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

AIMS e-commerce software

Subject: ITSS Software Development

Group 6: Nguyễn Đình Dũng – 20210230
Vũ Minh Dũng - 20205179
Trần Nam Dương - 20210263
Nguyễn Mạnh Dũng - 20194745
Trịnh Tiến Dũng – 20215187

Hanoi, tháng 6 năm 2024

Mục lục

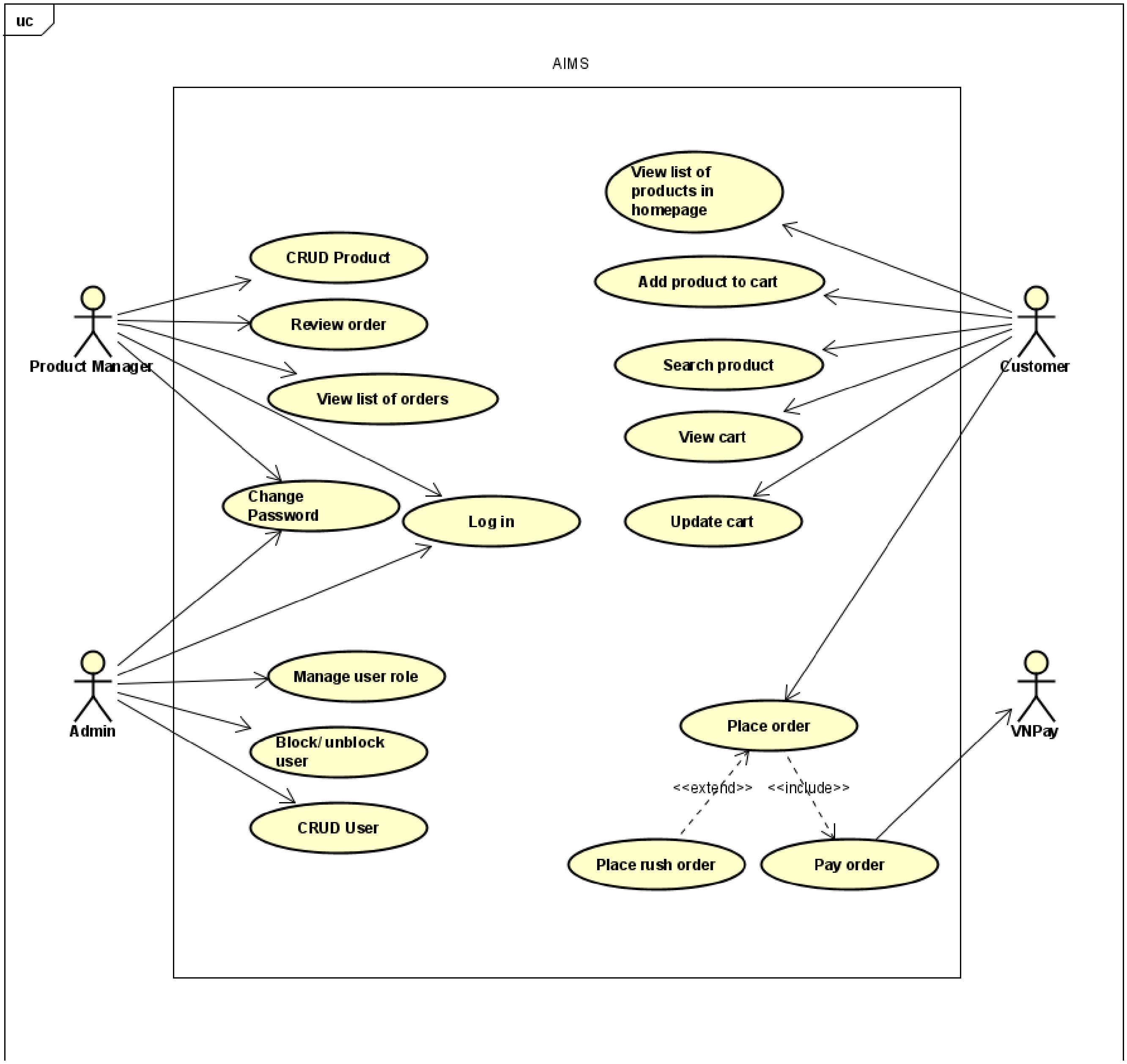
1. Assign working 4
2. Usecase Specification 5
    2.1. General use case diagram 5
    2.2. Usecase specification Search and sort product 5
    2.3. Usecase specification View product detail 8
    1.4. Usecase Spectification Manage Cart 10
    2.5. Usecase Specification Place Order 14
    2.6. Usecase Speccification Place Rush Order 17
    2.6. Usecase Specification View list of products on home screen 22
3. Usecase Analysis 27
    3.1. Usecase Analysis Search/Sort product. 27
    3.2. Usecase Analysis View product detail. 28
    3.3. Usecase Analysis Manage Cart 29
    3.4. Usecase Analysis Place Order. 30
    3.1. Usecase Analysis Place Rush Order 32
4. Interface Design 34
    4.1. Home Screen 34
    4.2. View Product detail 35
    4.3. Cart 36
    4.4. Shipping information Screen 37
    4.5. Delivery Method 38
    4.6. Invoice Screen 39
    4.7. PaymentScreen 40
    4.8. Error alert 41
5. ANALYSIS CLASS DIAGRAM 43
    5.1. General Class Diagram 43
    5.2.1 Class Diagram for Package View 44
6. Data Modeling 47
    6.1. Conceptual Data Model 47
    6.2. Logical Data Model 49
    6.3. Physical Data Model 49

1. Assign working

Nguyễn Đình Dũng	Design place order and place rush order functions
Trần Nam Dương	Design payment functions, vnpay api , database
Nguyễn Mạnh Dũng	Design the product management functions of the product manager
Vũ Minh Dũng	Design the administrator's user management function, login
Trịnh Tiến Dũng	UC view list of products; view product detail; search, sort products; CRUD products in cart

2. Usecase Specification

2.1. General use case diagram


uc
AIMS
View list of products in homepage
CRUD Product
Add product to cart
Review order
Search product
View list of orders
View cart
Change Password
Log in
Update cart
Product Manager
Customer
Admin
Manage user role
Place order
VNPay
Block/ unblock user
<<extend>>
<<include>>
CRUD User
Place rush order
Pay order

2.2. Usecase specification Search and sort product

Use Case "Search, Sort products"

1. **Use case code**

   UC002

2. **Brief Description**

   This use case describes the interaction between customer and AIMS when customer wish(es) to search or sort products on the home screen

3. Actors

3.1 Customer

4. Preconditions

Customer successfully view products on the home screen.

5. Basic Flow of Events

5.1 Search product

1. The customer searches for products by title or category. (see Table A)
2. AIMS looks up in database products satisfy the customer request.
3. AIMS displays result to home screen. (see Table B)

5.2 Sort products

1. The customer requests ascending sort of products.
2. AIMS sorts products prices in ascending order.
3. AIMS displays sorted products on the home screen. (see Table B)

6. Alternative flows

Table 1-Alternative flows of events for UC Search products

No	Location	Condition	Action	Resume location
1.	At Step 3	If there is no product match search request	▪ System displays no product found	End use case

Table 2-Alternative flows of events for UC Sort products

No	Location	Condition	Action	Resume location
1.	At Step 1	If customer requests descending sort.	▪ AIMS sorts products prices in descending order	Resumes at Step 3

7. Input data

Table A-Input data of UC Search products

No	Data fields	Description	Mandatory	Valid condition	Example
1	Product title	Title of searched product	No	Text	Harry Potter
	Product category	Category of searched product	No	Choose from list	Book

8. Output data

Table B-Output data of UC Searched products

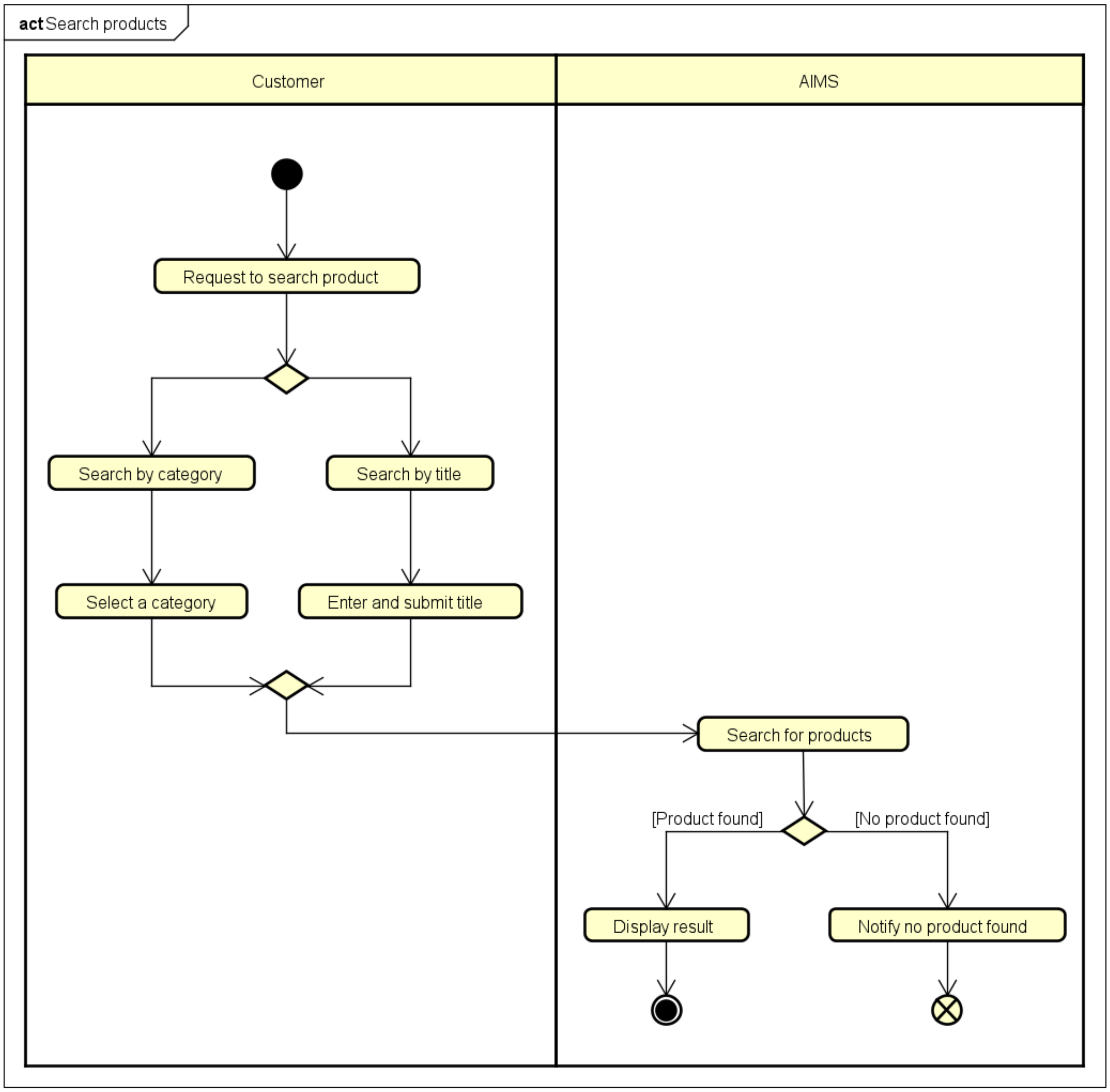
No	Data fields	Description	Display format	Example
1.	Image	Image of product's cover	Image	
2.	Title	Title of product	Text	Harry Potter

3.	Price	Price of product	- Comma for thousands Separator - Positive integer - Right alignment	100.000 vnd
4.	Avail	Available quantity of product	- Positive integer - Right alignment	10

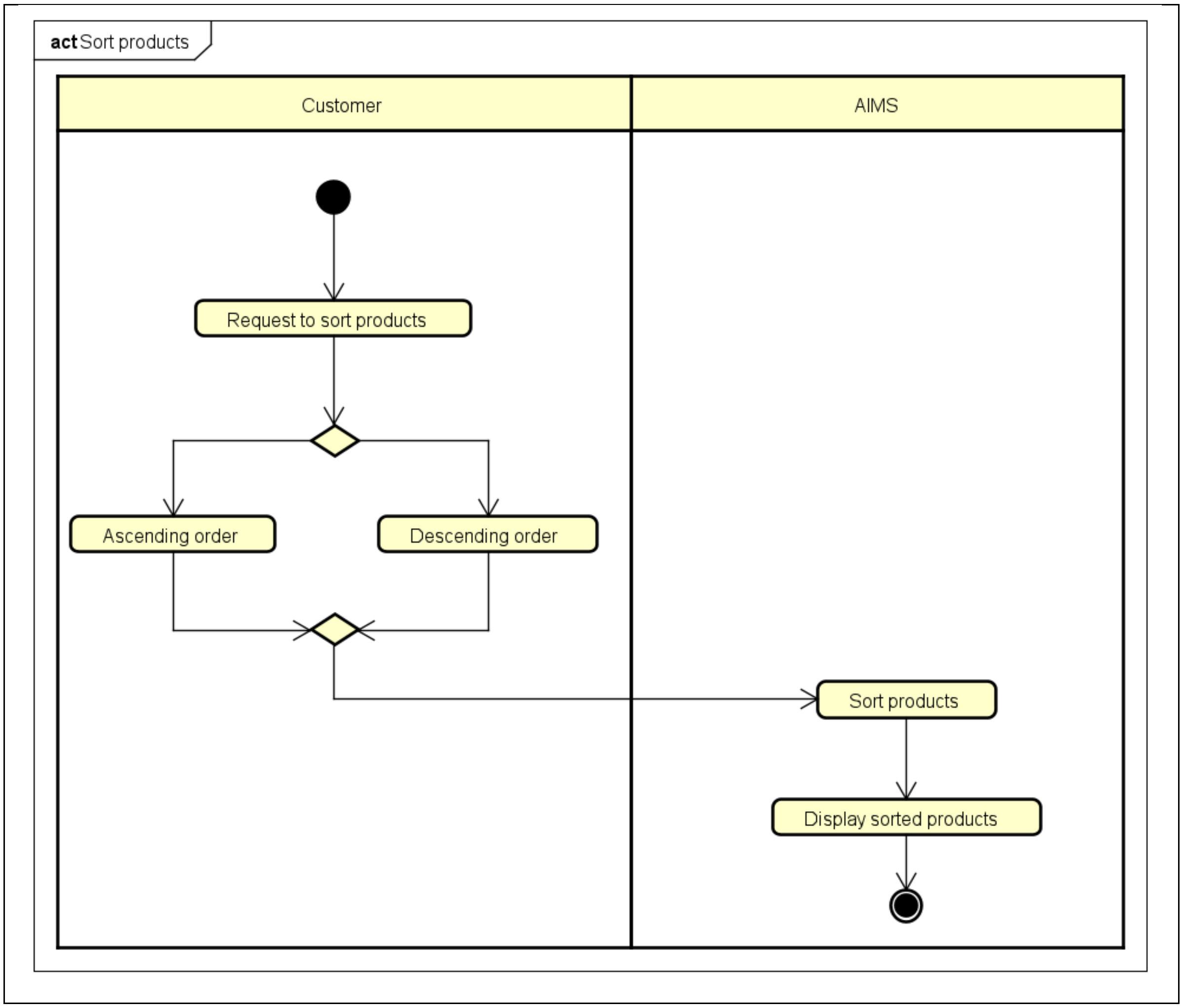
9. Postconditions

10. Activity Diagrams

10.1 Search products


act Search products
Customer
AIMS
Request to search product
Search by category
Search by title
Select a category
Enter and submit title
Search for products
[Product found]
[No product found]
Display result
Notify no product found

10.2 Sort products


act Sort products
Customer
AIMS
Request to sort products
Ascending order
Descending order
Sort products
Display sorted products

2.3. Usecase specification View product detail

Use Case “View product detail”

1. Use case code

UC003

2. Brief Description

This use case describes the interaction between customer and AIMS when customer wish(es) to view product detail

3. Actors

3.1 Customer

4. Preconditions

Customer successfully views list of products on the home screen.

5. Basic Flow of Events

1. The customer requests to view detail information of a product

2. AIMS checks for category of the product
3. AIMS shows detail information of the product (if Book see Table A, if DVD see Table B, if CD see Table C)

6. Alternative flows

7. Input data

8. Output data

Table A-Output data of Book's detail information

No	Data fields	Description	Display format	Example
1.	Image	Image of book's cover	Image	
2.	Title	Title of book	Text	Harry Potter
3.	Author	Author of book	Text	J.K.Rowling
4.	Publisher	Publisher of book	Text	Bloomsbury
5.	Publish Date	Publish Date of book	dd/mm/yyyy	26/06/1997
6.	Language	Language used in book	Text	English
7.	Category	Category of book	Text	Fantasy
8.	Numbers of Pages	Numbers of Pages of book	Positive Integer	700

Table B-Output data of DVD's detail information

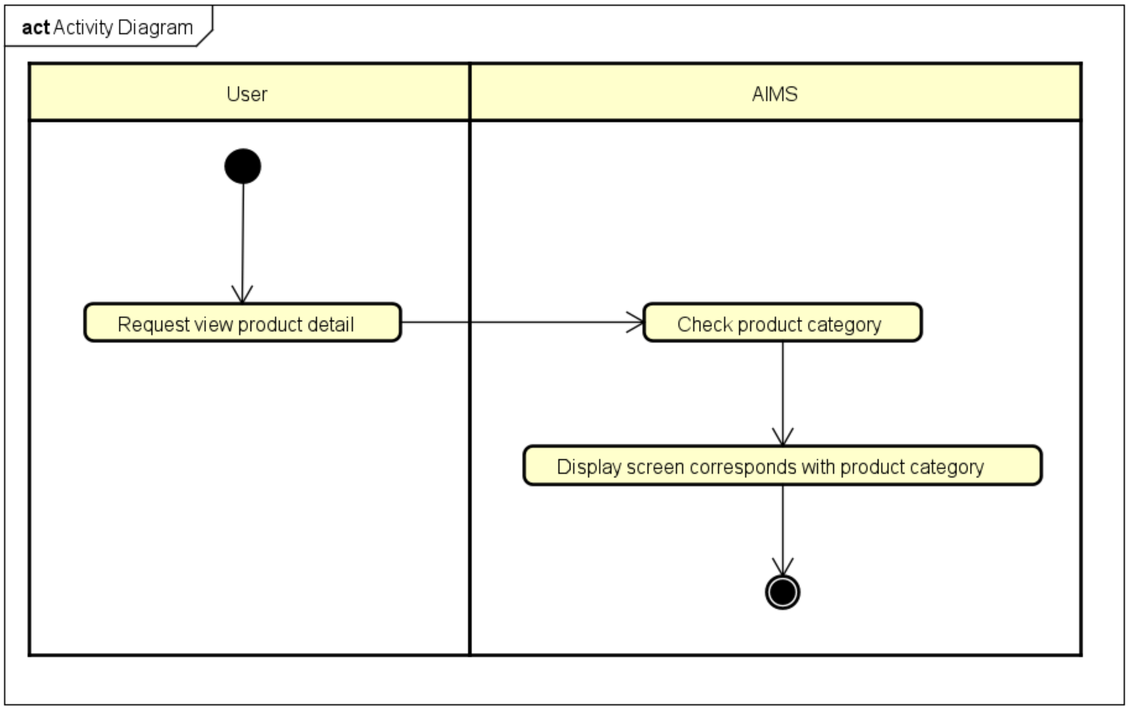
No	Data fields	Description	Display format	Example
1.	Image	Image of DVD's cover	Image	
2.	Title	Title of DVD	Text	Harry Potter
3.	Director	Director of DVD	Text	Chris Columbus
4.	Studio	Name of DVD production studio	Text	Warner Bros. Pictures
5.	Released Date	Released Date of DVD	dd/mm/yyyy	11/05/2002
6.	Type	Type of DVD	Text	Fantasy

Table C-Output data of CD's detail information

No	Data fields	Description	Display format	Example
1.	Image	Image of CD's cover	Image	
2.	Title	Title of CD	Text	1989
3.	Artist	Artist of CD	Text	Taylor Swift
4.	Record Label	Record Label of CD	Text	Big Machine Records
5.	Record Date	Record Date of CD	dd/mm/yyyy	27/10/2014
6.	Music Type	Music Type of CD	Text	Pop

9. Postconditions

10.Activity Diagrams


act Activity Diagram
User
AIMS
Request view product detail
Check product category
Display screen corresponds with product category

1.4. Usecase Specification Manage Cart

Use Case "CRUD product in Cart"

1. **Use case code**

   UC00X

2. **Brief Description**

   This use case describes the interaction between customer and AIMS when customer wish(es) to CRUD product in cart

3. **Actors**

   a. **Customer**

4. **Preconditions**

   Customer successfully views list of products on the home screen.

5. **Basic Flow of Events**

   a. **Add product to cart**

   1. The customer selects a product and the quantity they want to purchase.
   2. The customer requests to add the product to cart.
   3. AIMS displays a notification that the product has been successfully added.

   b. **View product in cart**

   1. The customer requests to view the shopping cart.
   2. AIMS displays a list of products in the shopping cart.

3. AIMS displays the total amount for the products. (see Table A)

c. Delete product in cart

1. The customer views products in cart
2. The customer delete products from cart
3. AIMS displays list of products after delete.
4. AIMS updates and displays the total amount for the products. (see Table A)

d. Update product in cart

1. The customer views products in cart.
2. The customer updates the products’ quantity.
3. AIMS updates and displays the total amount for the products. (see Table A)

6. Alternative flows

Table 1-Alternative flows of events for Add product to cart

No	Location	Condition	Action	Resume location
2.	At Step 2	The quantity added to the cart exceeds the quantity in stock	▪ AIMS displays error message: The number of products in stock is not enough	Resumes at Step 1

Table 2-Alternative flows of events for Update product in cart

No	Location	Condition	Action	Resume location
1.	At Step 2	The updated quantity exceeds the quantity in stock	▪ AIMS displays error message: The number of products in stock is not enough	Resumes at Step 1

7. Input data

8. Output data

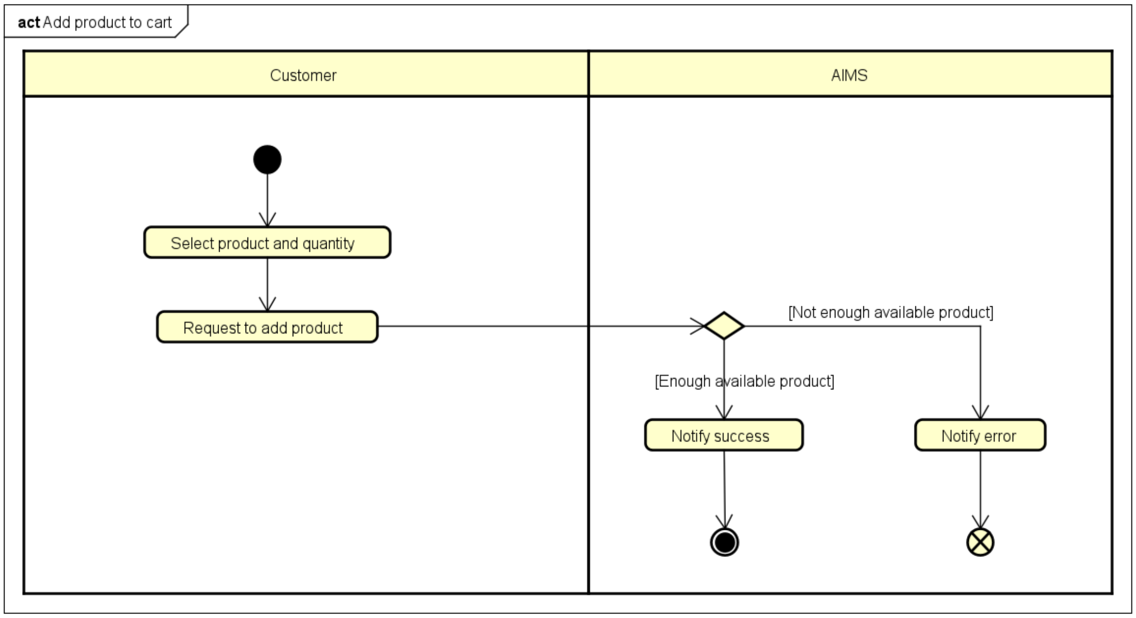
Table A-Output data of View products in cart

No	Data fields	Description	Display format	Example
9.	Image	Image of product’s cover	Image	
10.	Title	Title of product	Text	Harry Potter
11.	Quantity	Quantity of product	- Positive integer - Right alignment	10
12.	Price	Price of product	- Comma for thousands Separator - Positive integer - Right alignment	100.000 vnd
13.	Subtotal	Total price of products in the cart before VAT		100.000 vnd
14.	VAT	Value Added Tax		10.000 vnd
15.	Amount	Total price of products in the cart after VAT		110.000 vnd

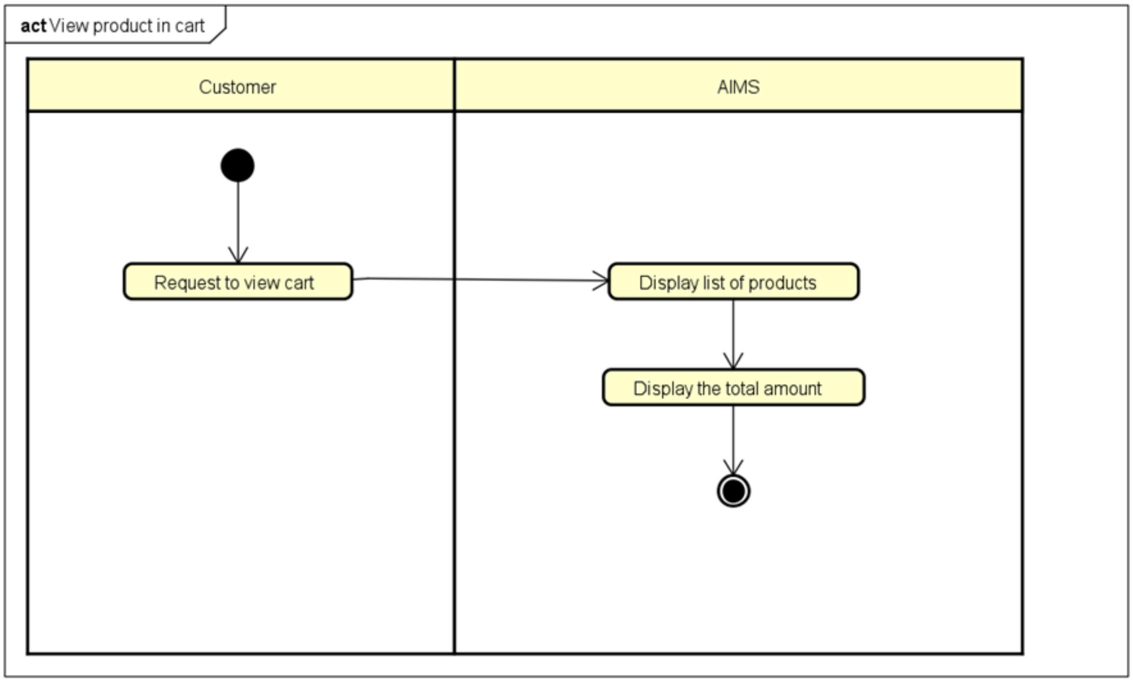
9. Postconditions

10. Activity Diagrams

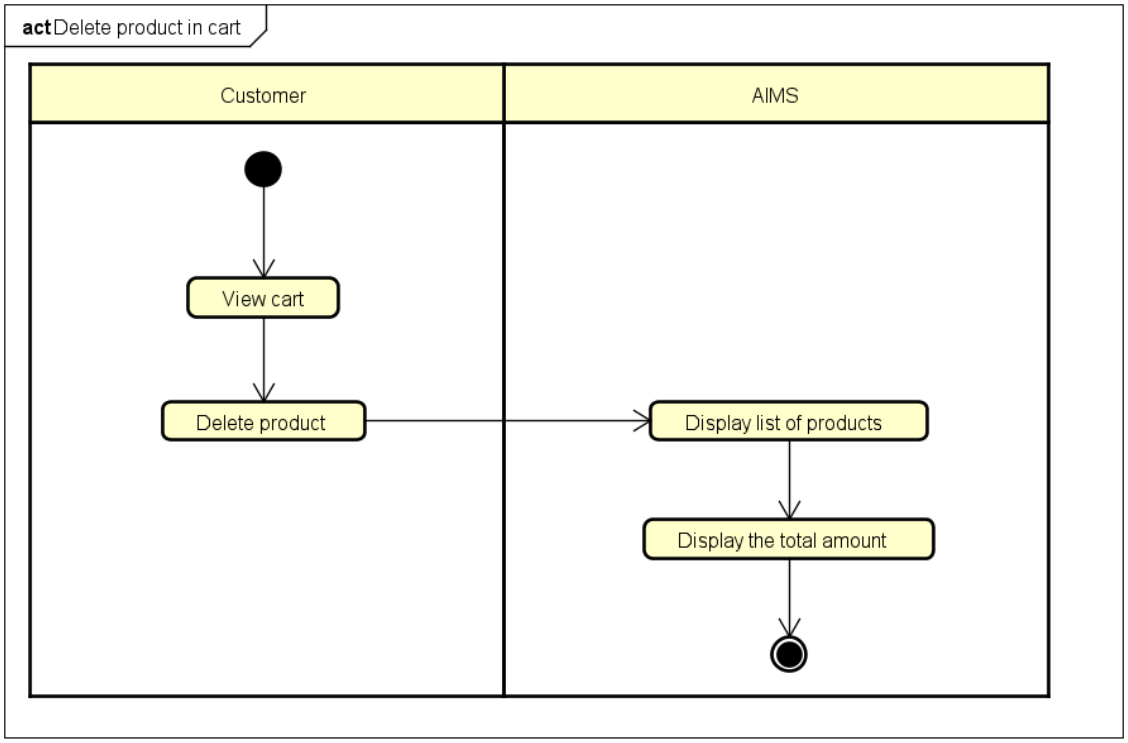
a. Add product to cart


act Add product to cart
Customer
AIMS
Select product and quantity
Request to add product
[Not enough available product]
[Enough available product]
Notify success
Notify error

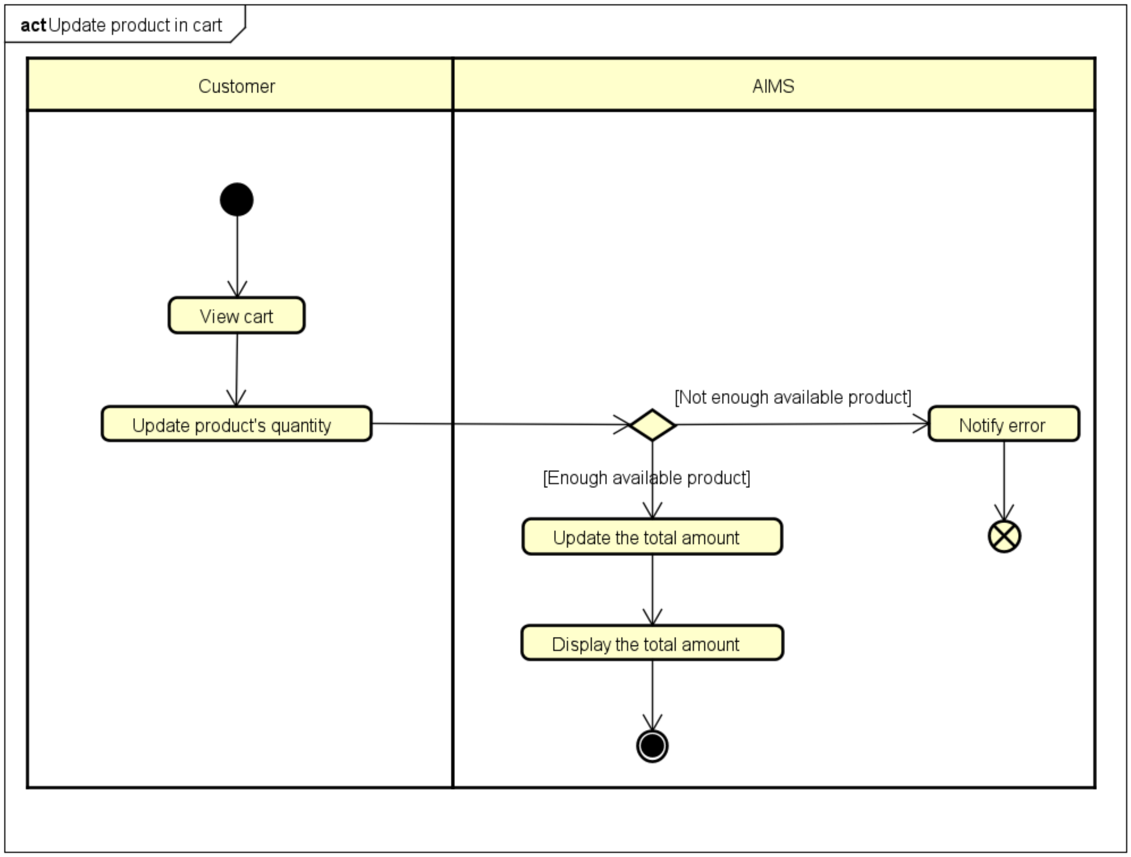
b. View product in cart


act View product in cart
Customer
AIMS
Request to view cart
Display list of products
Display the total amount

c. Delete product in cart


act Delete product in cart
Customer
AIMS
View cart
Delete product
Display list of products
Display the total amount

d. Update product in cart


act Update product in cart
Customer
AIMS
View cart
Update product's quantity
[Not enough available product]
Notify error
[Enough available product]
Update the total amount
Display the total amount

2.5. Usecase Specification Place Order

Use Case “Place Order”

1. **Use case code**

   UC005

2. **Brief Description**

   This use case describes the interaction between customer and AIMS when customer wish(es) to place order

3. **Actors**
   a. **Customer**
4. **Preconditions**

None

5. **Basic Flow of Events**
   1. Customers click to view cart.
   2. The system calculates the total product price.
   3. The system checks whether the products in the cart are still in stock.
   4. The system will display a list of items the customer wants to order (product name, quantity, and price).
   5. Customer clicks on the “Place Order” button.
   6. The system checks whether the products in the cart are still in stock.
   7. The system displays a delivery information form, asking the customer to update shipment details.
   8. Customers fill in the necessary information and do not select "fast delivery".
   9. Click “Update”.
   10. The system checks input information.
   11. Delivery fee calculation system.
   12. The system will display temporary order information.
   13. Call Usecase “Payment”.
   14. System empties cart.
6. **Alternative flows**

No	Location	Condition	Action	Resume location
3.	At Step 3	If system check in insufficient warehouse products.	▪ The system will notify customers the product is not enough in stock and requires the customer to update	Resumes at Step 4.

			the cart with missing products.	
4.	At step 8	If the customer misses the required information fields blank or write in the wrong format.	▪ The system will ask the customer to enter complete information.	Resumes at Step 8.
5.	At step 8	If the customer selects “Place Rush Order”	▪ Insert into usecase “Place Rush Order”	Continue usecase “Place Rush Order”
6.	At step 1	If there are no products in the cart.	▪ The system will notify you that there are no products in the cart product	Resumes at Step 1.

Table A-Input data of “delivery information form”

No	Data fields	Description	Mandatory	Valid condition	Example
	Address	Address of customer	Yes	Text	Số 1, Đường Tạ Quang Bửu, quận Hai Bà Trưng.
	Name of customer		Yes	Maximum 30 characters	Nguyen Van A
	Phone number		Yes	From 9 – 11 numbers (first number need be 0)	0998716388
	Province		Yes	Choose from list	Ha Noi

7. Output data

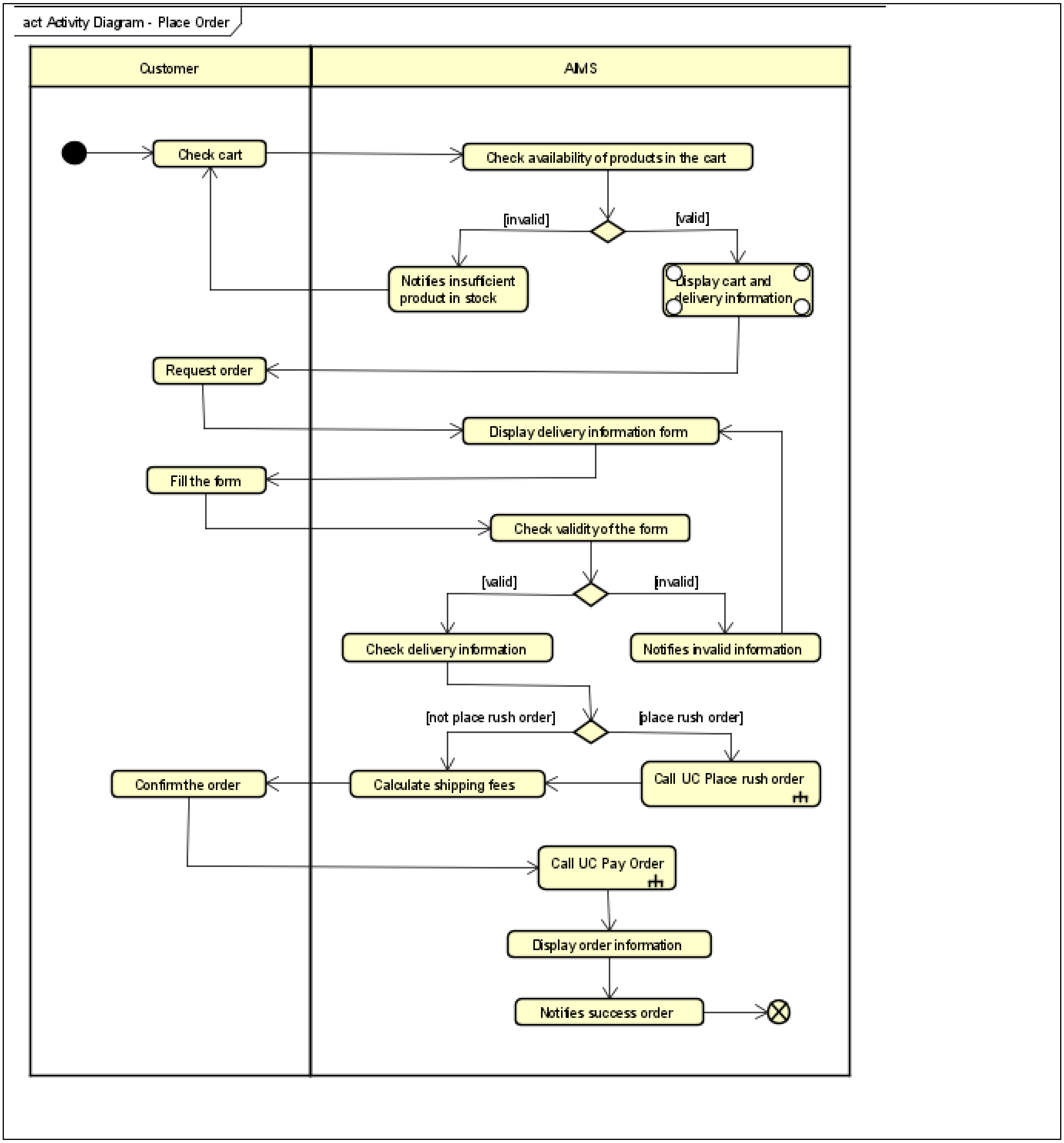
Table B-Output data of “Temporary order information sheet”

No	Data fields	Description	Display format	Example
16.	Title	Name of Product	Text	Harry Potter Book
17.	Price	Price of product	- Comma for thousands Separator - Positive integer - Right alignment	100.000 vnd
18.	Quanity	Quantity of product	- Positive integer	10
19.	Sum price of product	Sum price of each product	- Comma for thousands Separator - Positive integer - Right alignment	300,00 vnđ

20.	Total amount payable before calculation shipping fee	Total price of all products before calculation shipping fee	- Comma for thousands Separator - Positive integer - Right alignment	20,000 vnđ
21.	Total amount money	Total amout money of all products after calculation shipping fee	- Comma for thousands Separator - Positive integer - Right alignment	320,000 vnđ

8. Postconditions

9. Activity Diagrams


act Activity Diagram - Place Order
Customer
AIMS
Check cart
Check availability of products in the cart
[invalid]
[valid]
Notifies insufficient product in stock
Display cart and delivery information
Request order
Display delivery information form
Fill the form
Check validity of the form
[valid]
[invalid]
Check delivery information
Notifies invalid information
[not place rush order]
[place rush order]
Confirm the order
Calculate shipping fees
Call UC Place rush order
Call UC Pay Order
Display order information
Notifies success order

2.6. Usecase Speccification Place Rush Order

Use Case “Place Rush Order”
1. Use case code

UC006

2. Brief Description

This use case describes the interaction between customer and AIMS when customer wish(es) to place order

3. Actors

a. Customer

4. Preconditions

None

5. Basic Flow of Events

15. The system displays the delivery address information fieldsThe system calculates the total product price.
16. User enters information fields.
17. User confirms
18. The system checks whether the fields are valid.
19. The system switches to the delivery method selection screen.
20. The user selects the fast delivery method
21. User confirms.
22. The system calculates delivery fees and displays the invoice screen.
23. User confirms.
24. The system switches to payment.

6. Alternative flows

No	Location	Condition	Action	Resume location
1	At step 4	If the customer misses the required information fields blank or write in the wrong format	▪ The system will ask the customer to enter complete information.	Resumes at Step 2.
2	At step 6	The address checking system does not support fast shipping	▪ Fast delivery is not allowed	Resumes at Step 6.
3	At step 6	There are no product which is fast dilivery sopported	▪ Fast delivery is not allowed products in the cart product	Resumes at Step 6.

Table A-Input data of "delivery information form"

No	Data fields	Description	Mandatory	Valid condition	Example
	Address	Address of customer	Yes	Text	Số 1, Đường Tạ Quang Bửu, quận Hai Bà Trưng.

	Name of customer		Yes	Maximum 30 characters	Nguyen Van A
	Phone number		Yes	From 9 – 11 numbers (first number need be 0)	0998716388
	Province		Yes	Choose from list	Ha Noi
	Shipping Instructions		No	Maximum 50 characters	Khong

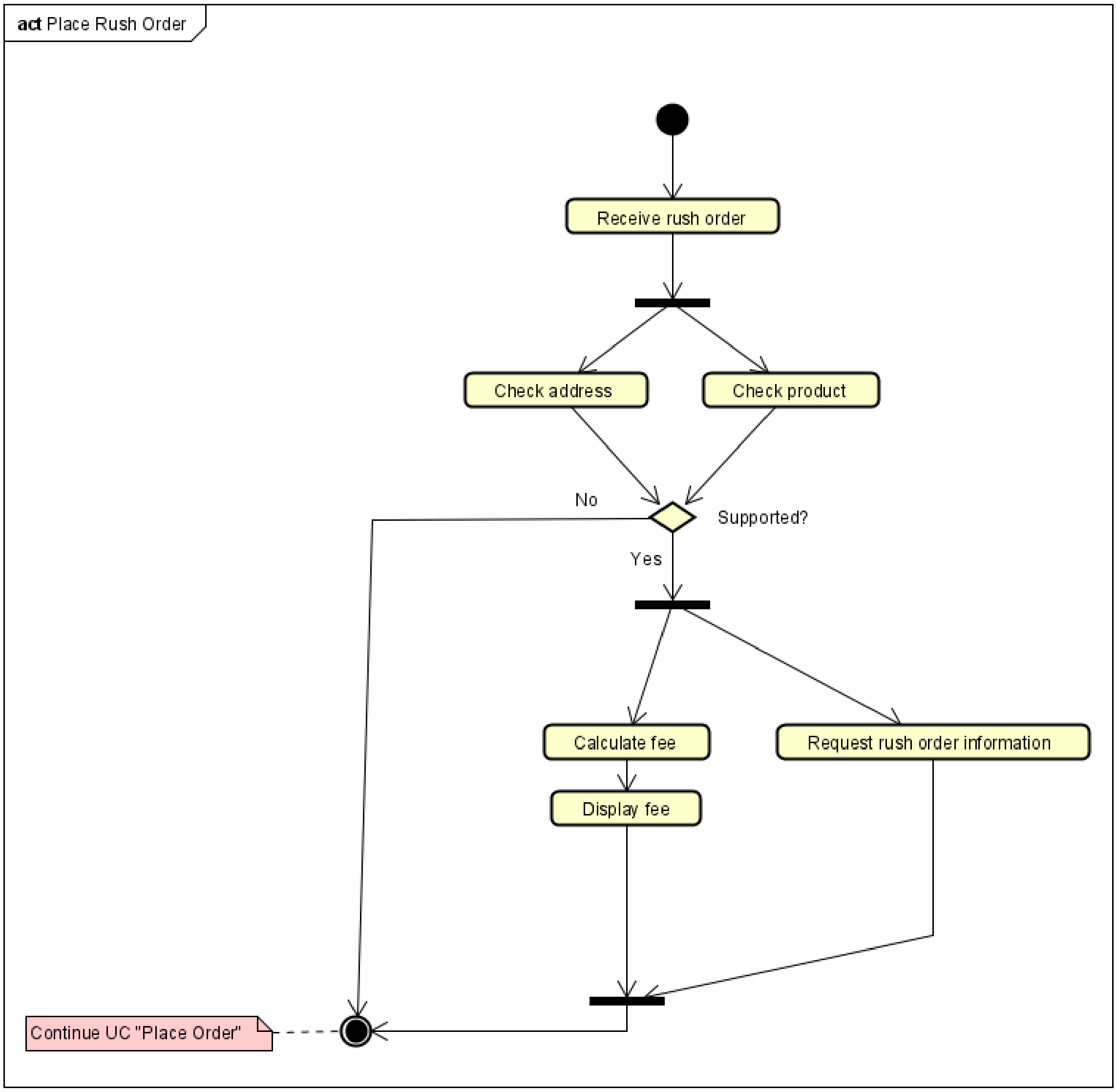
7. Output data

Table B-Output data of “Temporary rush order information sheet”

No	Data fields	Description	Display format	Example
22.	Title	Name of Product	Text	Harry Potter Book
23.	Price	Price of product	- Comma for thousands Separator - Positive integer - Right alignment	100.000 vnd
24.	Quanity	Quantity of product	- Positive integer	10
25.	Type of delivery		Normal or Fast	Fast
26.	Sum price of product	Sum price of each product	- Comma for thousands Separator - Positive integer - Right alignment	300,00 vnđ
27.	Total amount payable before calculation shipping fee	Total price of all products before calculation shipping fee	- Comma for thousands Separator - Positive integer - Right alignment	20,000 vnđ
28.	Total amount fee of rush delivery	10,000 vnđ for each rush delivery product	- Comma for thousands Separator - Positive integer - Right alignment	50,000 vnđ
29.	Total amount money	Total amout money of all products after calculation shipping fee	- Comma for thousands Separator - Positive integer - Right alignment	370,000 vnđ

8. **Postconditions**

9. **Activity Diagrams**


act Place Rush Order
Receive rush order
Check address
Check product
No
Supported?
Yes
Calculate fee
Request rush order information
Display fee
Continue UC "Place Order"

2.7 Use case specification: Pay order

Usecase code:

UC007

Brief Description:

This usecase describes the interaction between the customer and the AIMS system when the user pay for the order.

Actor:

Customer, VNPay subsystem.

Preconditions:

UC Place Order after user confirm shipping method.

Flow of action:

1. The customer confirm the order.
2. AIMS redirect the customer to VNPay payment page.
3. The customer finish payment for the order through VNPay
4. AIMS display Payment Successful page.

Alternate flows for Searching

No	Location	Condition	Action	Resume location
1.	At Step 3	User cancel the payment	▪ AIMS display Payment Failed Page	End usecase

Input data

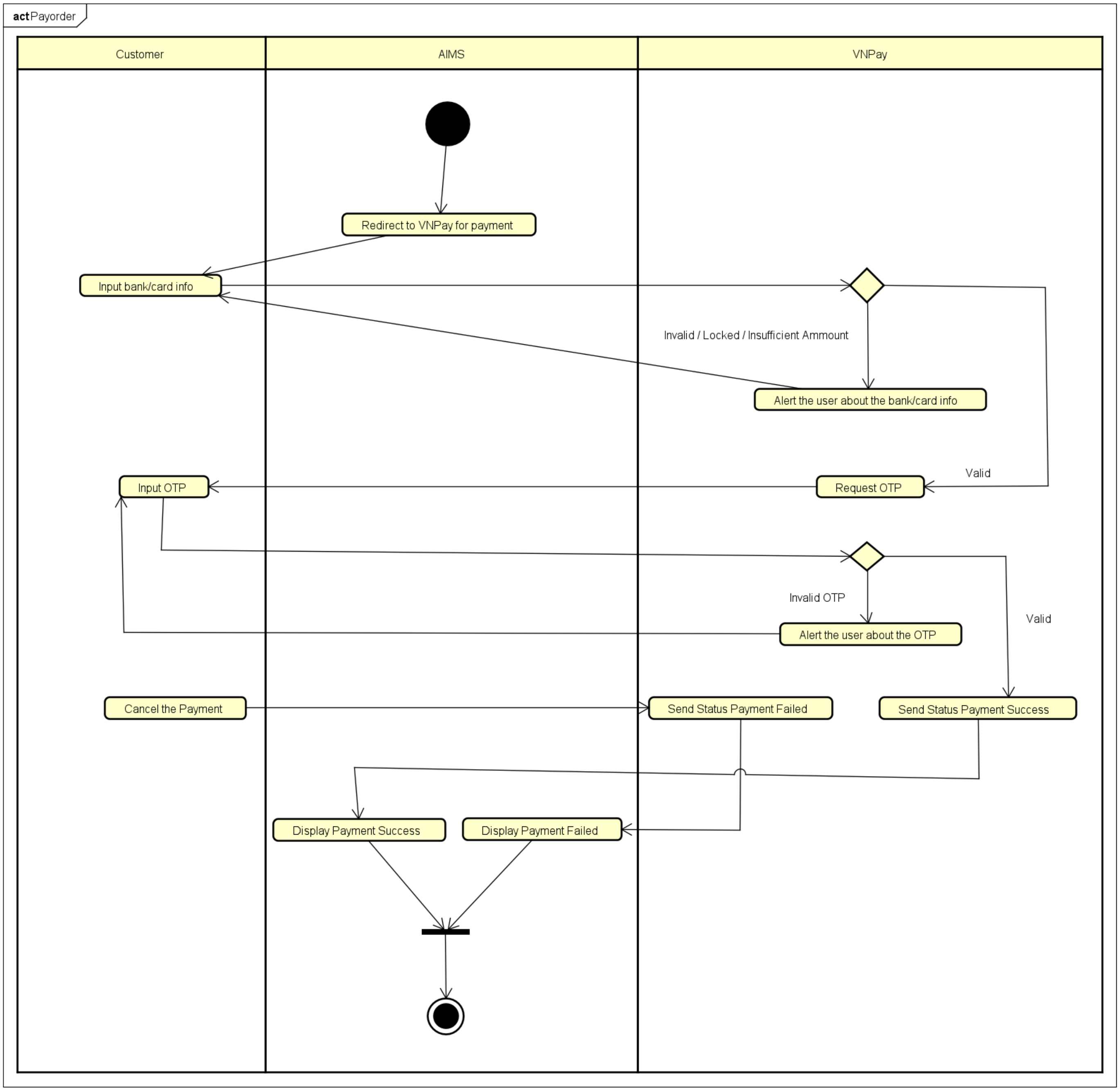
Output data

No	Data fields	Description	Display format	Example
1.	Payment Status	Outcome status of the payment	String	PAYMENT SUCCESSFUL
2.	Message	Message from the system about the payment	String	You have successfully paid for the order

Postconditions:

None

Activity Diagram:


actPayorder
Customer
AIMS
VNPay
Redirect to VNPay for payment
Input bank/card info
Invalid / Locked / Insufficient Ammount
Alert the user about the bank/card info
Valid
Input OTP
Request OTP
Invalid OTP
Valid
Alert the user about the OTP
Cancel the Payment
Send Status Payment Failed
Send Status Payment Success
Display Payment Success
Display Payment Failed

2.6. Usecase Specification View list of products on home screen

Use Case “View list of products”

1. Use case code

UC008

2. Brief Description

This use case describes the interaction between customer and AIMS when customer wish(es) to view product detail

3. Actors

3.1 Customer

4. Preconditions

5. Basic Flow of Events

1. The customer access AIMS
2. AIMS initialize home screen
3. AIMS get all media information from database
4. AIMS displays list of 20 products on each page

6. Alternative flows

7. Input data

8. Output data

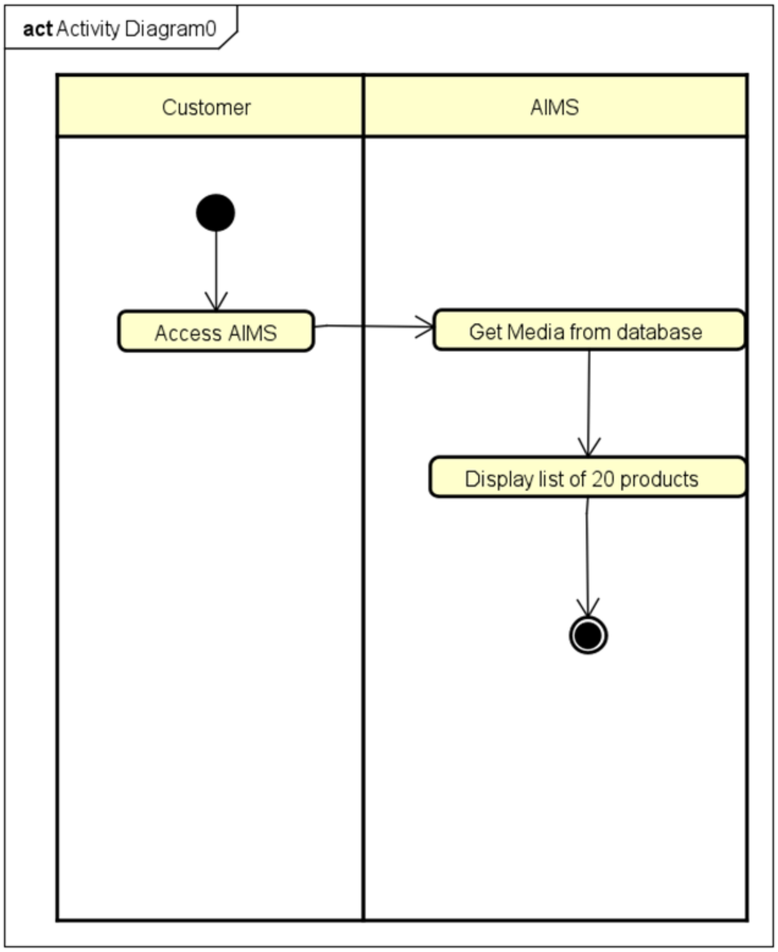
Table A-Output data of Book’s detail information

No	Data fields	Description	Display format	Example
1.	Image	Image of product’s cover	Image	
2.	Title	Title of product	Text	Harry Potter
3.	Price	Price of product	- Comma for thousands Separator - Positive integer - Right alignment	100.000 vnd

4.	Avail	Available quantity of product	- Positive integer - Right alignment	10

9. Postconditions

10. Activity Diagrams


act Activity Diagram0
Customer
AIMS
Access AIMS
Get Media from database
Display list of 20 products

2.7. Usecase Specification for Login

Use case "Login"

1. **Use case code**
   UC009
2. **Brief description**
   The use case describes the interaction between user and system when the user wishes to login.
3. **Actors**
   User, system
4. **Preconditions**
   None
5. **Basic flows of event**
   a. The user clicks the "Login" button.
   b. The application displays the login form
   c. The user fills in the login form **(Table A)** and clicks to "login" button to send information to the application
   d. The system checks username and password of the user
   e. The system checks whether that user is block
   f. The system checks the roles of the user
   g. The system displays the manager screen with tabs based on the user's role.
6. **Alternative flows**

No	Location	Condition	Action	Resumes location
1	At step d	If the user is not existed or the user's information is wrong	The system displays the error "Wrong username or password"	at step c
2	At step e	If the user is blocked	The system displays the error "This user is blocked"	at step c

7. **Input data**

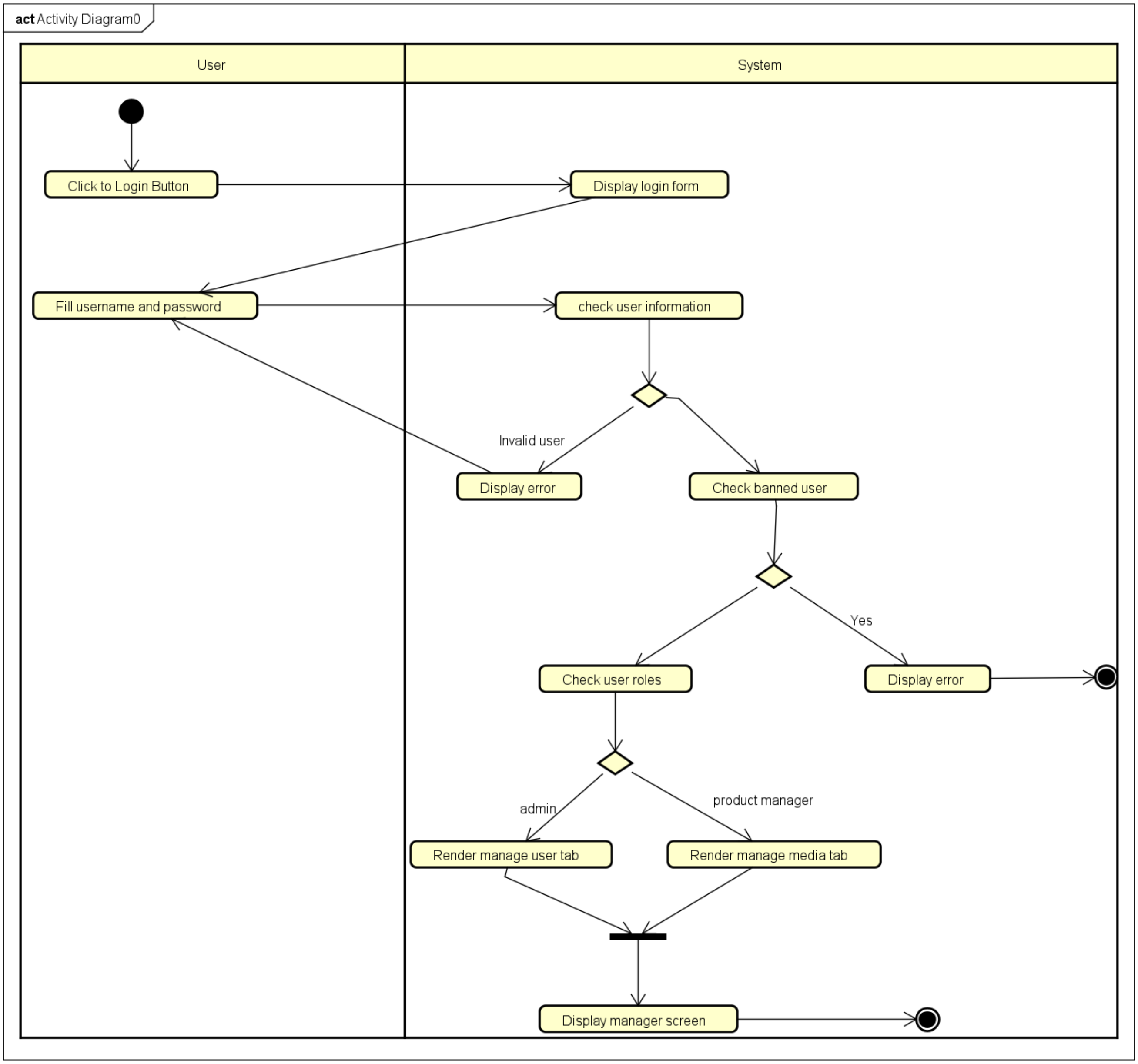
No	Data fields	Description	Display format	Example

1.	username	username of the user	text	admin
2.	password		password	123@123

8. **Output data**

   None

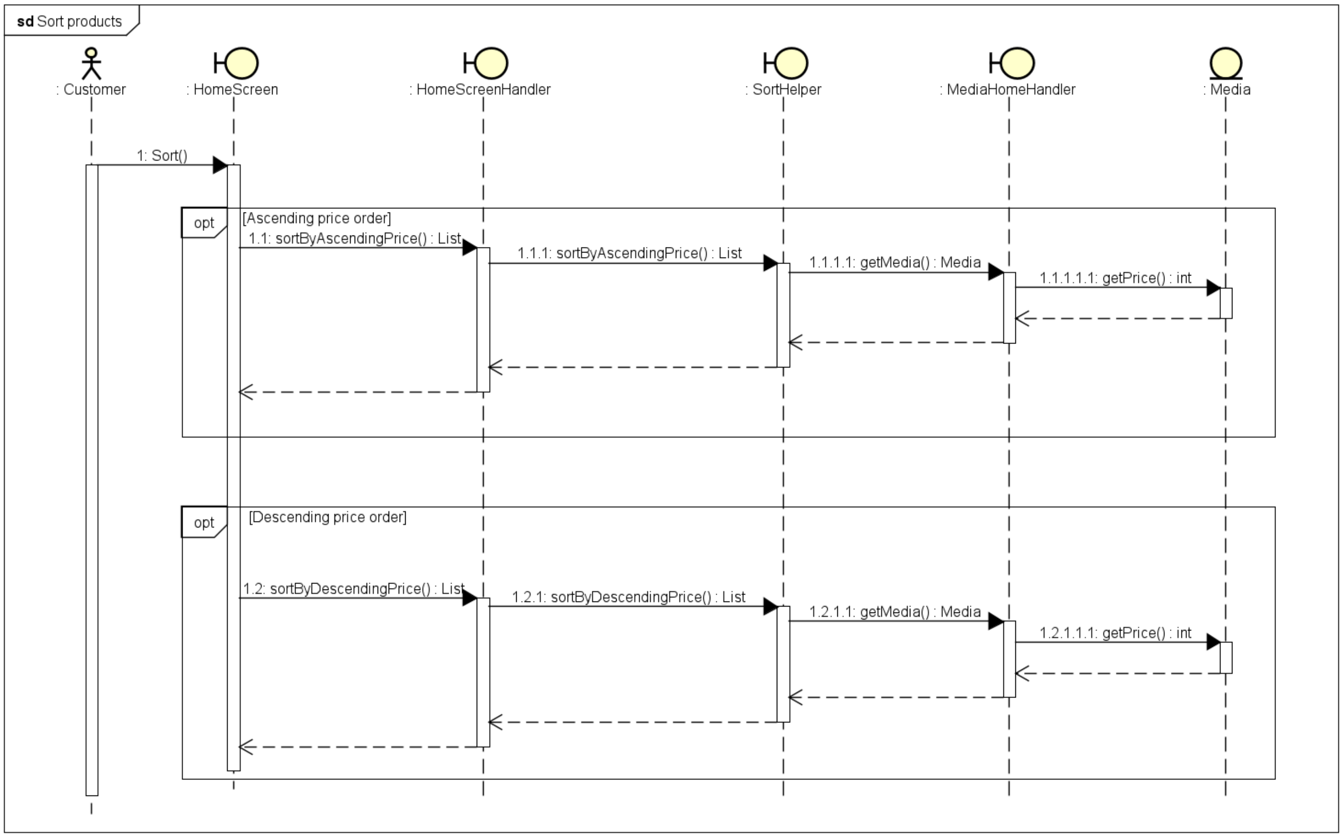
9. **Post condition**
10. **Activity diagram**

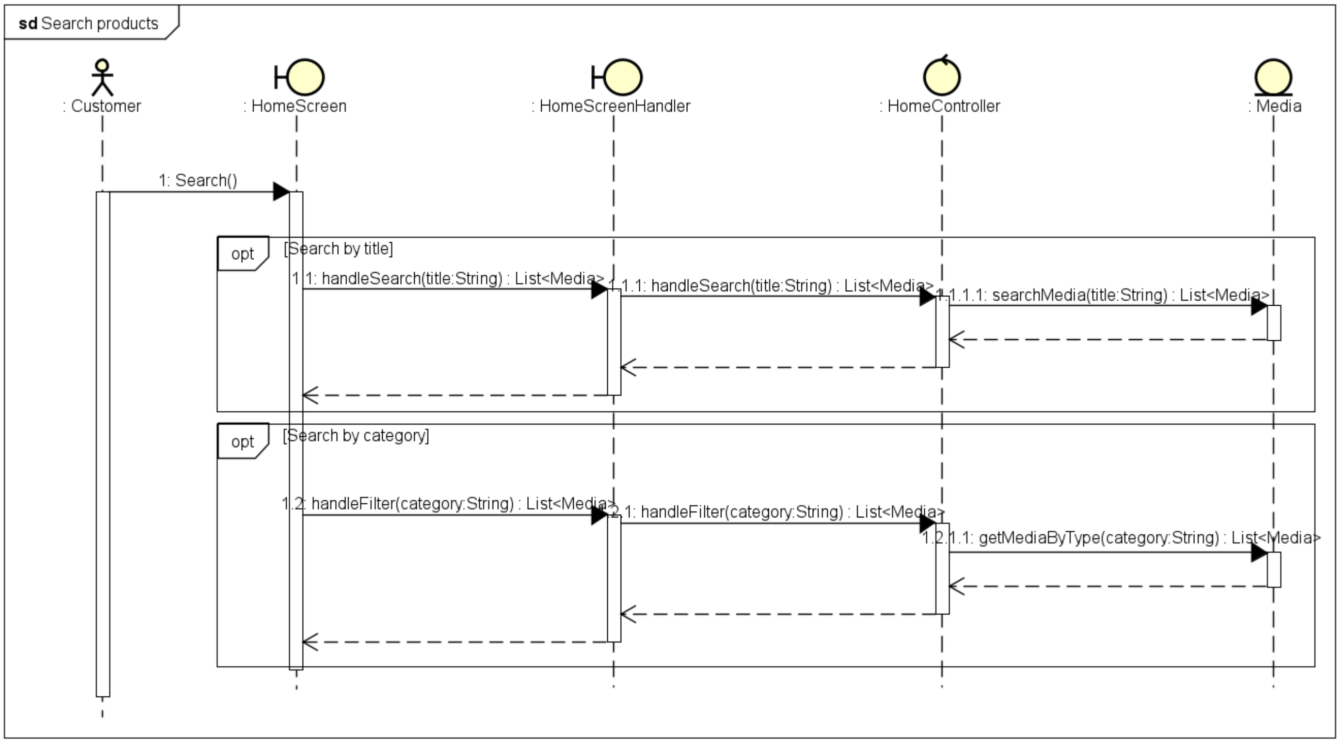

act Activity Diagram0
User
System
Click to Login Button
Display login form
Fill username and password
check user information
Invalid user
Display error
Check banned user
Yes
Check user roles
Display error
admin
product manager
Render manage user tab
Render manage media tab
Display manager screen

3. Usecase Analysis

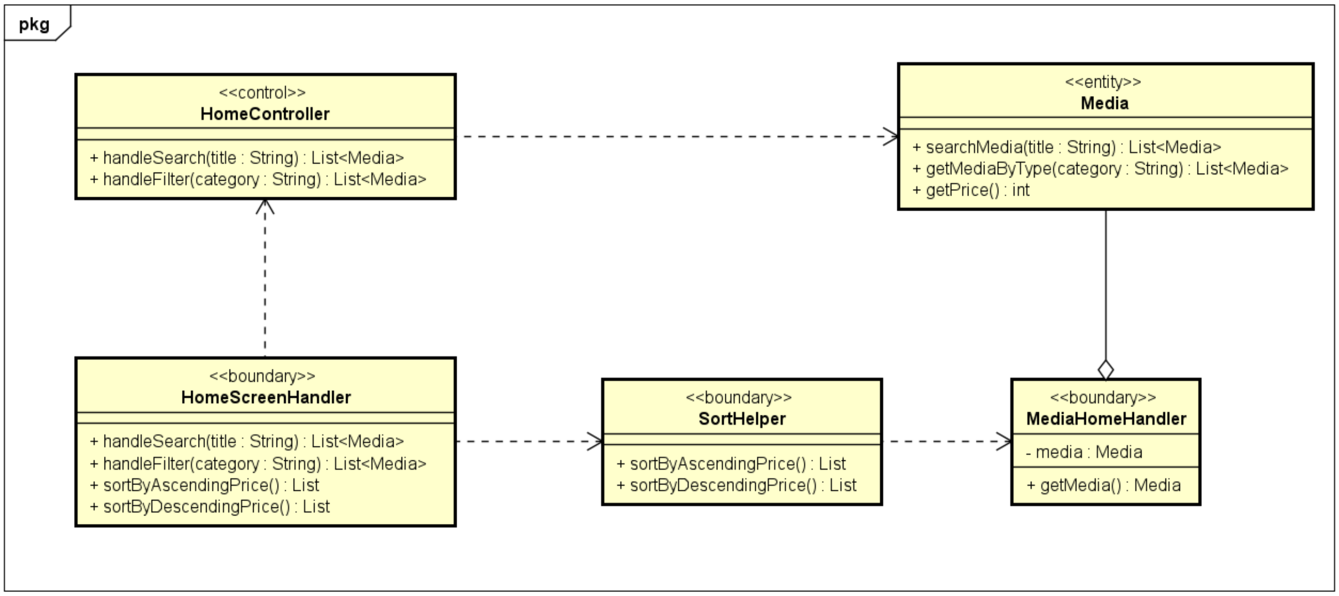
3.1. Usecase Analysis Search/Sort product.

Sequence Diagram:


sd Sort products
: Customer
: HomeScreen
: HomeScreenHandler
: SortHelper
: MediaHomeHandler
: Media
1: Sort()
opt
[Ascending price order]
1.1: sortByAscendingPrice() : List
1.1.1: sortByAscendingPrice() : List
1.1.1.1: getMedia() : Media
1.1.1.1.1: getPrice() : int
opt
[Descending price order]
1.2: sortByDescendingPrice() : List
1.2.1: sortByDescendingPrice() : List
1.2.1.1: getMedia() : Media
1.2.1.1.1: getPrice() : int

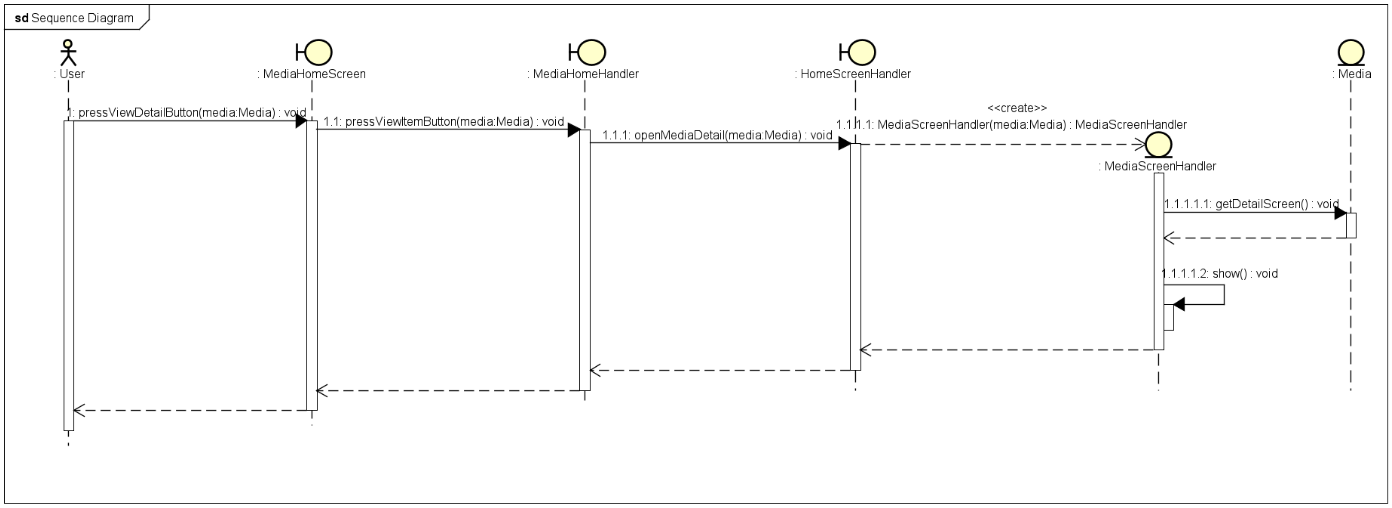

sd Search products
: Customer
: HomeScreen
: HomeScreenHandler
: HomeController
: Media
1: Search()
opt
[Search by title]
1.1: handleSearch(title:String) : List<Media>
1.1.1: handleSearch(title:String) : List<Media>
1.1.1.1: searchMedia(title:String) : List<Media>
opt
[Search by category]
1.2: handleFilter(category:String) : List<Media>
1.2.1: handleFilter(category:String) : List<Media>
1.2.1.1: getMediaByType(category:String) : List<Media>

Class Diagram:

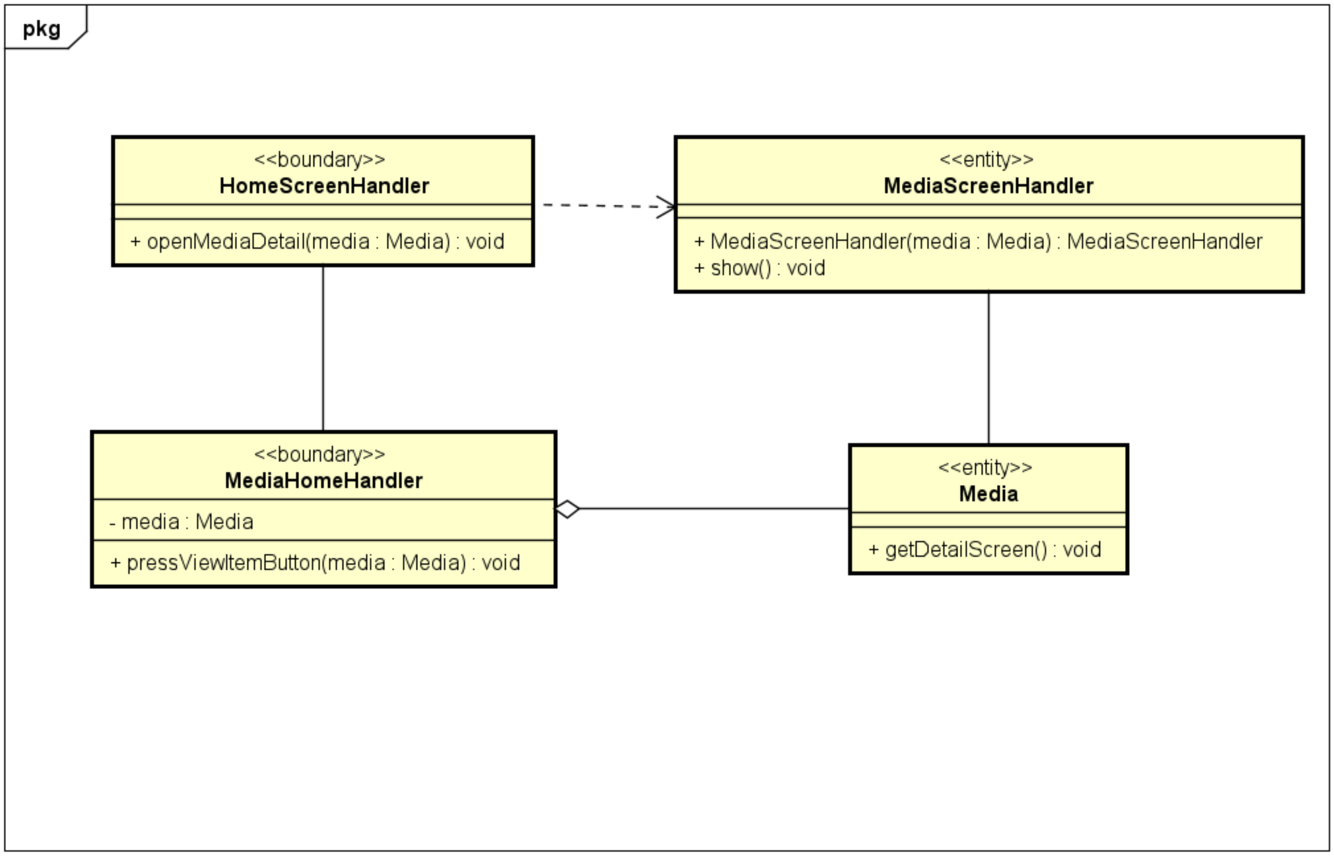

pkg
<<control>>
HomeController
+ handleSearch(title : String) : List<Media>
+ handleFilter(category : String) : List<Media>
<<entity>>
Media
+ searchMedia(title : String) : List<Media>
+ getMediaByType(category : String) : List<Media>
+ getPrice() : int
<<boundary>>
HomeScreenHandler
+ handleSearch(title : String) : List<Media>
+ handleFilter(category : String) : List<Media>
+ sortByAscendingPrice() : List
+ sortByDescendingPrice() : List
<<boundary>>
SortHelper
+ sortByAscendingPrice() : List
+ sortByDescendingPrice() : List
<<boundary>>
MediaHomeHandler
- media : Media
+ getMedia() : Media

3.2. Usecase Analysis View product detail.

Sequence Diagram:

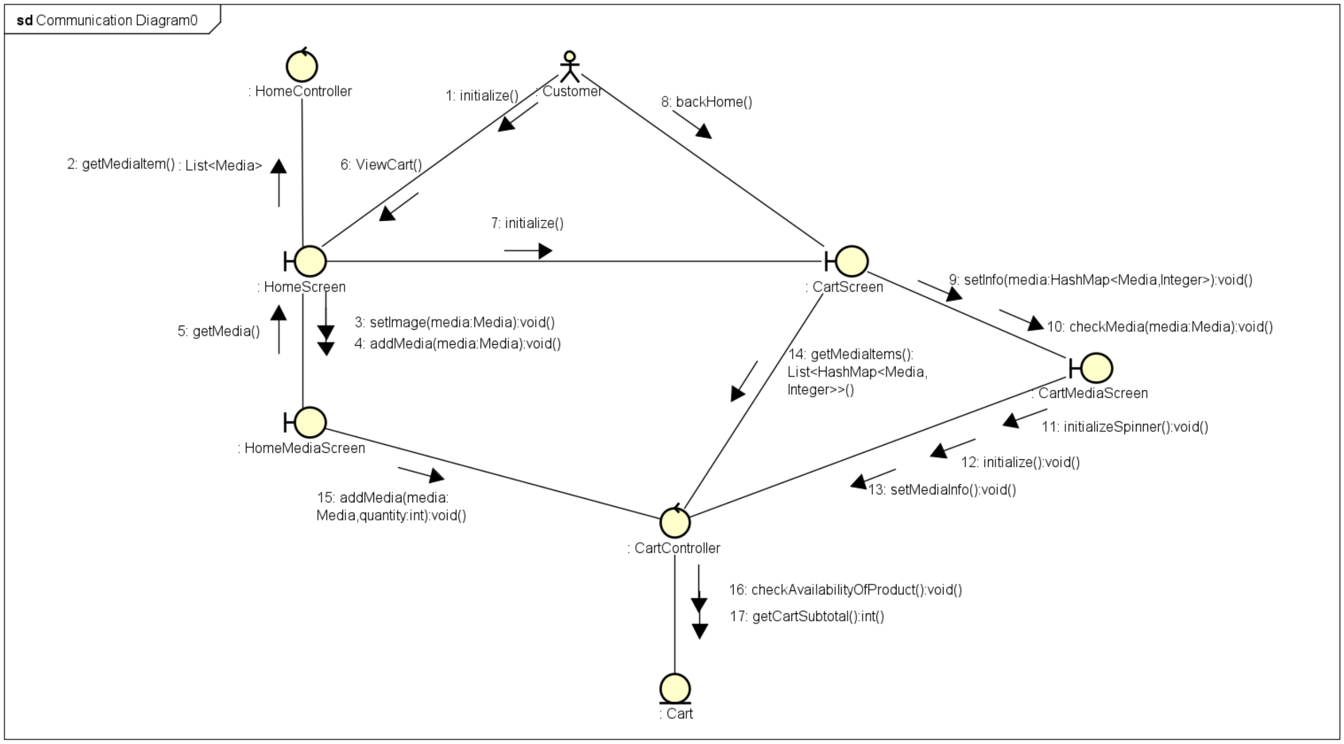

sd Sequence Diagram
: User
: MediaHomeScreen
: MediaHomeHandler
: HomeScreenHandler
: Media
1: pressViewDetailButton(media:Media) : void
1.1: pressViewItemButton(media:Media) : void
1.1.1: openMediaDetail(media:Media) : void
<<create>>
1.1.1.1: MediaScreenHandler(media:Media) : MediaScreenHandler
: MediaScreenHandler
1.1.1.1.1: getDetailScreen() : void
1.1.1.1.2: show() : void

Class Diagram:

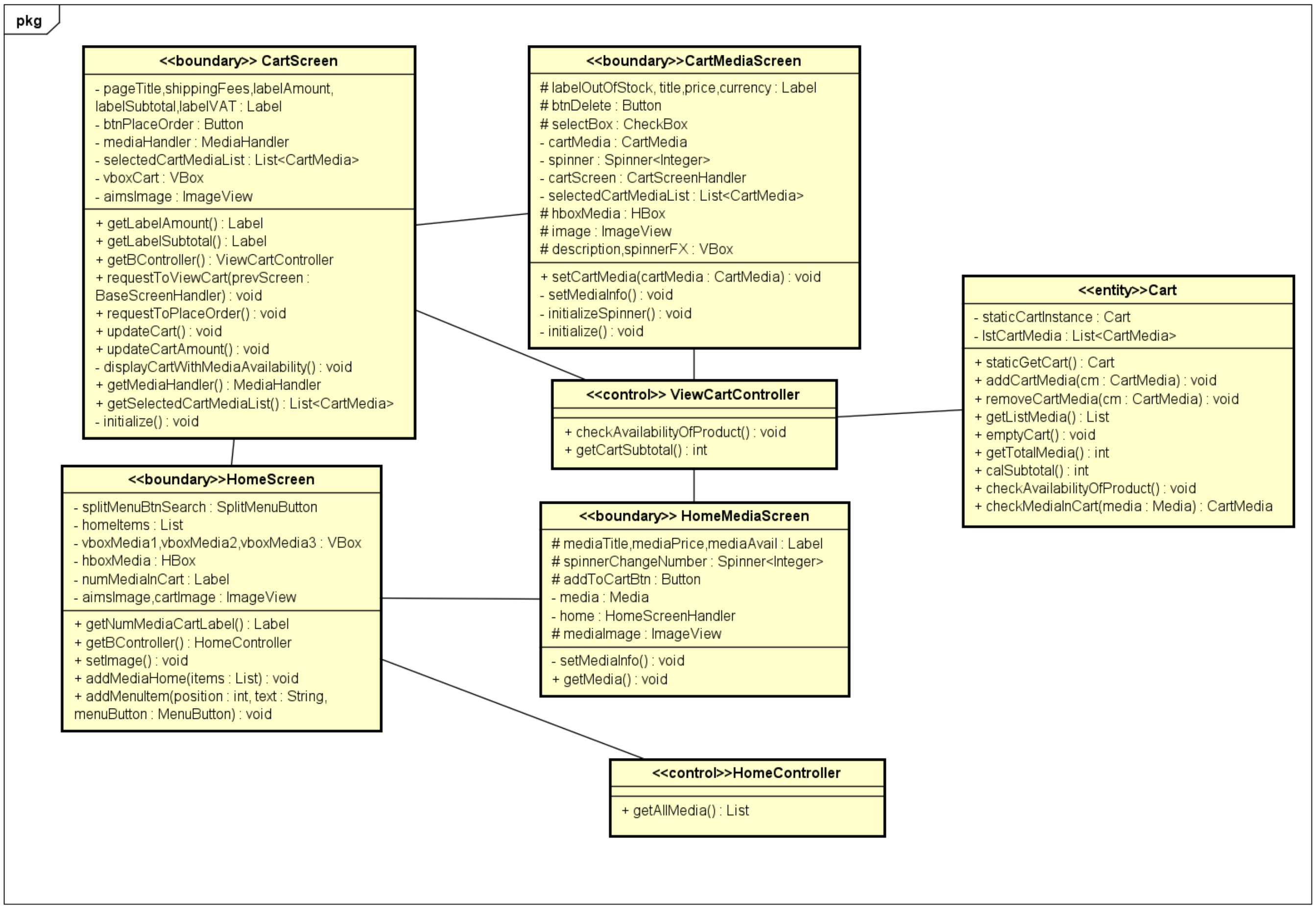

pkg
<<boundary>>
HomeScreenHandler
+ openMediaDetail(media : Media) : void
<<entity>>
MediaScreenHandler
+ MediaScreenHandler(media : Media) : MediaScreenHandler
+ show() : void
<<boundary>>
MediaHomeHandler
- media : Media
+ pressViewItemButton(media : Media) : void
<<entity>>
Media
+ getDetailScreen() : void

3.3. Usecase Analysis Manage Cart

Sequence Diagram:

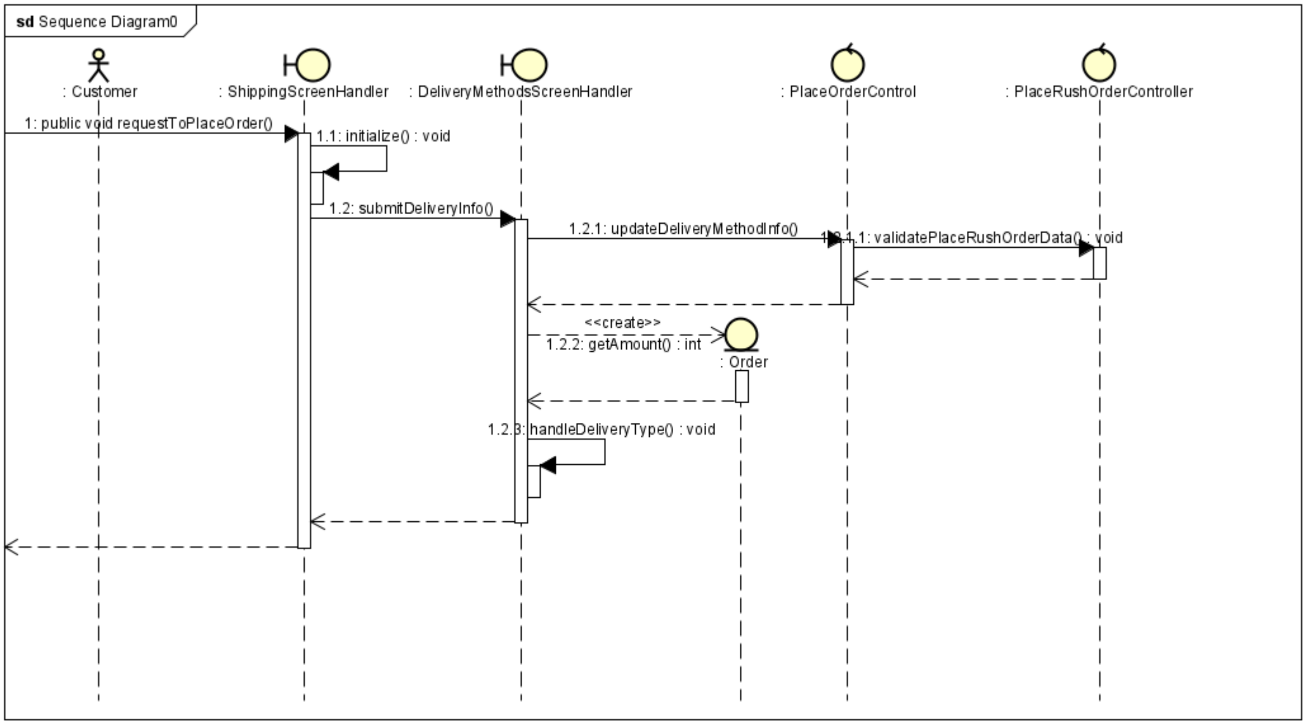

sd Communication Diagram0
: HomeController
: Customer
1: initialize()
8: backHome()
2: getMediaItem() : List<Media>
6: ViewCart()
7: initialize()
: HomeScreen
: CartScreen
9: setInfo(media:HashMap<Media,Integer>):void()
10: checkMedia(media:Media):void()
5: getMedia()
3: setImage(media:Media):void()
4: addMedia(media:Media):void()
14: getMediaItems(): List<HashMap<Media, Integer>>()
: CartMediaScreen
: HomeMediaScreen
11: initializeSpinner():void()
12: initialize():void()
13: setMediaInfo():void()
15: addMedia(media: Media,quantity:int):void()
: CartController
16: checkAvailabilityOfProduct():void()
17: getCartSubtotal():int()
: Cart

Class Diagram:

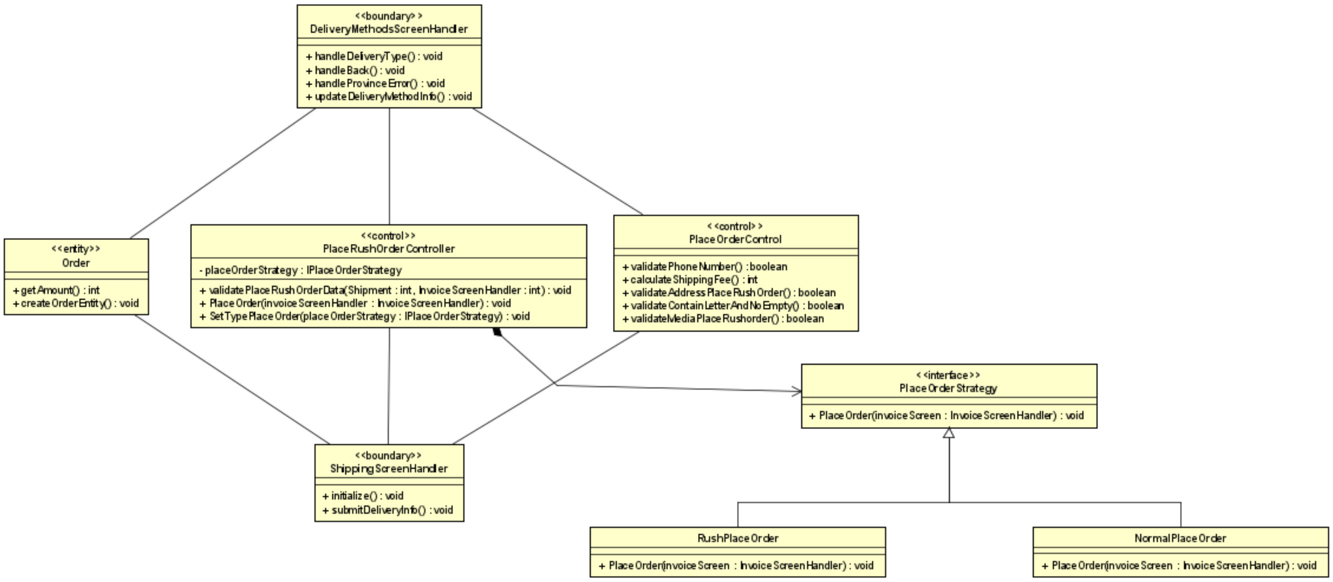

pkg
<<boundary>> CartScreen
- pageTitle,shippingFees,labelAmount, labelSubtotal,labelVAT : Label
- btnPlaceOrder : Button
- mediaHandler : MediaHandler
- selectedCartMediaList : List<CartMedia>
- vboxCart : VBox
- aimsImage : ImageView
+ getLabelAmount() : Label
+ getLabelSubtotal() : Label
+ getBController() : ViewCartController
+ requestToViewCart(prevScreen : BaseScreenHandler) : void
+ requestToPlaceOrder() : void
+ updateCart() : void
+ updateCartAmount() : void
- displayCartWithMediaAvailability() : void
+ getMediaHandler() : MediaHandler
+ getSelectedCartMediaList() : List<CartMedia>
- initialize() : void
<<boundary>>CartMediaScreen
# labelOutOfStock, title,price,currency : Label
# btnDelete : Button
# selectBox : CheckBox
- cartMedia : CartMedia
- spinner : Spinner<Integer>
- cartScreen : CartScreenHandler
- selectedCartMediaList : List<CartMedia>
# hboxMedia : HBox
# image : ImageView
# description,spinnerFX : VBox
+ setCartMedia(cartMedia : CartMedia) : void
- setMediaInfo() : void
- initializeSpinner() : void
- initialize() : void
<<entity>>Cart
- staticCartInstance : Cart
- lstCartMedia : List<CartMedia>
+ staticGetCart() : Cart
+ addCartMedia(cm : CartMedia) : void
+ removeCartMedia(cm : CartMedia) : void
+ getListMedia() : List
+ emptyCart() : void
+ getTotalMedia() : int
+ calSubtotal() : int
+ checkAvailabilityOfProduct() : void
+ checkMediaInCart(media : Media) : CartMedia
<<control>> ViewCartController
+ checkAvailabilityOfProduct() : void
+ getCartSubtotal() : int
<<boundary>>HomeScreen
- splitMenuBtnSearch : SplitMenuButton
- homeItems : List
- vboxMedia1,vboxMedia2,vboxMedia3 : VBox
- hboxMedia : HBox
- numMediaInCart : Label
- aimsImage,cartImage : ImageView
+ getNumMediaCartLabel() : Label
+ getBController() : HomeController
+ setImage() : void
+ addMediaHome(items : List) : void
+ addMenuItem(position : int, text : String, menuButton : MenuButton) : void
<<boundary>> HomeMediaScreen
# mediaTitle,mediaPrice,mediaAvail : Label
# spinnerChangeNumber : Spinner<Integer>
# addToCartBtn : Button
- media : Media
- home : HomeScreenHandler
# mediaImage : ImageView
- setMediaInfo() : void
+ getMedia() : void
<<control>>HomeController
+ getAllMedia() : List

3.4. Usecase Analysis Place Order.

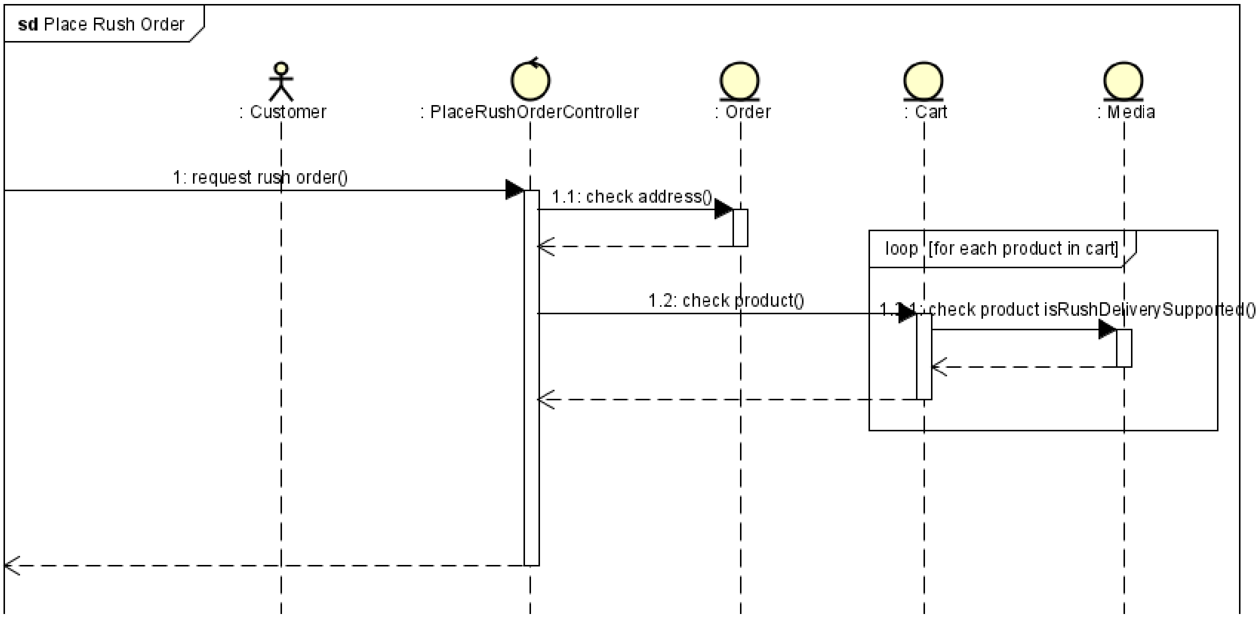
Sequence Diagram:


sd Sequence Diagram0
: Customer
: ShippingScreenHandler
: DeliveryMethodsScreenHandler
: PlaceOrderControl
: PlaceRushOrderController
1: public void requestToPlaceOrder()
1.1: initialize() : void
1.2: submitDeliveryInfo()
1.2.1: updateDeliveryMethodInfo()
1.2.1.1: validatePlaceRushOrderData() : void
<<create>>
1.2.2: getAmount() : int
: Order
1.2.3: handleDeliveryType() : void

Class diagram:

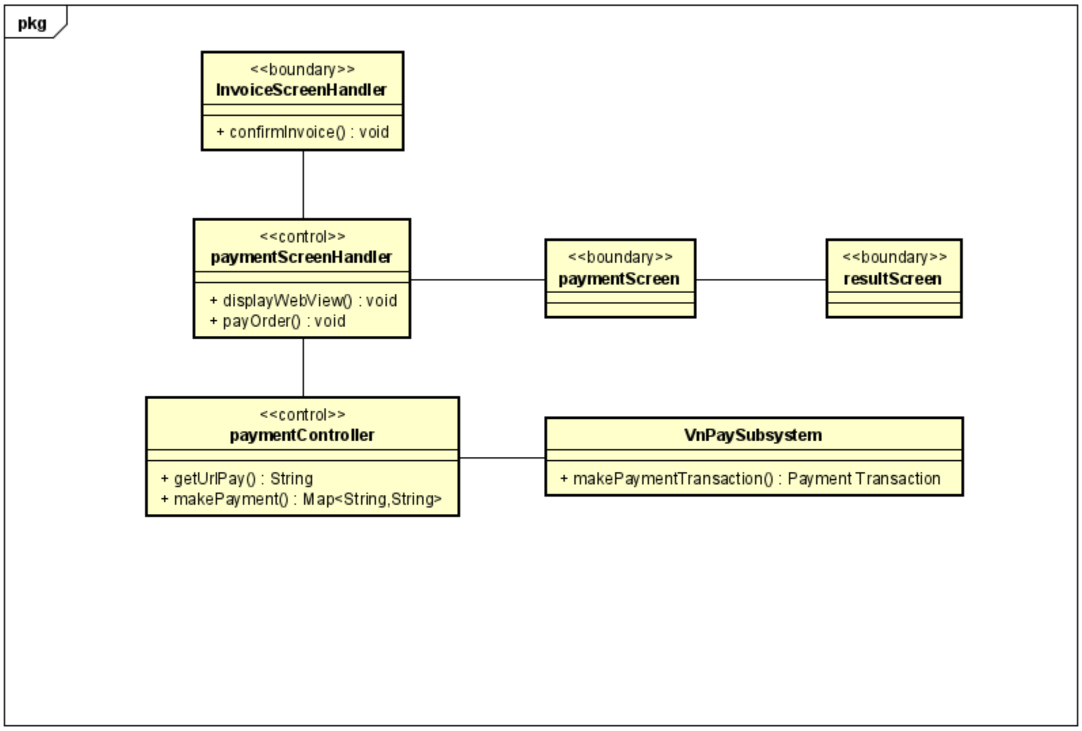

<<boundary>>
DeliveryMethodsScreenHandler
+ handleDeliveryType() : void
+ handleBack() : void
+ handleProvinceError() : void
+ updateDeliveryMethodInfo() : void
<<entity>>
Order
+ getAmount() : int
+ createOrderEntity() : void
<<control>>
PlaceRushOrderController
- placeOrderStrategy : IPlaceOrderStrategy
+ validatePlaceRushOrderData(Shipment : int, InvoiceScreenHandler : int) : void
+ PlaceOrder(invoiceScreenHandler : InvoiceScreenHandler) : void
+ SetTypePlaceOrder(placeOrderStrategy : IPlaceOrderStrategy) : void
<<control>>
PlaceOrderControl
+ validatePhoneNumber() : boolean
+ calculateShippingFee() : int
+ validateAddressPlaceRushOrder() : boolean
+ validateContainLetterAndNoEmpty() : boolean
+ validateMediaPlaceRushorder() : boolean
<<interface>>
PlaceOrderStrategy
+ PlaceOrder(invoiceScreen : InvoiceScreenHandler) : void
<<boundary>>
ShippingScreenHandler
+ initialize() : void
+ submitDeliveryInfo() : void
RushPlaceOrder
+ PlaceOrder(invoiceScreen : InvoiceScreenHandler) : void
NormalPlaceOrder
+ PlaceOrder(invoiceScreen : InvoiceScreenHandler) : void

3.1. Usecase Analysis Place Rush Order

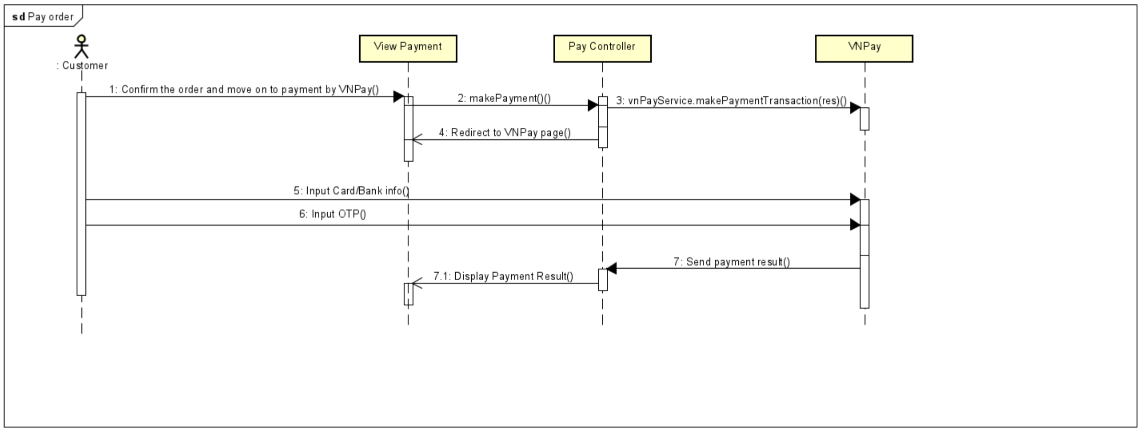

sd Place Rush Order
: Customer
: PlaceRushOrderController
: Order
: Cart
: Media
1: request rush order()
1.1: check address()
loop [for each product in cart]
1.2: check product()
1.1.1: check product isRushDeliverySupported()

3.2 Usecase Analysis: Pay order.

Class diagram:

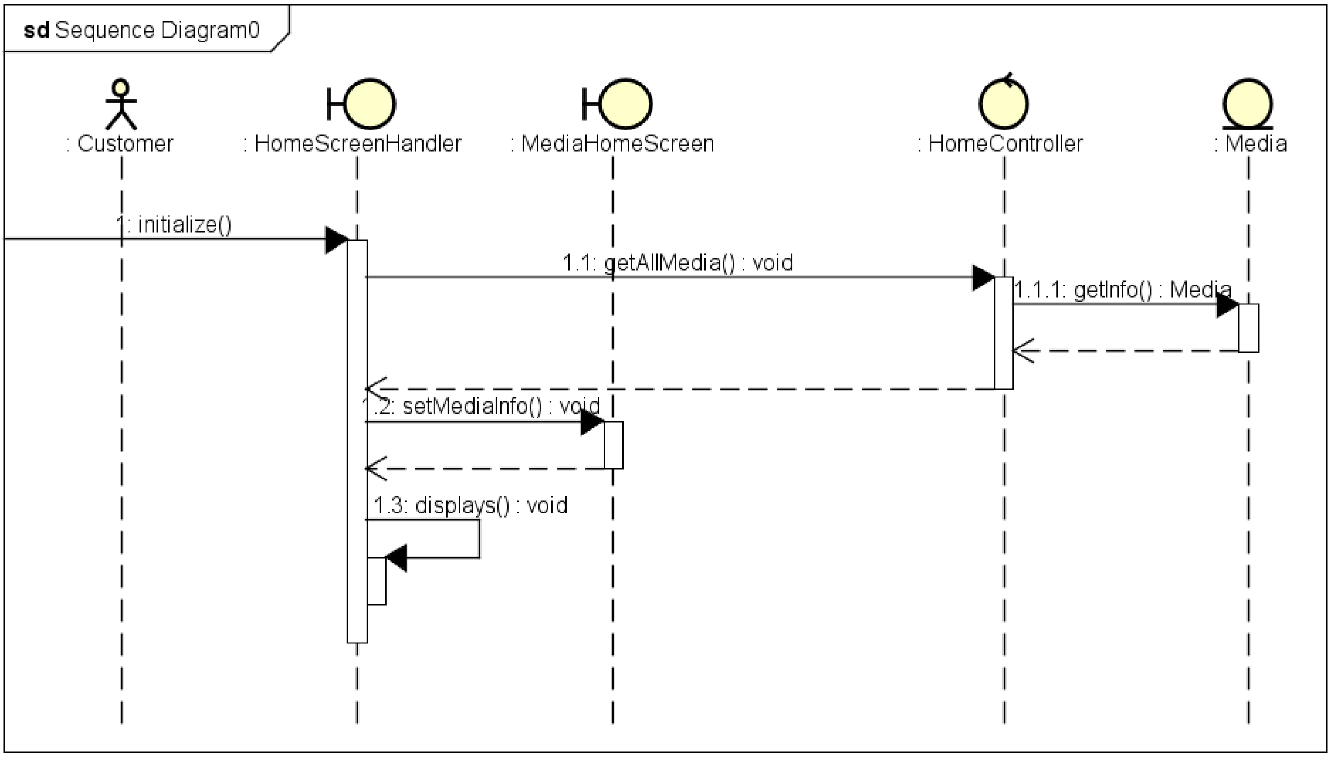

pkg
<<boundary>>
InvoiceScreenHandler
+ confirmInvoice() : void
<<control>>
paymentScreenHandler
+ displayWebView() : void
+ payOrder() : void
<<boundary>>
paymentScreen
<<boundary>>
resultScreen
<<control>>
paymentController
+ getUrlPay() : String
+ makePayment() : Map<String,String>
VnPaySubsystem
+ makePaymentTransaction() : Payment Transaction

Sequence Diagram:

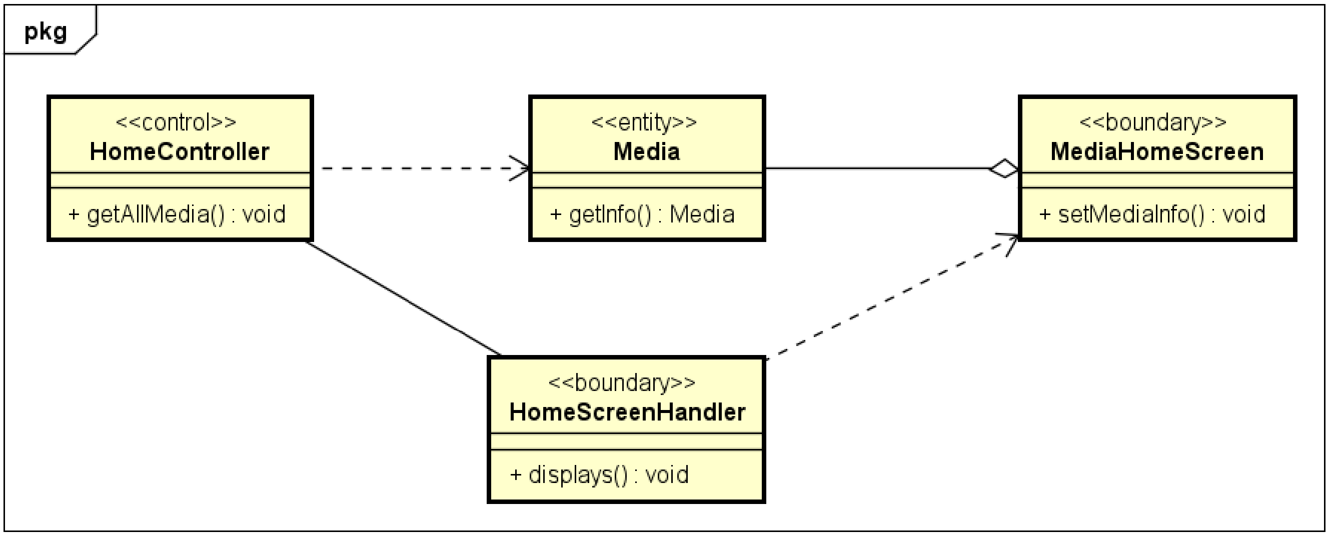

sd Pay order
: Customer
View Payment
Pay Controller
VNPay
1: Confirm the order and move on to payment by VNPay()
2: makePayment()()
3: vnPayService.makePaymentTransaction(res)()
4: Redirect to VNPay page()
5: Input Card/Bank info()
6: Input OTP()
7: Send payment result()
7.1: Display Payment Result()

3.5. Usecase Analysis View list of products

Sequence Diagram:

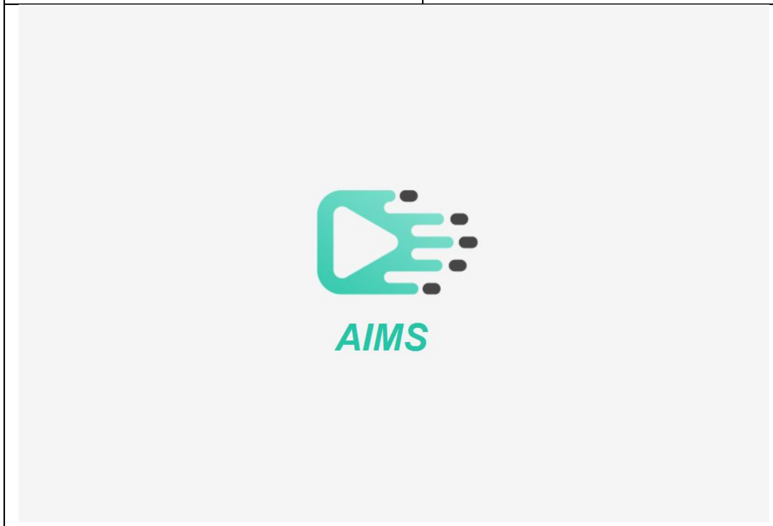

sd Sequence Diagram0
: Customer
: HomeScreenHandler
: MediaHomeScreen
: HomeController
: Media
1: initialize()
1.1: getAllMedia() : void
1.1.1: getInfo() : Media
1.2: setMediaInfo() : void
1.3: displays() : void

Class Diagram:


pkg
<<control>>
HomeController
+ getAllMedia() : void
<<entity>>
Media
+ getInfo() : Media
<<boundary>>
MediaHomeScreen
+ setMediaInfo() : void
<<boundary>>
HomeScreenHandler
+ displays() : void

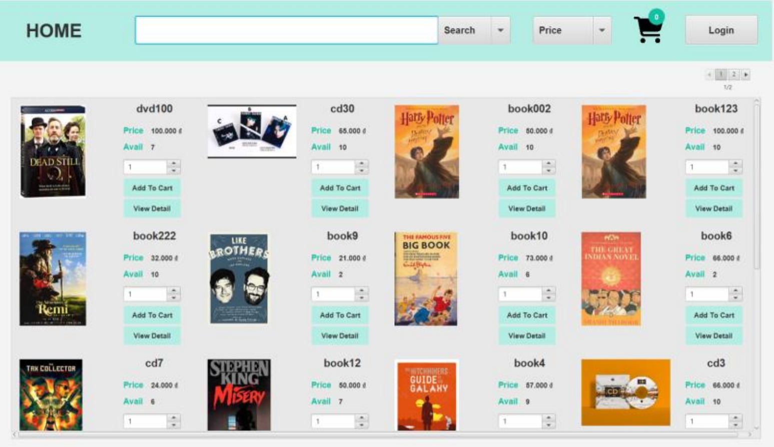
4. Interface Design

4.1. Splash Screen

<table>
<tr><td colspan="2">Aims Software</td><td>Date of creation</td><td>Approved by</td><td>Reviewed by</td><td>Persion in charge</td></tr>
<tr><td>Screen specification</td><td>Splash screen</td><td>17/06/2024</td><td></td><td></td><td>Trinh Tien Dung</td></tr>
<tr><td colspan="2" rowspan="2">
</td><td>Control</td><td>Operation</td><td colspan="2">Function</td></tr>
<tr><td>Main area</td><td>None</td><td colspan="2">Loading</td></tr>
</table>

4.2. Home Screen

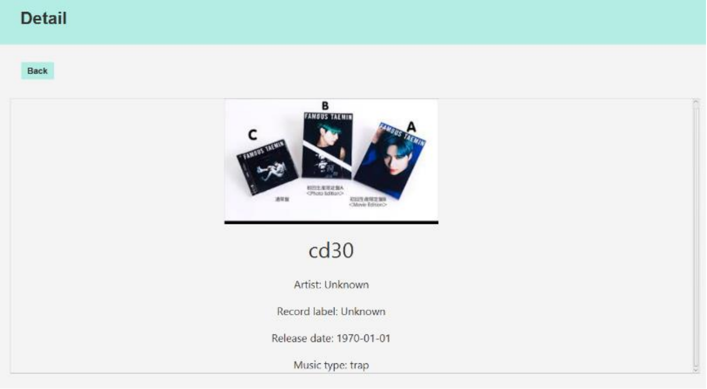
Aims Software	**Date of creation**	**Approved by**	**Reviewed by**	**Persion in charge**

Screen specification	**Home screen**	17/06/2024			Trinh Tien Dung
 HOME Search Price 0 Login 1/2 dvd100 Price 100.000 đ Avail 7 cd30 Price 65.000 đ Avail 10 book002 Price 50.000 đ Avail 10 book123 Price 100.000 đ Avail 10 book222 Price 32.000 đ Avail 10 book9 Price 21.000 đ Avail 2 book10 Price 73.000 đ Avail 6 book6 Price 66.000 đ Avail 2 cd7 Price 24.000 đ Avail 6 book12 Price 50.000 đ Avail 7 book4 Price 57.000 đ Avail 9 cd3 Price 66.000 đ Avail 10 Add To Cart View Detail		**Control**	**Operation**	**Function**	
		Page title	Click	Reload home screen	
		Search bar	Type and Click	Category or title of search items and search	
		Cart logo	Click	View items in the cart	
		Scroll bar	Scroll	Scroll to see hidden items	
		Login button	Click	Login as user	
		Price button	Select and click	Sort items by price	
		Pagination	Click	Go to other pages	
		Main area	Click	Items detail info and add to cart	

Screen	**Main screen**			
Field name	**Type**	**Limitation**	**Attribute**	**Remarks**
Media title	Text	40 characters	Bold, black	Left-justified
Price	Digits	10 digits	Bold, black	Left-justified Dot for thousand separation
Avail	Digits	5 digits	Bold, black	Left-justified
Item image	Image	158x178 pixels	None	None

4.3. Product detail screen

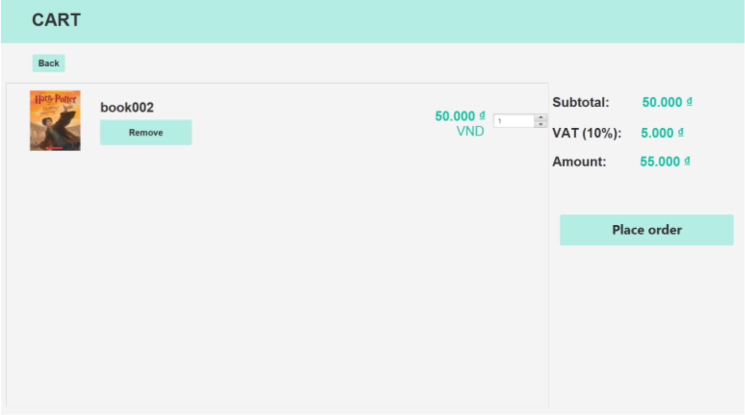
Aims Software	**Date of creation**	**Approved by**	**Reviewed by**	**Persion in charge**

Screen specification	**Product detail screen**	17/06/2024			Trinh Tien Dung
 cd30 Artist: Unknown Record label: Unknown Release date: 1970-01-01 Music type: trap		**Control**	**Operation**	**Function**	
		Area for display items in the cart	Initial	Displays detail information base on category	
		Back button	Click	Go to back to home screen	

Screen	**Main screen**			
Field name	**Type**	**Limitation**	**Attribute**	**Remarks**
Media title	Text	40 characters	Black	None
Item image	Image	400 width pixels	None	None
Media detail information	Text	1000 characters	Black	None

4.4. Cart

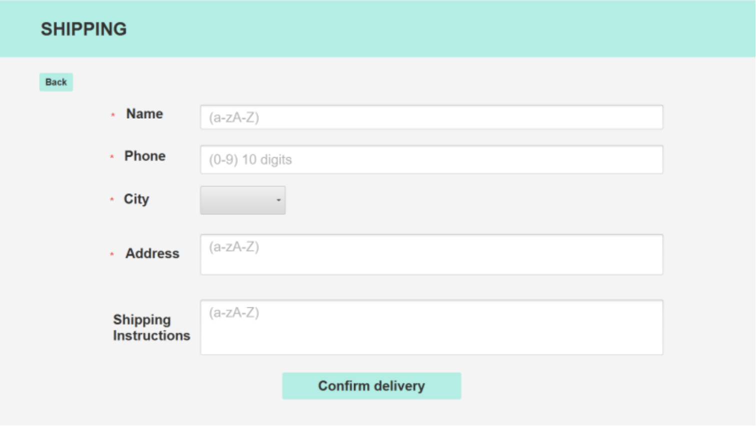
Aims Software		**Date of creation**	**Approved by**	**Reviewed by**	**Persion in charge**
Screen specification	**View cart screen**	17/06/2024			Trinh Tien Dung
		Control	**Operation**	**Function**	
		Area for displaying the subtotal	Initial	Display the subtotal	
		Area for display items in the cart	Initial	***Display the product with corresponding information***	

 CART Back book002 Remove 50.000 đ VND 1 Subtotal: 50.000 đ VAT (10%): 5.000 đ Amount: 55.000 đ Place order	Place order button	Click	Display the Delivery form
	Edit number button	Click/Input	Change number of selected products
	Remove button	Click	Remove product from cart
	Back button	Click	Back to home screen

Screen	**Cart screen**			
Field name	**Type**	**Limitation**	**Attribute**	**Remarks**
Media title	Text	40 characters	Bold, black	Left-justified
Quantity	Digits	5 digits	None	Left-justified
Price	Digits	10 digits	Green	Left-justified Dot for thousand separation
Subtotal				
Currency	Text	3 characters	All caps	None
Item thumbnail	Image	95x103 pixels	None	None

4.5. Shipping information Screen

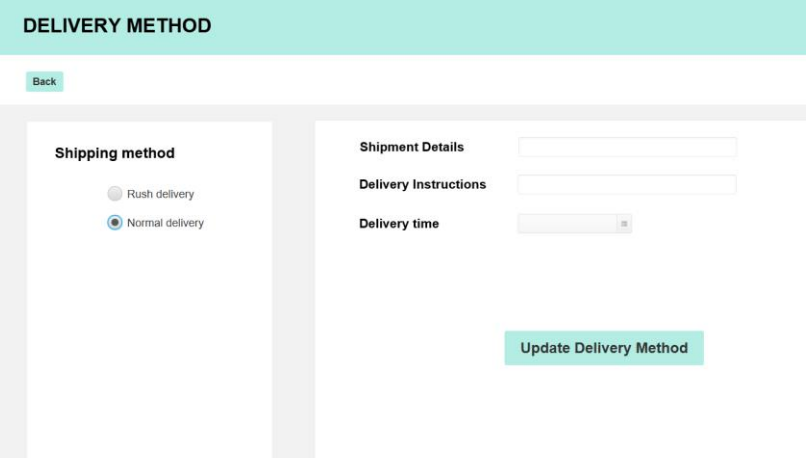
Aims Software		**Date of creation**	**Approved by**	**Reviewed by**	**Persion in charge**
Screen specification	**Delivery Form**	17/06/2024			Trinh Tien Dung
		Control	**Operation**	**Function**	
		Area for filling the name	Input	***Fill name of the customer***	

 SHIPPING Back Name (a-zA-Z) Phone (0-9) 10 digits City Address (a-zA-Z) Shipping Instructions (a-zA-Z) Confirm delivery	***Area for filling phone***	Input	***Fill phone of the customer***
	Area for filling address	Input	***Fill specific address of the customer***
	City selecting dropdown	Select	Change city of customer
	Area for filling shipping instruction	Input	***Fill shipping instruction of the customer***
	Submit button	Click	Save information and process to check rush order
	Back button	Click	Back to previous page

Screen	**Deliver screen**			
Field name	**Type**	**Limitation**	**Attribute**	**Remarks**
Province/City	Select from list	None	None	None
Delivery option	Select from list	None	None	None

4.6. Delivery Method

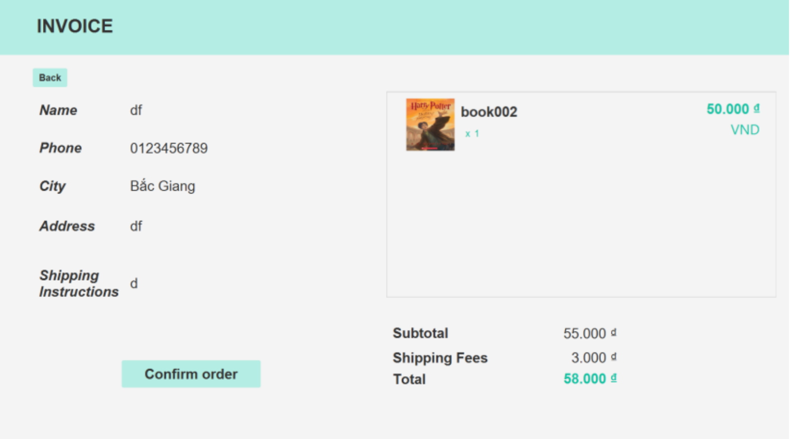
Aims Software		**Date of creation**	**Approved by**	**Reviewed by**	**Person in charge**
Screen specification	**Delivery Method**	17/06/2024			Trinh Tien Dung
		Control	**Operation**	**Function**	
		Shipping method	Select	Select delivery method	
		Area for inputting Shipment Detail	Input	Input shipment detail	
		Area for inputting delivery instruction	Input	***Update the delivery instruction of order***	

	Area for selected expected delivery time	Select	***Update the delivery expected time of order***
	Confirm button	Click	Save information and process to invoice screen
	Back button	Click	Back to previous screen

DELIVERY METHOD
Back
Shipping method
Rush delivery
Normal delivery
Shipment Details
Delivery Instructions
Delivery time
Update Delivery Method

Shipping method: ○ Rush delivery ● Normal delivery

4.7. Invoice Screen

Aims Software		**Date of creation**	**Approved by**	**Reviewed by**	**Persion in charge**
Screen specification	**Invoice Screen**	07/04/2024			Trinh Tien Dung
		**Control**	**Operation**	**Function**	
		Area for displaying the selected address	Initial	Display the selected shipping address	
		Area for displaying the customer's phone number	Initial	Display the customer's phone number	
		Area for displaying the customer's name	Initial	Display the customer's name	
		Area for displaying the customer's instruction	Initial	Display the customer's instruction	

INVOICE
Back
Confirm order

Name df
Phone 0123456789
City Bắc Giang
Address df
Shipping Instructions d

book002 x 1 50.000 ₫ VND

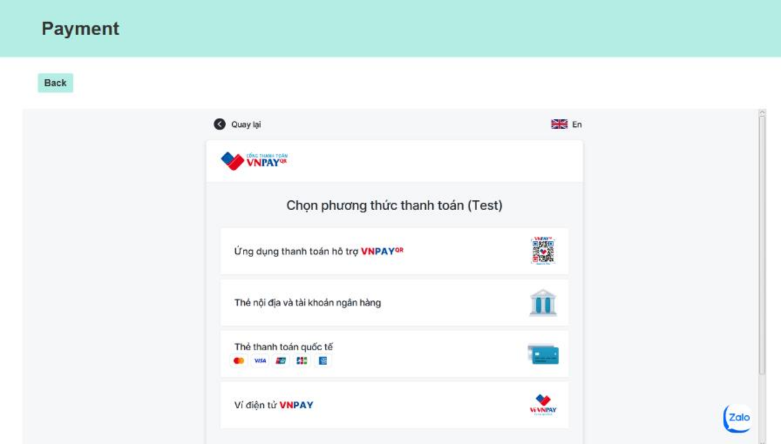
Subtotal	55.000 ₫
Shipping Fees	3.000 ₫
Total	58.000 ₫

	Area for displaying the shipping fee	Initial	Display the customer’s shipping fee
	Area for displaying the product list with specified info for each product	Initial	Display the customer’s product list
	Confirm button	Click	Save information and process to invoice screen

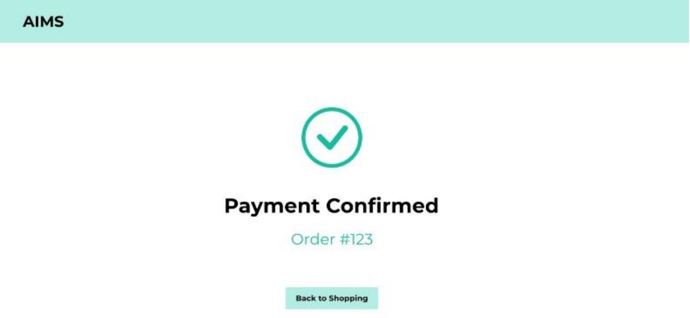
Screen	**Normal order invoice & Rush order invoice screen**			
Field name	**Type**	**Limitation**	**Attribute**	**Remarks**
Media title	Text	40 characters	Bold, black	Left-justified
Subtotal	Digits	10 digits	None	Left-justified Dot for thousand separation
Shipping fee				
Price			Green	
Items				
Total				
Item thumbnail	Image	163x128 pixels	None	None
Customer name	Text	50 characters	Italic	None
Province/City		30 characters		
Address		100 characters		
Shipping instruction		100 characters		
Customer phone	Digits	10 digits	Italic	None

4.9. PaymentScreen

Aims Software		**Date of creation**	**Approved by**	**Reviewed by**	**Persion in charge**
Screen specification	**Payment Screen**	17/06/2024			Trinh Tien Dung

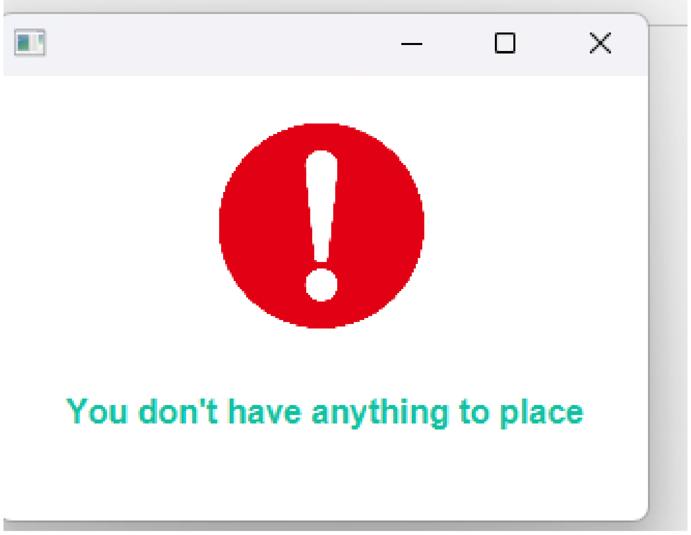
	Control	Operation	Function
 Payment Back Quay lại En VNPAY Chọn phương thức thanh toán (Test) Ứng dụng thanh toán hỗ trợ VNPAY Thẻ nội địa và tài khoản ngân hàng Thẻ thanh toán quốc tế Ví điện tử VNPAY Zalo	***Area for redirect to VNPay***	***Initial***	***Pay order through VNPay***

4.9. PaymentResultScreen

Aims Software		**Date of creation**	**Approved by**	**Reviewed by**	**Persion in charge**
Screen specification	**Result Screen**	17/06/2024			Trinh Tien Dung
 AIMS Payment Confirmed Order #123 Back to Shopping		**Control**	**Operation**	**Function**	
		Area for displaying successful order	***Initial***	***Display successful order***	
		Area for displaying message	***Initial***	***Displays message***	
		Button	***Click***	***Back to home screen***	

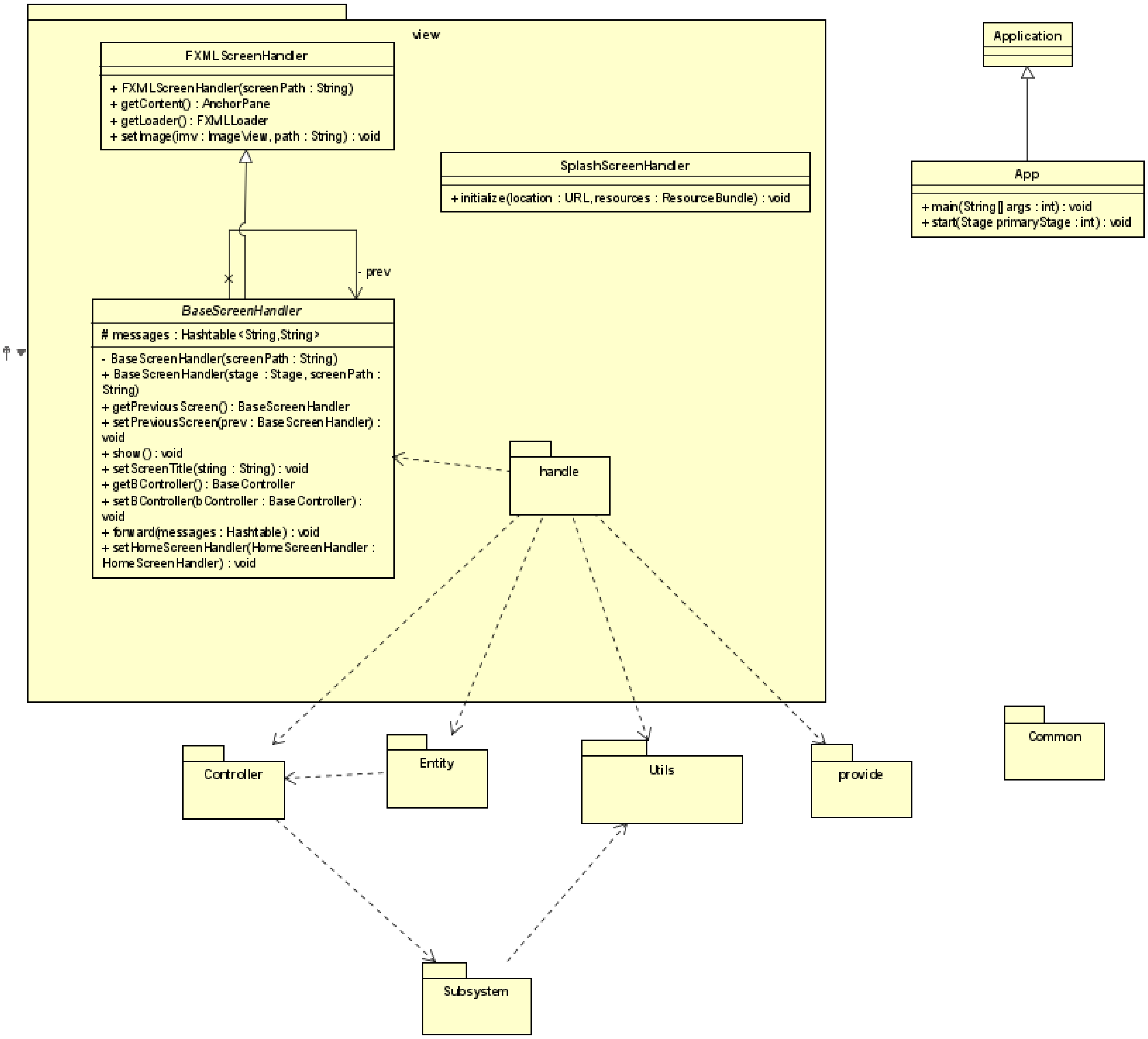
4.10. Popup Screen

Aims Software		**Date of creation**	**Approved by**	**Reviewed by**	**Persion in charge**
Screen specification	**Popup Screen**	17/06/2024			Trinh Tien Dung
		Control	**Operation**	**Function**	

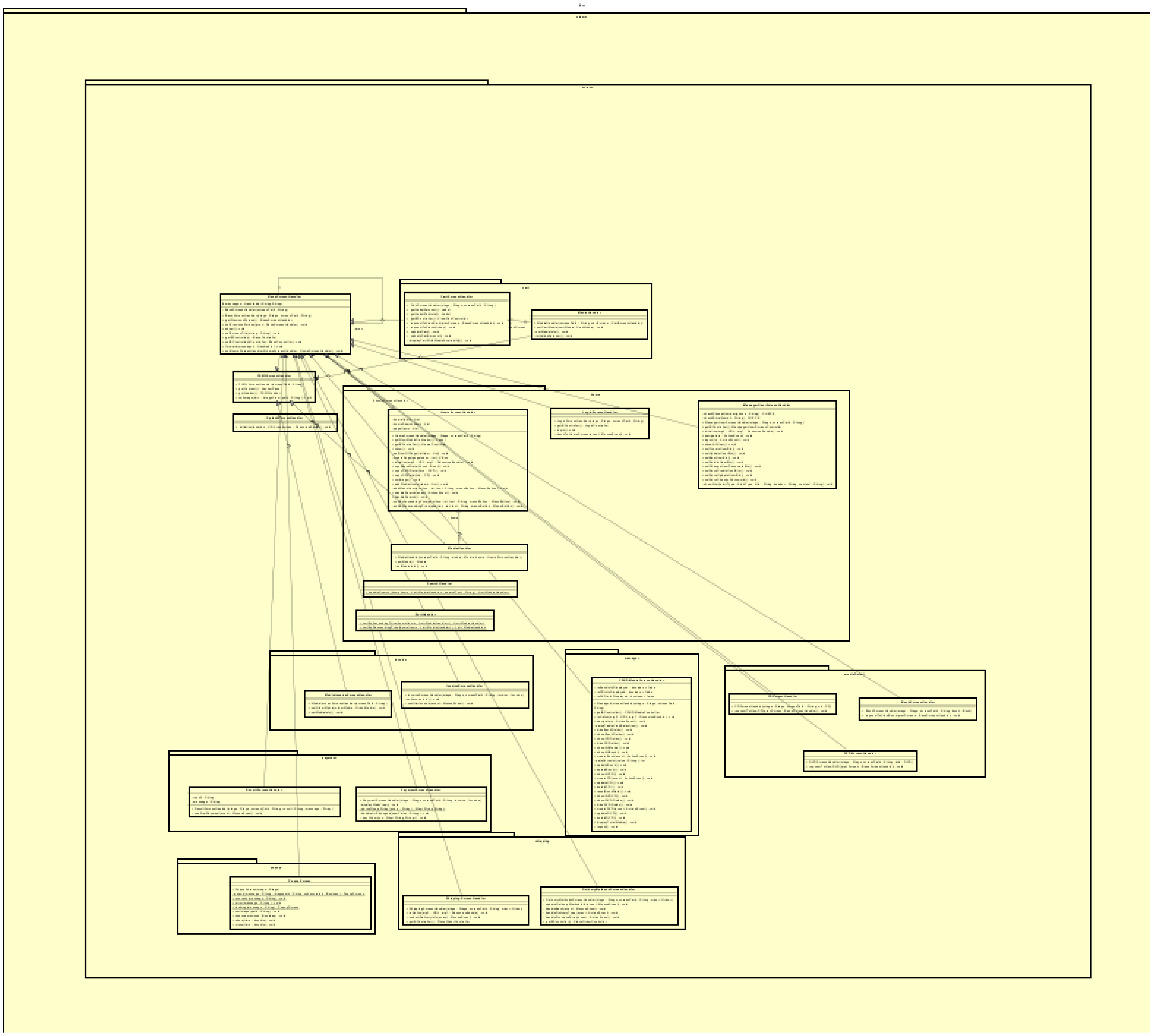
 You don't have anything to place	***Area for displaying icon***	***Initial***	***Display icon***
	Area for displaying message	***Initial***	***Displays message***
	Button	***Click***	***Close popup window***

5. ANALYSIS CLASS DIAGRAM

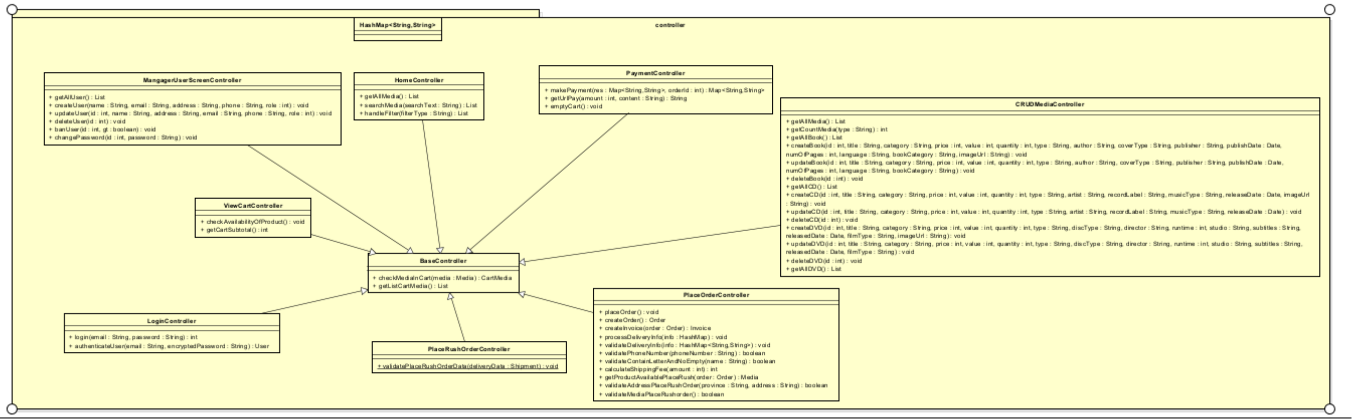
5.1. General Class Diagram


view
FXMLScreenHandler
+ FXMLScreenHandler(screenPath : String)
+ getContent() : AnchorPane
+ getLoader() : FXMLLoader
+ setImage(imv : ImageView, path : String) : void
SplashScreenHandler
+ initialize(location : URL, resources : ResourceBundle) : void
Application
App
+ main(String [] args : int) : void
+ start(Stage primaryStage : int) : void
- prev
BaseScreenHandler
# messages : Hashtable<String,String>
- BaseScreenHandler(screenPath : String)
+ BaseScreenHandler(stage : Stage, screenPath : String)
+ getPreviousScreen() : BaseScreenHandler
+ setPreviousScreen(prev : BaseScreenHandler) : void
+ show() : void
+ setScreenTitle(string : String) : void
+ getBController() : BaseController
+ setBController(bController : BaseController) : void
+ forward(messages : Hashtable) : void
+ setHomeScreenHandler(HomeScreenHandler : HomeScreenHandler) : void
handle
Controller
Entity
Utils
provide
Common
Subsystem

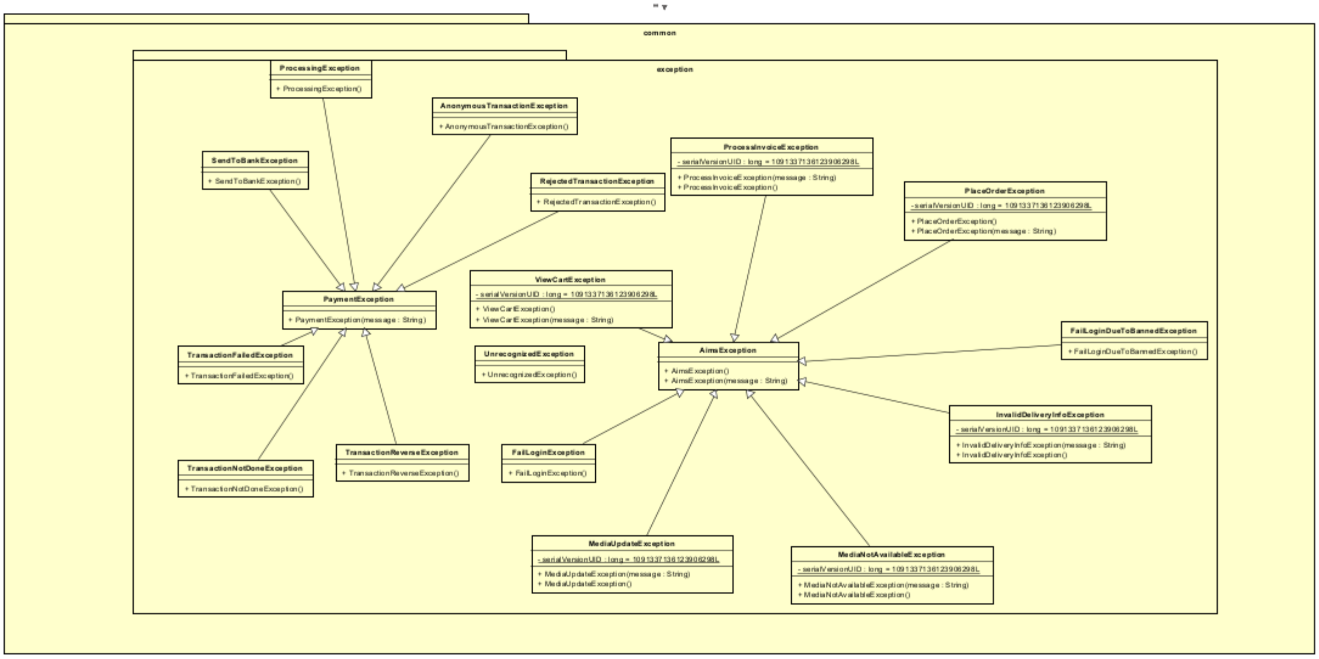
5.2.1 Class Diagram for Package View



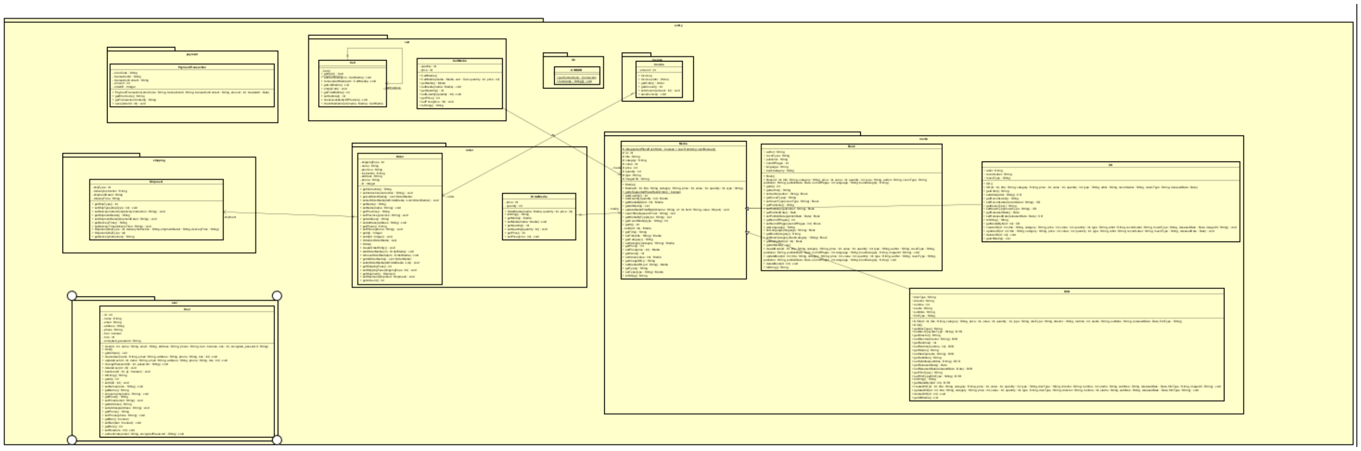
5.2.2 Class Diagram for Package Controller


HashMap<String,String>
controller
ManagerUserScreenController
+ getAllUser() : List
+ createUser(name : String, email : String, address : String, phone : String, role : int) : void
+ updateUser(id : int, name : String, address : String, email : String, phone : String, role : int) : void
+ deleteUser(id : int) : void
+ banUser(id : int, gt : boolean) : void
+ changePassword(id : int, password : String) : void
HomeController
+ getAllMedia() : List
+ searchMedia(searchText : String) : List
+ handleFilter(filterType : String) : List
PaymentController
+ makePayment(res : Map<String,String>, orderId : int) : Map<String,String>
+ getUrlPay(amount : int, content : String) : String
+ emptyCart() : void
CRUDMediaController
+ getAllMedia() : List
+ getCountMedia(type : String) : int
+ getAllBook() : List
+ createBook(id : int, title : String, category : String, price : int, value : int, quantity : int, type : String, author : String, coverType : String, publisher : String, publishDate : Date, numOfPages : int, language : String, bookCategory : String, imageUrl : String) : void
+ updateBook(id : int, title : String, category : String, price : int, value : int, quantity : int, type : String, author : String, coverType : String, publisher : String, publishDate : Date, numOfPages : int, language : String, bookCategory : String) : void
+ deleteBook(id : int) : void
+ getAllCD() : List
+ createCD(id : int, title : String, category : String, price : int, value : int, quantity : int, type : String, artist : String, recordLabel : String, musicType : String, releaseDate : Date, imageUrl : String) : void
+ updateCD(id : int, title : String, category : String, price : int, value : int, quantity : int, type : String, artist : String, recordLabel : String, musicType : String, releaseDate : Date) : void
+ deleteCD(id : int) : void
+ createDVD(id : int, title : String, category : String, price : int, value : int, quantity : int, type : String, discType : String, director : String, runtime : int, studio : String, subtitles : String, releasedDate : Date, filmType : String, imageUrl : String) : void
+ updateDVD(id : int, title : String, category : String, price : int, value : int, quantity : int, type : String, discType : String, director : String, runtime : int, studio : String, subtitles : String, releasedDate : Date, filmType : String) : void
+ deleteDVD(id : int) : void
+ getAllDVD() : List
ViewCartController
+ checkAvailabilityOfProduct() : void
+ getCartSubtotal() : int
BaseController
+ checkMediaInCart(media : Media) : CartMedia
+ getListCartMedia() : List
LoginController
+ login(email : String, password : String) : int
+ authenticateUser(email : String, encryptedPassword : String) : User
PlaceRushOrderController
+ validatePlaceRushOrderData(deliveryData : Shipment) : void
PlaceOrderController
+ placeOrder() : void
+ createOrder() : Order
+ createInvoice(order : Order) : Invoice
+ processDeliveryInfo(info : HashMap) : void
+ validateDeliveryInfo(info : HashMap<String,String>) : void
+ validatePhoneNumber(phoneNumber : String) : boolean
+ validateContainLetterAndNoEmpty(name : String) : boolean
+ calculateShippingFee(amount : int) : int
+ getProductAvailablePlaceRush(order : Order) : Media
+ validateAddressPlaceRushOrder(province : String, address : String) : boolean
+ validateMediaPlaceRushorder() : boolean

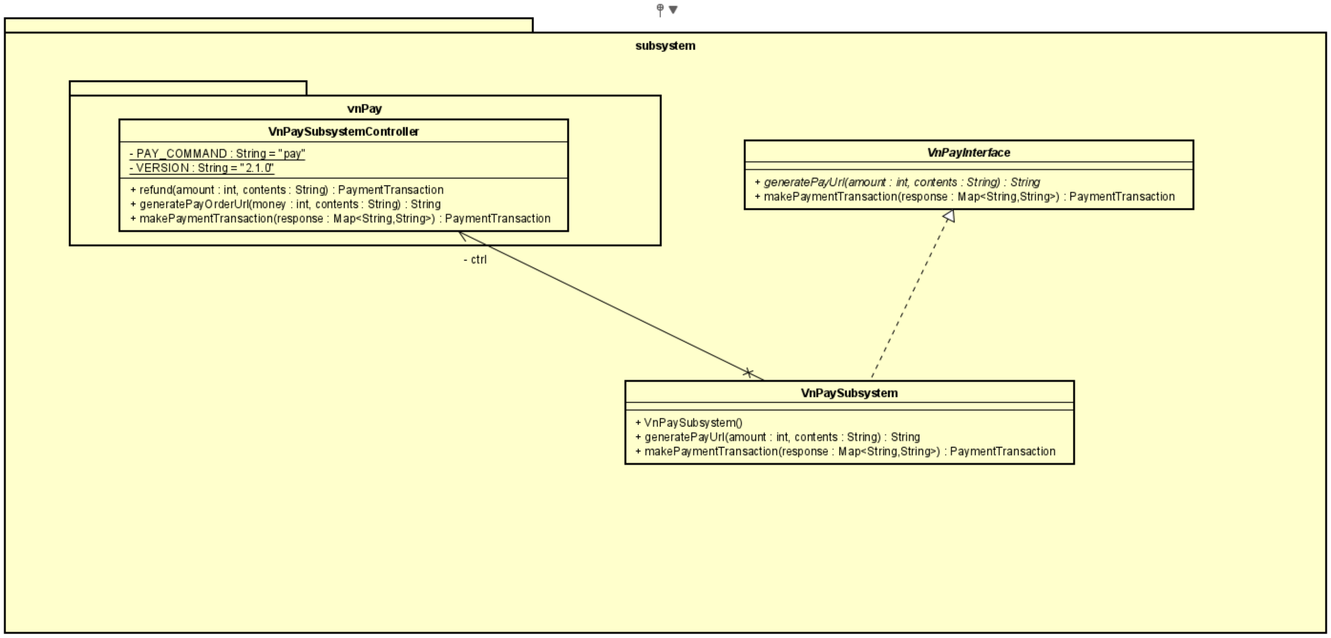
5.2.3 Class Diagram for Package Common


common
exception
ProcessingException
+ ProcessingException()
AnonymousTransactionException
+ AnonymousTransactionException()
SendToBankException
+ SendToBankException()
RejectedTransactionException
+ RejectedTransactionException()
ProcessInvoiceException
- serialVersionUID : long = 1091337136123906298L
+ ProcessInvoiceException(message : String)
+ ProcessInvoiceException()
PlaceOrderException
- serialVersionUID : long = 1091337136123906298L
+ PlaceOrderException()
+ PlaceOrderException(message : String)
PaymentException
+ PaymentException(message : String)
ViewCartException
- serialVersionUID : long = 1091337136123906298L
+ ViewCartException()
+ ViewCartException(message : String)
FailLoginDueToBannedException
+ FailLoginDueToBannedException()
TransactionFailedException
+ TransactionFailedException()
UnrecognizedException
+ UnrecognizedException()
AimsException
+ AimsException()
+ AimsException(message : String)
InvalidDeliveryInfoException
- serialVersionUID : long = 1091337136123906298L
+ InvalidDeliveryInfoException(message : String)
+ InvalidDeliveryInfoException()
TransactionReverseException
+ TransactionReverseException()
FailLoginException
+ FailLoginException()
TransactionNotDoneException
+ TransactionNotDoneException()
MediaUpdateException
- serialVersionUID : long = 1091337136123906298L
+ MediaUpdateException(message : String)
+ MediaUpdateException()
MediaNotAvailableException
- serialVersionUID : long = 1091337136123906298L
+ MediaNotAvailableException(message : String)
+ MediaNotAvailableException()

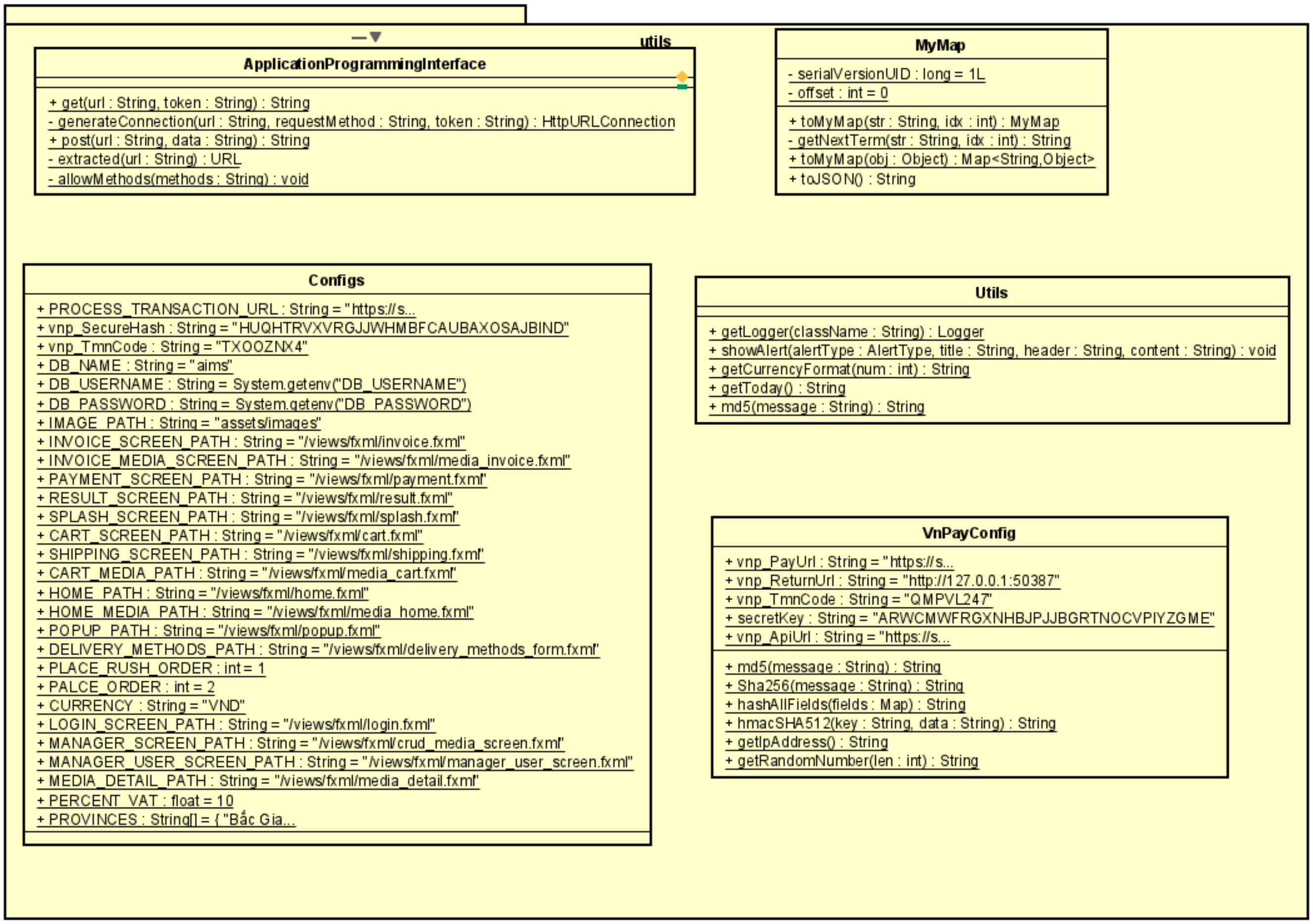
5.2.4 Class Diagram for Package Entity



5.2.5 Class Diagram for Package Subsystem

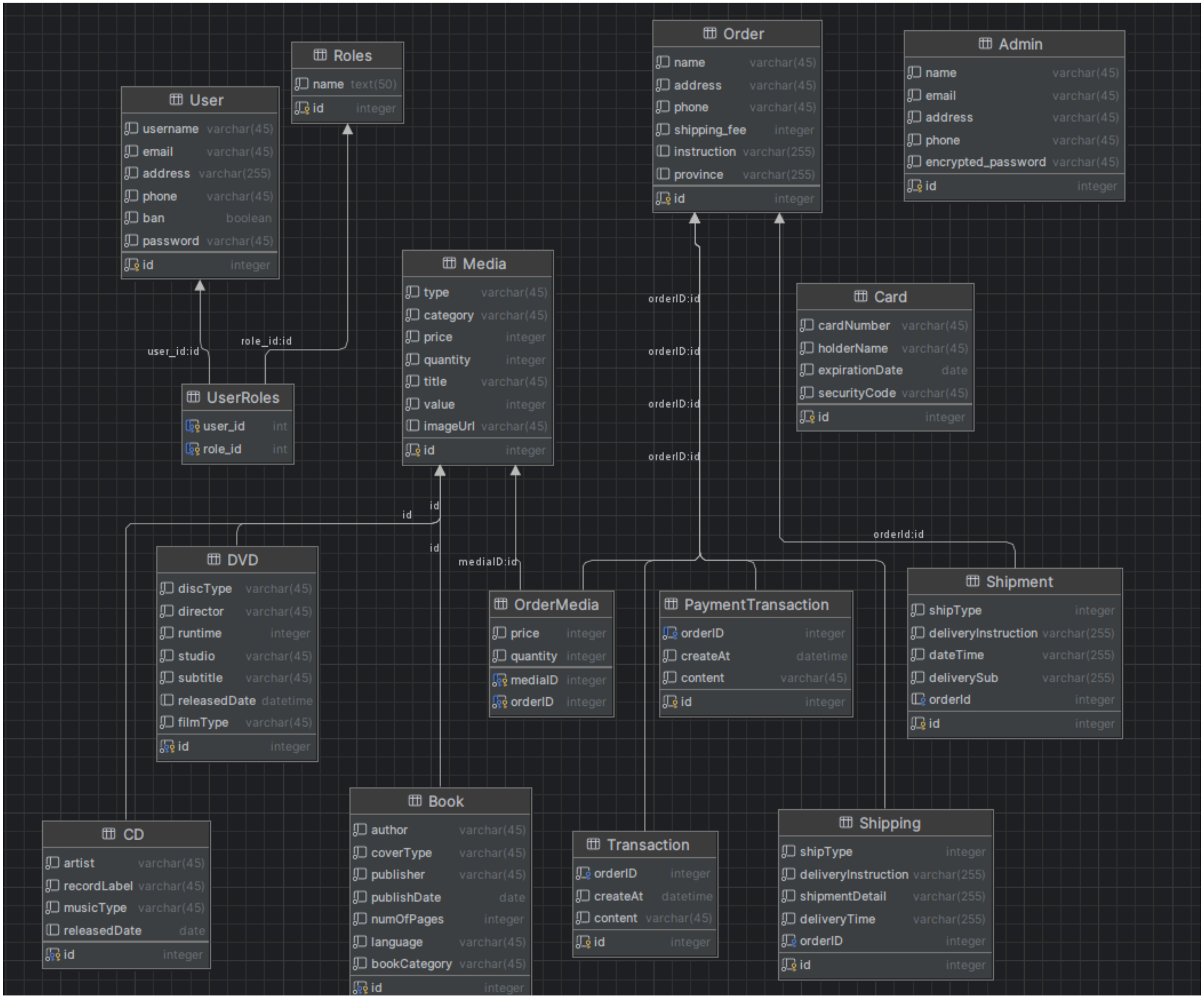

subsystem
vnPay
VnPaySubsystemController
- PAY_COMMAND : String = "pay"
- VERSION : String = "2.1.0"
+ refund(amount : int, contents : String) : PaymentTransaction
+ generatePayOrderUrl(money : int, contents : String) : String
+ makePaymentTransaction(response : Map<String,String>) : PaymentTransaction
VnPayInterface
+ generatePayUrl(amount : int, contents : String) : String
+ makePaymentTransaction(response : Map<String,String>) : PaymentTransaction
- ctrl
VnPaySubsystem
+ VnPaySubsystem()
+ generatePayUrl(amount : int, contents : String) : String
+ makePaymentTransaction(response : Map<String,String>) : PaymentTransaction

5.2.6 Class Diagram for Package Utils

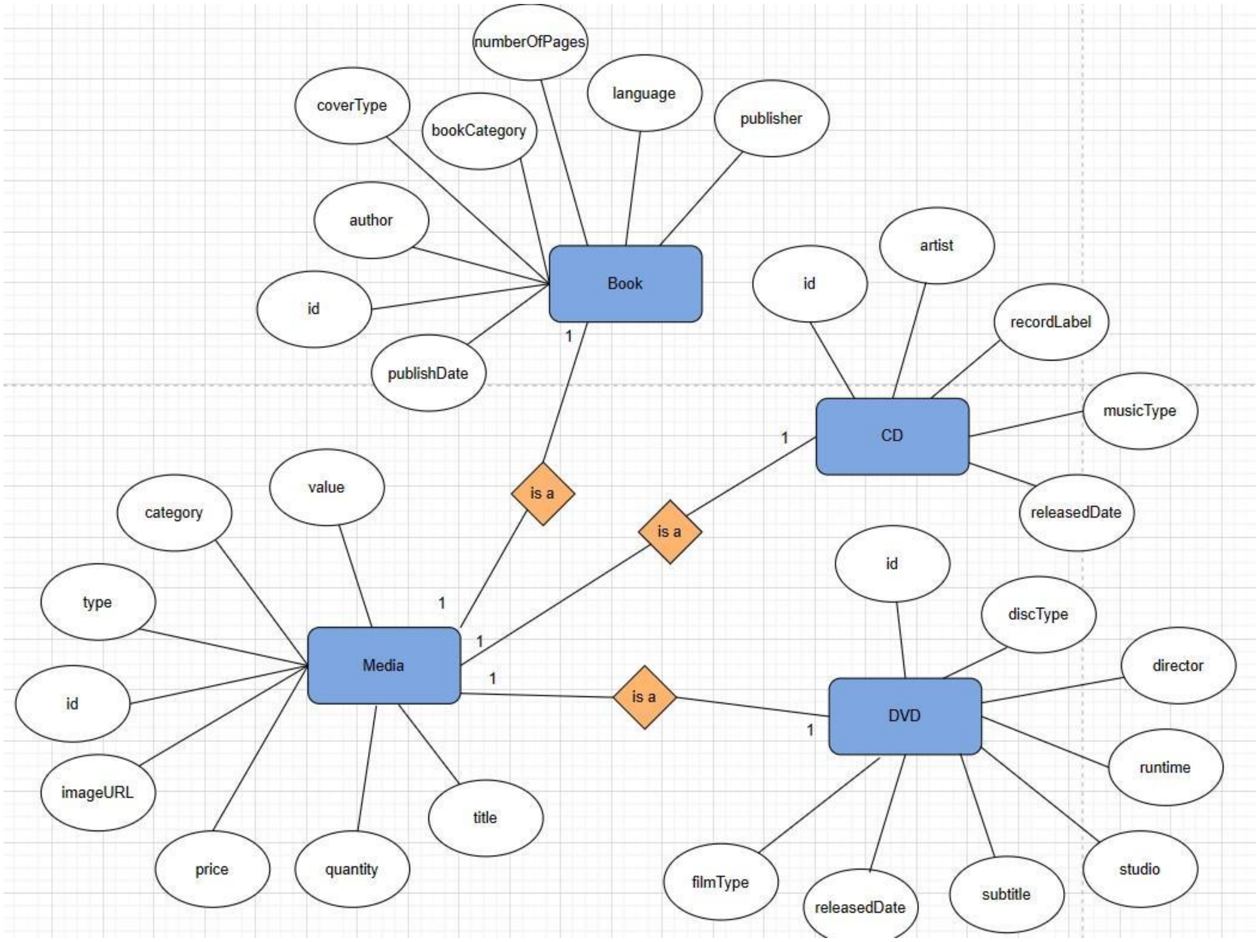

utils
ApplicationProgrammingInterface
+ get(url : String, token : String) : String
- generateConnection(url : String, requestMethod : String, token : String) : HttpURLConnection
+ post(url : String, data : String) : String
- extracted(url : String) : URL
- allowMethods(methods : String) : void
MyMap
- serialVersionUID : long = 1L
- offset : int = 0
+ toMyMap(str : String, idx : int) : MyMap
- getNextTerm(str : String, idx : int) : String
+ toMyMap(obj : Object) : Map<String,Object>
+ toJSON() : String
Configs
+ PROCESS_TRANSACTION_URL : String = "https://s...
+ vnp_SecureHash : String = "HUQHTRVXVRGJJWHMBFCAUBAXOSAJBIND"
+ vnp_TmnCode : String = "TXOOZNX4"
+ DB_NAME : String = "aims"
+ DB_USERNAME : String = System.getenv("DB_USERNAME")
+ DB_PASSWORD : String = System.getenv("DB_PASSWORD")
+ IMAGE_PATH : String = "assets/images"
+ INVOICE_SCREEN_PATH : String = "/views/fxml/invoice.fxml"
+ INVOICE_MEDIA_SCREEN_PATH : String = "/views/fxml/media_invoice.fxml"
+ PAYMENT_SCREEN_PATH : String = "/views/fxml/payment.fxml"
+ RESULT_SCREEN_PATH : String = "/views/fxml/result.fxml"
+ SPLASH_SCREEN_PATH : String = "/views/fxml/splash.fxml"
+ CART_SCREEN_PATH : String = "/views/fxml/cart.fxml"
+ SHIPPING_SCREEN_PATH : String = "/views/fxml/shipping.fxml"
+ CART_MEDIA_PATH : String = "/views/fxml/media_cart.fxml"
+ HOME_PATH : String = "/views/fxml/home.fxml"
+ HOME_MEDIA_PATH : String = "/views/fxml/media_home.fxml"
+ POPUP_PATH : String = "/views/fxml/popup.fxml"
+ DELIVERY_METHODS_PATH : String = "/views/fxml/delivery_methods_form.fxml"
+ PLACE_RUSH_ORDER : int = 1
+ PALCE_ORDER : int = 2
+ CURRENCY : String = "VND"
+ LOGIN_SCREEN_PATH : String = "/views/fxml/login.fxml"
+ MANAGER_SCREEN_PATH : String = "/views/fxml/crud_media_screen.fxml"
+ MANAGER_USER_SCREEN_PATH : String = "/views/fxml/manager_user_screen.fxml"
+ MEDIA_DETAIL_PATH : String = "/views/fxml/media_detail.fxml"
+ PERCENT_VAT : float = 10
+ PROVINCES : String[] = { "Bắc Gia...
Utils
+ getLogger(className : String) : Logger
+ showAlert(alertType : AlertType, title : String, header : String, content : String) : void
+ getCurrencyFormat(num : int) : String
+ getToday() : String
+ md5(message : String) : String
VnPayConfig
+ vnp_PayUrl : String = "https://s...
+ vnp_ReturnUrl : String = "http://127.0.0.1:50387"
+ vnp_TmnCode : String = "QMPVL247"
+ secretKey : String = "ARWCMWFRGXNHBJPJJBGRTNOCVPIYZGME"
+ vnp_ApiUrl : String = "https://s...
+ md5(message : String) : String
+ Sha256(message : String) : String
+ hashAllFields(fields : Map) : String
+ hmacSHA512(key : String, data : String) : String
+ getIpAddress() : String
+ getRandomNumber(len : int) : String

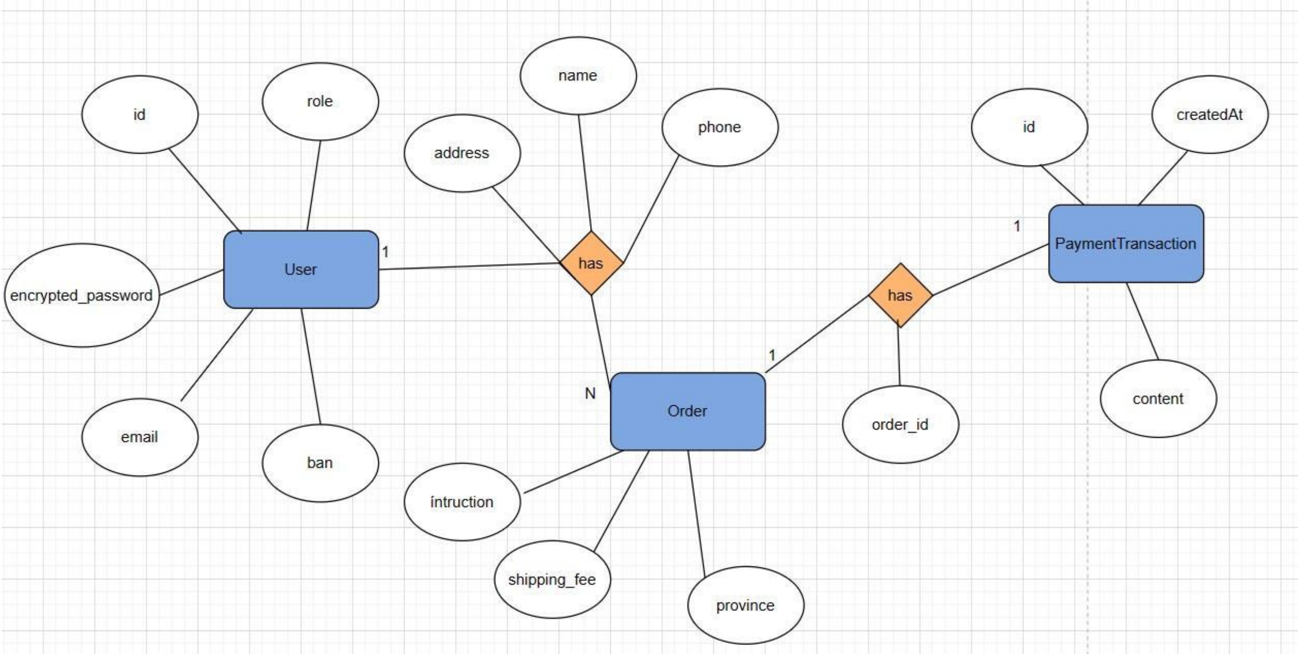
6. Data Modeling

6.1 Database Schema Diagram:

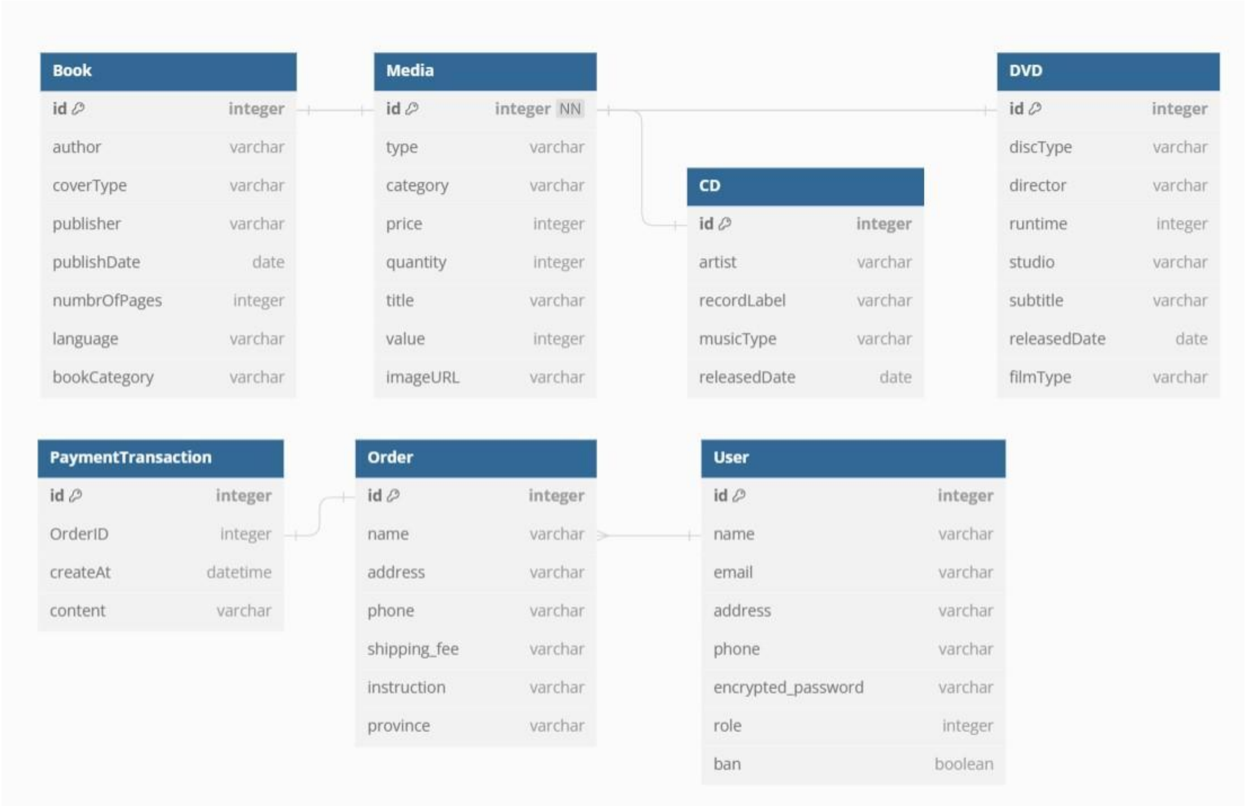

Roles
name text(50)
id integer
User
username varchar(45)
email varchar(45)
address varchar(255)
phone varchar(45)
ban boolean
password varchar(45)
id integer
Order
name varchar(45)
address varchar(45)
phone varchar(45)
shipping_fee integer
instruction varchar(255)
province varchar(255)
id integer
Admin
name varchar(45)
email varchar(45)
address varchar(45)
phone varchar(45)
encrypted_password varchar(45)
id integer
Media
type varchar(45)
category varchar(45)
price integer
quantity integer
title varchar(45)
value integer
imageUrl varchar(45)
id integer
Card
cardNumber varchar(45)
holderName varchar(45)
expirationDate date
securityCode varchar(45)
id integer
user_id:id
role_id:id
UserRoles
user_id int
role_id int
orderID:id
orderID:id
orderID:id
orderID:id
id
id
id
medialD:id
orderId:id
DVD
discType varchar(45)
director varchar(45)
runtime integer
studio varchar(45)
subtitle varchar(45)
releasedDate datetime
filmType varchar(45)
id integer
OrderMedia
price integer
quantity integer
medialD integer
orderID integer
PaymentTransaction
orderID integer
createAt datetime
content varchar(45)
id integer
Shipment
shipType integer
deliveryInstruction varchar(255)
dateTime varchar(255)
deliverySub varchar(255)
orderId integer
id integer
Book
author varchar(45)
coverType varchar(45)
publisher varchar(45)
publishDate date
numOfPages integer
language varchar(45)
bookCategory varchar(45)
id integer
CD
artist varchar(45)
recordLabel varchar(45)
musicType varchar(45)
releasedDate date
id integer
Transaction
orderID integer
createAt datetime
content varchar(45)
id integer
Shipping
shipType integer
deliveryInstruction varchar(255)
shipmentDetail varchar(255)
deliveryTime varchar(255)
orderID integer
id integer

6.1. Conceptual Data Model


numberOfPages
coverType
bookCategory
language
publisher
author
id
publishDate
Book
1
id
artist
recordLabel
CD
1
musicType
releasedDate
value
category
is a
is a
type
1
1
Media
1
id
discType
director
is a
DVD
1
id
imageURL
runtime
title
price
quantity
filmType
releasedDate
subtitle
studio


name
id
role
address
phone
id
createdAt
User
1
has
1
PaymentTransaction
encrypted_password
has
1
N
Order
order_id
content
email
ban
intruction
shipping_fee
province

6.2. Logical Data Model



Book	
id	integer
author	varchar
coverType	varchar
publisher	varchar
publishDate	date
numbrOfPages	integer
language	varchar
bookCategory	varchar

Media	
id	integer NN
type	varchar
category	varchar
price	integer
quantity	integer
title	varchar
value	integer
imageURL	varchar

CD	
id	integer
artist	varchar
recordLabel	varchar
musicType	varchar
releasedDate	date

DVD	
id	integer
discType	varchar
director	varchar
runtime	integer
studio	varchar
subtitle	varchar
releasedDate	date
filmType	varchar

PaymentTransaction	
id	integer
OrderID	integer
createAt	datetime
content	varchar

Order	
id	integer
name	varchar
address	varchar
phone	varchar
shipping_fee	varchar
instruction	varchar
province	varchar

User	
id	integer
name	varchar
email	varchar
address	varchar
phone	varchar
encrypted_password	varchar
role	integer
ban	boolean

6.3. Physical Data Model

Media

STT	PK	FK	Trường	Kiểu dữ liệu	Bắt buộc	Mô tả
1	*		id	Integer	Có	ID, auto increment
2			type	Varchar(45)	Có	Loại sản phẩm
3			category	Varchar(45)	Có	Loại nội dung sản phẩm
4			price	Integer	Có	Giá sản phẩm
5			quantity	Integer	Có	Số lượng sản phẩm
6			title	Varchar(45)	Có	Tên sản phẩm
7			value	Integer	Có	

8			imageURL	Varchar(45)	Có	Đường dẫn hình ảnh sản phẩm

Book

STT	PK	FK	Trường	Kiểu dữ liệu	Bắt buộc	Mô tả
1	*		id	Integer	Có	ID, auto increment
2			author	Varchar(45)	Có	Tên tác giả
3			coverType	Varchar(45)	Có	Kiểu bìa sách
4			publisher	Varchar(45)	Có	Tên nhà xuất bản
5			publishDate	Date	Có	Ngày xuất bản
6			numberOfPages	Integer	Có	Số trang
7			languge	Varchar(45)	Có	Ngôn ngữ
8			bookCategory	Varchar(45)	Có	Thể loại nội dung

CD

STT	PK	FK	Trường	Kiểu dữ liệu	Bắt buộc	Mô tả
1	*		id	Integer	Có	ID, auto increment
2			artist	Varchar(45)	Có	Tên nghệ sĩ
3			recordLabel	Varchar(45)	Có	Tên hãng sản xuất
4			musicType	Varchar(45)	Có	Thể loại nhạc
5			releasedDate	Date	Có	Ngày phát hành

DVD

STT	PK	FK	Trường	Kiểu dữ liệu	Bắt buộc	Mô tả
1	*		id	Integer	Có	ID, auto increment
2			discType	Varchar(45)	Có	Kiểu đĩa
3			director	Varchar(45)	Có	Đạo diễn
4			runtime	Integer	Có	Thời lượng
5			studio	Varchar(45)	Có	Hãng sản xuất
6			subtitle	Varchar(45)	Có	Phụ đề
7			releasedDate	Date	Có	Ngày phát hành
8			filmType	Varchar(45)	Có	Thể loại nội dung

PaymentTransaction

STT	PK	FK	Trường	Kiểu dữ liệu	Bắt buộc	Mô tả
1	*		id	Integer	Có	ID, auto increment
2		*	orderID	Integer	Có	ID đơn hàng
3			createAt	DateTime	Có	Thời gian giao dịch
4			content	Varchar(45)	Có	Nội dung

Order

STT	PK	FK	Trường	Kiểu dữ liệu	Bắt buộc	Mô tả
1	*		id	Integer	Có	ID, auto increment
2			name	Varchar(45)	Có	Tên người đặt
3			address	Varchar(45)	Có	Địa chỉ nhận hàng
4			phone	Varchar(45)	Có	Số điện thoại người đặt
5			shipping_fee	Integer	Có	Phí vận chuyển
6			instruction	Varchar(255)	Có	Yêu cầu
7			province	Varchar(255)	Có	Tỉnh thành

User

STT	PK	FK	Trường	Kiểu dữ liệu	Bắt buộc	Mô tả
1	*		id	Integer	Có	ID, auto increment
2			name	Varchar(45)	Có	Tên người dùng
3			email	Varchar(45)		Email người dùng
4			address	Varchar(45)	Có	Địa chỉ người dùng
5			phone	Varchar(45)	Có	Số điện thoại người dùng
6			encrypted_password	Varchar(45)	Có	Mật khẩu mã hóa người dùng
7			role	Integer	Có	Vai trò
8			ban	Boolean	Có	Tình trạng ban