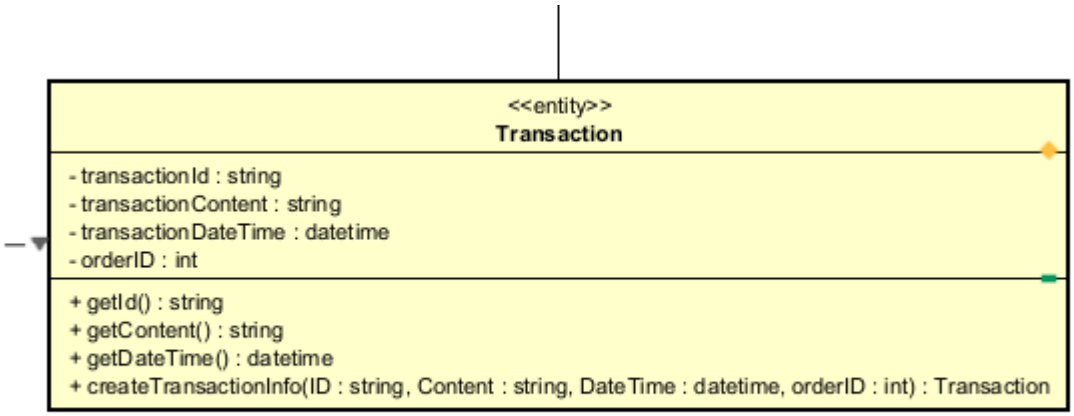
Class Design for Use Case “Pay Order”

1. Design for class “Transaction”



1.1. Attribute design example

#	Name	Data type	Default Value	Description
1	transactionId	string	Generated by system	ID of transaction
2	transactionContent	string	None	Content of transaction inputted by User
3	transactionDateTime	datetime	Generated by system	Time and Date of transaction
4	orderID	int	Generated by system	ID of Order

1.2. Operation Design example

#	Name	Return type	Description
1	createdTransactionInfo	Transaction	Create a new Transaction
2	getID	string	Getter to retrieve ID of transaction
3	getContent	string	Getter to retrieve Content of transaction
4	getDateTime	datetime	Getter to retrieve Date time of transaction

1.2.1. createdTransactionInfo method:

- Parameter:

    + ID: ID of transaction

+ Content: Content of transaction

+ DateTime: Datetime of Transaction

+ orderID: ID of order for transaction

- Exception:
    - InvalidTypeException if invalid type of parameters inputted appear.
    - DuplicatedIDException if TransactionID or OrderID is duplicated.
- How to use parameters/attributes: Assign function parameters to corresponding object attributes
- Flowchart / activity diagram / sequence diagram: None
- State: None

1.2.2. getID, getContent, getDateTime method:

- Parameter: none

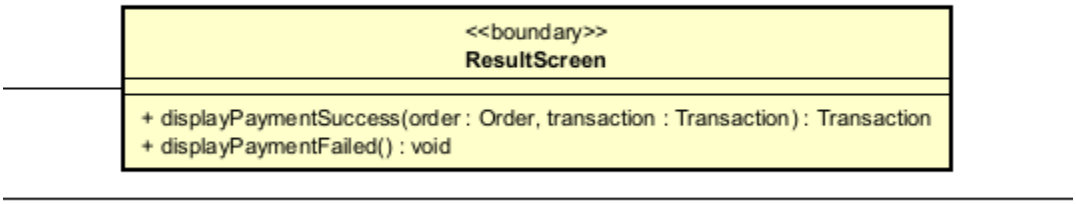
- Exception: none

- How to use parameters/attributes: Assign function parameters to corresponding object attributes

- Flowchart / activity diagram / sequence diagram: None

- State: None

2. Design for class “ResultScreen”



2.1. Attribute design example

#	Name	Data type	Default Value	Description

2.2. Operation Design example

#	Name	Return type	Description
1	displayPaymentFailed	void	Display message to announce payment failed.

2	displayPaymentSuccess	void	Display all information of transaction, order, customer

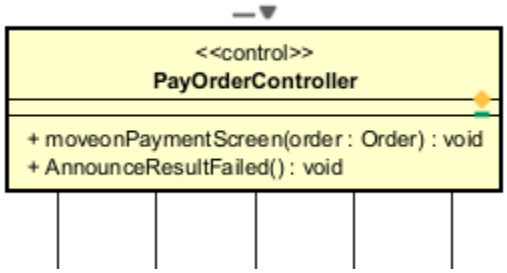
2.2.1. displayPaymentFailed method:

- Parameter: None
- Exception: None
- How to use parameters/attributes: Assign function parameters to corresponding object attributes
- Flowchart / activity diagram / sequence diagram: None
- State: None

2.2.2. displayPaymentSuccess method:

- Parameter:
    - order: Order of this payment
    - transaction: Transaction of this payment
- Exception:
    - RetrieveException if parameters return null.
- How to use parameters/attributes: Assign function parameters to corresponding object attributes
- Flowchart / activity diagram / sequence diagram: None
- State: None

3. Design for class “PayOrderController”



3.1. Attribute design example

#	Name	Data type	Default Value	Description

3.2. Operation Design example

#	Name	Return type	Description
1	moveonPaymentScreen	void	
2	AnnounceResultFailed	void	Call to PaymentFailed Screen.

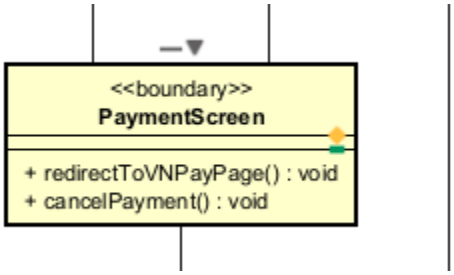
3.2.1. moveonPaymentScreen method:

- Parameter:
    - order: Order for payment
- Exception:
    - ViewException if can not access boundary class due to setting or connection.
- How to use parameters/attributes: Assign function parameters to corresponding object attributes
- Flowchart / activity diagram / sequence diagram: None
- State: None

3.2.2. AnnounceResultFailed method:

- Parameter: none
- Exception: none
- How to use parameters/attributes: Assign function parameters to corresponding object attributes
- Flowchart / activity diagram / sequence diagram: None
- State: None

4. Design for class “PaymentScreen”



4.1. Attribute design example

#	Name	Data type	Default Value	Description

4.2. Operation Design example

#	Name	Return type	Description
1	redirectToVNPayPage	void	Open to VNPay page
2	cancelPayment	void	

4.2.1. redirectToVNPayPage method:

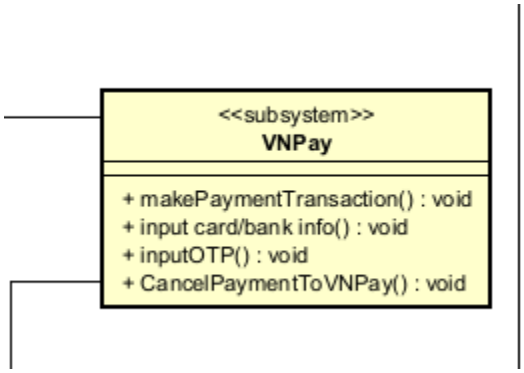
- Parameter: None
- Exception:

    - ConnectionException: can not connect to VNPay
- How to use parameters/attributes: Assign function parameters to corresponding object attributes
- Flowchart / activity diagram / sequence diagram: None
- State: None

4.2.2. cancelPayment method:

- Parameter: none
- Exception: none
- How to use parameters/attributes: Assign function parameters to corresponding object attributes
- Flowchart / activity diagram / sequence diagram: None
- State: None

5. Design for class "PaymentScreen"



5.1. Attribute design example

#	Name	Data type	Default Value	Description

5.2. Operation Design example

#	Name	Return type	Description
1	makePaymentTransaction	void	Send request to Subsystem to make a transaction
2	Input card/bank info	void	Make users input card/bank info
3	inputOTP	void	Make users input OTP to complete a transaction
4	CancelPaymentToVNPay	void	Customers cancel the payment to VNPay.

5.2.1. makePaymentTransaction method:

- Parameter: None
- Exception:
    - ConnectionException: can not connect to VNPay
- How to use parameters/attributes: Assign function parameters to corresponding object attributes
- Flowchart / activity diagram / sequence diagram: None
- State: None

5.2.2. Input card/bank info method:

- Parameter: none
- Exception: None
- How to use parameters/attributes: Assign function parameters to corresponding object attributes
- Flowchart / activity diagram / sequence diagram: None
- State: None

5.2.3. inputOTP method:

- Parameter: None
- Exception:
    - ReceiveOTPException if it takes too long to receive OTP
- How to use parameters/attributes: Assign function parameters to corresponding object attributes
- Flowchart / activity diagram / sequence diagram: None
- State: None

5.2.4. CancelPaymentToVNPay method:

- Parameter: None
- Exception:
    - ConnectionException: can not connect to VNPay
- How to use parameters/attributes: Assign function parameters to corresponding object attributes
- Flowchart / activity diagram / sequence diagram: None
- State: None