Nhóm 12 – Class Desgin

1. Class "RentBikeController"

<control>> RentBikeController

- + processSelectParking(): void
- + processBarcode(): void

Attribute

None

Operation

#	Name	Return type	Description
1	processSelectParking	void	Quá trình xử lý từ việc chọn bãi
			xe
2	processBarcode	void	Quá trình xử lý việc nhập
			Barcode

Parameter

None

Exception

None

Method

2. GiveBikeBackController

<<control>>

GlveBikeBackController

- + CheckRentBike(): boolean
- + requestToGiveBikeBack(): int

Attribute

None

Operation

#	Name	Return type	Description
1	CheckRentBike	boolean	Kiểm tra quá trình Thuê xe
2	requestToGiveBac	int	Quá trình xử lý yêu cầu trả xe
	k		

Parameter

None

Exception

None

Method

3. PaymentController

<control>> PaymentController

- + requestToPayment(): void
- + DepositBike(): void

Attribute

None

Operation

#	Name	Return type	Description
1	requestToPayment	void	Quá trình xử lý yêu cầu thanh toán
2	DepositBike	void	

Parameter

None

Exception

None

Method

4. Bike

<<entity>> Bike

- Battery : int

- Category: string

- BikelD : int - Price : int

- Status : int

+ getBikeInfo(): void

+ checkBarcode(): void

+ changeBikeStatus(): void

+ checkRentBike(): boolean

+ checkBikeLock(): boolean

Attribute

#	Name	Data type	Default Value	Description
1	Battery	int	NULL	Lượng pin của xe điện
2	Category	string	NULL	Loại xe
3	BikeID	int	NULL	
4	Price	int	NULL	Giá tiền của xe (Dựa
				vào giá để tính tiền cọc)
5	Status	int	NULL	Trạng thái của xe (đã
				thuê hoặc chưa thuê)

Operation

#	Name	Return type	Description
1	getBikeInfo	void	

2	checkBarcode	void	
3	changeBikeStatus	void	
4	checkRentBike	boolean	
5	checkBikeLock	Boolean	

Parameter

None

Exception

None

Method

None

5. Class "Parking"

<<entity>> Parking

- listBike : bike

- Address : string

- Maxslot : int

- Slot : int

+ checkAvailcble(): void

+ updatePakingLot(): void

Attribute

#	Name	Date Type	Default value	Description
1	listBike	Bike	NULL	Danh sách xe trong bãi
2	Address	String	NULL	Địa chỉ bãi đỗ xe
3	Maxslot	int	20	Số chỗ để xe tối đa
4	Slot	int	NULL	Số chỗ đã có xe

Operation

#	Name	Return Type	Description
1	checkAvailable	boolen	Kiểm tra xe có trong bãi
2	updatePakingLot	Void	Cập nhật vị trí để xe

Parameter

None

Exception

None

Method

None

6. Class "PaymentTransaction"

<<entity>> PaymentTransaction

- CustomerID : int

- BikeID : int

- TransactionType : string

- Cost : int

- TransactionID : int

- CardID : int

 $+\ Save Payment Transaction (): void$

+ getPaymentTransaction() : void

Attribute

#	Name	Date	Default value	Description
		Type		
1	TransactionID	Int	NULL	Mã giao dịch
2	CustomerID	Int	NULL	Mã khách hàng
3	BikeID	Int	NULL	Mã xe
4	TransactionTyp	String	NULL	Loại giao dịch
	e			
5	Cost	Int	NULL	Số tiền
6	CardID	int	NULL	Mã thẻ

Operation

#	Name	Return Type	Description
1	SavePaymentTransaction	Void	Lưu giao dịch
2	getPaymentTransaction	Void	Lấy thông tin giao dịch

Parameter

None

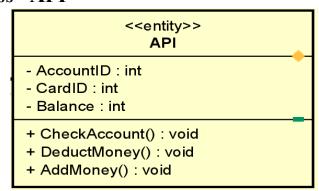
Exception

None

Method

None

7. Class "API"



Attribute

#	Name	Date	Default value	Description
		Type		
1	AccountID	Int	NULL	Mã tài khoản
2	CardID	int	NULL	Mã thẻ
3	Balance	Int	1000000	Số tiền trong thẻ

Operation

#	Name	Return Type	Description
1	CheckAccount	Boolean	Kiểm tra tài khoản
2	DeductMoney	Void	Trừ tiền trong tài khoản (để đặt cọc)
3	AddMoney	Void	Cộng tiền trong tài khoản (để trả xe)

Parameter

None

Exception

None

Method

8. Class "User"

User

CustomerID : intName : stringAddress : stringCardID : int

+ CheckCard() : void

+ CustomerAvailable(): boolean

Attribute

#	Name	Date	Default value	Description
		Type		
1	CustomerID	Int	NULL	Mã khách hàng
2	Name	String	NULL	Tên khách hàng
3	Address	String	NULL	Địa chỉ khách hàng
4	CardID	Int	NULL	Mã thẻ

Operation

#	Name	Return Type	Description
1	CheckCard	Boolean	Kiểm tra thẻ
2	CustomerAvailable	Boolean	Kiểm tra có tồn tại khách hàng không

Parameter

None

Exception

None

Method