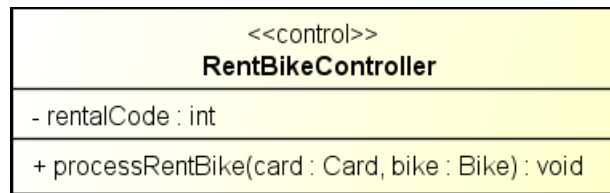


## Lớp thiết kế

### 1. Lớp RentBikeController



Xử lý các yêu cầu thuê xe được gửi về từ phía giao diện người dùng

#### Attribute

#	Name	Data type	Default value	Description
1	rentalCode	int	NULL	Mã thuê xe

#### Operation

#	Name	Return type	Description
1	processRentBike	NULL	Xử lý yêu cầu thuê xe

#### Parameters:

card: Chỉ thực thể thẻ dùng để thuê xe

bike: Thực thể xe mà người dùng muốn thuê

## 2. Lớp ReturnBike Controller

<<control>> ReturnBikeController	
+ processReturnBike(rentalCode : String, newDockId : String) : void + getRentBikeTransaction(rentalCode : String) : RentBikeTransaction + checkRentalCodeValid() : boolean + getBike(bikeCode : String) : Bike	

Xử lý các yêu cầu trả xe từ phía giao diện người dùng

Attribute

Không

Operation

#	Name	Return type	Description
1	processReturnBike	NULL	Xử lý yêu cầu trả xe
2	getRentBikeTransaction	RentBikeTransaction	Lấy thông tin của ca thuê xe từ mã thuê xe
3	checkRentalCodeValid	Boolean	Mã thuê xe có hợp lệ ?
4	getBike	Bike	Lấy thông tin của xe từ mã xe

Parameters:

rentalCode: Mã của ca thuê xe

newDockId: Mã của bãi xe mà người dùng trả xe về đó (có thể trả ở bãi khác bãi mượn xe)

bikeCode: Mã đại diện cho xe người dùng thuê

Method:

Không

### 3. Lớp PaymentController:

<<control>> <b>PaymentController</b>	
- card : Card - interbank : Interbank	
+ processTransaction(amount : int, content : String, expirationDate : int) : void - getExpirationDate() : Date	

Xử lý các yêu cầu thanh toán được gửi về từ người dùng

#### Attribute

#	Name	Data type	Default value	Description
1	Card	Card	NULL	Mô tả thông tin của thẻ dùng để thanh toán
2	Interbank	Interbank	NULL	Cung cấp các hàm thực hiện chức năng thanh toán

#### Operation

#	Name	Return type	Description
1	processTransaction	NULL	Thực hiện xử lý yêu cầu thanh toán
2	getExpirationDate	Date	Lấy ra thông tin về ngày hết hạn của giao dịch thanh toán

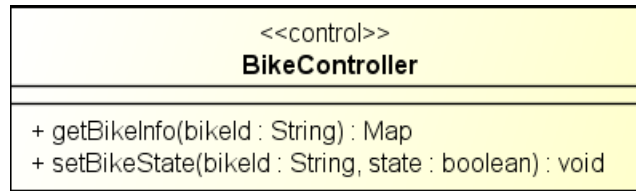
#### Parameters:

Amount: số tiền thanh toán

Content: Nội dung thanh toán

expirationDate: Ngày hết hạn của giao dịch thanh toán

#### 4. Lớp BikeController



Xử lý các yêu cầu liên quan đến việc xem xe của người dùng

Attribute: Không

Operation

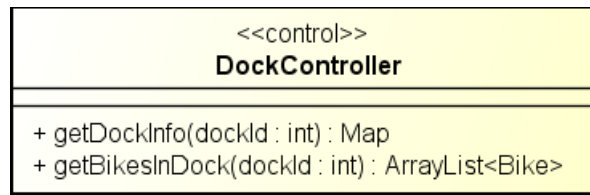
#	Name	Return type	Description
1	getBikeInfo	Map	Lấy ra thông tin của xe
2	setBikeState	NULL	Thiết lập trạng thái của xe

Parameters:

bikeld: Mã xe

state: Trạng thái của xe

## 5. Lớp DockController



Xử lý các yêu cầu liên quan đến việc xem bãi xe của người dùng

Attribute: Không

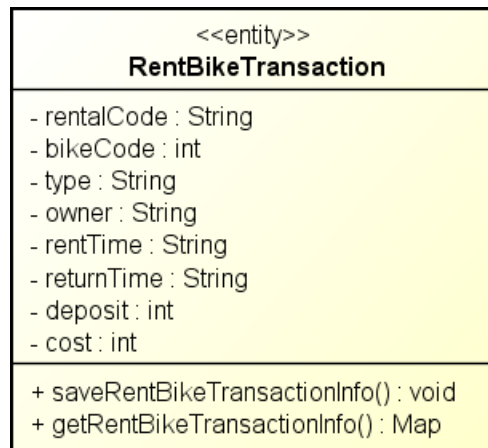
Operation

#	Name	Return type	Description
1	getDockInfo	Map	Lấy ra thông tin của bãi xe
2	setBikesInDock	NULL	Hiển thị danh sách các xe có trong bãi

Parameters:

dockId: Mã bãi xe

## 6. Lớp RentBikeTransaction



### Attribute

#	Name	Data type	Default value	Description
1	rentalCode	String	NULL	Mã của ca thuê xe
2	bikeCode	String	NULL	Mã xe được thuê
3	Type	String	NULL	Loại
4	Owner	String	NULL	Tên chủ thuê xe
5	rentTime	String	NULL	Thời gian bắt đầu thuê xe
6	returnTime	String	NULL	Thời gian trả xe
7	Deposit	Int	NULL	Phí đặt cọc
8	Cost	Int	NULL	Phí thuê xe

### Operation

#	Name	Return type	Description
1	saveRentBikeTransactionInfo	NULL	Lưu thông tin của giao dịch cho thuê xe
2	GetRentBikeTransactionInfo	Map	Lấy ra thông tin về giao dịch cho thuê xe

Parameters:

Không

## 7. Lớp PaymentTransaction

PaymentTransaction
- rentalCode : String - cardCode : String - owner : String - content : String - amount : int - time : String
+ savePaymentTransaction() : void

### Attribute

#	Name	Data type	Default value	Description
1	rentalCode	String	NULL	Mã của ca thuê xe
2	cardCode	String	NULL	Mã thẻ
3	Owner	String	NULL	Tên chủ thẻ
4	Content	String	NULL	Nội dung giao dịch
5	Amount	Int	NULL	Giá trị giao dịch thanh toán
6	Time	String	NULL	Thời gian giao dịch

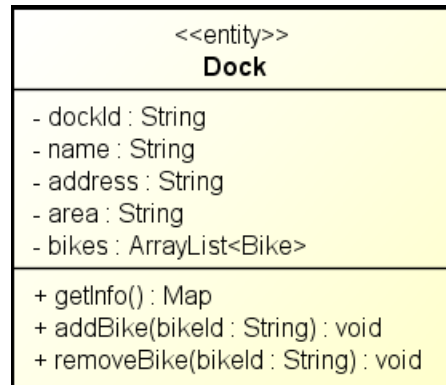
### Operation

#	Name	Return type	Description
1	savePaymentTransaction	NULL	Lưu thông tin của giao dịch thanh toán

Parameters:

Không

## 8. Lớp Dock



### Attribute

#	Name	Data type	Default value	Description
1	dockId	String	NULL	Mã của bãi xe
2	Name	String	NULL	Tên bãi xe
3	Address	String	NULL	Địa chỉ bãi xe
4	Area	Float	NULL	Diện tích bãi xe
5	Bikes	ArrayList<Bike>	NULL	Danh sách các xe trong bãi

### Operation

#	Name	Return type	Description
1	getInfo	Map	Lấy thông tin của bãi xe
2	addBike	NULL	Thêm xe vào bãi
3	removeBike	NULL	Xóa xe khỏi bãi

### Parameters:

bikeld: Mã định danh của xe



## 9. Lớp Bike

<<entity>> <b>Bike</b>	
<ul style="list-style-type: none"><li>- bikeCode : String</li><li>- available : boolean</li><li>- cost : int</li><li>- licensePlate : String</li><li>- numPedals : int</li><li>- numSaddle : int</li><li>- numBicycleSeat : int</li><li>- dockId : String</li></ul>	
<ul style="list-style-type: none"><li>+ updateInfo(bikeCode : String, available : boolean, dockID : boolean, cost : int) : void</li><li>+ getInfo() : Map</li></ul>	

### Attribute

#	Name	Data type	Default value	Description
1	bikeCode	String	NULL	Mã định danh của xe
2	Available	Bool	NULL	Xe có sẵn để phục vụ hay không
3	Cost	Int	NULL	Giá thành xe
4	licensePlate	String	NULL	Biển số xe
5	numPedal	Int	NULL	Số lượng bàn đạp
6	numSaddle	Int	NULL	Số lượng yên xe
7	numBicycleSeat	Int	NULL	Số lượng ghế sau
8	dockId	String	NULL	Mã bãi xe mà xe đang được đặt

### Operation

#	Name	Return type	Description
1	getInfo	Map	Lấy thông tin của xe
2	updateInfo	NULL	Cập nhật các thông tin của xe

### Parameters:

bikeCode: Mã định danh của xe

available: Xe còn sẵn để thuê hay không

dockID: Mã định danh của bãi xe mà xe được đặt

cost: Giá thành xe

## 10. Lớp Card

<b>&lt;&lt;entity&gt;&gt;</b> <b>Card</b>
- cardCode : String - owner : int - CVV : String - expiredDate : String
+ saveCardInfo() : void + deleteFromDB() : void

### Attribute

#	Name	Data type	Default value	Description
1	cardCode	String	NULL	Mã thẻ
2	Owner	String	NULL	Tên chủ thẻ
3	CVV	String	NULL	Mã xác minh thẻ
4	expiredDate	String	NULL	Ngày hết hạn

### Operation

#	Name	Return type	Description
1	saveCardInfo	NULL	Lưu thông tin của thẻ
2	deleteFromDB	NULL	Xóa thẻ khỏi DB

Parameters:

Không