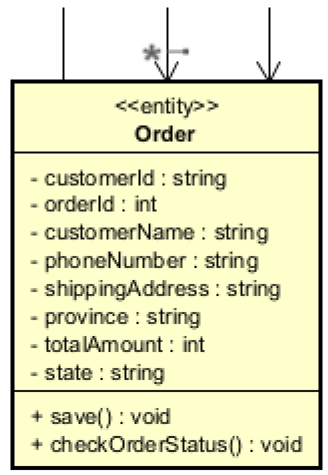
Class Design for Use Case “Cancel Order”

1. Design for class “Order”



1.1. Attribute design example

#	Name	Data type	Default Value	Description
1	customerId	string	Generated by system	ID of Customer
2	orderId	int	Generated by system	ID of Order
3	customerName	string	Input by User	Name of Customer
4	phoneNumber	string	Input by User	Phone number of Customer
5	shippingAddress	string	Input by User	Address of Customer
6	province	string	Input by User	Province
7	totalAmount	int	Calculated by taking the sum of all product prices.	
8	state	string		

1.2. Operation Design example

#	Name	Return type	Description
1	save	void	Order is saved into database
2	checkOrderStatus	void	Get information of order state. If state is not “PENDING”, user can not cancel order.

1.2.1. save method:

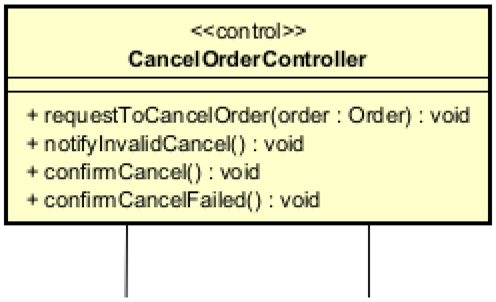
- Parameter: None

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

1.2.2. checkOrderStatus method:

- Parameter: none
- Exception: none
- How to use parameters/attributes: Assign function parameters to corresponding object attributes
- Flowchart / activity diagram / sequence diagram: None
- State:
  - "PENDING" -> call createCancelconfirmScreen method
  - "APPROVED" OR "REJECT" -> call notifyInvalidCancel method

2. Design for class "CancelOrderController"



<<control>>
CancelOrderController
+ requestToCancelOrder(order : Order) : void
+ notifyInvalidCancel() : void
+ confirmCancel() : void
+ confirmCancelFailed() : void

2.1. Attribute design example

#	Name	Data type	Default Value	Description

2.2. Operation Design example

#	Name	Return type	Description
1	requestToCancelOrder	void	Send request to cancel order
2	notifyInvalidCancel	void	If the state of order does not satisfy, order will notify Controller
3	confirmCancel	void	
4	confirmCancelFailed	void	Notify the Controller that fail to Cancel Order

2.2.1. requestToCancelOrder method:

- Parameter:
    - order: Order needs to be cancel
- Exception:
    - orderException if can not find this order.
- How to use parameters/attributes: Assign function parameters to corresponding object attributes
- Flowchart / activity diagram / sequence diagram: None
- State: None

2.2.2. notifyInvalidCancel 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

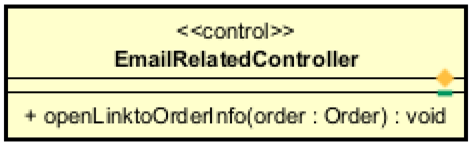
2.2.3. confirmCancel method:

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

2.2.4. confirmCancelFailed 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

3. Design for class "EmailRelatedController"



3.1. Attribute design example

#	Name	Data type	Default Value	Description

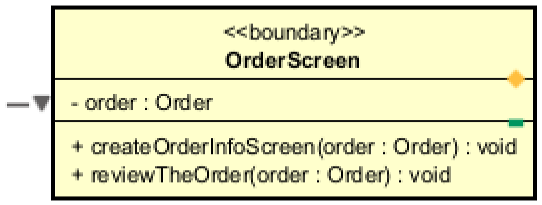
3.2. Operation Design example

#	Name	Return type	Description
1	openLinktoOrderInfo	void	

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

4. Design for class “OrderScreen”


<<boundary>>
OrderScreen
- order : Order
+ createOrderInfoScreen(order : Order) : void
+ reviewTheOrder(order : Order) : void

4.1. Attribute design example

#	Name	Data type	Default Value	Description
1	order	Order	None	Order information

4.2. Operation Design example

#	Name	Return type	Description
1	createOrderInfoScreen	void	
2	reviewTheOrder	void	

4.2.1. createOrderInfoScreen method:

- Parameter:
  - order: Order for payment
- 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

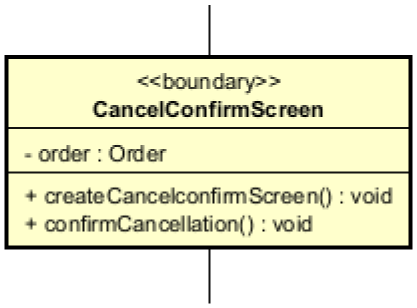
4.2.2. reviewTheOrder method:

- Parameter:
    - order: Order for payment

- 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 “CancelConfirmScreen”



<<boundary>>
CancelConfirmScreen
- order : Order
+ createCancelconfirmScreen() : void
+ confirmCancellation() : void

5.1. Attribute design example

#	Name	Data type	Default Value	Description
1	order	Order	None	Order information

5.2. Operation Design example

#	Name	Return type	Description
1	createCancelconfirmScreen	void	CancelConfirmScreen is created
2	confirmCancellation	void	User confirm cancellation via this interface

5.2.1. createCancelconfirmScreen method:

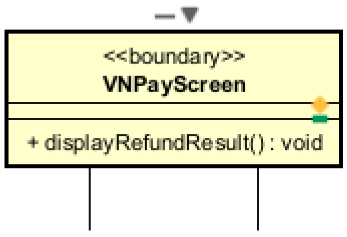
- Parameter: None
- Exception: None
- How to use parameters/attributes: Assign function parameters to corresponding object attributes

- Flowchart / activity diagram / sequence diagram: None
- State: None

5.2.2. confirmCancellation 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

6. Design for class "VNPayScreen"



6.1. Attribute design example

#	Name	Data type	Default Value	Description

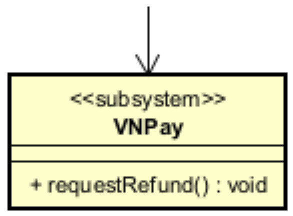
6.2. Operation Design example

#	Name	Return type	Description
1	displayRefundResult	void	Display outcome of refund result

6.2.1. displayRefundResult 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

7. Design for class "VNPay"



7.1. Attribute design example

#	Name	Data type	Default Value	Description

7.2. Operation Design example

#	Name	Return type	Description
1	requestRefund	void	Send request to Subsystem to make a refund

7.2.1. displayRefundResult 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