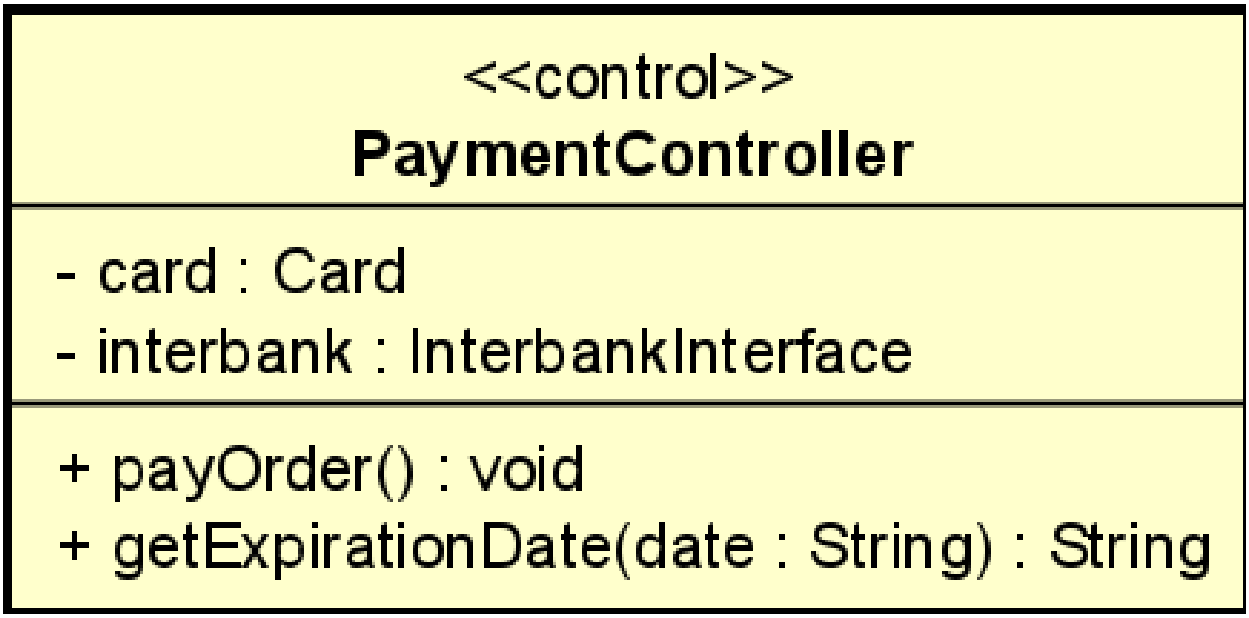
1) PaymentController



Attribute:

#	Name	Data type	Default value	Description
1	card	CreditCard	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

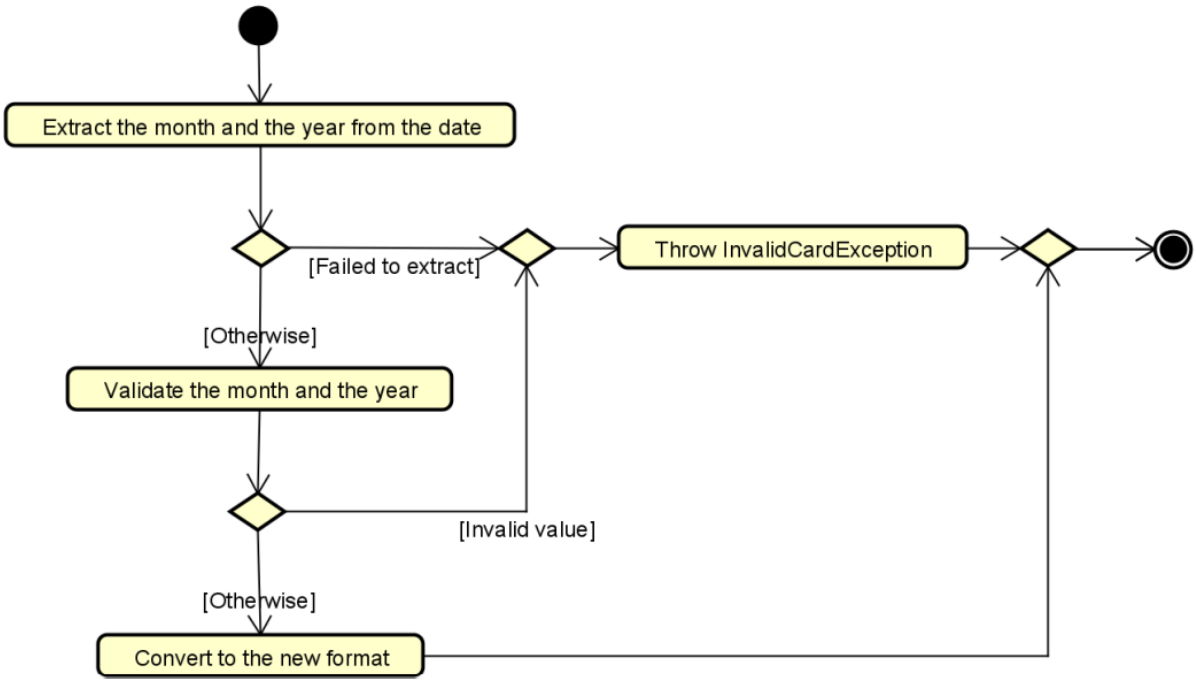
Parameter:

- 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

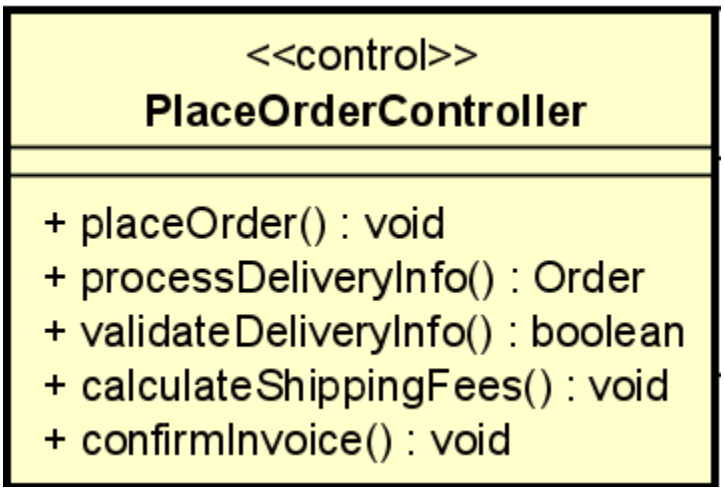
Exception: None

Method:

- getExpirationDate: Given the String “date” representing the expiration date in the format “mm/yy”, this method converts it into the required format “mmyy”. The algorithm is illustrated as follows.



2) Class “PlaceOrderController”



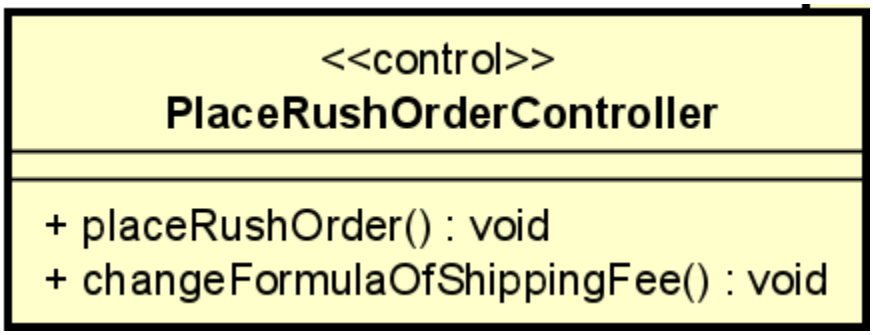
Attribute: None
Operation:

#	Name	Return type	Description (purpose)
1	placeOrder	void	Place an order, receiving the entity Cart
2	processDeliveryInfo	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	confirmInvoice	void	Confirming the Invoice, receiving the entity Invoice

Parameter: None
Exception: None
Method: None

3) Class "PlaceRushOrderController"

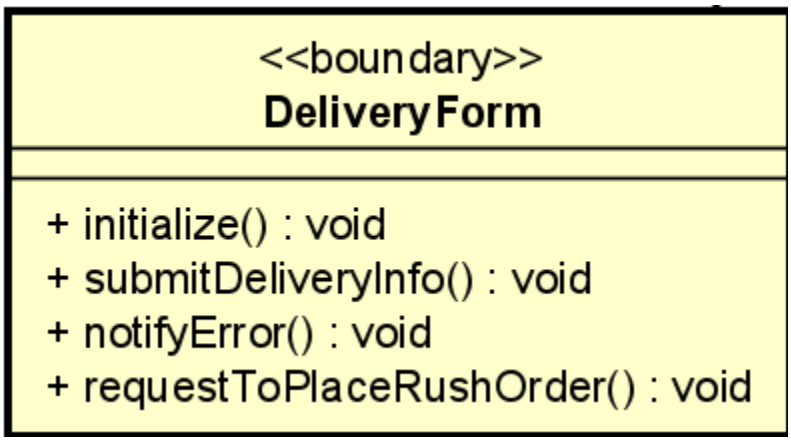


Attribute: None
Operation:

#	Name	Return Type	Description
1	placeRushOrder	void	Calculating the shipping fees when user uses rush order
2	changeFormulaOfShippingFee	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

4) Class "DeliveryForm"

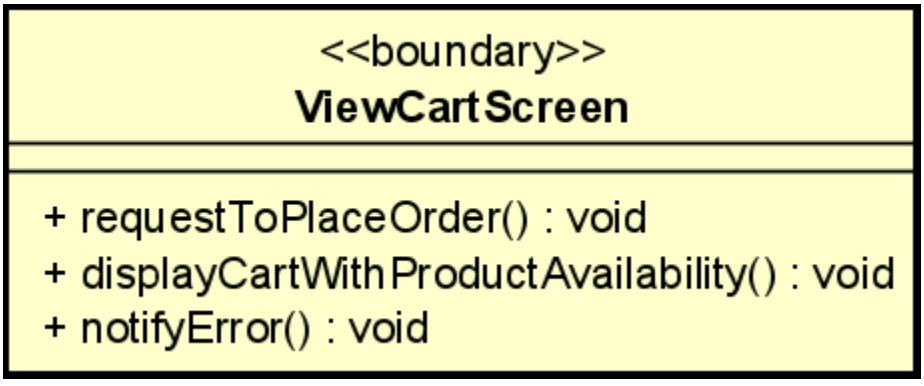


Attribute: None
Operation:

#	Name	Return Type	Description
1	initialize	void	Initializing the Delivery Form after clicking the button “Place Order” in class ViewCartScreen, receiving the entity Cart
2	submitDeliveryInfo	void	Storing the information of Order when user clicks button “Confirm Delivery”
3	notifyError	void	Showing error notification on the screen
4	requestToPlaceRushOrder	void	Requesting Place Rush Order when user clicks the button “Place Rush Order” Receiving the entity Order

Parameter: None
Exception: None
Method: None

5) Class “ViewCartScreen”



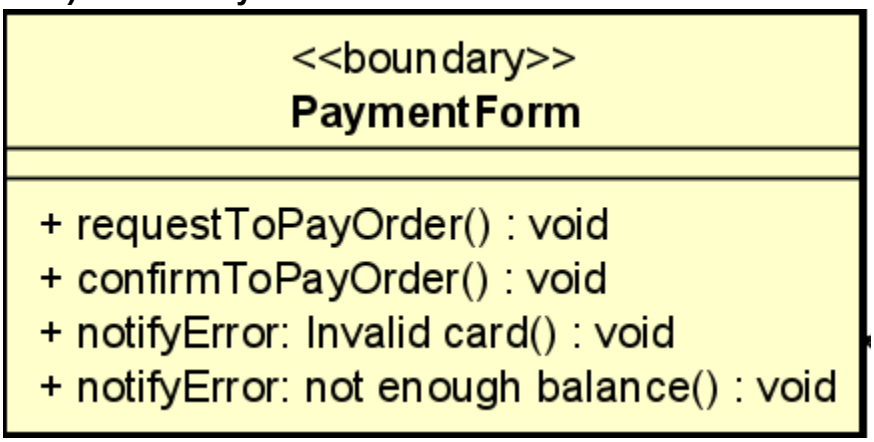
Attribute: None

Operation:

#	Name	Return type	Description
1	requestToPlaceOrder	void	Saving the information of Media when users in Cart Screen after clicking the button “Place Order”
2	displayCartWithProductAvailability	void	Display the Media that exists in the Cart
3	notifyError	void	Initializing Cart Screen

Parameter: None
Exception: None
Method: None

6) Class “PaymentForm”



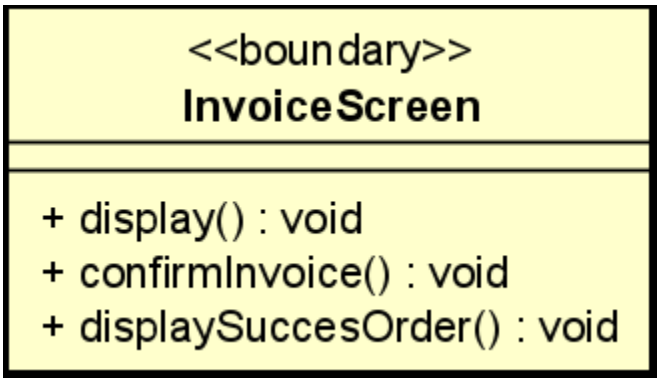
Attribute: None
Operation:

#	Name	Return Type	Description
1	requestToPayOrder	void	Taking the actions of paying Order, receiving the entity Invoice
2	confirmToPayOrder	void	Saving the information of Invoice when users clicks 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

7) Class “InvoiceScreen”



Attribute: None

Operation:

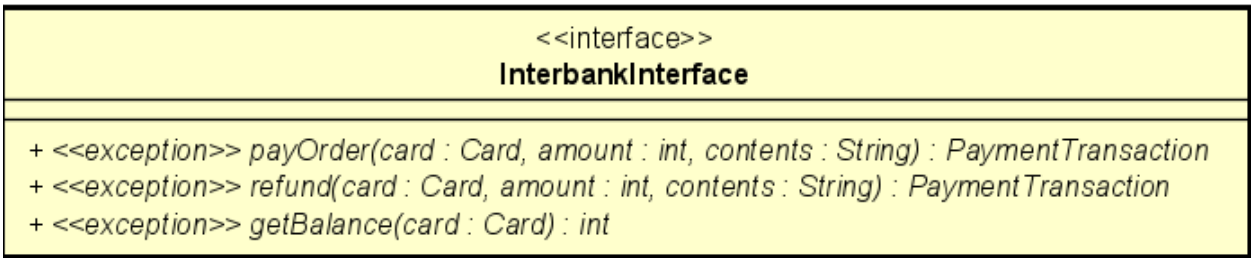
#	Name	Return type	Description
1	display	void	Displaying the InvoiceScreen and receiving the entity Invoice
2	confirmInvoice	void	Saving the information of Invoice when user clicks the button “Confirm Invoice”
3	displaySuccessOrder	void	Displaying the notification of Success Order after user completes the actions of place order

Parameter: None

Exception: None

Method: None

8) Class “InterbankInterface”



Attribute: None

Operation:

#	Name	Return Type	Description
1	payOrder	PaymentTransaction	Pay order and then return the payment transaction
2	refund	PaymentTransaction	Refund and then return the payment transaction

3	getBalance	int	Return the balance in the card

Parameter:

- card: the credit card used for payment/refund
- amount: the amount to pay/refund
- contents: the transaction contents

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