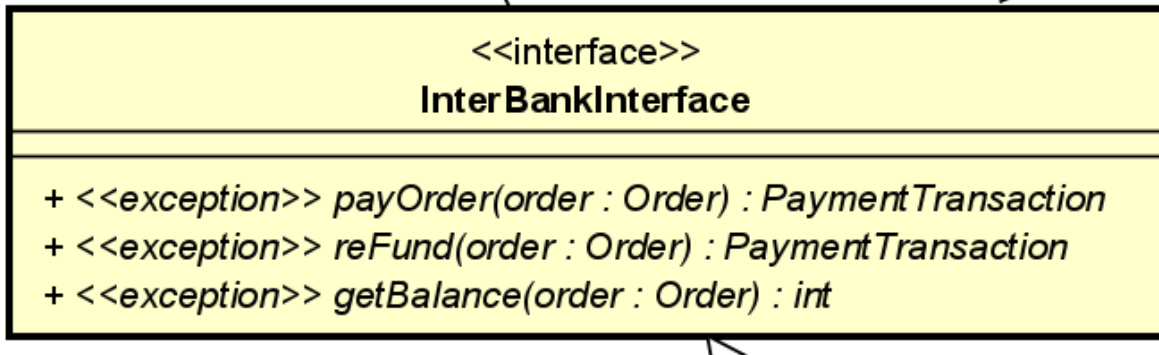


Class Diagram Design

1. Class “InterbankInterface”



Attribute

- None

Operation

#	Name	Return type	Description (purpose)
1	payOrder	PaymentTransaction	Pay order, and then return the payment transaction
2	refund	PaymentTransaction	Refund, and then return the payment transaction
3	getBalance	int	Get the difference between the money that a country receives from exports and the money that it spends on imports on this order

Parameter:

- Order: the order inform includes shipping and cart information.

Exception:

- PaymentException - if responded with a pre-defined error code
- UnrecognizedException - if responded with an unknown error code
or something goes wrong

Method

- None

State

- None

2. Class “PaymentController”

<<control>> PaymentController	
- card : Card	
- interbank : InterbankInterface	
+ payOrder(amount : int, contents : String, cardNumber : String, cardHolderName : String, expirationDate : String, securityCode : String) : Map<String,String>	
- getExpirationDate(date : String) : String	

Attribute

#	Name	Data type	Default value	Description
1	card	Card	NULL	Represent the card used for payment
2	interbank	InterbankInterface	NULL	Represent the Interbank subsystem

Operation

#	Name	Return type	Description (purpose)
1	payOrder	Map<String,String>	Pay order, and then return the result with a message

Parameter:

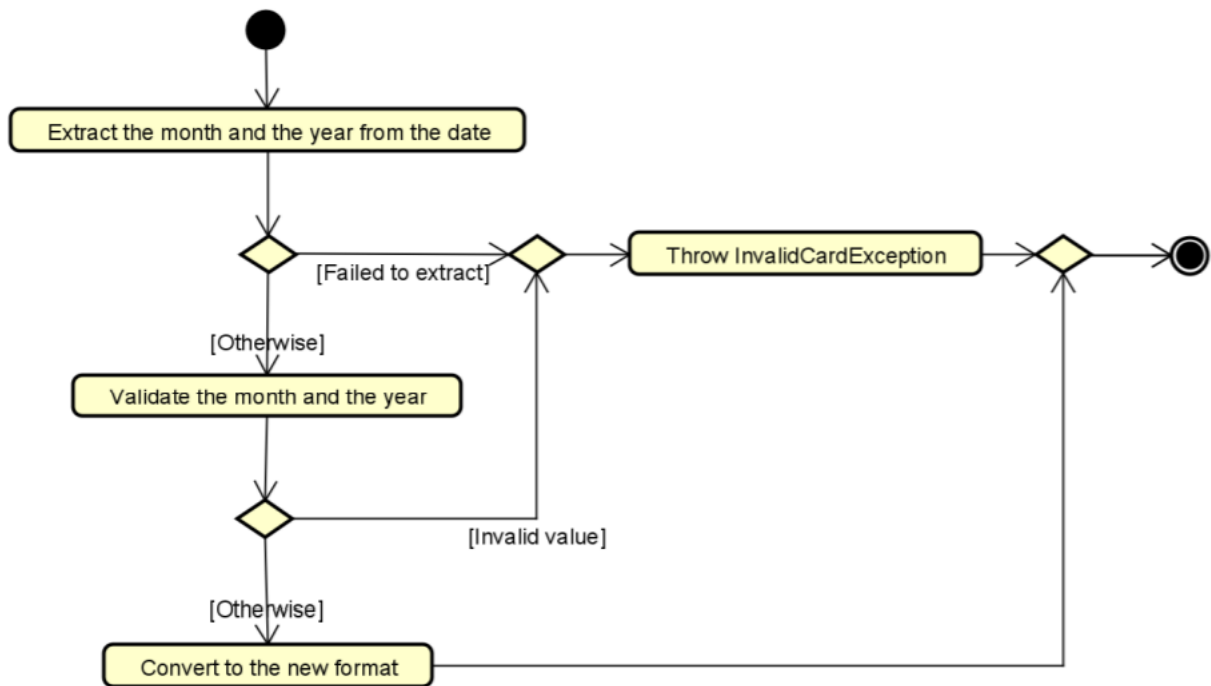
- amount - the amount to pay
- contents - the transaction contents
- cardNumber - the card number
- cardHolderName - the card holder name
- expirationDate - the expiration date in the format "mm/yy"
- securityCode - the cvv/cvc code of the credit card

Exception:

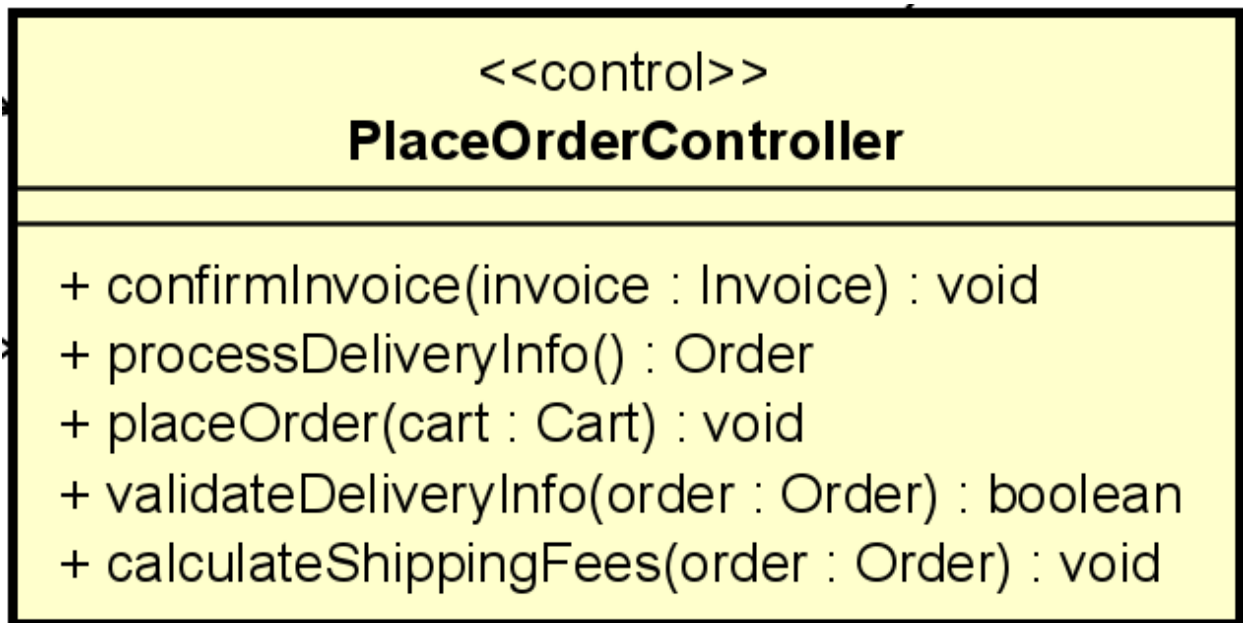
- None

Method

getExpirationDate: Given the String "date" representing the expiration date in the format "mm/yy", this method convert it into the required format "mmyy."



3. Class “PlaceOrderController”



Attribute

- None

Operation

#	Name	Return type	Description (purpose)
---	------	-------------	-----------------------

1	placeOrder	void	Place an order, receiving the entity Cart
2	processDeliveryInfor	Order	Represent DeliveryForm, store the information of User, Utils, and Cart and then export to an order
3	validateDeliveryInfo	boolean	Validating the Delivery information, receiving the entity Order
4	calculateShippingFees	void	Calculating the sum of Media and the shipping fees, receiving the entity Order
5	confimInvoice	void	Confirming the Invoice, receiving the entity Invoice

Parameter:

- None

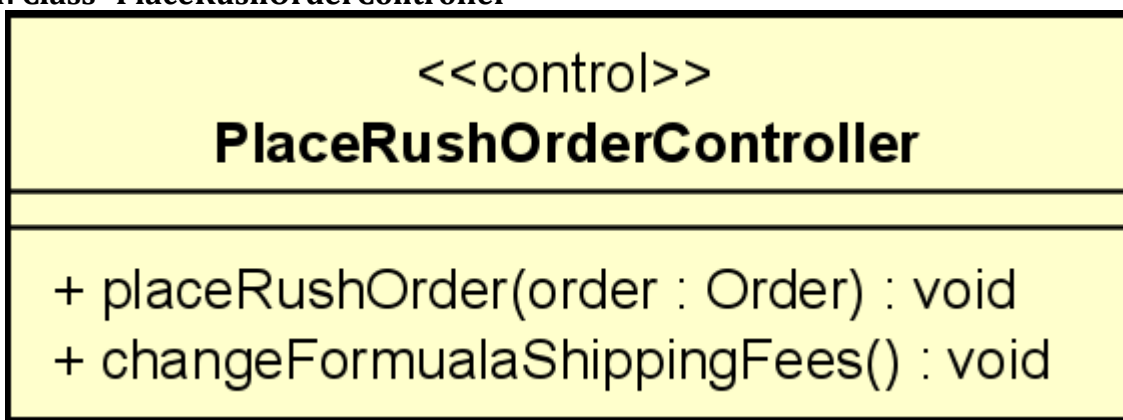
Exception:

- None

Method

- None

4. Class "PlaceRushOrderController"



Attribute

- None

Operation

#	Name	Return type	Description (purpose)
---	------	-------------	-----------------------

1	changeFormualaShippingFees	void	Calculating the shipping fees when user uses rush order
2	placeRushOrder	void	Operating the action of Place Rush Order, check the valid Media and Location of Delivery, receiving the entity Order

Parameter:

- None

Exception:

- None

Method

- None

5. Class “DeliveryForm”

<<boundary>> DeliveryForm	
+ notifyUpsupportedShippingFees() : void + requestToPlaceRushOrder(order : Order) : void + notifyInvalidDeliveryInfo() : void + chechSupportedShippingFees() : boolean + displayRushOrderScreen() : void + submitDeliveryForm() : void + notifyUnsupportedMedia() : void + Initialize(cart : Card) : void	

Attribute

- None

Operation

#	Name	Return type	Description (purpose)
1	initialize	void	Initializing the Delivery Form after clicking the button “Place Order” in class ViewCartScreen,

			receiving the entity Cart
2	submitDeliveryForm	void	Storing the information of Order when user clicks button “Confirm Delivery”
3	requestToPlaceRushOrder	void	Requesting Place Rush Order when user clicks the button “Place Rush Order” Receiving the entity Order
4	displayRushOrderScreen	void	After the sequence actions in PlaceRushOrderControlller, DeliveryForm shows the information of Rush Order
5	checkSupportedShippingFees	boolean	Checking whether Media is supported by Shipping
6	notifyInvalidDeliveryInfo	void	Notify Invalid delivery information, allowing the user to change some information
7	notifyUnsupportedMedia	void	Notify Media is not supported, allowing the user to change Media
8	notifyUpsupportedShippingFees	void	Notify Location is not supported Shipping Fees, allowing the user to change Location

Parameter:

- None

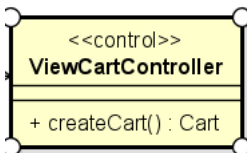
Exception:

- None

Method

- None

6. Class “ViewCartController”



Attribute

- None

Operation

#	Name	Return type	Description (purpose)
1	createCart	Cart	Creating the cart, which has the list of Media

Parameter:

- None

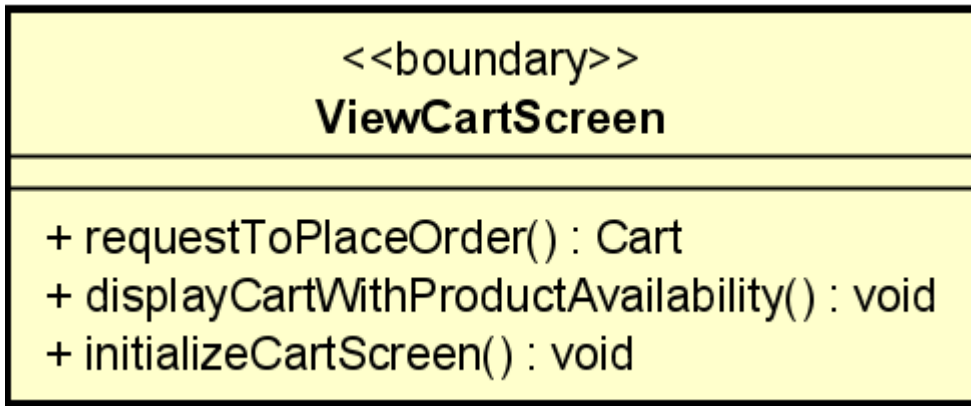
Exception:

- None

Method

- None

7. Class “ViewCartScreen”



Attribute

- None

Operation

#	Name	Return type	Description (purpose)
1	requestToPlaceOrder	Cart	Saving the information of Media when users choose in Cart Screen after clicking the button “Place Order”
2	displayCartWithProductAvailability	void	Displaying the Media that exists in the Cart

3	initializeCartScreen	void	Initializing Cart Screen
---	----------------------	------	--------------------------

Parameter:

- None

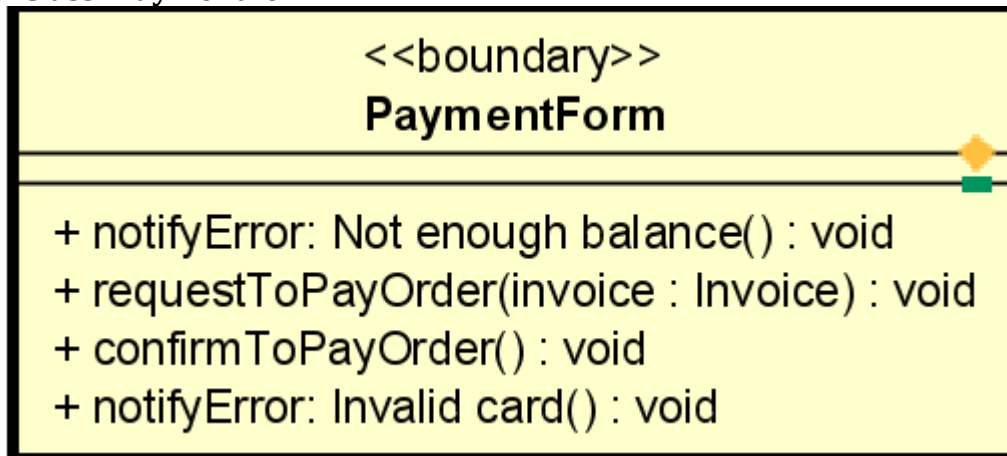
Exception:

- None

Method

- None

8. Class “PaymentForm”



Attribute

- None

Operation

#	Name	Return type	Description (purpose)
1	requestToPayOrder	void	Taking the actions of paying Order, receiving the entity Invoice
2	confirmToPayOrder	void	Saving the information of Invoice when users click the button “Confirm Payment” Receiving the entity Invoice
3	notifyError: Invalid card()	void	Notifying the Error: Invalid card

4	notifyError: Not enough balance()	void	Notifying the Error: Not enough balance
---	-----------------------------------	------	---

Parameter:

- None

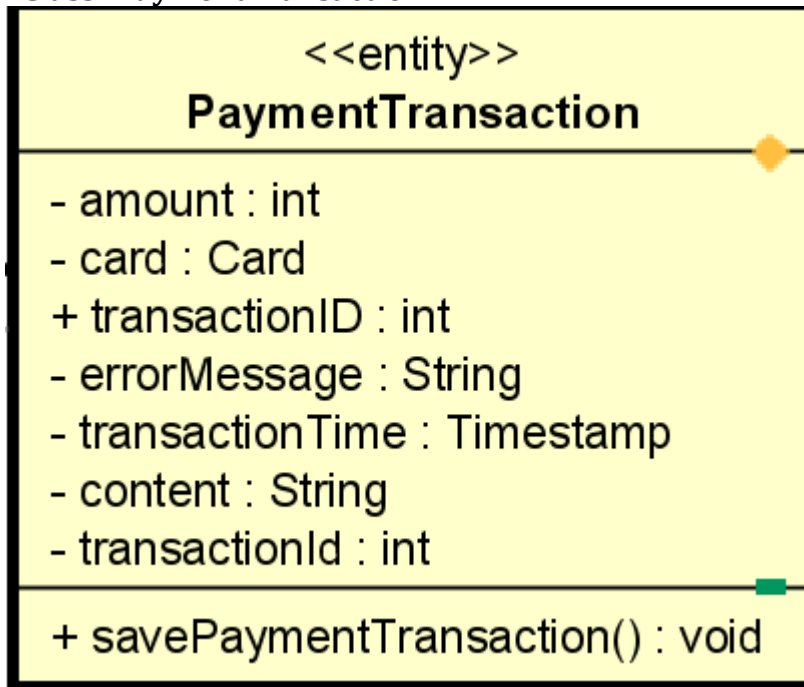
Exception:

- None

Method

- None

9. Class “PaymentTransaction”



Attribute

#	Name	Data type	Default value	Description
1	transactionID	int	NULL	Represent the Id transaction used for payment
2	card	Card	NULL	Represent the information of Card used for payment
3	amount	int	NULL	Represent the amount of the payment
4	transactionTime	Timestamp	NULL	Represent the transaction time of the payment

5	content	String	NULL	Represent the content of the payment
6	errorMessage	String	NULL	Represent the error during the transaction

Operation

#	Name	Return type	Description (purpose)
1	savePaymentTransaction	void	Saving the payment transaction

Parameter:

- None

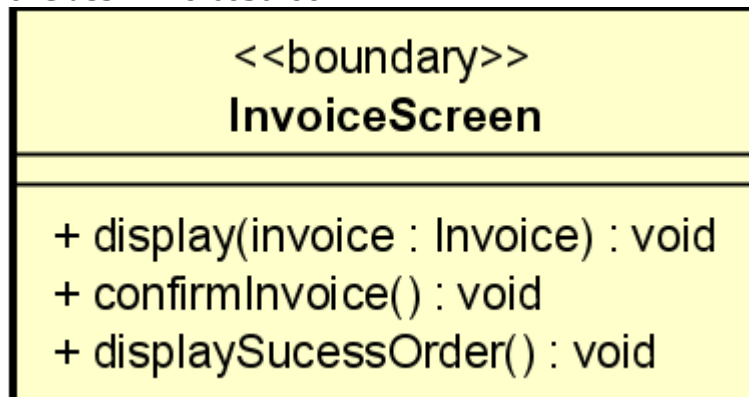
Exception:

- None

Method

- None

10. Class “InvoiceScreen”



Attribute

- None

Operation

#	Name	Return type	Description (purpose)
1	display	void	Displaying the InvoiceScreen and receiving the entity Invoice

2	confirmInvoice	void	Saving the information of Invoice when user click the button "Confirm Invoice"
3	displaySuccessOrder	void	Displaying the notification of Success Order after user completes the actions of place order

Parameter:

- None

Exception:

- None

Method

- None