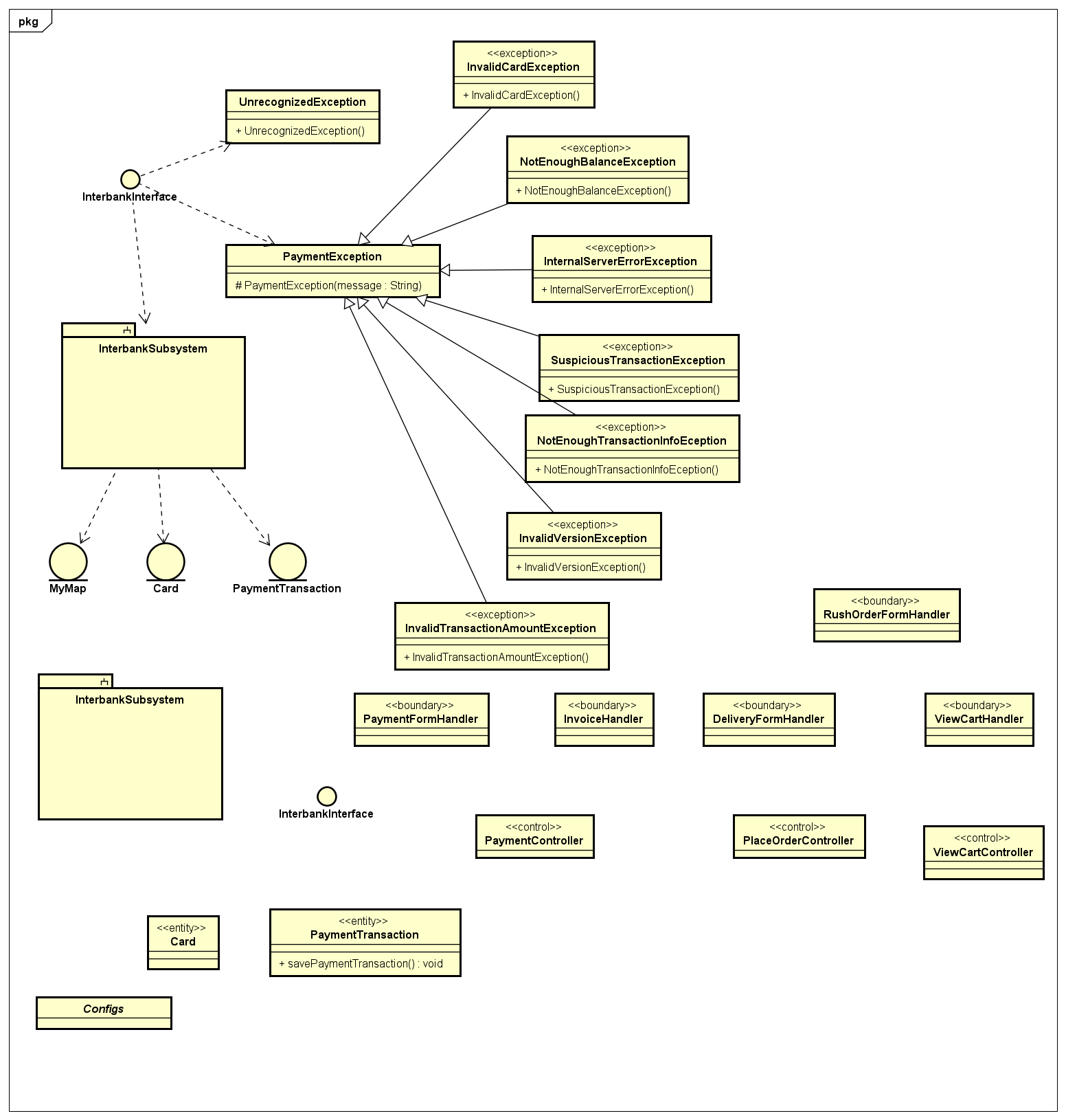
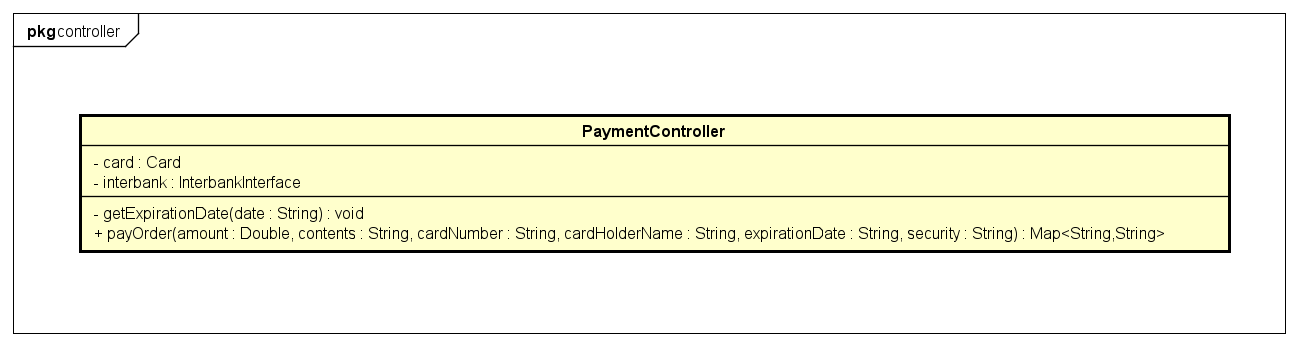
**Class Design**

1. **Define relationships between classes**



1. **Class design**
   1. Class “PaymentController”



* **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**

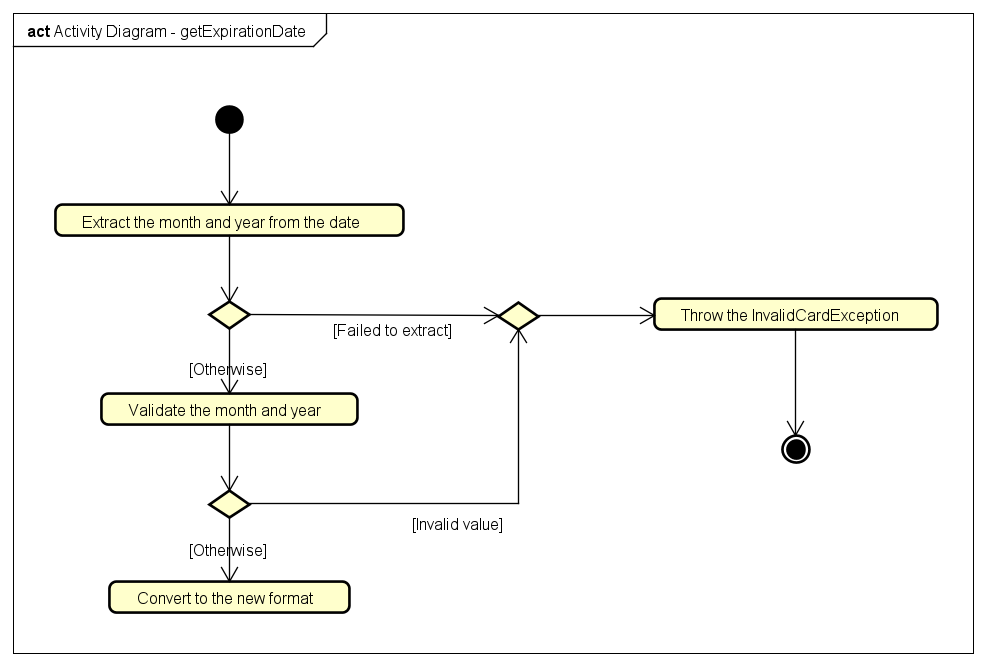
|  |  |  |  |
| --- | --- | --- | --- |
| # | Name | Return type | Description |
| 1 | payOrder | Map<String,String> | Pay order and return the result with a message |

Parameter:

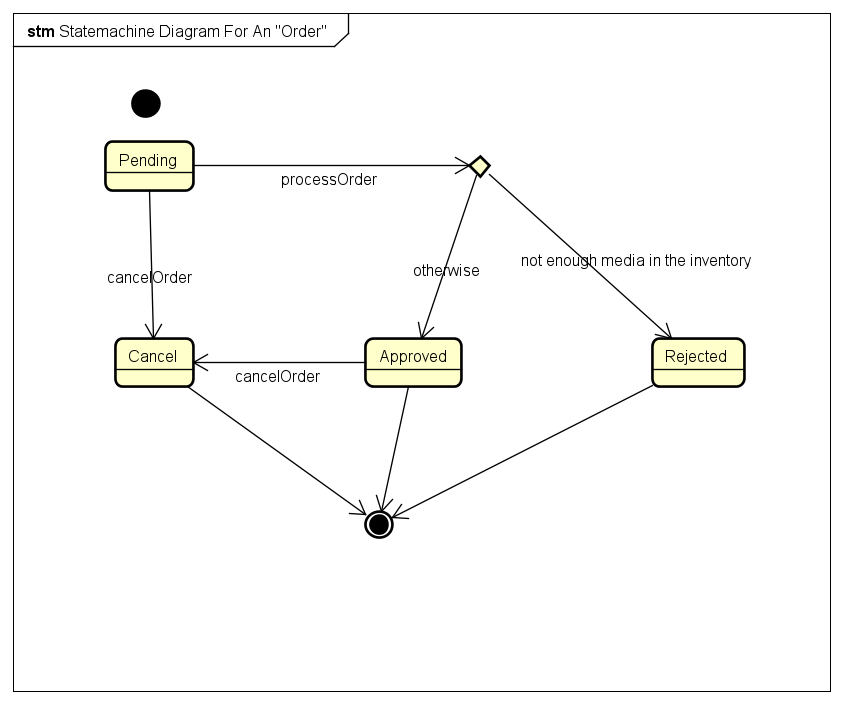
* amount: the amount to pay
* contents: the transaction contents
* cardNumber: the card number
* cardHolderName: the card holder name
* expirationDate: the expiration date in format “mm/yy”
* securityCode: the cvv/cvc of the credit card

Exception:

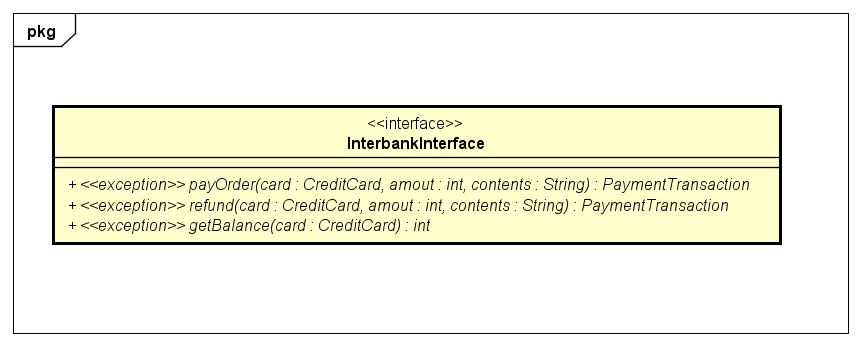
* None
* **Method**
* getExpirationDate: Given the String “date” representing the expiration date in the format “mm/yy”, this method converts it into the required format “mm//yy”. The algorithm is illustrated as follows:

****

* **State**



* 1. Class “InterbankInterface”



* **Attribute**

None

* **Operation**

|  |  |  |  |
| --- | --- | --- | --- |
| # | Name | Return type | Description |
| 1 | payOrder | Map<String,String> | Pay order and return the result with a message |
| 2 | refund | PaymentTransaction | Refund and then return the payment transaction |
| 3 | getBalance | int | Get the balance of the credit card and return the balance |

Parameter:

* card: the credit card used
* amount: the amount to pay/refund
* contents: the transaction contents

Exception:

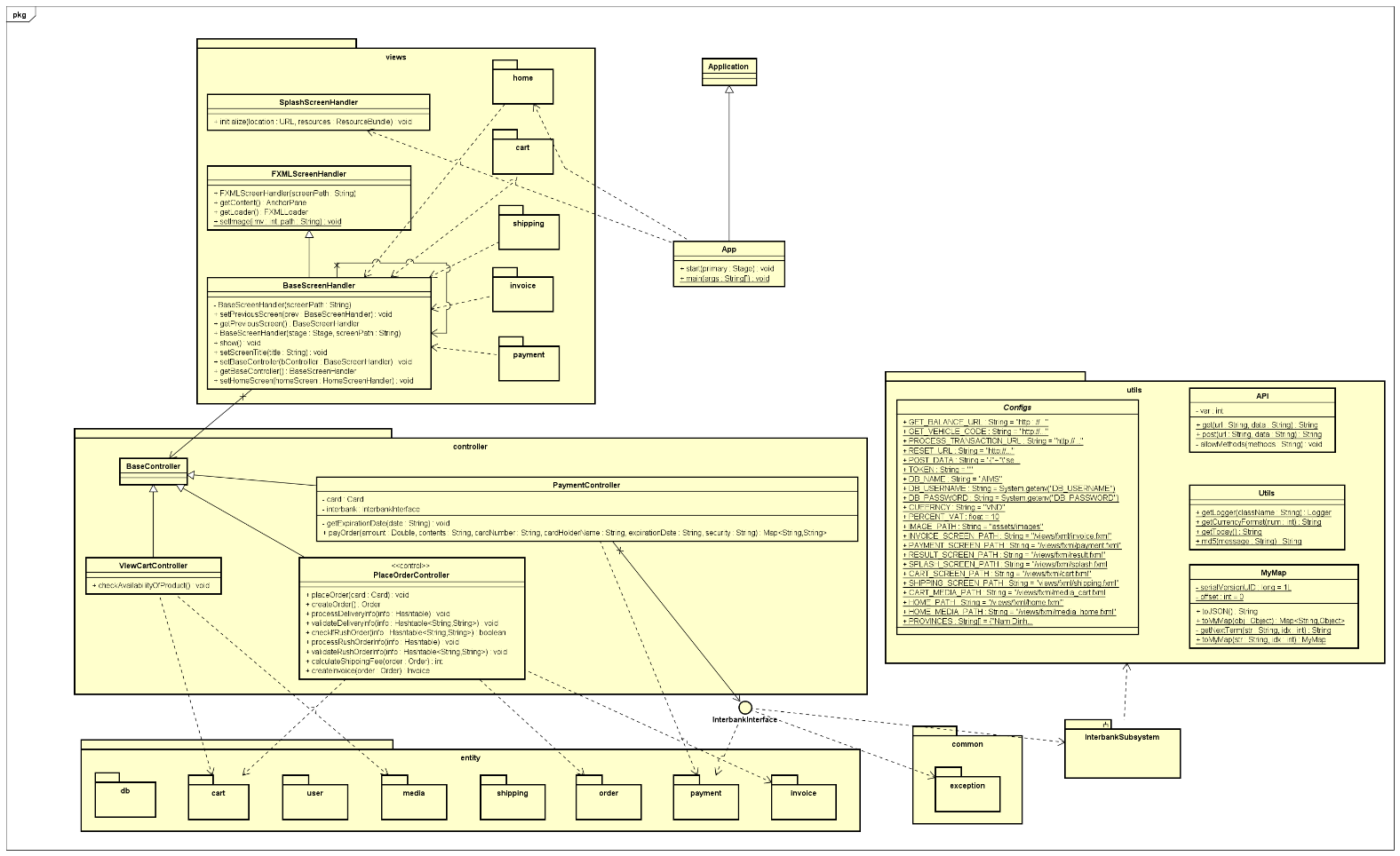
* PaymentExeption: if responded with a pre-defined error code
* UnrecognizedException: if responded with an unknown error code or something goes wrong
* **Method**

None

* **State**

None

1. **Class diagram**

****