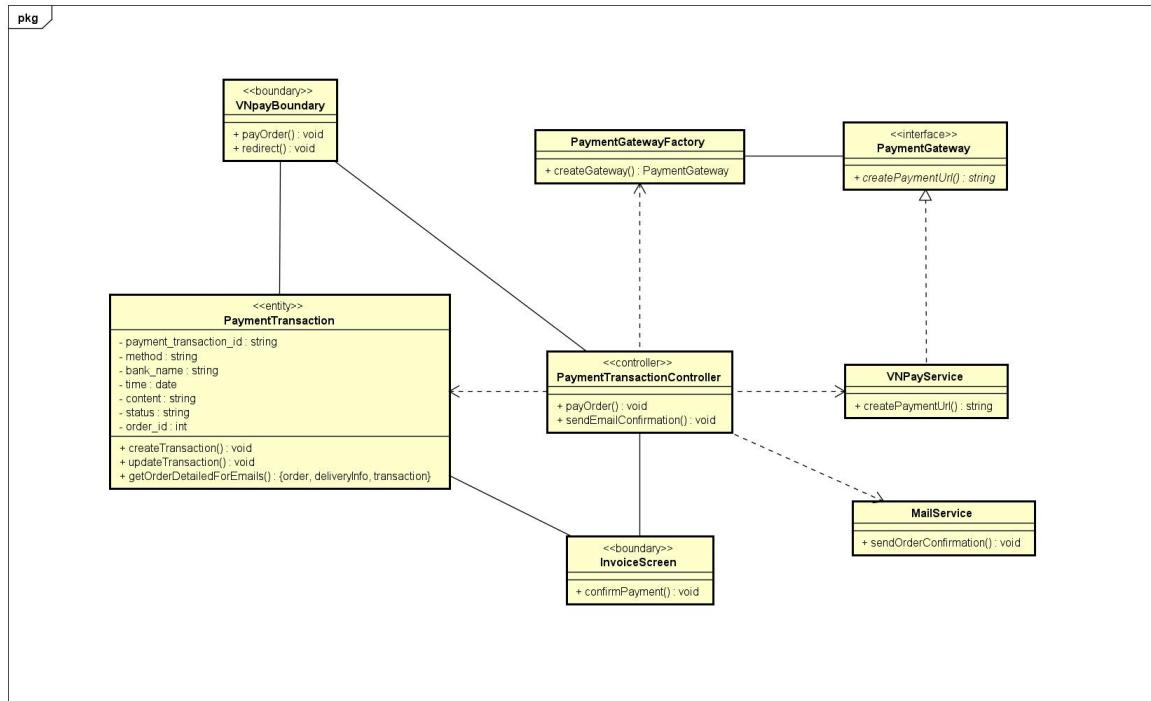


Usecase: PayOrder

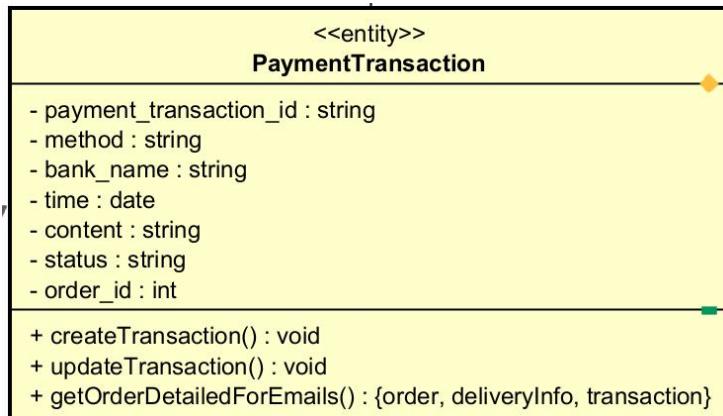
1. Class design

1.1 Pay Order Class diagram



1.2 Design Class

1.2.1 Entity: PaymentTransaction



Attribute design:

#	Name	Data Type	Default Value	Description
1	Payment_transaction_id	String		An unique Integer identifies

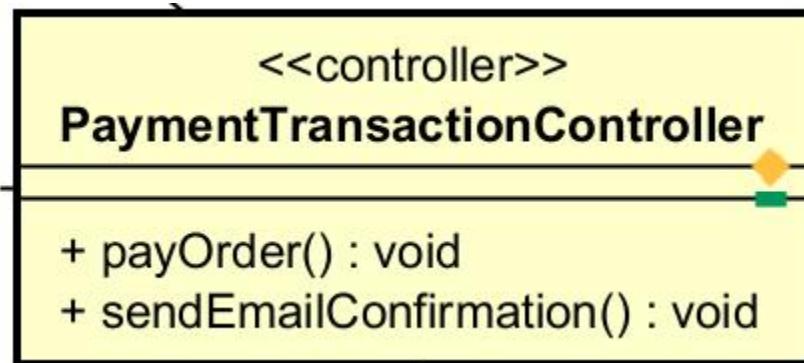
				define a transaction
2	Method	String		Payment method
3	Bank_name	String		Name of payment bank
4	Time	Date		Time pay
5	Content	String		Content payment
6	Status	String		Status of the transaction
7	Order_id	Int		An unique Integer identifies define an order

Operations Design

#	Name	Return Type	Description
1	createTransaction	void	A method that create a new information of a new transaction
2	updateTransaction	Void	A method that update The status of existed transaction
3	getOrderDetailedForEmails	{order, deliverInfo, transaction}	A method that return the information of order to sending email

1.2.2 Controller PayOrderController

UML Diagram:



The **PayOrderController** is responsible for managing the process of paying an order in an e-commerce or order management system. It acts as an intermediary between the user interface (UI) and the underlying business logic (models such as Order, Invoice, and DeliveryInfo).

It performs the following key tasks:

1. **payOrder**
2. **sendEmailConfirmation**

