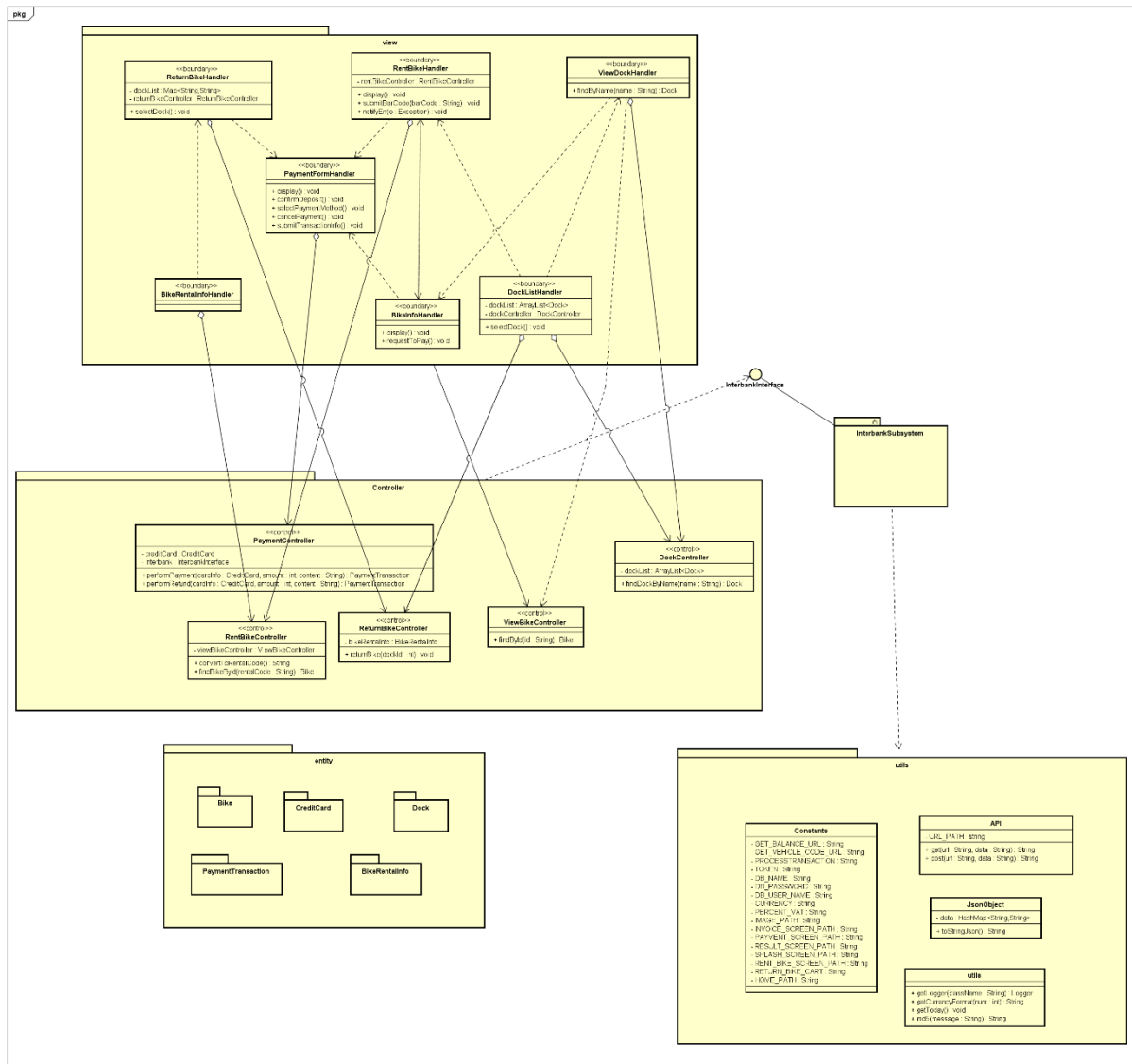


Thiết kế lớp (Class Design)

Biểu đồ quan hệ giữa các lớp



UC Thuê xe

a) Lớp RentBikeHandler

Attribute

- RentBikeController : đối tượng xử lý logic cho nghiệp vụ thuê xe

Operation

#	Tên	Parameter	Kiểu dữ liệu trả về	Mô tả (mục đích)
1	display		void	Khởi tạo màn hình giao diện thuê xe
2	submitBarcode	String barCode	void	Callback function , người dùng ấn nút nhập barcode tìm xe
3	notifyErr	Exception e	void	Báo lỗi cho người dùng liên quan đến quá trình tìm xe, lỗi mạng , đổi barcode Các lỗi gồm có <ul style="list-style-type: none"> - InvalidBarCodeException - NetworkError

Exception:

- UnrecognizedException – nếu không tìm thấy mã lỗi trả về hoặc có lỗi hệ thống
- InvalidBarcodeException – không tìm thấy hoặc lỗi barcode

Method

Không

State

Không

b) Lớp DockListHandler

Attribute

#	Tên	Kiểu dữ liệu	Giá trị mặc định	Mô tả
1	dockList	ArrayList<Map<String, String>>	NULL	Danh sách các trạm xe , đã được tải về
2	dockController	DockController	NULL	Biểu diễn đối tượng xử lý logic của Dock

Operation

#	Tên	Kiểu dữ liệu trả về	Mô tả (mục đích)
1	selectDock	String	Callback function truyền vào view DockListScreen , được gọi khi người dùng chọn trạm xe

Parameter:

Không

Exception:

- UnrecognizedException – nếu không tìm thấy mã lỗi trả về hoặc có lỗi hệ thống

Method

Không
State
Không

c) Lớp DockController

Attribute

#	Tên	Kiểu dữ liệu	Giá trị mặc định	Mô tả
1	dockList	ArrayList<Map<String, String>>	NULL	Danh sách các trạm xe , đã được tải về
2	dockController	DockController	NULL	Biểu diễn đối tượng xử lý logic của Dock

Operation

#	Tên	Kiểu dữ liệu trả về	Mô tả (mục đích)
1	findDockByName	Dock	Lấy dữ liệu của thực thể Dock , tìm theo tên

Parameter:

- name : Tên của Dock (trạm xe)

Exception:

- UnrecognizedException – nếu không tìm thấy mã lỗi trả về hoặc có lỗi hệ thống

Method

Không
State
Không

d) Lớp RentBikeController

Attribute

Không

Operation

#	Tên	Parameter	Kiểu dữ liệu trả về	Mô tả (mục đích)
1	convertToRentalCode	String barcode	String	Chuyển đổi từ barcode sang mã xe tương ứng

2	findBike	String rentalCode	Bike	Tìm kiếm và trả về đối tượng Bike với code tương ứng
---	----------	----------------------	------	--

Exception:

- UnrecognizedException – nếu không tìm thấy mã lỗi trả về hoặc có lỗi hệ thống
- InvalidBarcodeException – không tìm thấy hoặc lỗi barcode

Method

Không

State

Không

e) Lớp BikeInfoHandler

Attribute

Không

Operation

#	Tên	Parameter	Kiểu dữ liệu trả về	Mô tả (mục đích)
1	display	Bike bike	void	Khởi tạo giao diện màn hình xem chi tiết xe
2	requestToPay		void	Nhận yêu cầu thanh toán để thuê xe

Exception:

- UnrecognizedException – nếu không tìm thấy mã lỗi trả về hoặc có lỗi hệ thống
- NetworkError – Lỗi mạng

Method

Không

State

Không

f) Lớp PaymentHandler

Attribute

#	Tên	Kiểu dữ liệu	Giá trị mặc định	Mô tả
1	paymentController	PaymentController	NULL	Đối tượng xử lý logic nghiệp vụ thanh toán

Operation

#	Tên	Parameter	Kiểu dữ liệu trả về	Mô tả (mục đích)
1	display		void	Khởi tạo giao diện màn hình thanh toán để thuê xe
2	confirmDeposit		void	Người dùng xác nhận đồng ý thanh toán tiền cọc thuê xe
3	selectPaymentMethod		void	Người dùng chọn phương pháp thanh toán
4	cancelPayment		void	Người dùng hủy quá trình thanh toán
5	submitTransactionInfo		void	Người dùng nhập các thông tin thanh toán

Exception:

- UnrecognizedException – nếu không tìm thấy mã lỗi trả về hoặc có lỗi hệ thống
- NetworkError – Lỗi mạng

Method

Không

State

Không

g) Lớp PaymentController

Attribute

#	Tên	Kiểu dữ liệu	Giá trị mặc định	Mô tả
1	creditCard	CreditCard	NULL	Đối tượng thẻ tín dụng
2	interbank	InterbankInterface	NULL	

Operation

#	Tên	Parameter	Kiểu dữ liệu trả về	Mô tả (mục đích)
1	performPayment	CreditCard cardInfo , int amount , String content	PaymentTransaction	Xử lý thanh toán thuê xe
2	performRefund	CreditCard cardInfo , int amount , String content	PaymentTransactin	Xử lý hoàn tiền thuê xe

Exception:

- UnrecognizedException – nếu không tìm thấy mã lỗi trả về hoặc có lỗi hệ thống
- NetworkError – Lỗi mạng

Method

Không

State

Không

h) Lớp BikeRentalInfo

Attribute

Không

Operation

#	Tên	Parameter	Kiểu dữ liệu trả về	Mô tả (mục đích)
1	save		void	Lưu lại thông tin phiên thuê xe

Exception:

- UnrecognizedException – nếu không tìm thấy mã lỗi trả về hoặc có lỗi hệ thống
- NetworkError – Lỗi mạng

Method

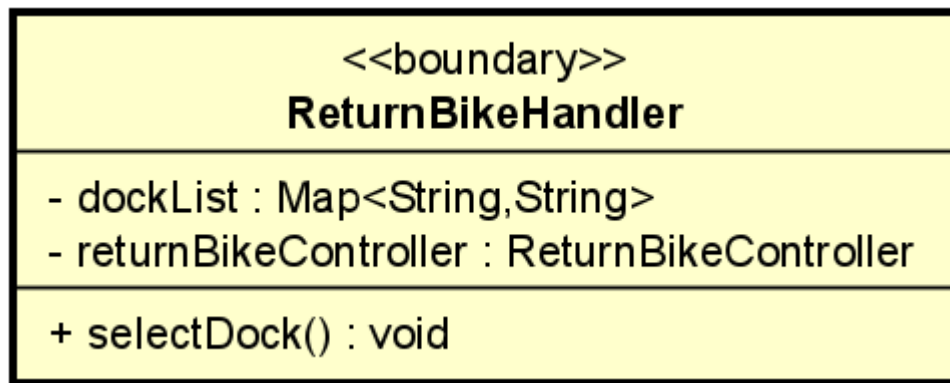
Không

State

Không

UC trả xe

a. Lớp ReturnBikeHandler



Attribute

#	Tên	Kiểu dữ liệu	Giá trị mặc định	Mô tả
1	dockList	Map<String, String>	NULL	Danh sách các trạm xe , đã được tải về
2	returnBikeController	ReturnBikeController	NULL	Đối tượng điều khiển xử lý sự kiện cho màn hình trả xe

Operation

#	Tên	Kiểu dữ liệu trả về	Mô tả (mục đích)
1	selectDock	void	Callback function truyền vào màn hình trả xe, được gọi khi người dùng chọn trạm xe để trả xe

Parameter:

Không

Exception:

- Không

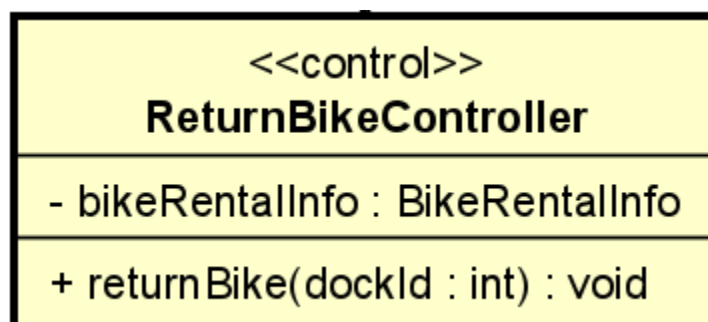
Method

Không

State

Không

b. Lớp ReturnBikeController



Attribute

#	Tên	Kiểu dữ liệu	Giá trị mặc định	Mô tả
1	bikeRentalInfo	BikeRentalInfo	NULL	Đối tượng thực thể biểu diễn thông tin thuê xe

Operation

#	Tên	Kiểu dữ liệu trả về	Mô tả (mục đích)
1	returnBike	void	Lấy dữ liệu của thực thể Dock , tìm theo tên

Parameter:

- dockId : Mã của trạm xe người dùng chọn để trả xe

Exception:

- ReturnBikeException- nếu có lỗi xảy ra trong quá trình trả xe

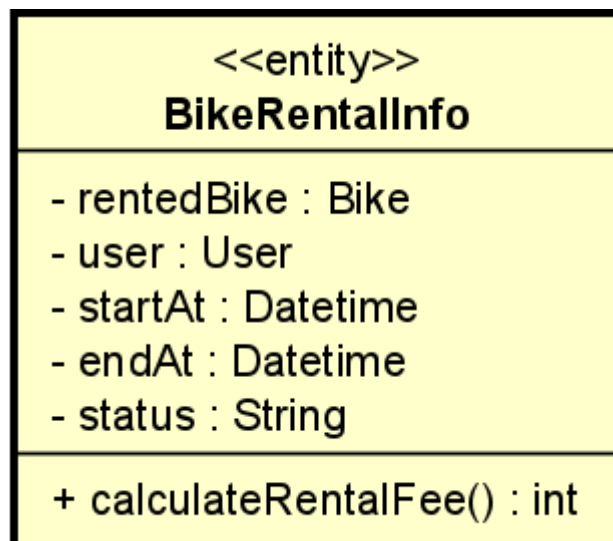
Method

Không

State

Không

c. Lớp BikeRentalInfo



Attribute

#	Tên	Kiểu dữ liệu	Giá trị mặc định	Mô tả
1	rentedBike	Bike	NULL	Đối tượng xe được thuê
2	user	User	NULL	Đối tượng biểu diễn người thuê xe
3	startAt	Datetime	NULL	Thời gian bắt đầu thuê xe
4	endAt	Datetime	NULL	Thời gian kết thúc thuê xe
5	status	String	ONGOING	Trạng thái của phiên thuê xe

Operation

#	Tên	Kiểu dữ liệu trả về	Mô tả (mục đích)
1	calculateRentalFee	int	Tính toán số tiền thuê xe cần trả

Parameter:

Không

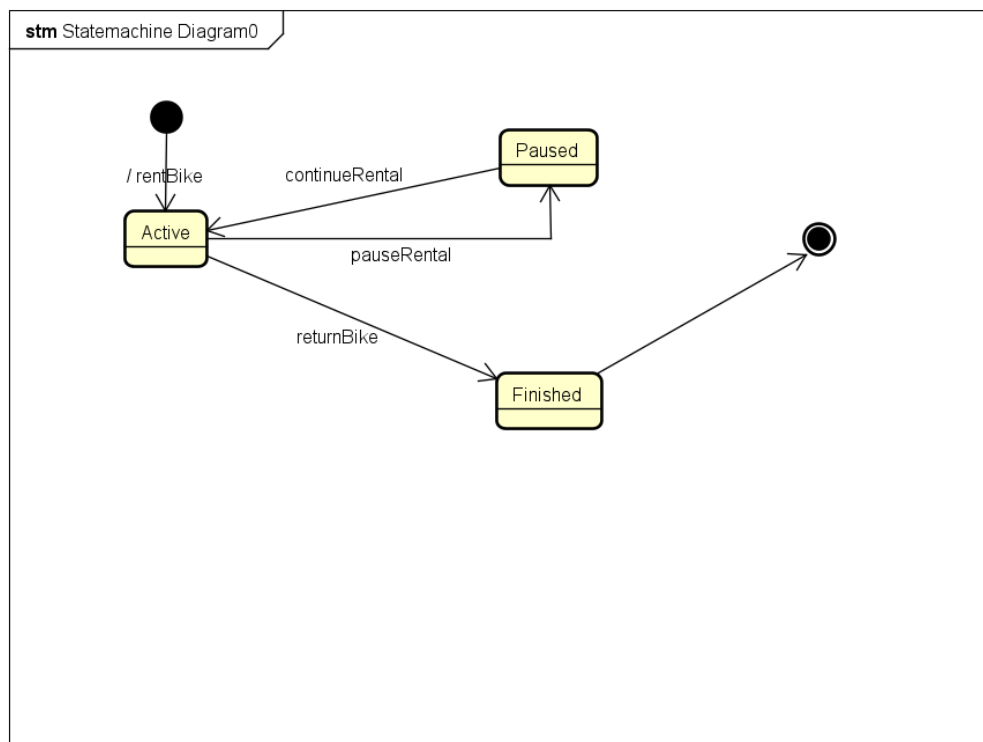
Exception:

- Không

Method

Không

State



Interbank Subsystem

a. Lớp InterbankInterface

<<interface>> InterbankInterface	
+ performPayment(cardInfo : CreditCard, amount : int, content : String) : PaymentTransaction + performRefund(cardInfo : CreditCard, amount : int, content : String) : PaymentTransaction	

Attribute

Không

Operation

#	Tên	Kiểu dữ liệu trả về	Mô tả (mục đích)
1	performPayment	PaymentTransaction	Thực hiện thanh toán và trả về giao dịch thanh toán
2	performRefund	PaymentTransaction	Thực hiện hoàn tiền và trả về giao dịch

Parameter:

- cardInfo - thông tin thẻ tín dụng để giao dịch
- amount - số tiền giao dịch
- content - nội dung giao dịch

Exception:

- PaymentException - nếu mã lỗi trả về đã biết
- UnrecognizedException - nếu không tìm thấy mã lỗi trả về hoặc có lỗi hệ thống

Method

Không

State

Không

b. Lớp InterbankController

InterbankSystemController	
- interbankBoundary : InterbankBoundary	
+ performPayment(cardInfo : CreditCard, amount : int, content : String) : PaymentTransaction + performRefund(cardInfo : CreditCard, amount : int, content : String) : PaymentTransaction	

Attribute

#	Tên	Kiểu dữ liệu	Giá trị mặc định	Mô tả
1	interbankBoundary	InterbankBoundary	NULL	Đối tượng biểu diễn giao diện hệ thống giao tiếp với API

Operation

#	Tên	Kiểu dữ liệu trả về	Mô tả (mục đích)
1	performPayment	PaymentTransaction	Thực hiện thanh toán và trả về giao dịch thanh toán
2	performRefund	PaymentTransaction	Thực hiện hoàn tiền và trả về giao dịch

Parameter:

- cardInfo - thông tin thẻ tín dụng để giao dịch
- amount - số tiền giao dịch
- content - nội dung giao dịch

Exception:

- PaymentException - nếu mã lỗi trả về đã biết
- UnrecognizedException - nếu không tìm thấy mã lỗi trả về hoặc có lỗi hệ thống

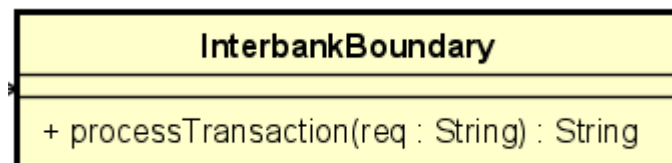
Method

Không

State

Không

c. Lớp InterbankBoundary



Attribute

Không

Operation

#	Tên	Kiểu dữ liệu trả về	Mô tả (mục đích)
---	-----	---------------------	------------------

1	processTransaction	String	Thực hiện xử lý giao dịch và trả về kết quả dưới dạng JSON string
---	--------------------	--------	---

Parameter:

- req - nội dung yêu cầu gửi đến API dưới dạng JSON string

Exception:

Không

Method

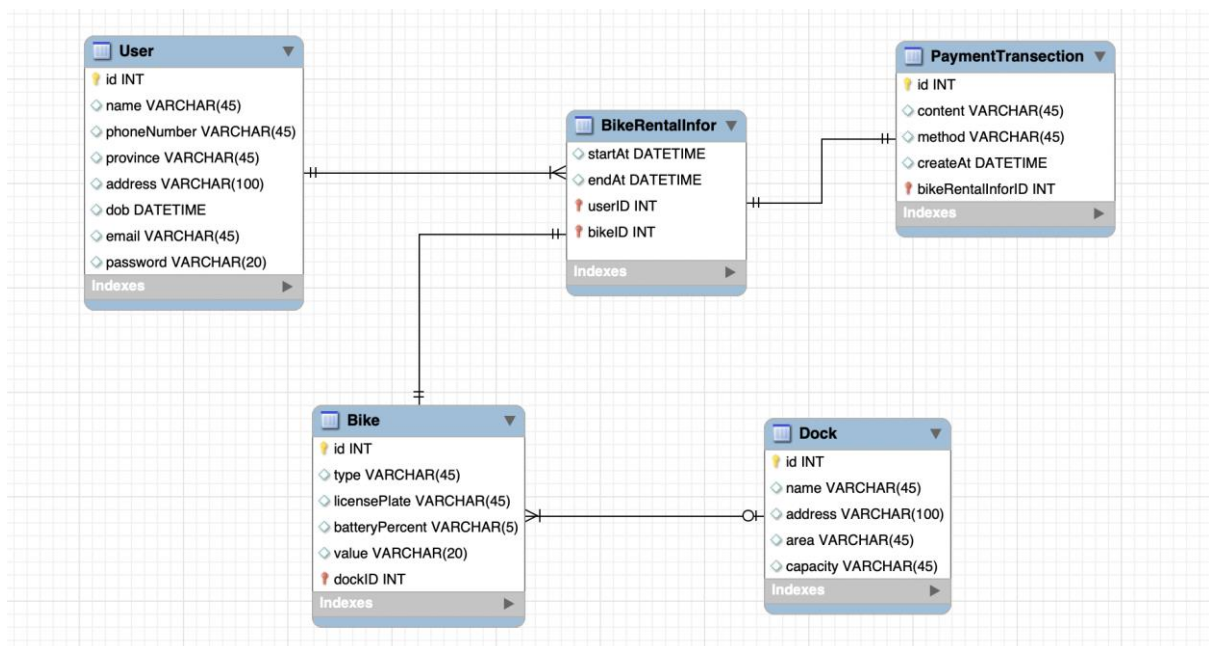
Không

State

Không

MÔ HÌNH DỮ LIỆU

Mô hình dữ liệu logic



Mô hình dữ liệu vật lý

Chú thích:

PK: Primary Key

FK: Foreign Key

- User

#	PK	FK	Column	Data type	Mandatory	Description
1	x		id	interger	Yes	ID, auto increment
2			name	VARCHAR(45)	Yes	Tên của khách
3			phoneNumber	VARCHAR(45)	Yes	Số. điện thoại của khách
4			province	VARCHAR(45)	Yes	Tỉnh
5			address	VARCHAR(100)	Yes	Địa chỉ khách
6			dob	DATETIME	Yes	Ngày sinh của khách
7			email	VARCHAR(45)	Yes	Email của khách
8			password	VARCHAR(45)	Yes	Mật khẩu khách đăng ký

- BikeRentalInfor

#	PK	FK	Column	Data type	Mandatory	Description
1			startAt	DATETIME	Yes	Thời gian khách đã thuê
2			endAt	DATETIME	Yes	Thời gian khách trả xe
3		x	userID	interger	Yes	ID của khách
4		x	bikeID	interger	Yes	ID của xe

- PaymentTransection

#	PK	FK	Column	Data type	Mandatory	Description
---	----	----	--------	-----------	-----------	-------------

1	x		id	interger	Yes	ID, not null, auto increment
2			content	VARCHAR(45)	Yes	Nội dung thanh toán
3			method	VARCHAR(45)	Yes	Phương thức thanh toán
4			createAt	DATETIME	Yes	Thời điểm thanh toán
5		x	bikeRentalInfoID	interger	No	ID của thông tin xe đã thuê

▪ Bike

#	PK	FK	Column	Data type	Mandatory	Description
1	x		id	interger	Yes	ID, same as ID of Dock of which type is Bike
2			type	VARCHAR(45)	Yes	Loại xe
3			licensePlate	VARCHAR(45)	Yes	Biển số xe
4			batteryPercent	VARCHAR(45)	No	Phần trăm pin của xe
5			value	VARCHAR(45)	Yes	Giá thuê xe
6		x	dockID	interger	Yes	ID của bãi xe

▪ Dock

#	PK	FK	Column	Data type	Mandatory	Description
1	x		id	interger	Yes	ID, auto increment
2			name	VARCHAR(45)	Yes	Tên bãi xe
3			address	VARCHAR(45)	Yes	Địa chỉ bãi xe
4			area	VARCHAR(45)	Yes	Khu vực bãi xe
5			capacity	VARCHAR(45)	No	Sức chứa của bãi xe

Database script

```
CREATE TABLE IF NOT EXISTS `Ecobike-logic-data-model`.`User` (  
  `id` INT NOT NULL,  
  `name` VARCHAR(45) NULL,  
  `phoneNumber` VARCHAR(45) NULL,  
  `province` VARCHAR(45) NULL,  
  `address` VARCHAR(100) NULL,  
  `dob` DATETIME NULL,  
  `email` VARCHAR(45) NULL,  
  `password` VARCHAR(20) NULL,  
  PRIMARY KEY (`id`))
```

```
CREATE TABLE IF NOT EXISTS `Ecobike-logic-data-model`.`Dock` (  
  `id` INT NOT NULL,  
  `name` VARCHAR(45) NULL,  
  `address` VARCHAR(100) NULL,  
  `area` VARCHAR(45) NULL,  
  `capacity` VARCHAR(45) NULL,  
  PRIMARY KEY (`id`))
```

```
CREATE TABLE IF NOT EXISTS `Ecobike-logic-data-model`.`Bike` (  
  `id` INT NOT NULL,  
  `type` VARCHAR(45) NULL,  
  `licensePlate` VARCHAR(45) NULL,  
  `batteryPercent` VARCHAR(5) NULL,  
  `value` VARCHAR(20) NULL,  
  `dockID` INT NOT NULL,  
  PRIMARY KEY (`id`, `dockID`),  
  INDEX `fk_Bike_Dock1_idx` (`dockID` ASC) VISIBLE,  
  CONSTRAINT `fk_Bike_Dock1`  
    FOREIGN KEY (`dockID`)
```

```
REFERENCES `Ecobike-logic-data-model`.`Dock` (`id`)
```

```
CREATE TABLE IF NOT EXISTS `Ecobike-logic-data-model`.`BikeRentalInfor` (  
  `startAt` DATETIME NULL,  
  `endAt` DATETIME NULL,  
  `userID` INT NOT NULL,  
  `bikeID` INT NOT NULL,  
  PRIMARY KEY (`userID`, `bikeID`),  
  INDEX `fk_BikeRentalInfor_Bike1_idx` (`bikeID` ASC) VISIBLE,  
  CONSTRAINT `fk_BikeRentalInfor_User1`  
    FOREIGN KEY (`userID`)  
      REFERENCES `Ecobike-logic-data-model`.`User` (`id`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT `fk_BikeRentalInfor_Bike1`  
    FOREIGN KEY (`bikeID`)  
      REFERENCES `Ecobike-logic-data-model`.`Bike` (`id`)
```

```
CREATE TABLE IF NOT EXISTS `Ecobike-logic-data-model`.`PaymentTransection`  
(  
  `id` INT NOT NULL,  
  `content` VARCHAR(45) NULL,  
  `method` VARCHAR(45) NULL,  
  `createAt` DATETIME NULL,  
  `bikeRentalInforID` INT NOT NULL,  
  PRIMARY KEY (`id`, `bikeRentalInforID`),  
  INDEX `fk_PaymentTransection_BikeRentalInfor1_idx` (`bikeRentalInforID` ASC)  
  VISIBLE,  
  CONSTRAINT `fk_PaymentTransection_BikeRentalInfor1`  
    FOREIGN KEY (`bikeRentalInforID`)  
      REFERENCES `Ecobike-logic-data-model`.`BikeRentalInfor` (`userID`)
```