5.3.2. MediaDetailHandler



Attribute

#	Name	Data type	Default value	Description
1	title	String	NULL	Title of media
2	type	String	NULL	Type of media
3	category	String	NULL	Category of media
4	price	int	0	Price of media
5	quantity	int	0	Remaining quantity of media

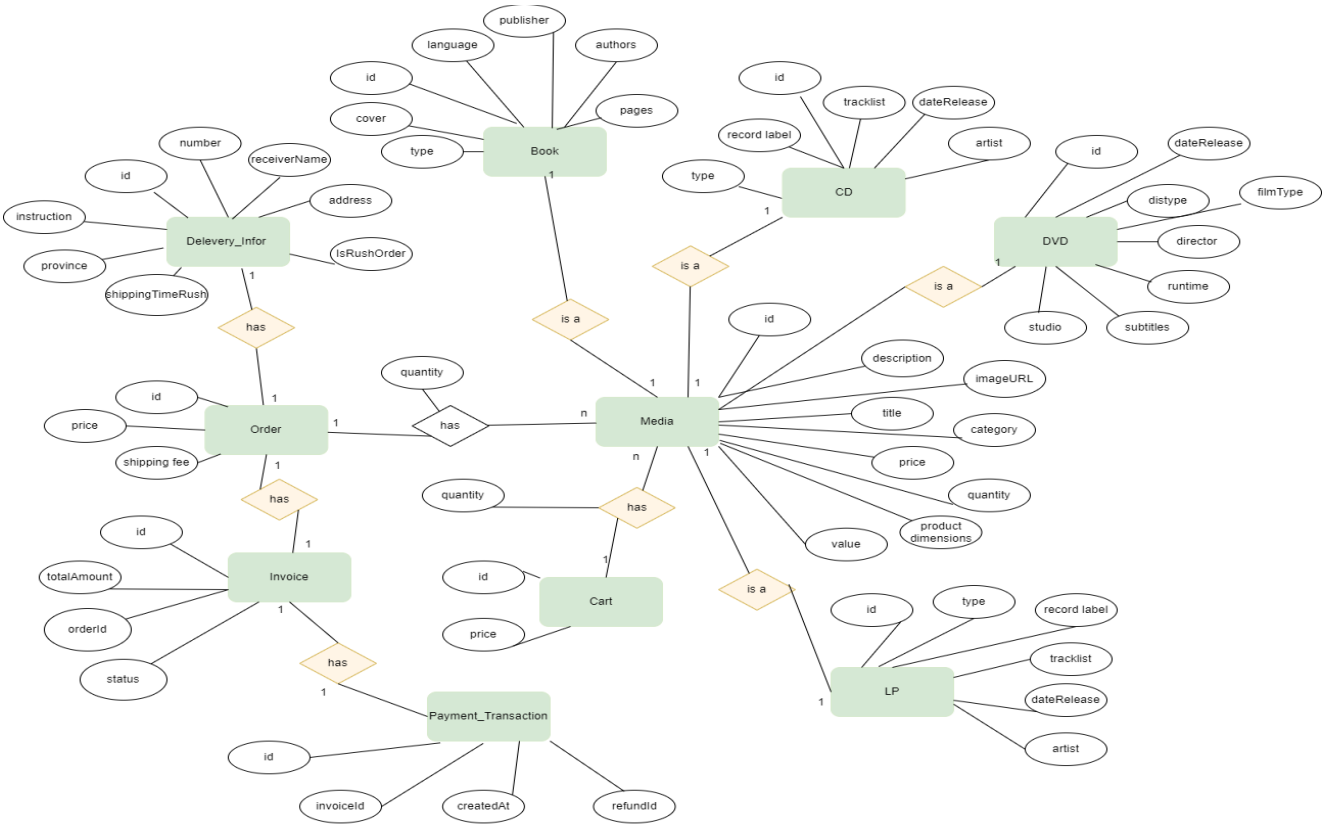
Operation

#	Name	Return type	Description (purpose)
1	requestToDetail	void	Request to detail media

6. Data Modeling

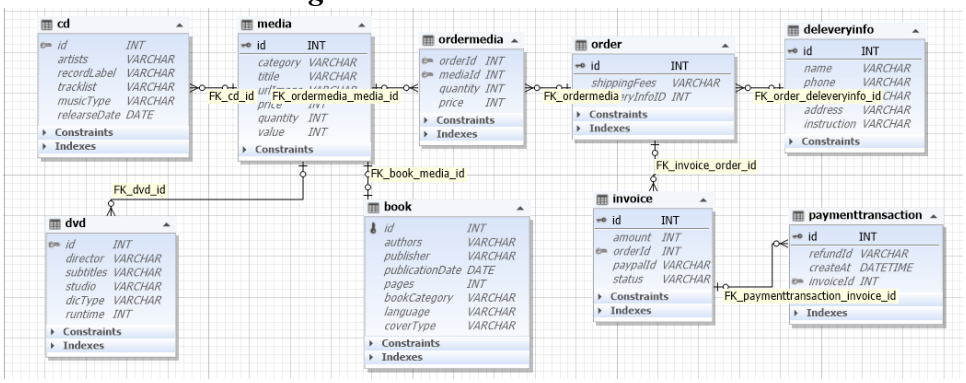
6.1. Conceptual Data Model

AIMS System ERD



6.2. 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.			productDimensions	Varchar(45)	yes	horizontal, length, width dimensions
8.			description	Varchar(45)	yes	Product description
9.			imageURL	Varchar(45)	yes	Product image path
10.			createAt	timestamp	yes	The time the product is added to the system
11.			updateAt	timestamp	no	The time the product is updated to the system

- **CD**

#	PK	FK	Column Name	Data type	Mandatory	Description
1.		x	id	int	yes	ID, same as ID of Media of twhich type is CD
2.			type	Varchar(45)	yes	Music genres
3.			artist	Varchar(45)	yes	Artist's name
4.			dateRelease	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 twhich type is CD
2.			type	Varchar(45)	yes	Music genres
3.			artist	Varchar(45)	yes	Artist's name
4.			dateRelease	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.			authors	Varchar(45)	yes	Authors of the book
3.			publisher	Varchar(45)	yes	Publishing house
4.			language	Varchar(45)	yes	Language
5.			type	Varchar(45)	yes	Cover type
6.			cover	Varchar(45)	yes	Book cover
7.			page	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

- **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.			isRushOrder	tinyint(1)	yes	Is Place Rush Order

8.			shippingTimeRush	datetime	no	Delivery Time for RO
9.			createAt	timestamp	yes	
10.			updateAt	timestamp	yes	

- **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		
7.			updateAt	timestamp		

- **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.			quanity	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.			amount	int	yes	Total
3.		x	orderId	int	yes	Order ID
4.			staus	Varchar(45)	yes	Order status

- **paymenttransaction**

#	PK	FK	Column Name	Data type	Mandatory	Description
5.	x		id	int	yes	ID
6.			createAt	timestamp	yes	Date of creation

7.			refundId	Varchar(45)	yes	Transaction contents
8.		x	invoiceId	int	yes	Invoice ID

SQL:

```
BEGIN TRANSACTION;
CREATE TABLE IF NOT EXISTS "Media" (
"id"	INTEGER NOT NULL,
"type"	VARCHAR(45) NOT NULL,
"category"	VARCHAR(45) NOT NULL,
"price"	INTEGER NOT NULL,
"quantity"	INTEGER NOT NULL,
"title"	VARCHAR(45) NOT NULL,
"value"	INTEGER NOT NULL,
"imageUrl"	VARCHAR(45) NOT NULL,
PRIMARY KEY("id" AUTOINCREMENT)
);
CREATE TABLE IF NOT EXISTS "CD" (
"id"	INTEGER NOT NULL,
"artist"	VARCHAR(45) NOT NULL,
"recordLabel"	VARCHAR(45) NOT NULL,
"musicType"	VARCHAR(45) NOT NULL,
"releasedDate"	DATE,
CONSTRAINT "fk_cd_media" FOREIGN KEY("id") REFERENCES "Media"("id"),
PRIMARY KEY("id")
);
CREATE TABLE IF NOT EXISTS "Book" (
"id"	INTEGER NOT NULL,
"author"	VARCHAR(45) NOT NULL,
"coverType"	VARCHAR(45) NOT NULL,
"publisher"	VARCHAR(45) NOT NULL,
"publishDate"	DATETIME NOT NULL,
"numOfPages"	INTEGER NOT NULL,
"language"	VARCHAR(45) NOT NULL,
"bookCategory"	VARCHAR(45) NOT NULL,
CONSTRAINT "fk_book_media" FOREIGN KEY("id") REFERENCES "Media"("id"),
PRIMARY KEY("id" AUTOINCREMENT)
);
CREATE TABLE IF NOT EXISTS "DVD" (
"id"	INTEGER NOT NULL,
"discType"	VARCHAR(45) NOT NULL,
"director"	VARCHAR(45) NOT NULL,
"runtime"	INTEGER NOT NULL,
"studio"	VARCHAR(45) NOT NULL,
"subtitle"	VARCHAR(45) NOT NULL,
"releasedDate"	DATETIME,
"filmType"	VARCHAR(45) NOT NULL,
CONSTRAINT "fk_dvd_media" FOREIGN KEY("id") REFERENCES "Media"("id"),
```

```
PRIMARY KEY("id")
);
CREATE TABLE IF NOT EXISTS "OrderMedia" (
"mediaID"	INTEGER NOT NULL,
"orderID"	INTEGER NOT NULL,
"price"	INTEGER NOT NULL,
"quantity"	INTEGER NOT NULL,
CONSTRAINT "fk_ordermedia_media" FOREIGN KEY("mediaID") REFERENCES "Media"("id"),
CONSTRAINT "fk_ordermedia_order" FOREIGN KEY("orderID") REFERENCES "Order"("id"),
PRIMARY KEY("mediaID","orderID")
);
CREATE TABLE IF NOT EXISTS "Order" (
"id"	INTEGER NOT NULL,
"name"	VARCHAR(45) NOT NULL,
"address"	VARCHAR(45) NOT NULL,
"phone"VARCHAR(45) NOT NULL,
"shipping_fee"	INTEGER NOT NULL,
PRIMARY KEY("id" AUTOINCREMENT)
);
CREATE TABLE IF NOT EXISTS "Transaction" (
"id"	INTEGER NOT NULL,
"invoiceId"	INTEGER NOT NULL,
"createAt"	DATETIME NOT NULL,
"refundId"	VARCHAR(45) NOT NULL,
CONSTRAINT "fk_transaction_invoice" FOREIGN KEY("orderId") REFERENCES "Invoice"("id"),
PRIMARY KEY("id" AUTOINCREMENT)
);
CREATE TABLE IF NOT EXISTS "Invoice" (
"id"	INTEGER NOT NULL,
"orderId"	INTEGER,
"amount"	INTEGER,
"paypalId"	VARCHAR(50),
"status"	VARCHAR(50),
CONSTRAINT "fk_invoice_order" FOREIGN KEY("orderId") REFERENCES "Order"("id"),
PRIMARY KEY("id" AUTOINCREMENT)
);
COMMIT;
```