

ĐẠI HỌC BÁCH KHOA HÀ NỘI
Trường Công nghệ thông tin và Truyền thông

Software Design Document

PHẦN MỀM THUÊ XE ĐẠP THEO GIỜ
Môn: Thiết kế và xây dựng phần mềm

Nhóm 2

Trịnh Tùng Dương.....	20190045
Lê Anh Đức.....	20190042
Nguyễn Hải Dương.....	20190044

Hà Nội, tháng 2 năm 2023

Table of Contents

Mục lục				11
Giới thiệu				
Error!	Bookmark	not	defined.	1.1
Mục tiêu				
				3
1.2 Phạm vi Error!	Bookmark	not	defined.	1.3
Thuật ngữ				3
1.4 Tài liệu tham khảo				32
Mô tả tổng quan Error!	Bookmark	not	defined.	2.1
Tổng quan				4
2.2 Giá thiết/Ràng buộc/Rủi ro				42.2.1
Giá thiết				42.2.2
Ràng buộc				52.2.3
Rủi ro				453
Kiến trúc hệ thống và thiết kế kiến trúc Error!	Bookmark	not	defined.	3.1
Khuôn mẫu thiết kế				6
3.2 Biểu đồ tương tác Error!	Bookmark	not	defined.	3.3
Biểu đồ phân tích lớp				8
3.4 Biểu đồ phân tích cho từng lớp	Bookmark	not	defined.	3.5
Bảo mật				12
4 Thiết kế chi tiết	Bookmark	not	defined.	4.1
13 Error!				13
4.1.1 Giao diện tổng quan	Bookmark	not	defined.	4.1.2
13 Error!				19
Đặc tả giao diện				
4.1.3 Biểu đồ dịch chuyển màn hình	Bookmark	not	defined.	4.2
31 Error!				32
Mô hình hóa dữ liệu				
4.2.1 Mô hình hóa mức khái niệm				32324.2.2
Thiết kế cơ sở dữ liệu				33334.3
Non-Database Management System Files				33374.4
Thiết kế lớp	Bookmark	not	defined.	4.4.1
37 Error!				37
Biểu đồ lớp tổng quan				

4.4.2	Biểu đồ lớp thiết kế tổng quan	38	Error!	Bookmark	not	defined.	4.4.3
	Biểu đồ thiết kế từng lớp chi tiết						38
5	Design Considerations						47
	Goals and Guidelines	47	Error!	Bookmark	not	defined.	5.2
	Architectural Strategies						47
5.3	Coupling and Cohesion	47	Error!	Bookmark	not	defined.	5.4
	Design Principles						48
5.5	Design Patterns						48

1 Giới thiệu

1.1 Mục tiêu

- Mục tiêu của tài liệu này là đưa ra các đặc tả chi tiết yêu cầu phần mềm cho ứng dụng EcoBikeRental cho thuê xe đạp theo giờ với nhiều bãi để xe để thuê/trả xe tự động trong khu đô thị Ecopark.
- Tài liệu đặc tả này phục vụ cho designers, developers và testers của hệ thống có thể thuận lợi hơn trong việc lập trình cũng như triển khai dự án
- Tài liệu này cũng có thể dùng bởi các bên có liên quan để đảm bảo sự thống nhất trong quá trình triển khai và sử dụng

1.2 Phạm vi

Thuê xe đạp dạo quanh là một trong những dịch vụ được thu hút tại khu đô thị sinh thái Ecopark. Nhằm giúp việc quản lý thuê xe trở nên đơn giản hơn đối với phía quản lý cũng như tạo ra một ứng dụng thân thiện với người dùng, phần mềm quản lý thuê xe EcoBikeRental được lên kế hoạch thực hiện. Hệ thống EcoBikeRental là hệ thống cho thuê xe của khu đô thị Ecopark. Nhờ có hệ thống thuê xe để sử dụng của người dân trở nên thuận tiện hơn. Phần mềm giúp dịch vụ thuê xe có thể trở nên phát triển, tiếp tục mở rộng. Ngoài ra, để dịch vụ này tiếp tục phát triển mở rộng ra, cần giải quyết khâu hạ tầng, bao gồm hệ thống làn đường cho xe đạp, điểm dừng, trông giữ, bảo quản, và đặc biệt là hệ thống thông tin thuê xe và trả xe tự động có thể hoạt động 24/7

1.3 Thuật ngữ

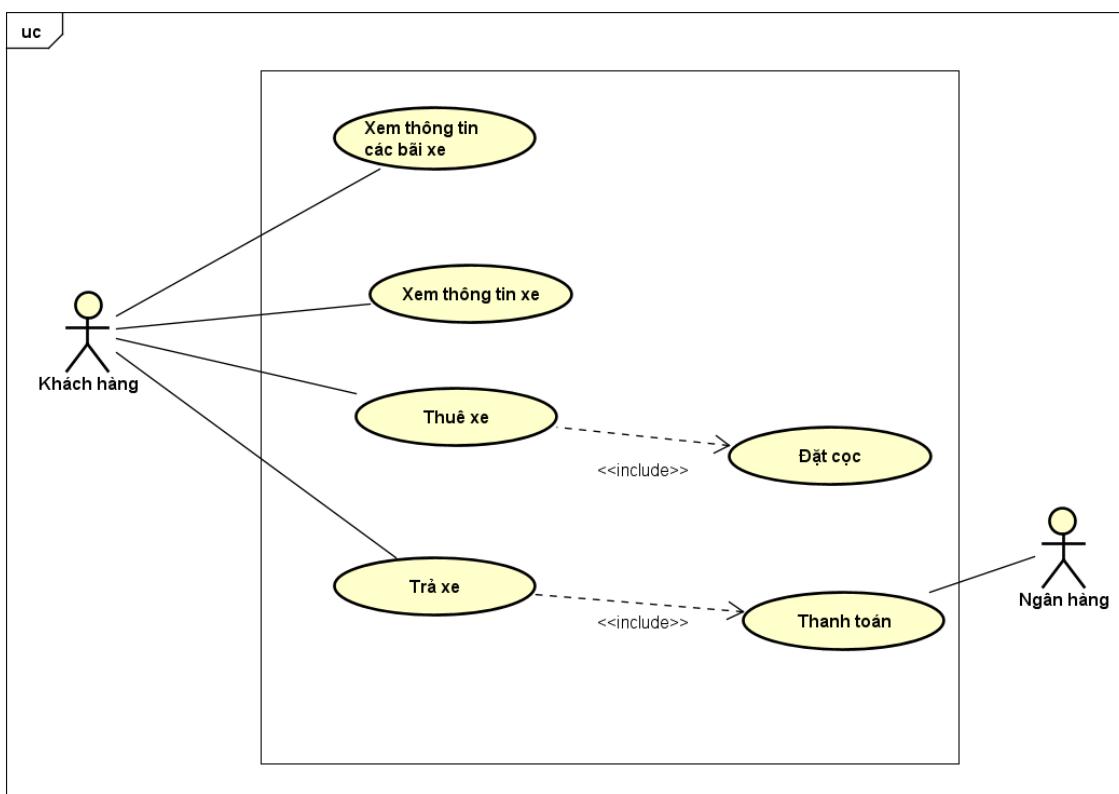
- API: (Application Programming Interface): là phần mềm trung gian cho phép kết nối 2 ứng dụng với nhau
- GUI(Graphical User Interface): Giao diện cho phép người dùng tương tác với hệ thống

1.4 Tài liệu tham khảo

2 Mô tả tổng quan

2.1 Tổng quan hệ thống

Về tổng quan, hệ thống có những chức năng cơ bản: Giao diện tương tác với người dùng; yêu cầu của người dùng được gửi và xử lý bằng cách nhấp vào giao diện và sau đó yêu cầu được xử lý bởi bộ điều khiển hệ thống, ngoài ra có một cơ sở dữ liệu (từ xa) để lưu trữ bất kỳ loại dữ liệu nào; mọi yêu cầu hoặc thay đổi liên quan đến dữ liệu sẽ được truy vấn trong cơ sở dữ liệu; sự thay đổi trong cơ sở dữ liệu sau đó được phản ánh trong UI (giao diện người dùng). Để làm được điều đó, hệ thống được xây dựng bao gồm 3 thành phần chính: view, controller, entity, mỗi thành phần đóng góp vai trò riêng biệt nhưng liên kết với nhau, điều đó giúp việc lập trình trở nên logic và thông nhất đồng thời cũng giúp cho hệ thống hoạt động ổn định hơn



Hình 1: Biểu đồ use case tổng quan

2.2 Giải thích/Ràng buộc/Rủi ro

2.2.1 Giải thích

Các yêu cầu đối với người dùng khi sử dụng phần mềm:

- Phần mềm chạy trên nền tảng web, do đó để sử dụng thì người dùng cần có kết nối Internet tốt
- Thời điểm hiện tại, phần mềm mới chỉ hỗ trợ trên Desktop, chưa hỗ trợ chạy trên các thiết bị điện thoại, bởi vậy người dùng cần sử dụng desktop để sử dụng.

2.2.2 Ràng buộc

- Nếu kết nối internet không tốt, phần mềm có thể hoạt động không chính xác
- Trong trường hợp có nhiều người dùng truy cập vào cùng một thời điểm, thời gian phản hồi có thể chậm hơn đáng kể
- Bộ nhớ RAM trên máy tính của người dùng nên tối thiểu là 2GB để có thể sử dụng ứng dụng một cách bình thường.

2.2.3 Rủi ro

Dữ liệu về ca thuê xe có thể mất nếu người dùng đóng ứng dụng nhưng chưa thực hiện trả xe

3 Kiến trúc hệ thống và thiết kế kiến trúc

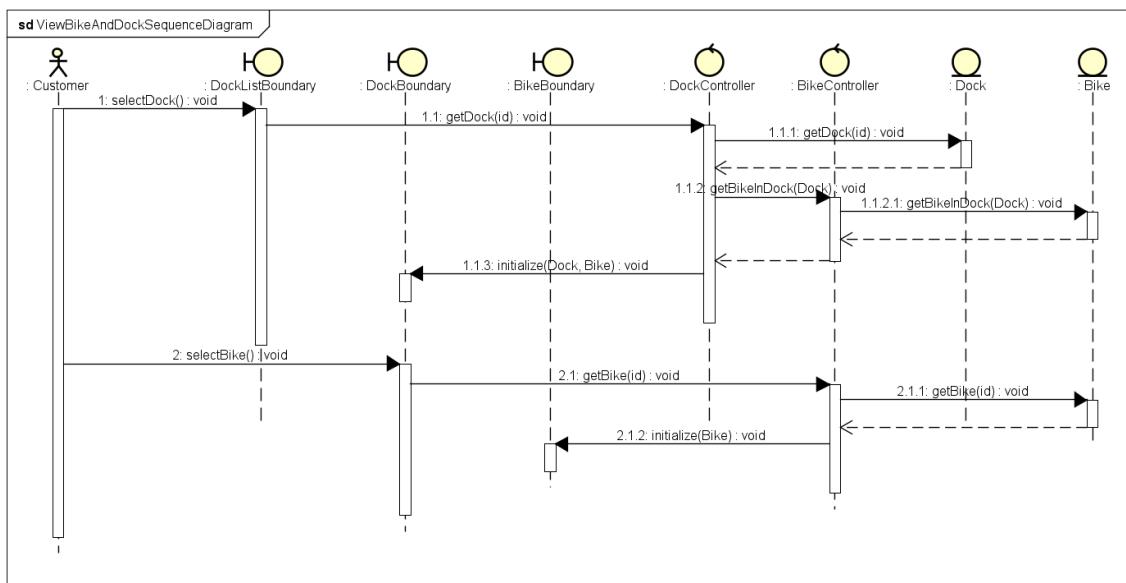
3.1 Khuôn mẫu kiến trúc

Trong Project, nhóm sử dụng mô hình 3 tầng để thực hiện thiết kế kiến trúc. Mô hình này giúp quá trình thực thi code trở nên tiện lợi hơn, đặc biệt giúp cho việc cập nhật và mở rộng hệ thống được dễ dàng hơn

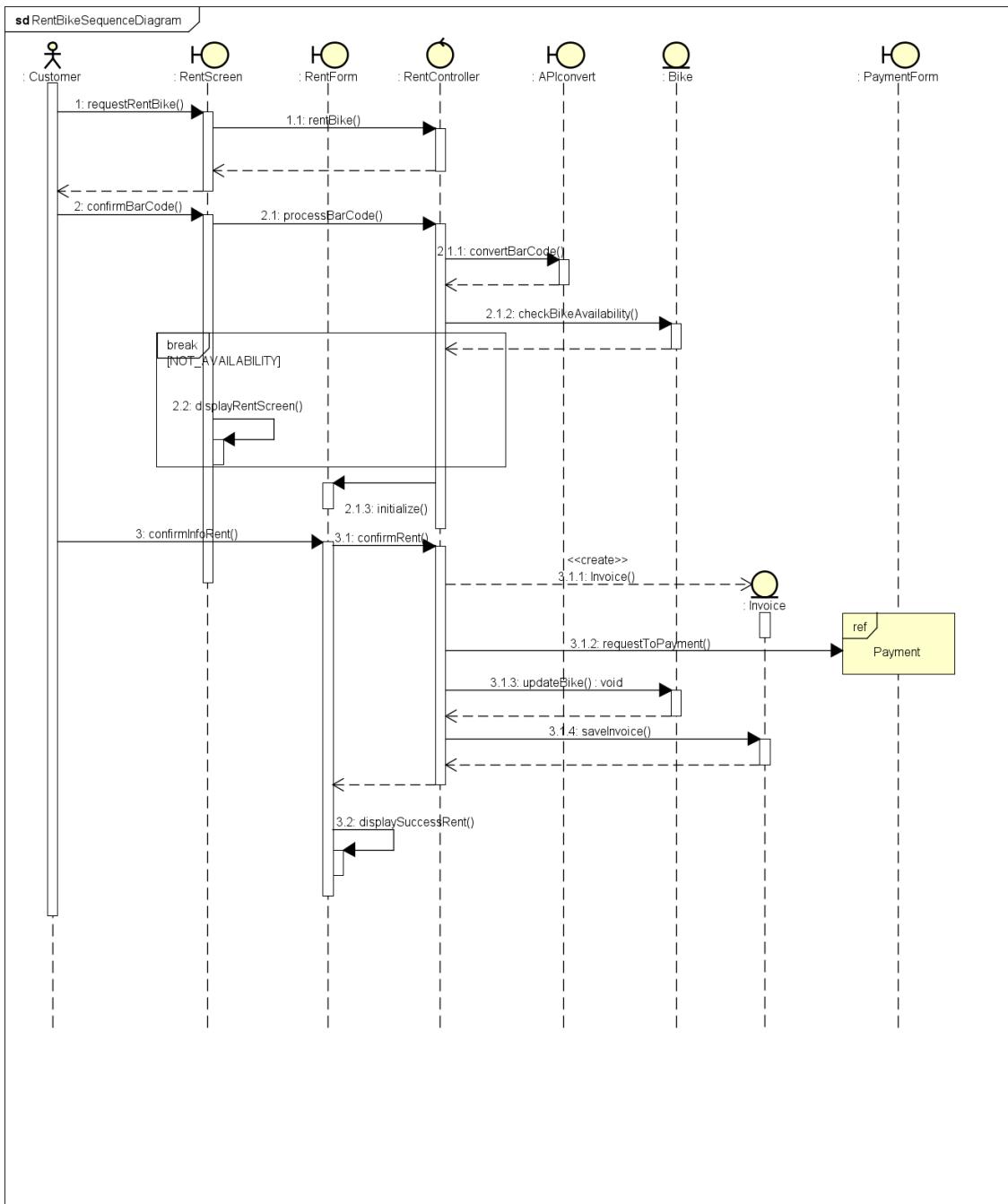
3.2 Biểu đồ tương tác

3.2.1 Biểu đồ trình tự

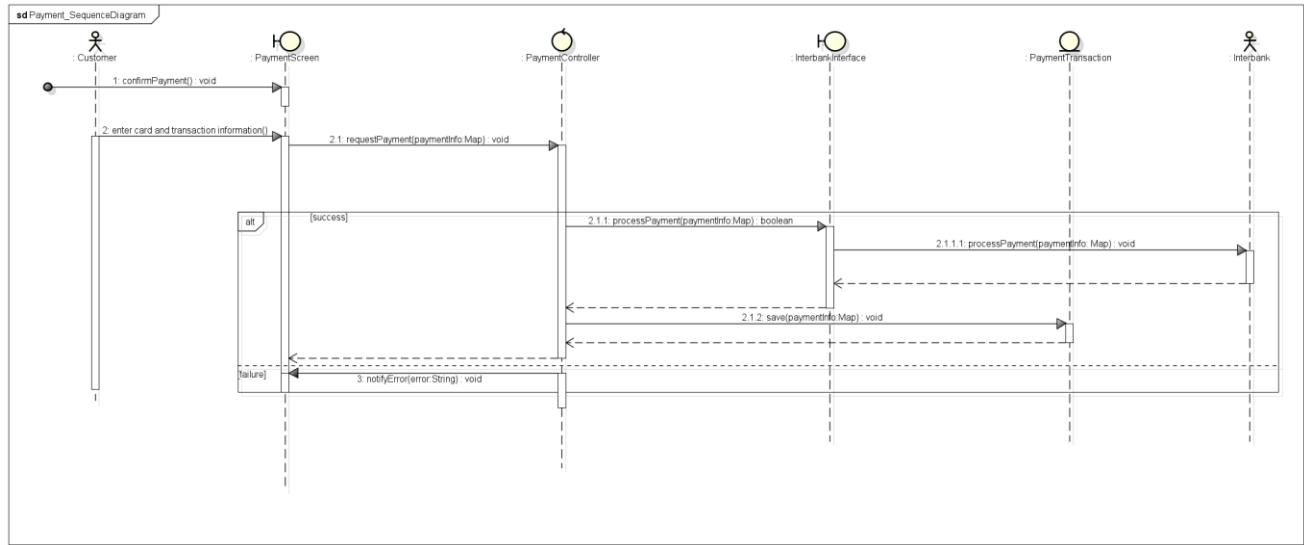
Quy trình xem xe và xem bãi xe của người dùng



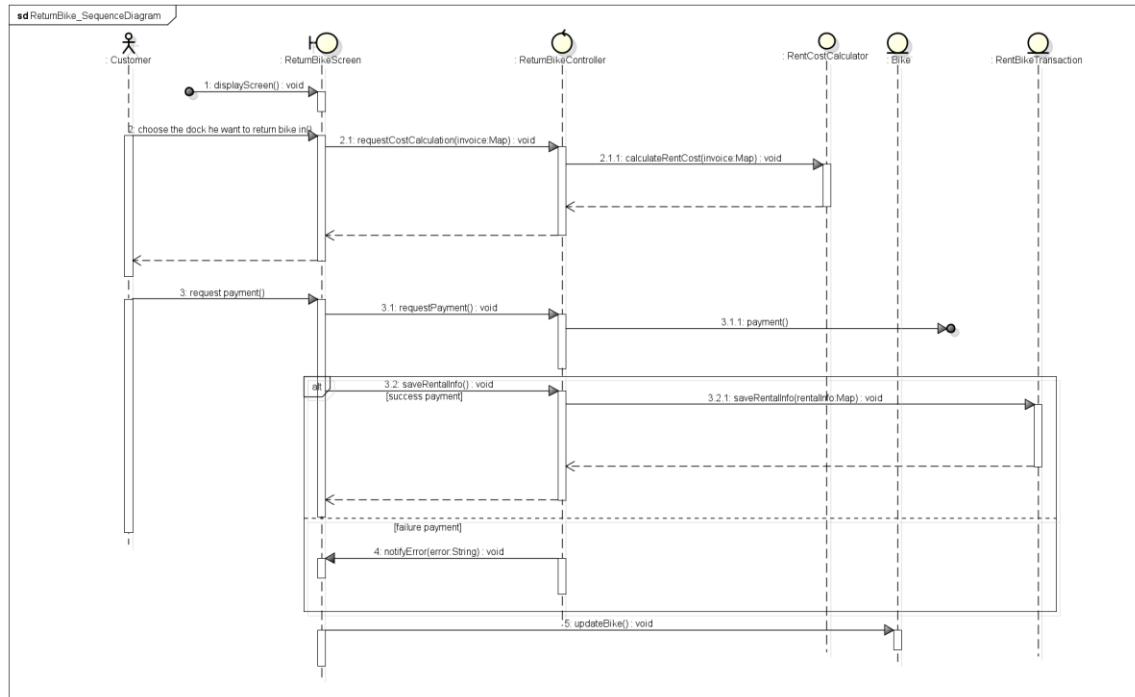
Quy trình thuê xe của người dùng:



Quy trình thanh toán của người dùng:

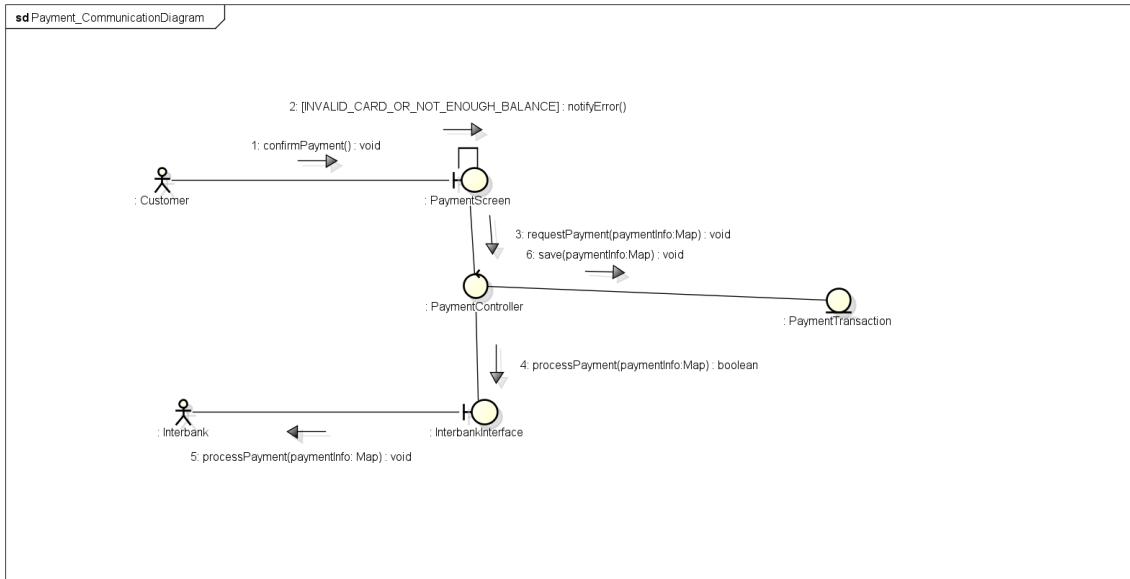


Quy trình trả xe

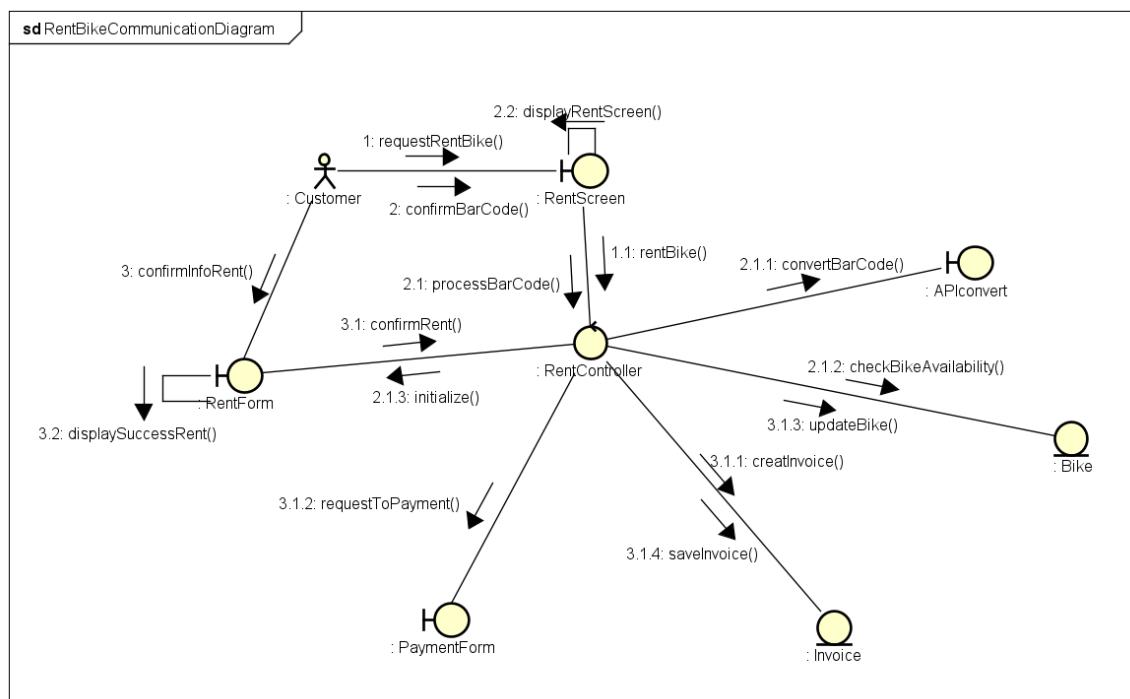


3.2.2 Biểu đồ giao tiếp

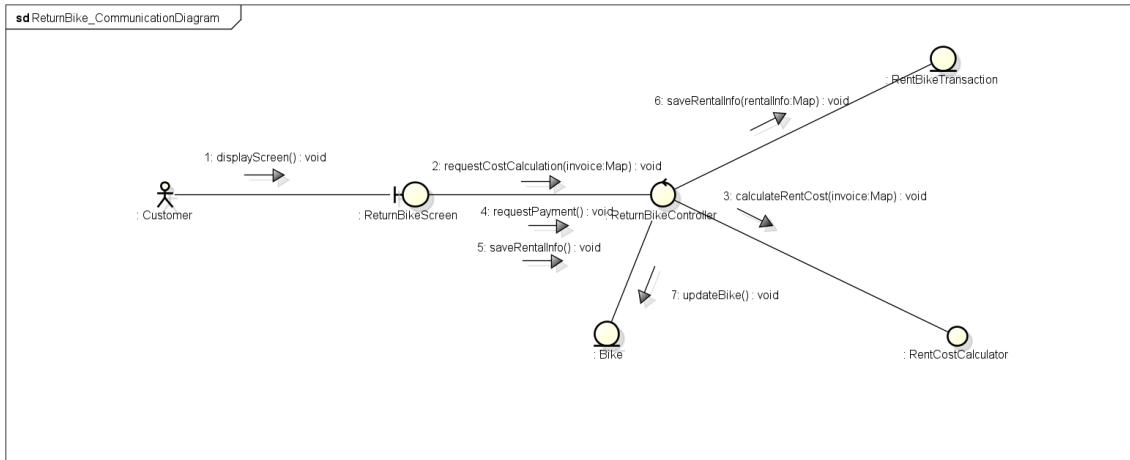
Thanh toán:



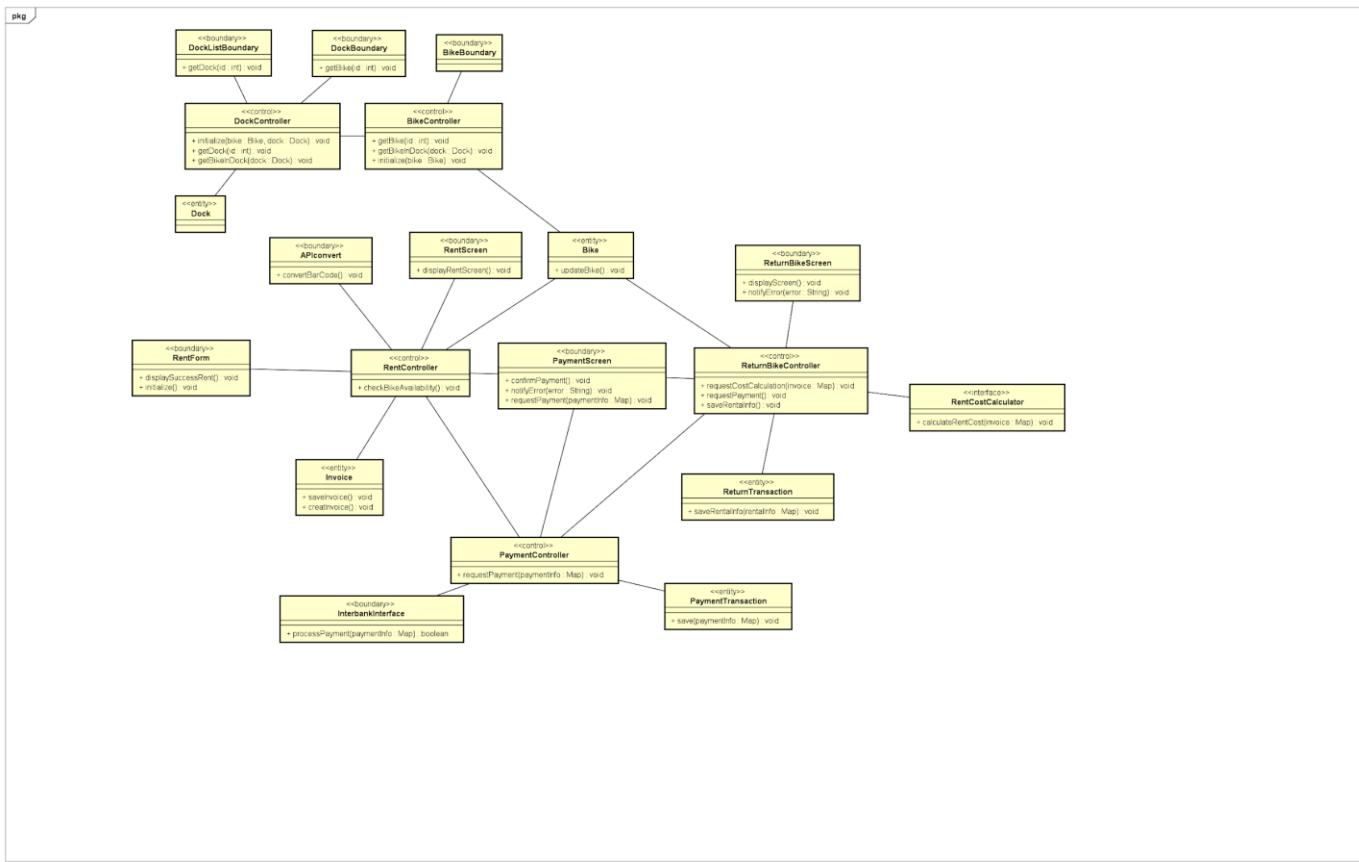
Thuê xe:



Trả xe:

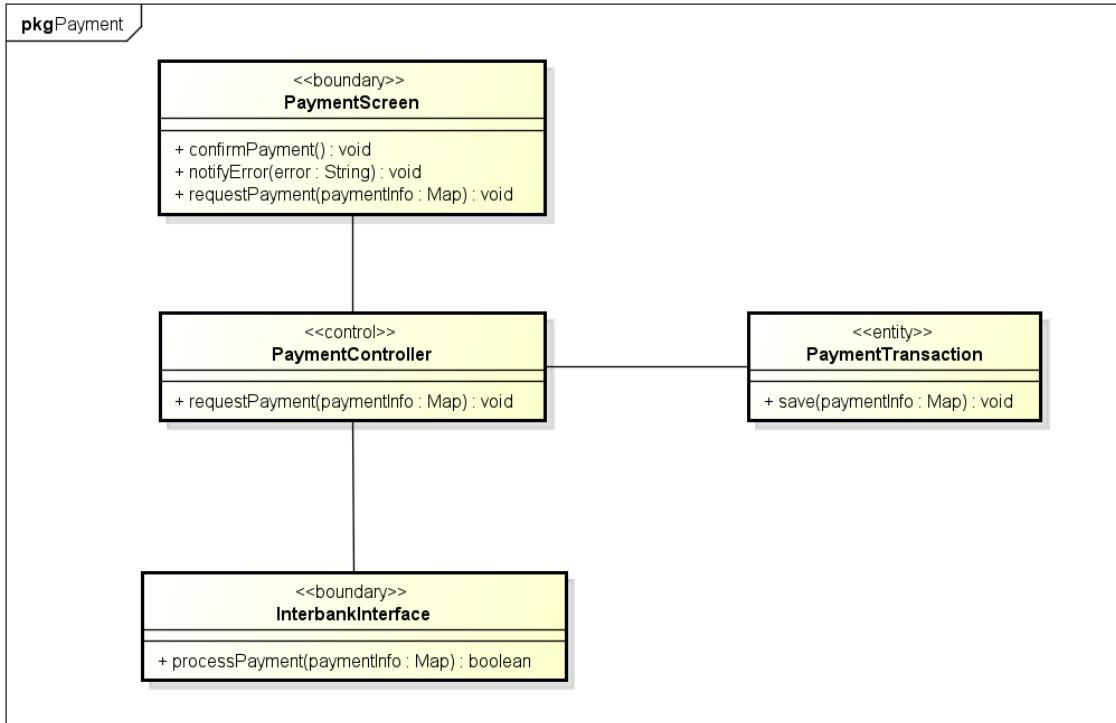


3.3 Biểu đồ phân tích lớp

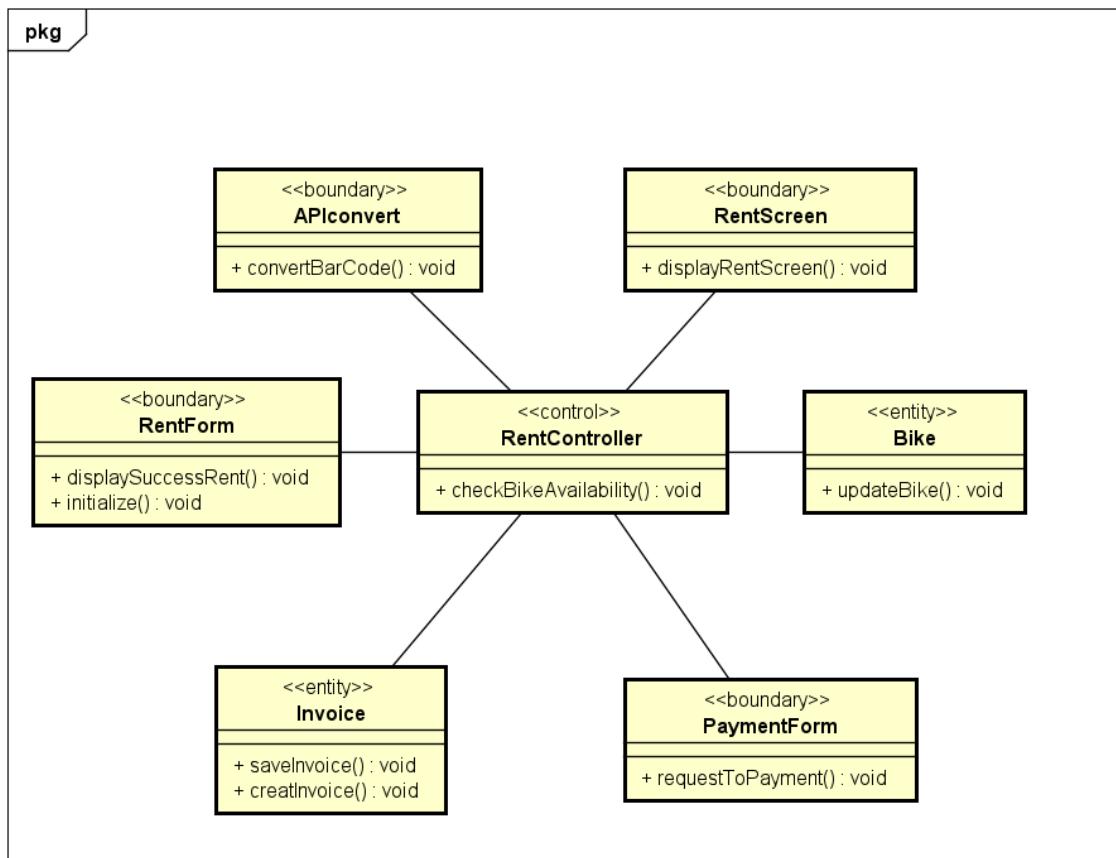


3.4 Biểu đồ thiết kế lớp

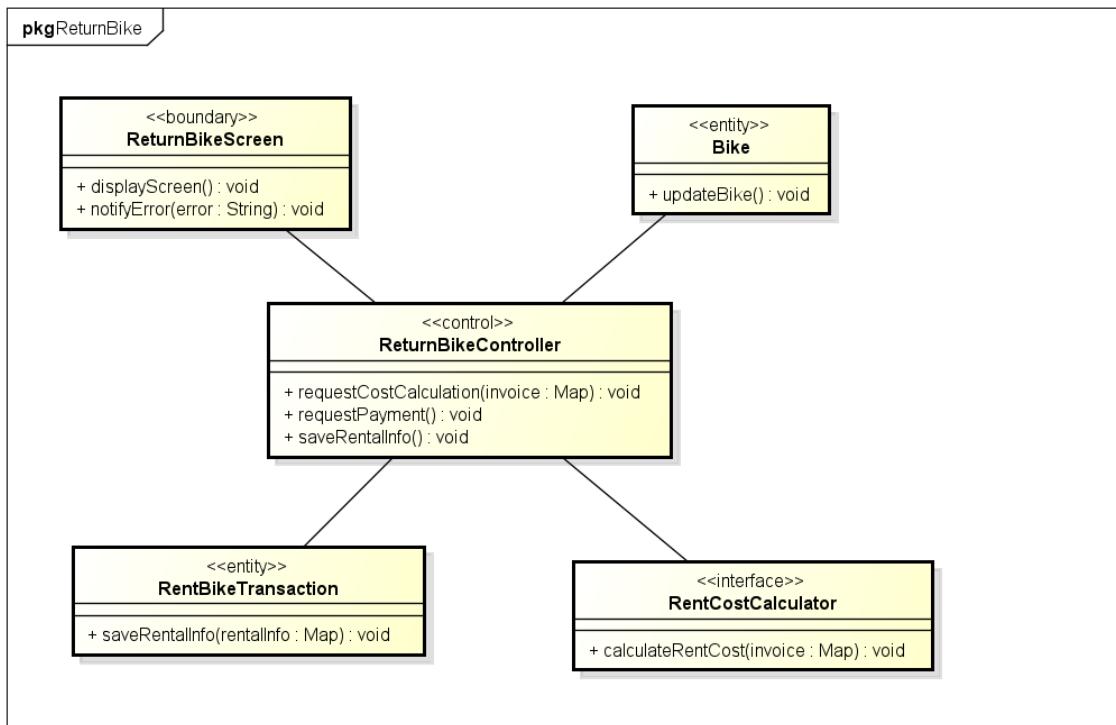
Thiết kế lớp cho chức năng Thanh toán:



Thiết kế lớp cho chức năng Thuê xe:



Thiết kế lớp cho chức năng Trả xe:



3.5 Security Software Architecture

Trong Project này, để tập trung vào mục tiêu môn học, nhóm tập trung vào những chức năng về thuê trả xe và thanh toán, không tập trung vào thực hiện các chức năng về bảo mật thông tin và xác thực người dùng

4 Thiết kế chi tiết

4.1 Thiết kế giao diện

4.1.1 Màn hình giao diện

1

Màn hình khi người dùng bắt đầu làm việc



2

Màn hình hiển thị danh sách bãi xe

The screenshot shows a software window titled "Ecobike Main Home". On the left, there's a logo with a bicycle icon and the word "ecobike". The main title is "Danh sách bãi xe". Below it, there are two entries for bike parking spots:

- Bãi xe 1**: Located at "Hai Bà Trưng, Hà Nội". It has 3 bikes and a capacity of 100. A "View" button is present.
- Bãi xe 2**: Located at "Cầu Giấy, Hà Nội". It has 3 bikes and a capacity of 100. A "View" button is present.

To the right of the list is a vertical orange sidebar containing three icons: a parking sign with a bicycle, an information icon (a circle with an 'i'), and a bicycle with a clock.

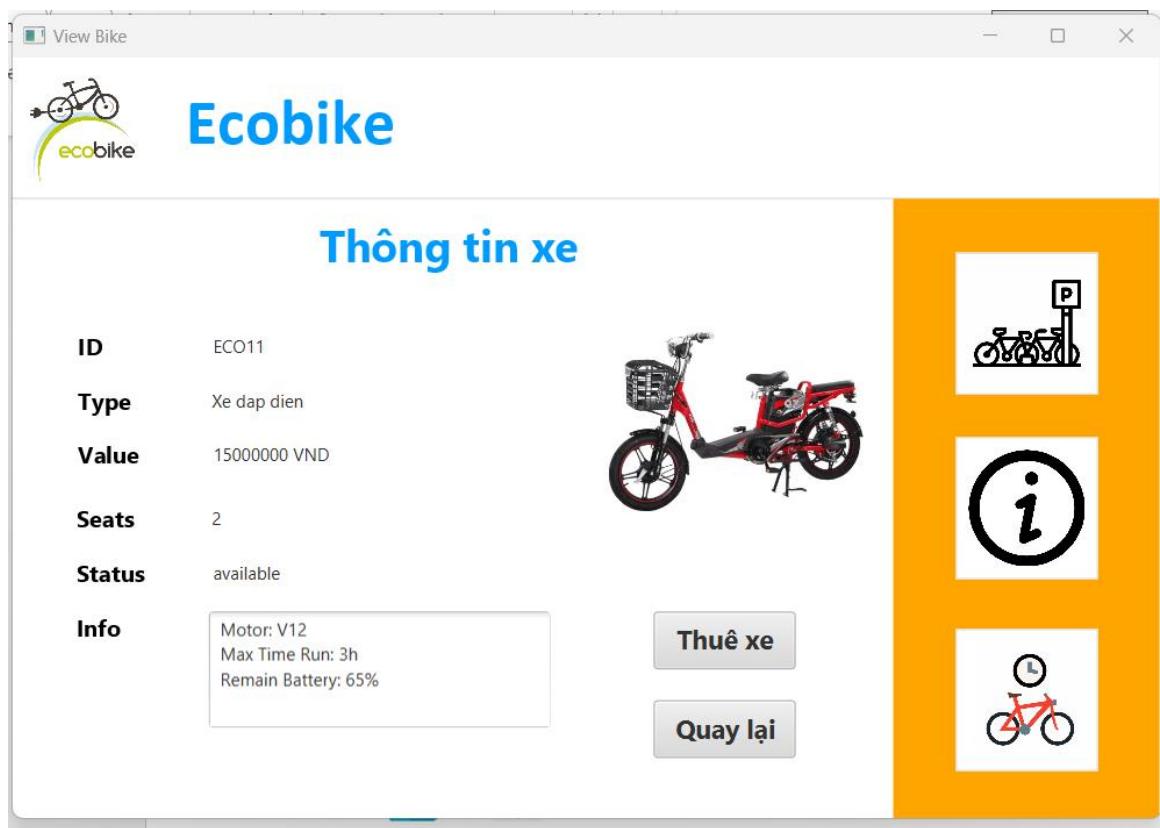
3

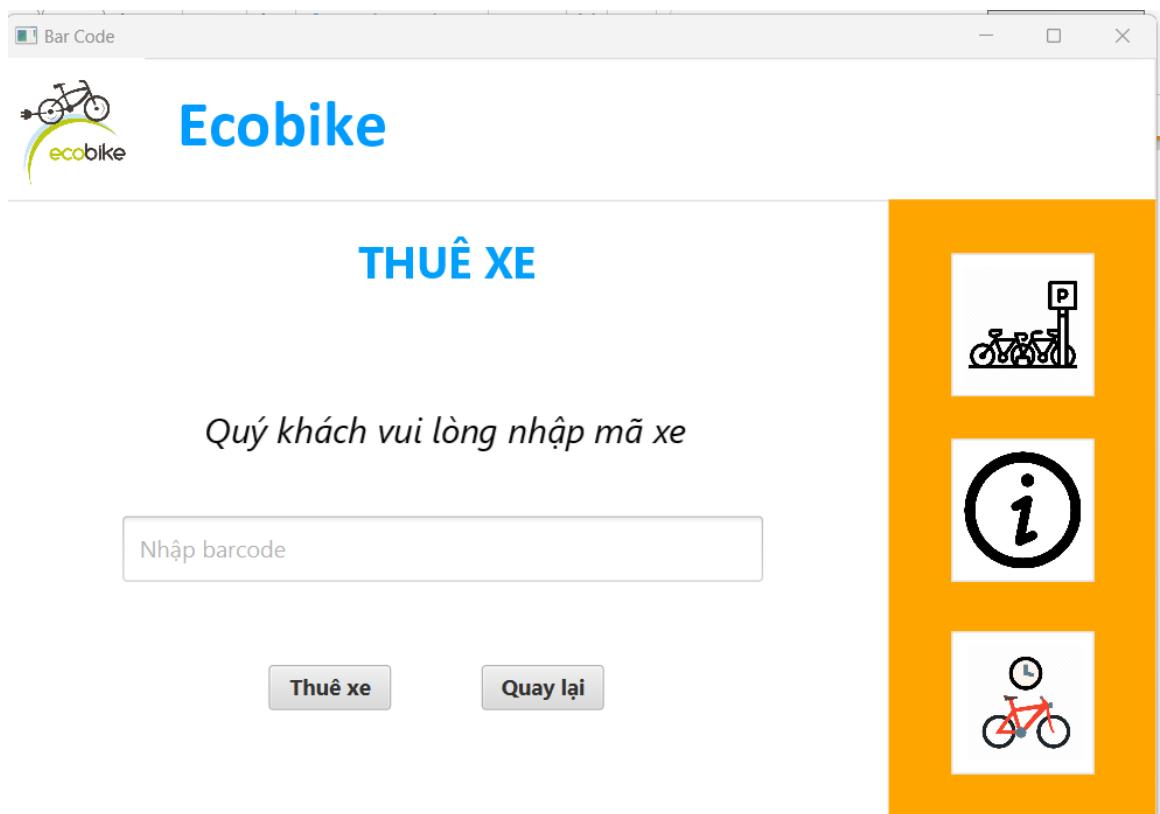
Màn hình hiển thị thông tin bãi xe:

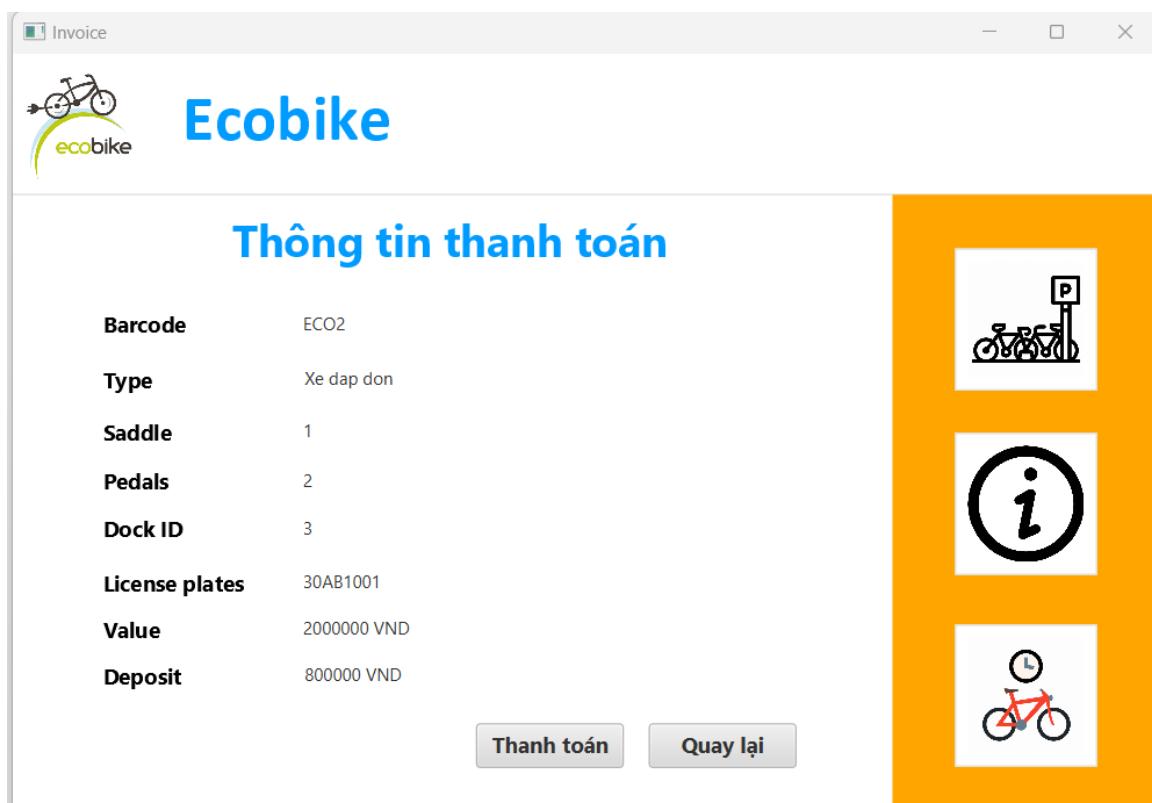
The screenshot shows a software window titled "Dock View". On the left, there's a logo with a bicycle icon and the word "ecobike". The main area is divided into two sections:

- Bãi xe 2**: Shows a photo of a bike parking area and some basic stats: Address: Cầu Giấy, Hà Nội; Bike: 3; Max: 100.
- Xe trong bãi**: Shows details for two bicycles:
 - A blue single-seater bicycle: Num of seat: 2, Type: Xe dap don, Value: 200000. A "Details" button is present.
 - A pink tandem bicycle: Num of seat: 2, Type: Xe dap doi, Value: 400000.

To the right of the list is a vertical orange sidebar containing three icons: a parking sign with a bicycle, an information icon (a circle with an 'i'), and a bicycle with a clock.

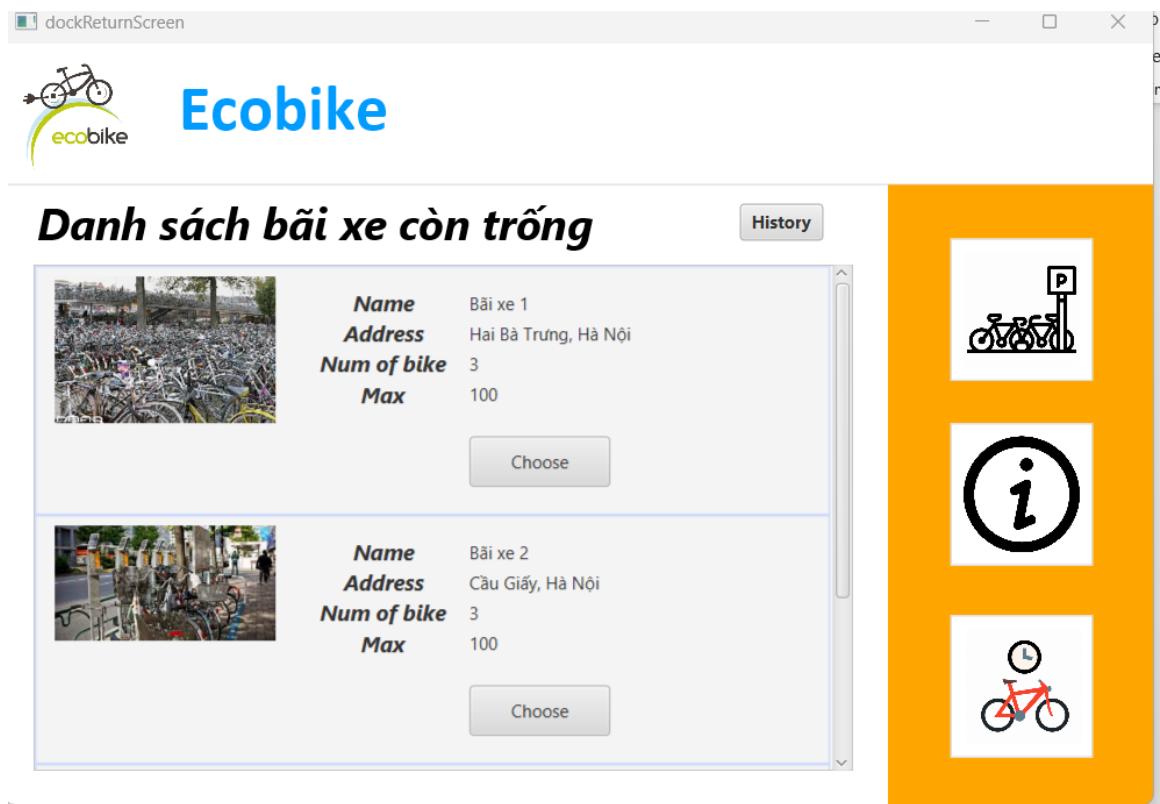








Màn hình chức năng trả xe:(Người dùng sẽ chọn bãi xe mà họ sẽ trả xe đang thuê)



9

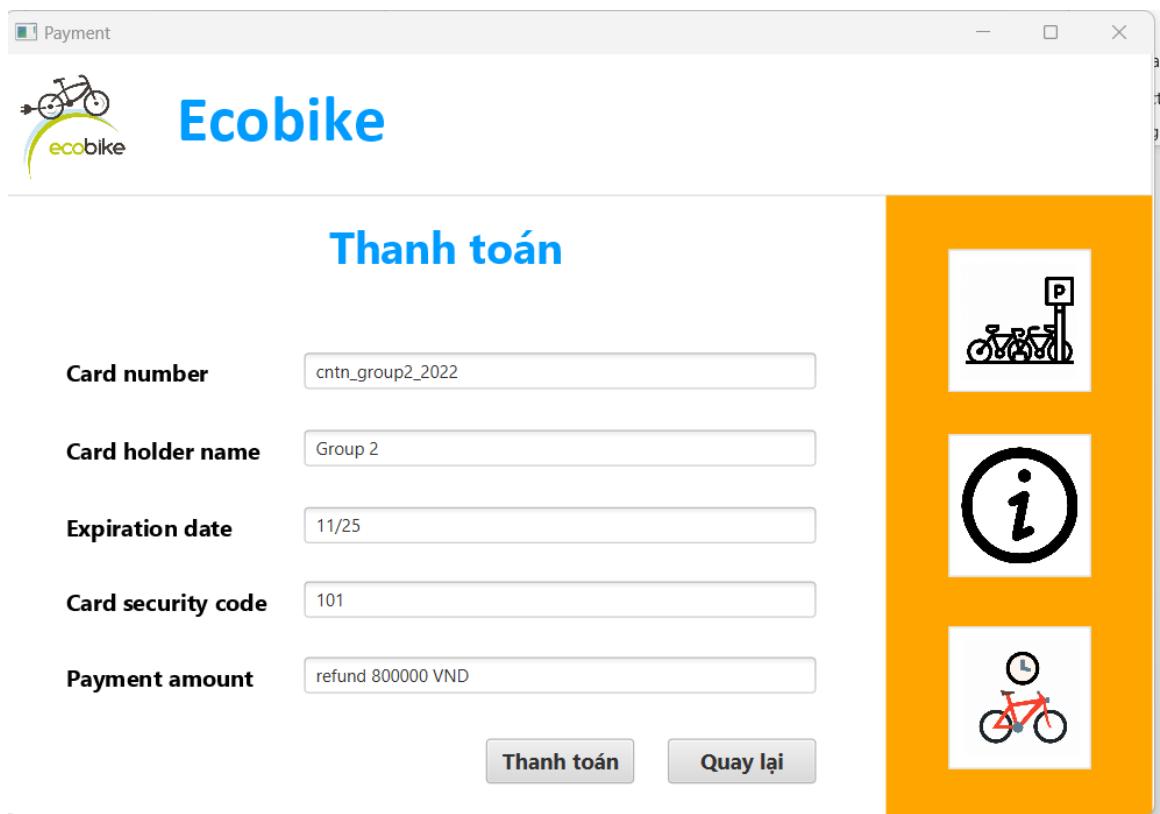
Màn hình hiển thị thông tin hóa đơn trả xe:

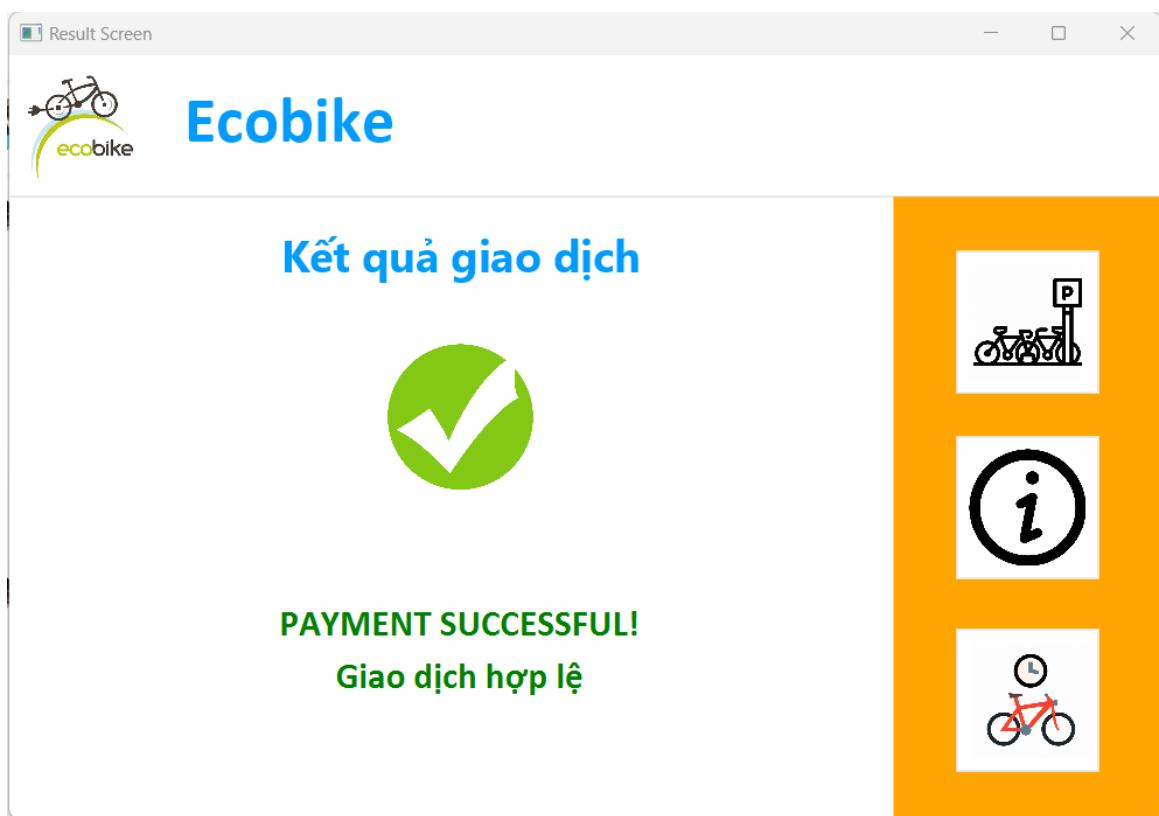
The screenshot shows a software window titled "Return Bike" with the "Ecobike" logo. The main content is a receipt titled "Hóa đơn trả xe". The receipt details are as follows:

Barcode	ECO2
Start Time	07/02/2023 20:11
End Time	07/02/2023 20:12
Deposit	800000
Rent Fee	0

Total: *Refund 800000 VNĐ*

At the bottom right are two buttons: "Thanh toán" and "Quay lại". To the right of the receipt is a vertical orange panel with three icons: a bicycle with a parking sign, an information icon (i), and a red bicycle.



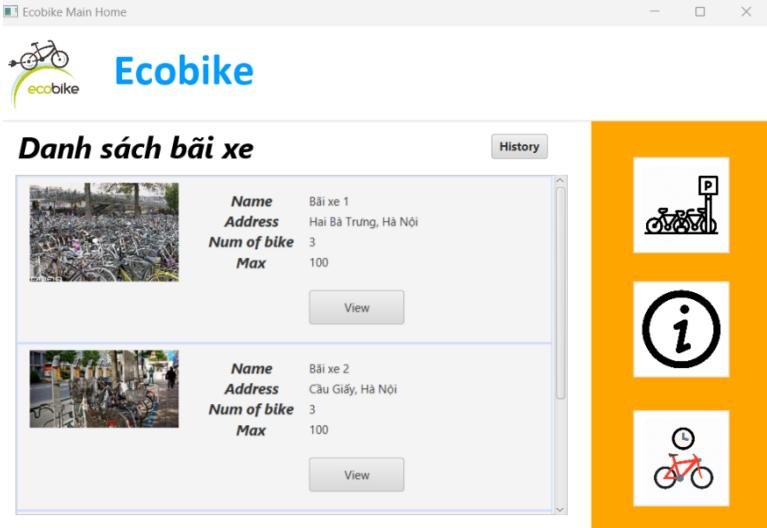


The screenshot shows a software application window titled "Transaction History". The title bar is at the top, and the main content area has a yellow header with the word "History" in large, dark blue font. Below the header is a table with six columns: "id", "bike code", "rent time", "return time", "deposit cost", and "rent cost". The table contains data for six transactions, with rows 7 through 12 being empty. The data is as follows:

id	bike code	rent time	return time	deposit cost	rent cost
1	ECO1	03/02/2023 11:10	03/02/2023 12:10	800000	16000
2	ECO4	03/02/2023 12:10	03/02/2023 12:13	800000	0
3	ECO1	03/02/2023 12:29	03/02/2023 12:35	800000	0
4	ECO10	03/02/2023 18:56	03/02/2023 18:56	6000000	0
5	ECO2	04/02/2023 13:10	04/02/2023 13:11	800000	0
6	ECO2	07/02/2023 20:11	07/02/2023 20:12	800000	0

4.1.2 Đặc tả thiết kế giao diện

Đặc tả màn hình ViewDock

Capstone		Date of creation	Approved by	Review ed by	Person in charge
Screen specification	ViewDock				
		Control	Operation	Function	
		Khu vực hiển thị logo và tên app	Khởi tạo	Hiển thị logo chương trình ecobike và tên apps ở trên cùng màn hình	
		Khu vực hiển thị thông tin bãi xe và xem chi tiết xe trong bãi.	Khởi tạo	Hiển thị thông tin bãi xe và xem chi tiết xe trong bãi.	
		Khu vực menu	Click	Thực hiện trở về màn hình Home, xem xe đang thuê, thông tin card.	

Đặc tả màn hình ViewBike

Capstone		Date of creation	Approved by	Review ed by	Person in charge
Screen specification	ViewBike				
		Control	Operation	Function	
		Khu vực hiển thị logo và slogan	Khởi tạo	Hiển thị logo chương trình ecobike và slogan ở trên cùng màn hình	
		Khu vực hiển thị thông tin xe	Khởi tạo	Hiển thị thông tin về mã xe, loại xe, giá trị,...	

	Khu vực hiển thị hình ảnh xe	Khởi tạo	Hiển thị hình ảnh của xe.
	Nút “Thuê xe”	Click	Gửi yêu cầu thuê xe.

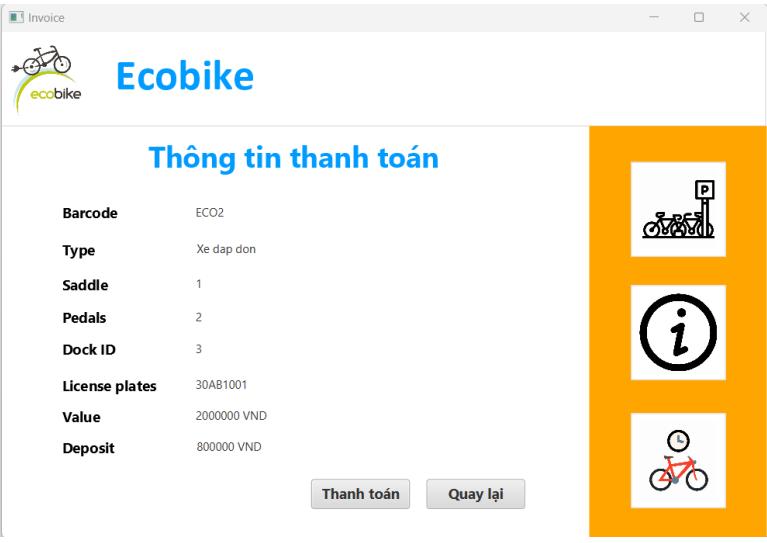
Đặc tả màn hình home

Capstone		Date of creation	Approved by	Review ed by	Person in charge
Screen specification	Home				
	Control	Operation	Function		
	Khu vực hiển thị logo và tên app	Khởi tạo	Hiển thị logo chương trình ecobike và tên app ở trên cùng màn hình		
	Khu vực hiển thị danh sách bãi xe	Khởi tạo	Hiển thị một số thông tin cơ bản của bãi xe		
	Nút “View” để xem bãi xe	Click	Chuyển đến ViewDock để xem chi tiết bãi xe		
	Nút history	Click	Hiển thị màn hình lịch sử giao diện		

Màn hình RentBikeScreen

Capstone		Date of creation	Approved by	Reviewed by	Person in charge
Screen specification	RentBikeScreen				
		Control	Operation	Function	
	Khu vực nhập mã xe	Nhập từ bàn phím		Người dùng nhập từ bàn phím mã xe muốn thuê	
	Khu vực menu	Click		Thực hiện trả về màn hình Home, xem xe đang thuê, thông tin card	
	Nút “Thuê xe” và nút “quay lại”	Click		Người dùng xác nhận muốn thuê xe với mã xe như đã điền để hệ thống kiểm tra hoặc quay lại trang chủ	

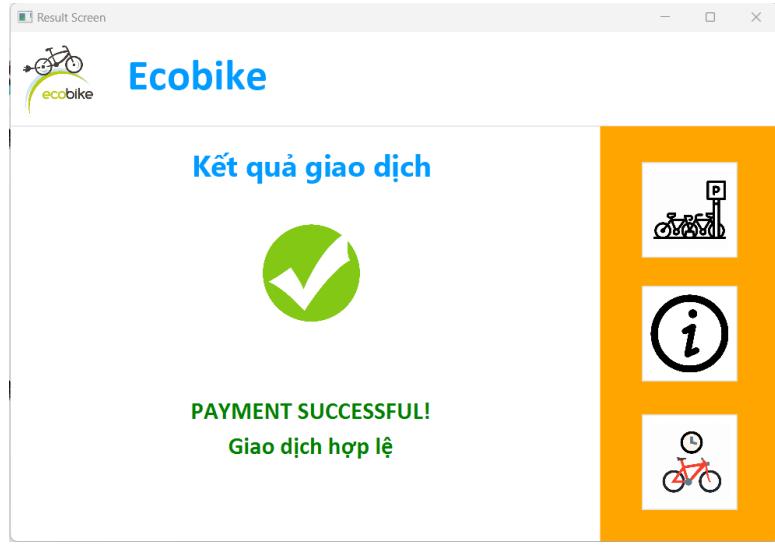
Màn hình RentBikeInfo

Capstone		Date of creation	Approved by	Review ed by	Person in charge
Screen specification	RentBikeInfoScreen				
		Control	Operation	Function	
 <p>Barcode: ECO2 Type: Xe đạp điện Saddle: 1 Pedals: 2 Dock ID: 3 License plates: 30AB1001 Value: 2000000 VND Deposit: 800000 VND</p> <p>Thanh toán Quay lại</p>					

Màn hình Payment

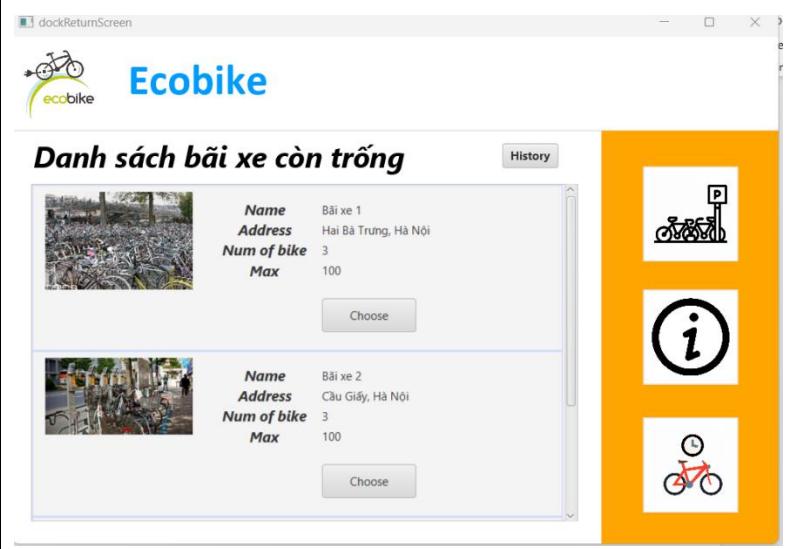
Capstone		Date of creation	Approved by	Reviewed by	Person in charge
Screen specification	Payment				
		Control	Operation	Function	
	Khu vực thông tin thanh toán	Nhập từ bàn phím	Người dùng nhập từ bàn phím mã thông tin thanh toán		
	Khu vực menu	Click	Thực hiện trở về màn hình Home, xem xe đang thuê, thông tin card		
	Nút “Thanh toán”	Click	Người dùng xác nhận thanh toán trả tiền đặt cọc để thuê xe.		

Màn hình thanh toán thành công

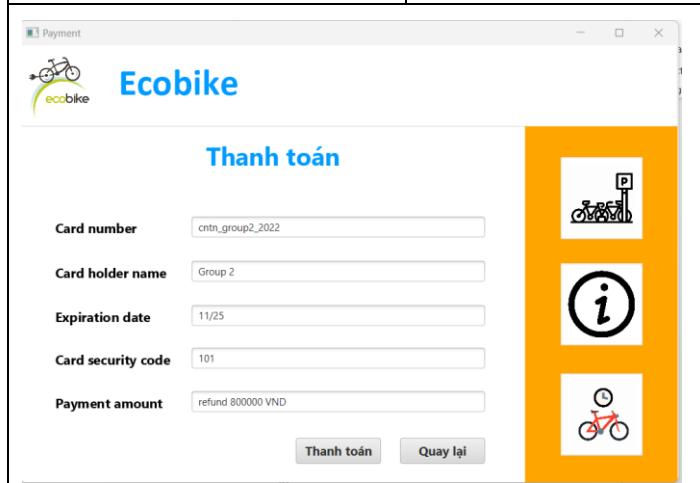
Capstone		Date of creation	Approved by	Reviewed by	Person in charge
Screen specification	RentBikeSuccessful				
		Control	Operation	Function	
		Khu vực hiển thị thông tin cơ bản của xe được thuê	Khởi tạo	Hiển thị thông tin cơ bản của xe được thuê cho người dùng nắm bắt	
		Khu vực menu	Click	Thực hiện trở về màn hình Home, xem xe đang thuê, thông tin card	
		Nút “OK”	Click	Người dùng xác nhận hoàn tất giao dịch thuê xe thành công	

Màn hình trả xe

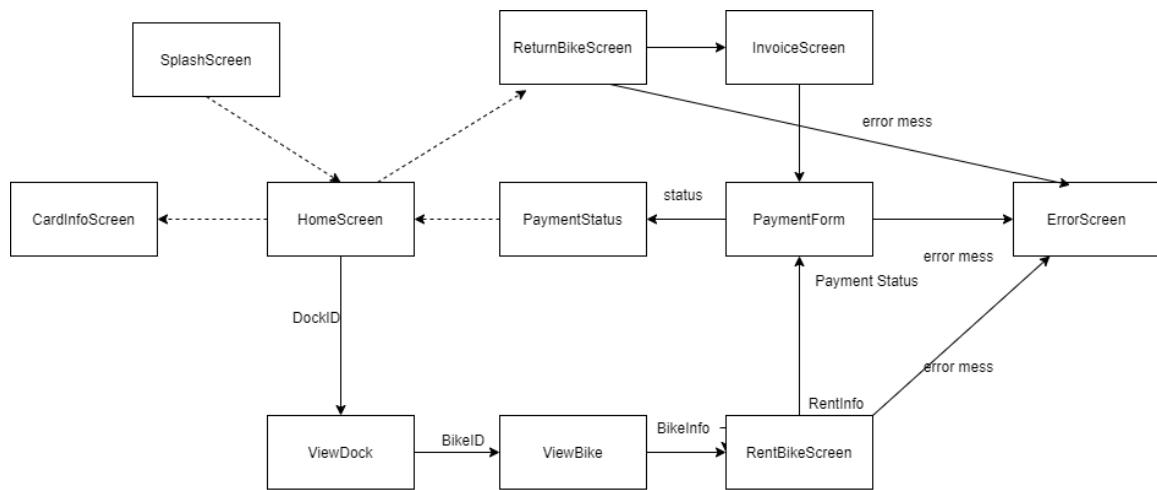
Capstone		Date of creation	Approved by	Reviewed by
Screen specification	Return Bike			
		Control	Operation	Function
		Khu vực hiển thị logo và slogan	Initial	Hiển thị logo chương trình ecobike ở trên cùng màn hình
		Khu vực hiển thị danh sách bãi xe	Initial	Hiển thị hình ảnh bãi xe, tên bãi xe để chọn bãi dùng để trả xe

	Nút chọn bãi xe để trả xe	Click	Chuyển đến trả xe
---	---------------------------	-------	-------------------

Màn hình hoá đơn trả xe

Capstone		Date of creation	Approved by	Reviewed by	Person in charge
Screen specification	Return Bike				
	Control Khu vực hiển thị logo và slogan	Operation Initial	Function Hiển thị logo chương trình ecobike và slogan ở trên cùng màn hình		
	Control Khu vực hiển thị thông tin hóa đơn	Operation Initial	Function Hiển thị thông tin của hóa đơn thuê xe của người dùng		
	Nút thanh toán	Click	Function Chuyển đến Payment		

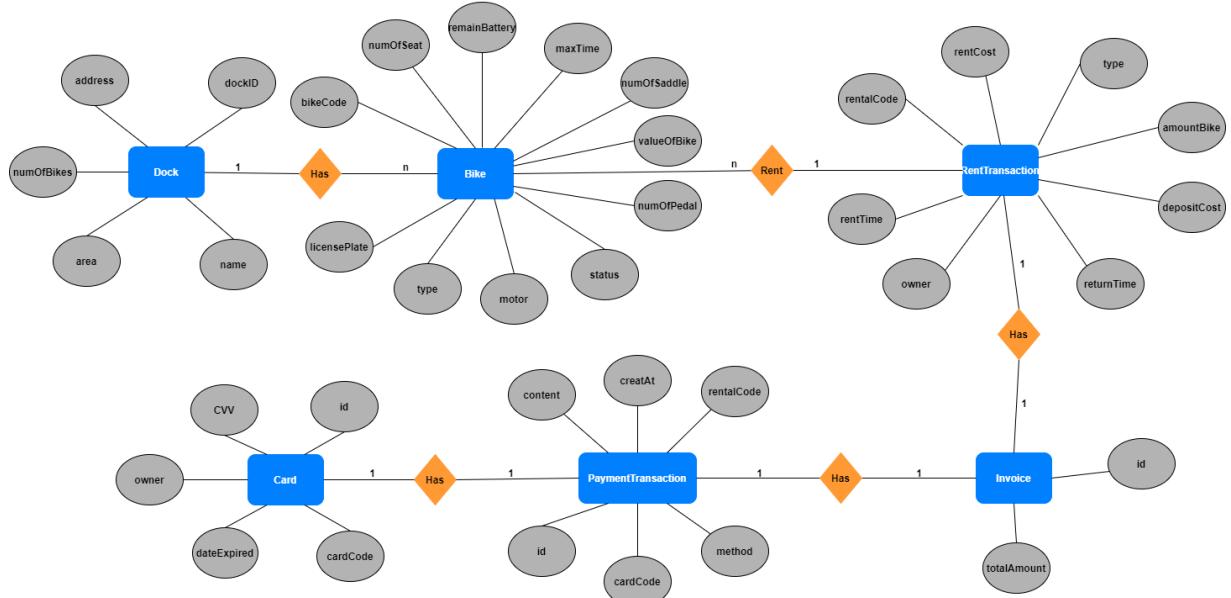
4.1.3 Biểu đồ dịch chuyển màn hình



4.2 Mô hình hóa dữ liệu

4.2.1 Mô hình hóa mức khai niệm

<E-R Diagram image and description of entities and relationships>



Các entity:

- Dock: Thực thể bãi gửi xe

- Bike: Thực thể xe đạp
- RentTransaction: Thực thể giao dịch thuê xe
- Invoice: Thực thể hóa đơn
- PaymentTransaction: Thực thể giao dịch thanh toán
- Card: Thực thể thẻ thanh toán

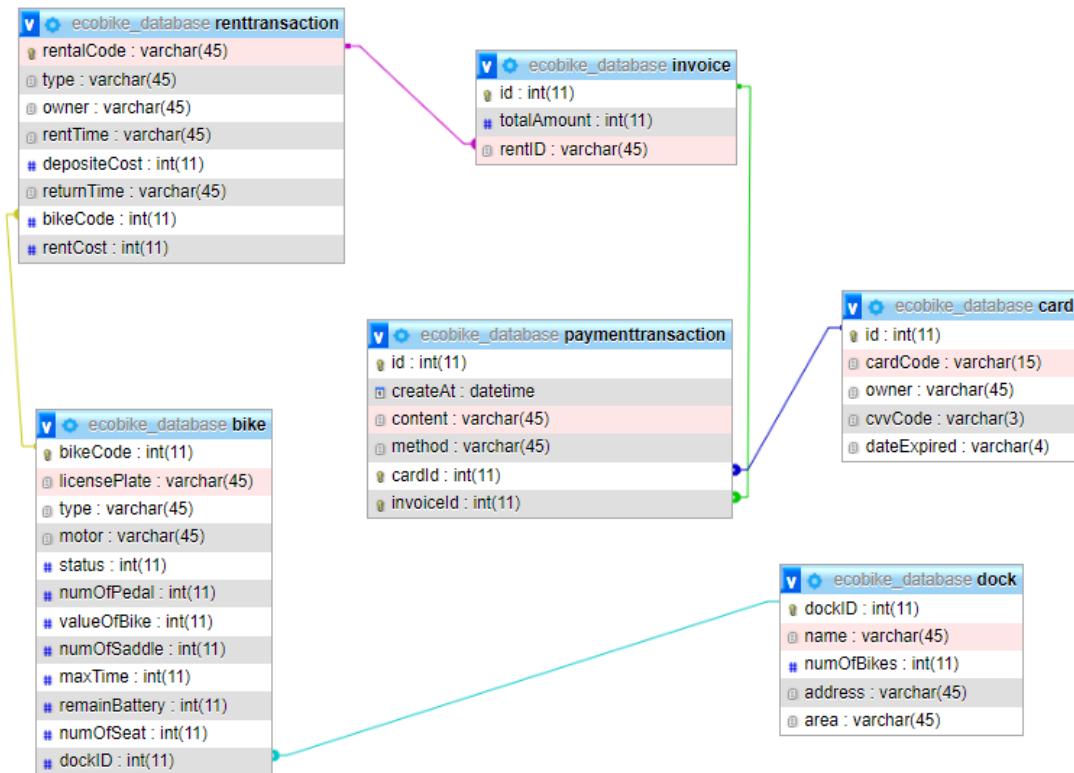
4.2.2 Thiết kế cơ sở dữ liệu

4.2.2.1 Hệ quản trị cơ sở dữ liệu

DBMS: SQLite

SQLite không chỉ nhẹ, nhanh, phổ biến, không cần cài đặt, có độ tin cậy cao, đầy đủ tính năng, và là một engine cơ sở dữ liệu SQL; ngoài ra còn ổn định, không phụ thuộc vào nền tảng, tương thích ngược và hỗ trợ lâu dài

4.2.2.2 Mô hình logic dữ liệu



4.2.2.3 Mô hình dữ liệu vật lý

Dock:

#	PK	FK	Tên cột	Kiểu dữ liệu	Giá trị mặc định	Bắt buộc ?	Mô tả
---	----	----	---------	--------------	------------------	------------	-------

1	x		dockID	int(11)	NULL	Có	ID của bãi xe
2			name	varchar(45)	NULL	Có	Tên bãi xe
3			numOfBikes	int(11)	NULL	Có	Số xe trong bãi
4			address	varchar(45)	NULL	Có	Địa chỉ, khu vực của bãi xe
5			area	varchar(45)	NULL	Có	Diện tích bãi xe

Card:

#	PK	FK	Tên cột	Kiểu dữ liệu	Giá trị mặc định	Bắt buộc ?	Mô tả
1	x		id	int(11)	NULL	Có	ID của thẻ
2			cardCode	varchar(15)	NULL	Có	Mã thẻ
3			owner	varchar(45)	NULL	Có	Tên chủ thẻ
4			cvvCode	varchar(45)	NULL	Có	Mã CVV
5			dateExpired	datetime	NULL	Có	Ngày hết hạn

RentTransaction:

#	PK	FK	Tên cột	Kiểu dữ liệu	Giá trị mặc định	Bắt buộc ?	Mô tả
1	x		rentalCode	varchar(45)	NULL	Có	Mã thuê xe
2			type	varchar(45)	NULL	Có	Loại xe
3			owner	varchar(45)	NULL	Có	Người thuê xe
4			rentTime	datetime	NULL	Có	Thời gian thuê
5			depositCode	int(11)	NULL	Có	Tiền đặt cọc
6		x	bikeCode	int(11)	NULL	Có	Mã xe thuê
7			returnTime	datetime	NULL	Có	Thời gian trả xe
8			rentCost	int(11)	NULL	Có	Tiền thuê xe

Bike:

#	PK	FK	Tên cột	Kiểu dữ liệu	Giá trị mặc định	Bắt buộc ?	Mô tả
1	x		bikeCode	int(11)	NULL	Có	Mã xe
2			licensePlate	varchar(45)	NULL	Có	Biển số xe
3			type	varchar(45)	NULL	Có	Loại xe
4			motor	varchar(45)	NULL	Có	Loại động cơ
5			status	int(11)	NULL	Có	Trạng thái xe (0: đã được thuê, 1: chưa được thuê)
6			numOfPedal	int(11)	NULL	Có	Số bàn đạp
7			value	int(11)	NULL	Có	Giá trị của xe
8			numOfSaddle	int(11)	NULL	Có	Số yên xe
9			maxTime	int(11)	NULL	Có	Thời gian đi được tối đa (tính bằng giờ đối với xe điện)
10			remainBattery	int(11)	NULL	Có	Lượng pin còn lại (đối với xe điện)
11			numOfSeat	int(11)	NULL	Có	Số ghế ngồi
12		x	dockID	int(11)	NULL	Có	ID bãi đỗ xe

Invoice:

#	PK	FK	Tên cột	Kiểu dữ liệu	Giá trị mặc định	Bắt buộc ?	Mô tả
1	x		id	int(11)	NULL	Có	ID của hóa đơn
2			totalAmount	int(11)	NULL	Có	Tổng chi phí
3		x	rentalCode	int(11)	NULL	Có	Mã thuê xe

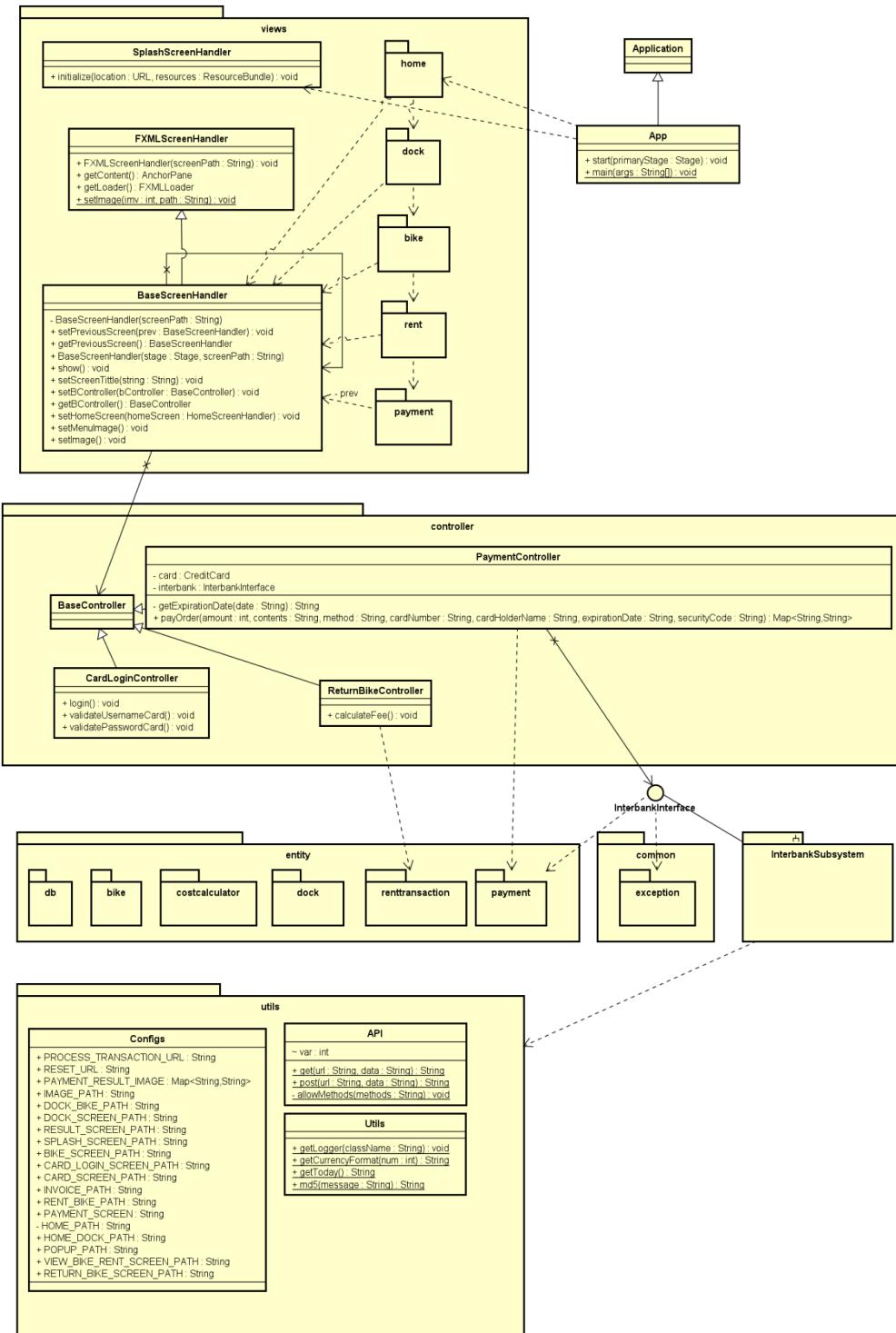
PaymentTransaction:

#	PK	FK	Tên cột	Kiểu dữ liệu	Giá trị mặc định	Bắt buộc ?	Mô tả
1	x		id	int(11)	NULL	Có	ID của giao dịch thanh toán
2			createAt	datetime	NULL	Có	Thời gian tạo giao dịch
3			content	varchar(45)	NULL	Không	Nội dung giao dịch
4			method	varchar(45)	NULL	CreditCard	Phương thức thanh toán
5		x	cardID	int(11)	NULL	Có	ID của thẻ
6		x	invoiceID	int(11)	NULL	Có	ID của hóa đơn

4.3 Non-Database Management System Files

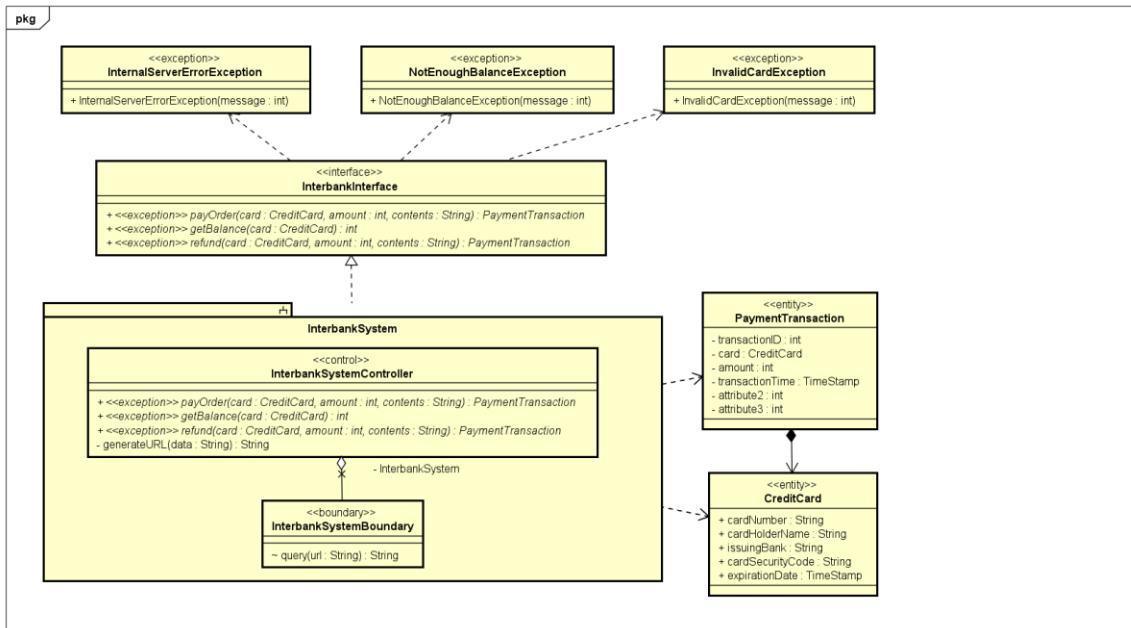
4.4 Thiết kế lớp

4.4.1 Biểu đồ lớp tổng quan



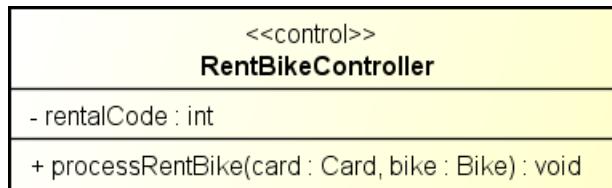
4.4.2 Biểu đồ lớp tổng quan

1 Biểu đồ Interbank Subsystem



4.4.3 Thiết kế chi tiết cho từng lớp

4.4.3.1 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

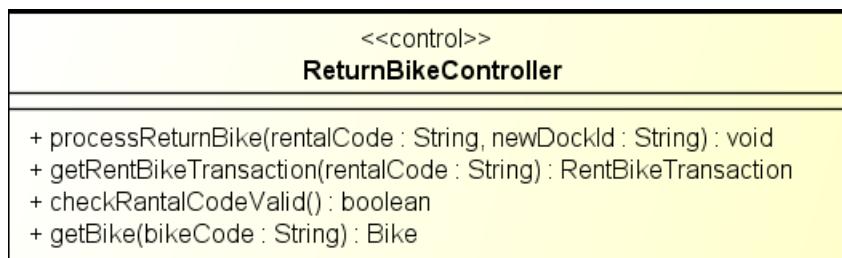
Operation

#	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

4.4.3.2 Class *ReturnBikeController*



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 bến xe mà người dùng trả xe về đó (có thể trả ở bến khác bến mượn xe)

bikeCode: Mã đại diện cho xe người dùng thuê

Method:

Không

4.4.3.3 Class PaymentController

<<control>> PaymentController	
- 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

4.4.3.4 Class BikeController



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

Attribute: Không

Operation

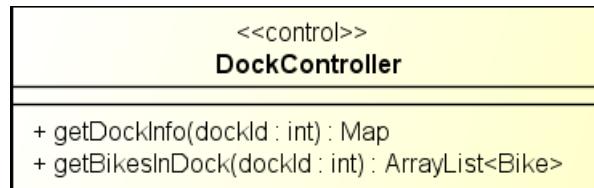
#	Name	Return type	Description
1	getBikeInfo	Map	Lấy ra thông tin của xe
2	setBikeState	NULL	Thiết lập trạng thái của xe

Parameters:

bikeId: Mã xe

state: Trạng thái của xe

4.4.3.5 Class 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

4.4.3.6 Class RentBikeController

RentBikeTransaction <i><<entity>></i>	
- rentalCode : String - bikeCode : int - type : String - owner : String - rentTime : String - returnTime : String - deposit : int - cost : int	
+ saveRentBikeTransactionInfo() : void + getRentBikeTransactionInfo() : Map	

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

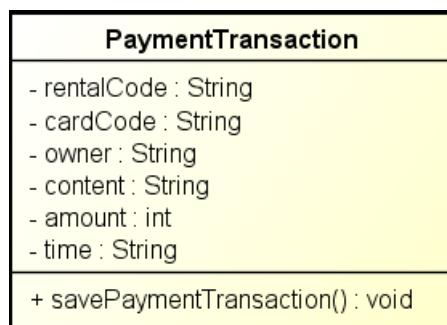
Operation

#	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

4.4.3.7 Class PaymentTransaction



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

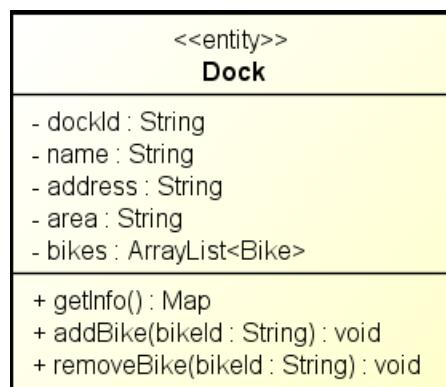
Operation

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

Parameters:

Không

4.4.3.8 Class 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:

bikeId: Mã định danh của xe

4.4.3.9 Class Bike

<<entity>> Bike	
- bikeCode : String - available : boolean - cost : int - licensePlate : String - numPedals : int - numSaddle : int - numBicycleSeat : int - dockId : String	
+ updateInfo(bikeCode : String, available : boolean, dockID : boolean, cost : int) : void + getInfo() : Map	

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:

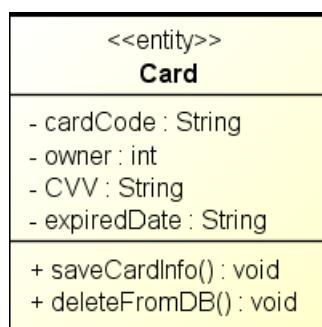
bikeCode: Mã định danh của xe

available: Xe còn sẵn để thuê hay không

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

cost: Giá thành xe

4.4.3.10 Class Card



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

5. Design Considerations

1. Goals and Guidelines

Mục tiêu:

- Mang lại trải nghiệm tốt cho người dùng
- Thời gian phản hồi của hệ thống thường ở mức 1s và chậm nhất là mức 2s (Khi thực hiện giao dịch)
- Mang lại cảm giác như đang dùng một sản phẩm thực tế

Nguyên tắc:

- Tuân thủ theo các nguyên tắc về convention (naming convention, coding convention, ...), các nguyên lý về OOP
- Các methods, class có javadoc đầy đủ
- Xem xét đến việc áp dụng các design pattern và chỉnh sửa, cải tiến để tuân theo các nguyên lý thiết kế, đảm bảo tính coupling và cohesion ở mức vừa phải

2. Architectural Strategies

- Programming Language: Java
- Database: SQLite
- Library: JavaFX, JUnit, OkHTTP, ...
- Tái sử dụng code: Tái sử dụng subsystem Interbank của Project AIMS để kết nối, thực hiện giao dịch.
- Kế hoạch phát triển trong tương lai: Thiết kế thêm role của Admin với chức năng thêm xe, xóa xe, thêm bãi xe, kiểm soát giao dịch, quản lý người dùng, ... làm cho hệ thống hoàn thiện và có thể áp dụng trong thực tế
- Hệ thống có khả năng chống chịu lỗi và recovery

3. Coupling and Cohesion

Trong quá trình triển khai, nhóm nhận ra một số vấn đề về mặt coupling và cohesion như sau:

- **Coupling:**

- Content Coupling: None
- Commons Coupling: None
- Control Coupling: None
- Stamps Coupling:

Related modules	Description	Improvement
CostCalculator	Chỉ cần tham số tiền của xe nhưng truyền vào cả xe	Thay tham số Bike bằng costScale

- Data Coupling: None

- **Cohesion:**
- **Coincidental Cohesion:**

Related modules	Description	Improvement
utils.Utils	Tất cả các hàm chức năng đặt trong một module	Tách ra thành các thành phần riêng biệt
utils.Configs	Có quá nhiều thành phần liên quan đến các module khác nhau đặt trong configs	Tách ra thành các lớp riêng biệt

4. Design Principles

Dánh giá project theo SOLID:

- *Single Responsibility Principle:*

•

Related modules	Description	Improvement
PaymentController, CardLoginController	Còn chứa các hàm validate nằm bên trong	Tách ra class Validate

- *Open/Closed Principle: None*
- *Liskov Substitution Principle: None*
- *Interface Segregation: None*
- *Dependency Inversion:*

•

Related modules	Description	Improvement
PaymentTransaction, Card	Đang bị phụ thuộc chặt chẽ vào nhau, khó mở rộng khi thêm nhiều hình thức thanh toán	Tạo một lớp abstract PaymentCard và để PaymentCard và CreditCard kế thừa

5. Design Patterns

- *Strategy Pattern: Sử dụng strategy pattern ở trong costcalculator, class Strategy1 implements Interface RentCostCalculator*
- *Façade Pattern: Sử dụng interbank interface để tương tác đến Interbank, giảm thiểu tính phức tạp khi tương tác với Interbank*

