

**ĐẠI HỌC BÁCH KHOA HÀ NỘI**  
**Trường công nghệ thông tin và truyền thông**



**Báo cáo**  
**Phân tích và thiết kế hệ thống EcoBikeRental**

Nhóm 6:

Hà Đức Tuấn – MSSV: 20190072

Nguyễn Văn Trung – MSSV: 20190071

Trần Quốc Việt – MSSV: 20190075

Giảng viên hướng dẫn: TS. Nguyễn Thị Thu Trang

Hà Nội, ngày 07 tháng 02 năm 2022

# Mục lục

<b>I. Introduction</b>	<b>4</b>
1. Objective	4
2. Scope	4
3. References	4
<b>II. Overall Description</b>	<b>5</b>
1. General Overview	5
2. Assumptions/Constraints/Risks	5
<b>III. Usecase</b>	<b>6</b>
1. Use Case diagram	6
2. Use Case specification	6
UC001 “Tìm kiếm xe”	6
<b>Biểu đồ hoạt động</b>	<b>8</b>
UC002 “Xem thông tin bãi xe”	8
UC003 “Thuê xe”	10
UC004 “Trả xe”	13
<b>IV. System Architecture and Architecture Design</b>	<b>16</b>
1. Architectural Patterns	16
2. Sequence Diagrams	16
3. Analysis Class Diagrams	18
<b>V. Detailed Design</b>	<b>19</b>
1. User Interface Design	19
2. Screen Transition Diagrams	20
3. Screen Specifications	21
<b>V. Data Modeling</b>	<b>27</b>
1. Conceptual Data Modeling	27
2. Database Design	28
a) Mô hình hoá dữ liệu (Logical data model)	28
b) Mô hình hoá dữ liệu vật lý (Physycal data model)	28
3. Class Design	32

Class Diagram	32
4. Class Specification	33
<b>VI. Design Considerations</b>	<b>38</b>
1. Goals and Guidelines	38
2. Architectural Strategies	38

## **I. Introduction**

### **1. Objective**

Tài liệu này mô tả phần thiết kế phần mềm sau bước phân tích ở tài liệu srs. Tài liệu được sử dụng cho programmers, testers, maintainers, systems integrators, vv. Nó bao gồm việc thiết kế chi tiết cho kiến trúc, thiết kế giao diện và thiết kế lớp cho từng chức năng của hệ thống, cũng như việc thiết kế cơ sở dữ liệu của cả hệ thống để từ đó người đọc sẽ có cái nhìn rõ ràng hơn về phần mềm cần xây dựng và nó sẽ là tài liệu chính thức để từ đó những người xây dựng phần mềm có thể xây dựng nên phần mềm dựa vào tài liệu này.

### **2. Scope**

Ứng dụng EcoBikeRental giả lập cho việc khách hàng có thể thuê và trả xe theo mô hình như trên mà không xét đến các chức năng như xác thực người dùng, chỉ quan tâm đến các chức năng liên quan đến thuê xe và trả xe.

### **3. References**

Tài liệu tham khảo bao gồm: javaFX doc, tài liệu được cung cấp trong học phần thiết kế và xây dựng phần mềm.

## **II. Overall Description**

### **1. General Overview**

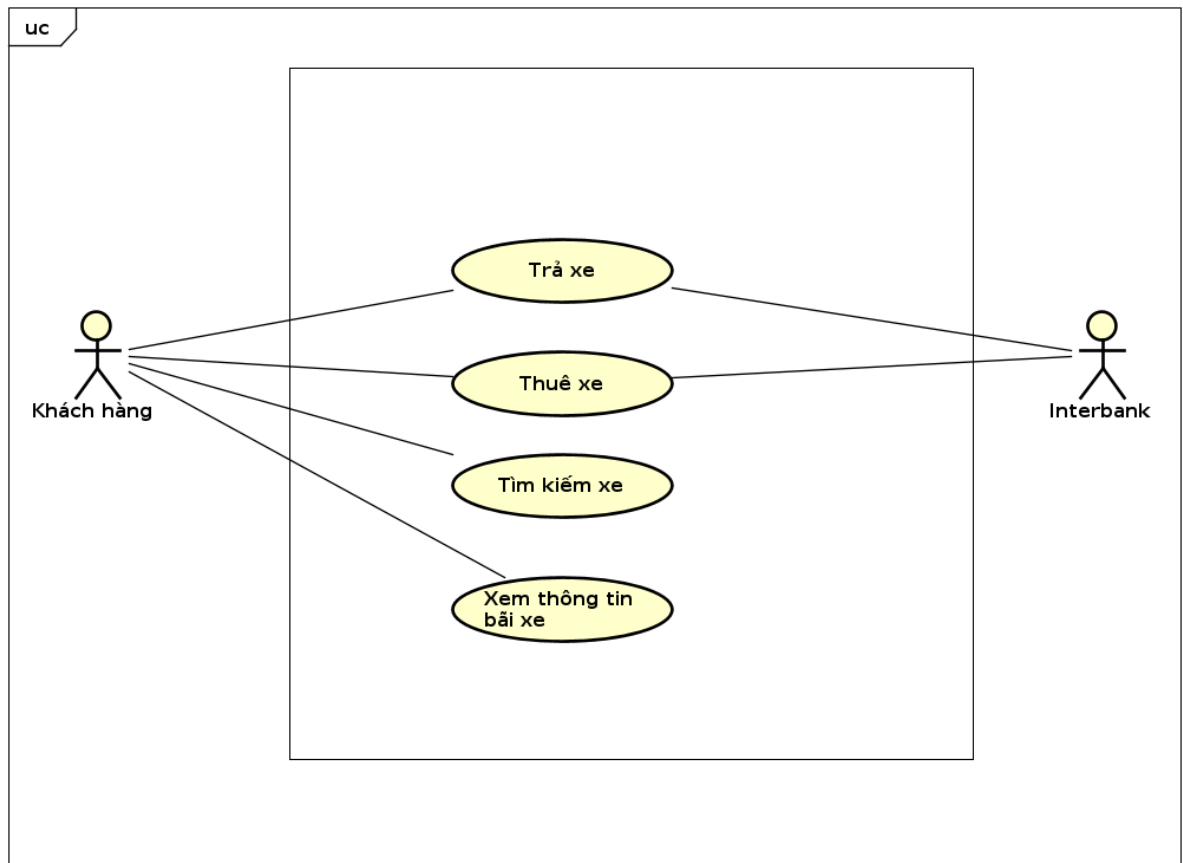
Phần mềm được thiết kế theo thiết kế 3 tầng, dùng trên desktop. Thiết kế này không thích hợp với người dùng trong thực tế vì không ai mang PC hay laptop đi để thuê xe đạp. Bên cạnh đó, laptop hay PC cũng không có chức năng quét barcode hay có mạng liên tục. Tuy nhiên, với khuôn khổ của bài tập, phần mềm được thiết kế phù hợp để demo.

### **2. Assumptions/Constraints/Risks**

Không có.

### III. Usecase

#### 1. Use Case diagram



#### 2. Use Case specification

##### ***UC001 “Tìm kiếm xe”***

*Mã usecase*

UC001

*Mô tả*

Tìm kiếm thông tin xe

*Tác nhân*

Người dùng

*Tiền điều kiện*

Không

*Luồng sự kiện chính*

1. Người dùng chọn “Tìm kiếm xe”
2. Hệ thống hiển thị màn hình “Tìm kiếm xe”
3. Hệ thống yêu cầu người dùng nhập các thông tin tìm kiếm
4. Người dùng nhập các thông tin tìm kiếm
5. Hệ thống hiển thị danh sách các xe tương ứng với thông tin tìm kiếm

*Luồng sự kiện thay thế*

STT	Vị trí	Điều kiện	Hành động	Trở về vị trí

*Dữ liệu vào*

STT	Trường dữ liệu	Mô tả	Bắt buộc	Điều kiện	Ví dụ
1	Loại xe	Loại xe cần tìm	Có	Không	Xe điện đôi
2	Màu xe	Màu xe	Không		Đen
3	Bến xe	Bến xe cần tìm	Có		Bến A

*Dữ liệu ra*

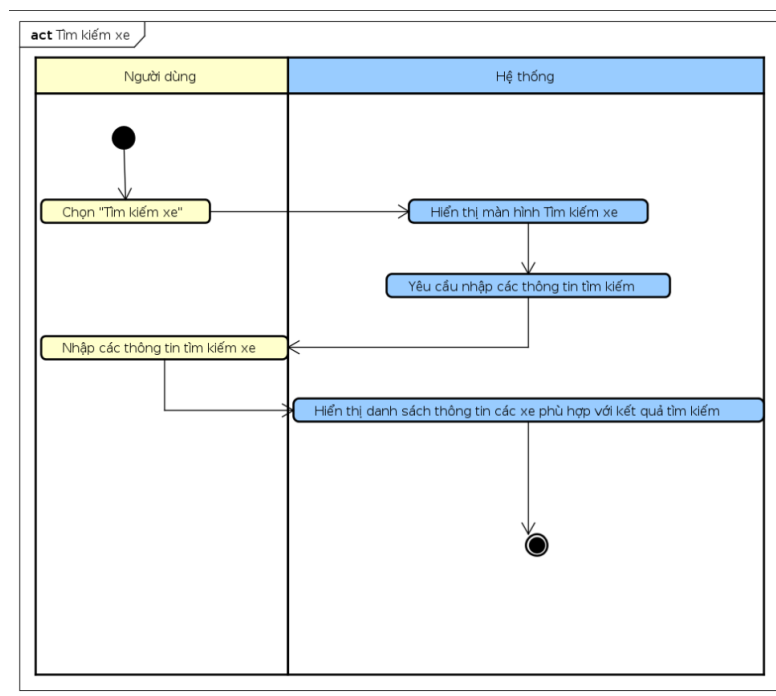
STT	Trường dữ liệu	Mô tả	Định dạng	Ví dụ
1	Loại xe	Tên loại xe	Chuỗi kí tự	Xe điện đôi
2	Màu xe	Màu xe	Chuỗi kí tự	Đen
3	Bến xe	Bến xe đang đậu	Chuỗi kí tự	Bến X

4	Biển số xe	Biển số xe	Chuỗi kí tự	01234
5	Pin	Số pin xe còn	Chuỗi kí tự	70%

*Hậu điều kiện*

Không

***Biểu đồ hoạt động***



***UC002 “Xem thông tin bãi xe”***

*Mã usecase*

UC002

*Mô tả*

Xem thông tin bãi xe

*Tác nhân*

Người dùng



*Tiền điều kiện*

Không

*Luồng sự kiện chính*

1. Người dùng chọn “Thông tin các bãi xe”
2. Hiện thị danh sách các bãi xe
3. Người dùng chọn bãi xe cần xem thông tin
4. Hiện thị thông tin bãi xe

*Luồng sự kiện thay thế*

STT	Vị trí	Điều kiện	Hành động	Trở về vị trí

*Dữ liệu vào*

STT	Trường dữ liệu	Mô tả	Bắt buộc	Điều kiện	Ví dụ

*Dữ liệu ra*

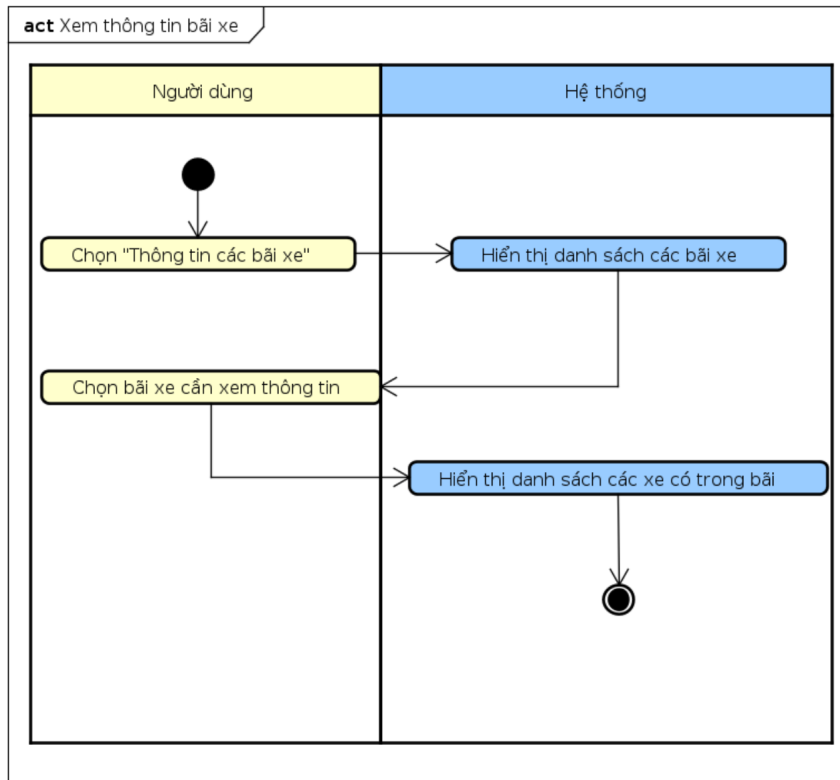
ST T	Trường dữ liệu	Mô tả	Định dạng	Ví dụ
1	Tên bãi xe	tên bãi xe	Chuỗi kí tự	Bãi A
2	Số lượng xe	Số lượng xe trong bãi	Số nguyên	10

3	Địa chỉ bãi xe	Địa chỉ bãi xe	Chuỗi kí tự	Địa chỉ A
---	----------------	----------------	-------------	-----------

*Hậu điều kiện*

Không

*Biểu đồ hoạt động*



***UC003 “Thuê xe”***

*Mã usecase*

UC003

*Mô tả*

Người dùng thuê xe

*Tác nhân*

Người dùng, Interbank

*Tiền điều kiện*

Không

*Luồng sự kiện chính*

1. Người dùng chọn “Thuê xe”
2. Hệ thống hiển thị giao diện nhập barcode
3. Người dùng nhập barcode
4. Hệ thống kiểm tra barcode
5. Hệ thống hiển thị thông tin xe
6. Hệ thống kiểm tra xe đã được thuê hay chưa
7. Hệ thống yêu cầu xác nhận thuê xe
8. Người dùng xác nhận thuê xe
9. Hệ thống yêu cầu nhập thông tin thẻ tín dụng
10. Người dùng nhập thông tin thẻ tín dụng
11. Hệ thống kiểm tra thông tin thẻ
12. Hệ thống yêu cầu thanh toán cọc
13. Người dùng xác nhận thanh toán cọc
14. Hệ thống gọi API thanh toán tiền cọc
15. Interbank kiểm tra thông tin giao dịch
16. Interbank thông báo thành công
17. Hệ thống lưu thông tin giao dịch
18. Hệ thống cập nhật trạng thái thuê xe
19. Hệ thống thông báo thuê thành công

*Luồng sự kiện thay thế*

STT	Vị trí	Điều kiện	Hành động	Trở về vị trí
1	5	Nếu thông tin barcode sai	Hệ thống thông báo barcode không hợp lệ	2
2	7	Nếu xe đã bị thuê	Hệ thống thông báo xe đã bị	5

			thuê	
3	12	Nếu thẻ không hợp lệ	Hệ thống thông báo thẻ không hợp lệ	9
4	16	Nếu giao dịch không hợp lệ	Hệ thống thông báo giao dịch không hợp lệ	9

### *Dữ liệu vào*

#### 1. Barcode

STT	Trường dữ liệu	Mô tả	Bắt buộc	Điều kiện	Ví dụ
1	Code	Barcode	Có		123123

### *Dữ liệu ra*

#### Thông tin xe

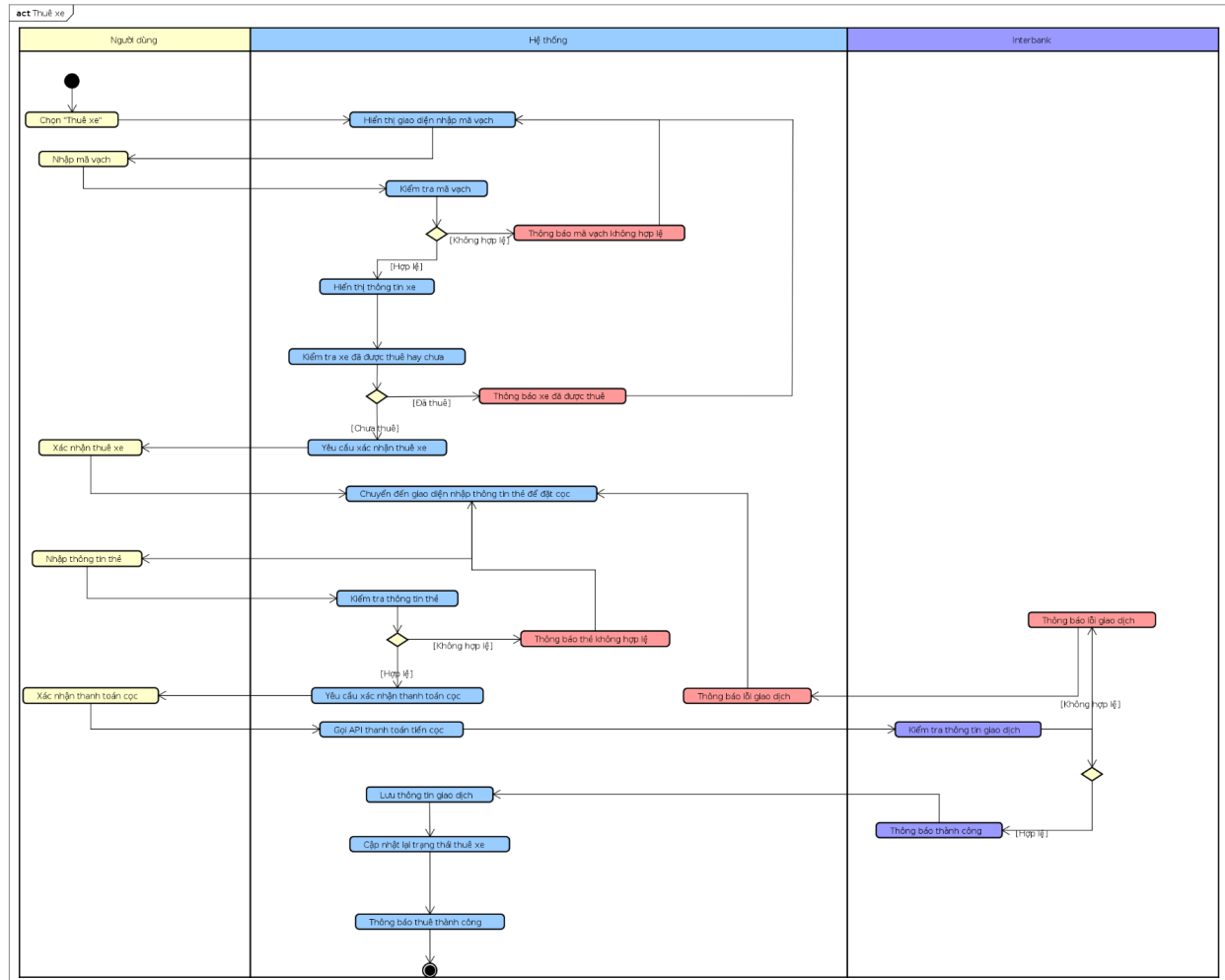
ST T	Trường dữ liệu	Mô tả	Định dạng	Ví dụ
1	Loại xe	Tên loại xe	Chuỗi kí tự	Xe điện đôi
2	Màu xe	Màu xe	Chuỗi kí tự	Đen
3	Bến xe	Bến xe đang đậu	Chuỗi kí tự	Bến X
4	Biển số xe	Biển số xe	Chuỗi kí tự	01234
5	Pin	Số pin xe còn	Chuỗi kí tự	70%

*Hậu điều kiện*

Cập nhật trạng thái xe thành công

Lưu thành công thông tin giao dịch

*Biểu đồ hoạt động*



**UC004 “Trả xe”**

*Mã usecase*

UC004

*Mô tả*

Người dùng trả xe đã thuê

*Tác nhân*

Người dùng, Interbank

*Tiền điều kiện*

Trước đó người dùng đã thuê xe

*Luồng sự kiện chính*

1. Người dùng chọn “Trả xe”
2. Hệ thống hiển thị các bãi xe
3. Người dùng chọn bãi xe để trả
4. Người dùng xác nhận trả xe
5. Hệ thống tín lại phí thuê xe
6. Hệ thống gọi API thanh toán tiền cọc và chi phí thuê xe
7. Interbank kiểm tra giao dịch
8. Hệ thống lưu thông tin giao dịch
9. Hệ thống cập nhật trạng thái xe
10. Hệ thống thông báo trả xe thành công

*Luồng sự kiện thay thế*

STT	Vị trí	Điều kiện	Hành động	Trở về vị trí
1	6	Nếu giao dịch không hợp lệ	Hệ thống thông báo giao dịch không hợp lệ	4

*Dữ liệu ra*

Không

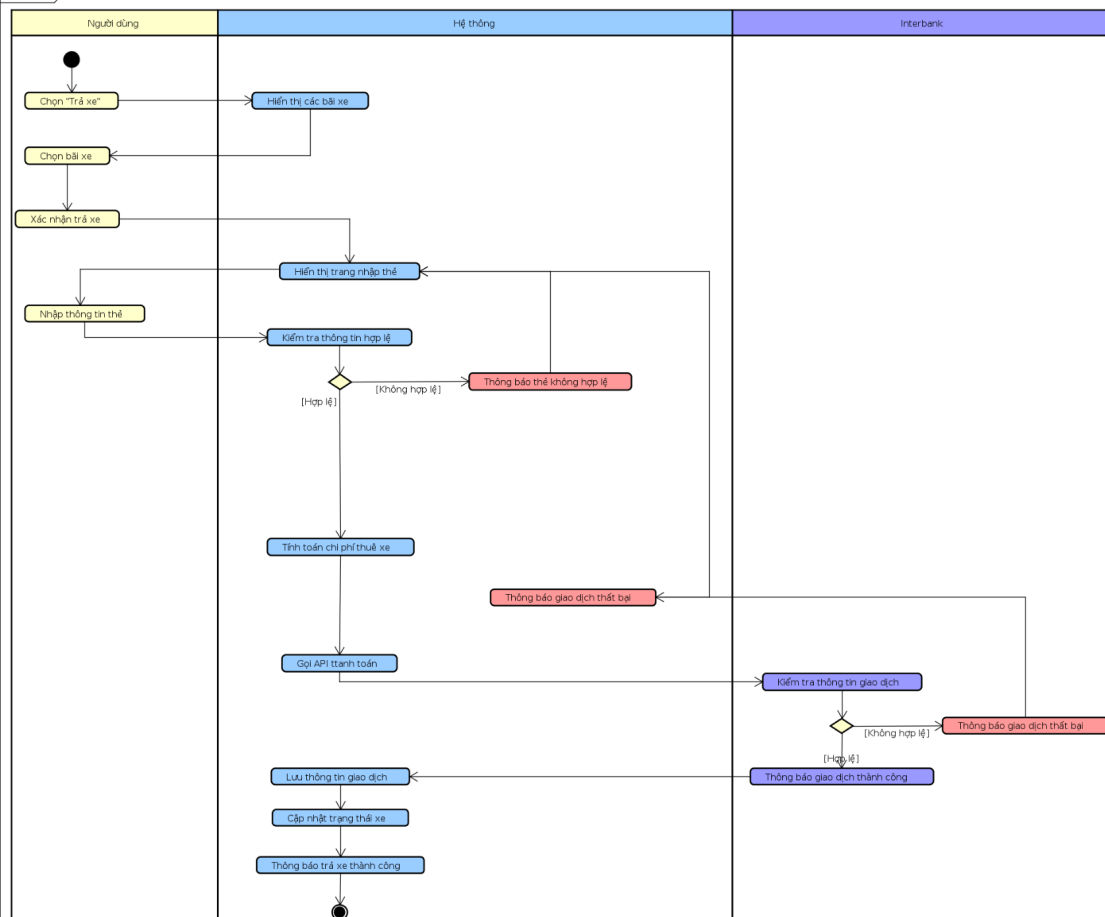
*Hậu điều kiện*

Cập nhật trạng thái xe thành công

Lưu thành công thông tin giao dịch

*Biểu đồ hoạt động*

act Trả xe

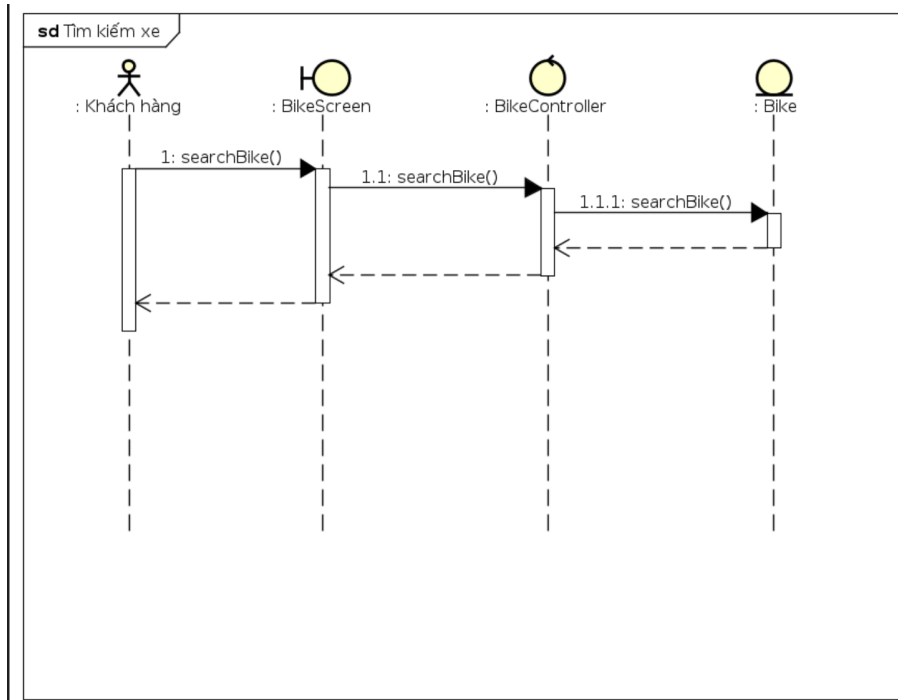


## IV. System Architecture and Architecture Design

### 1. Architectural Patterns

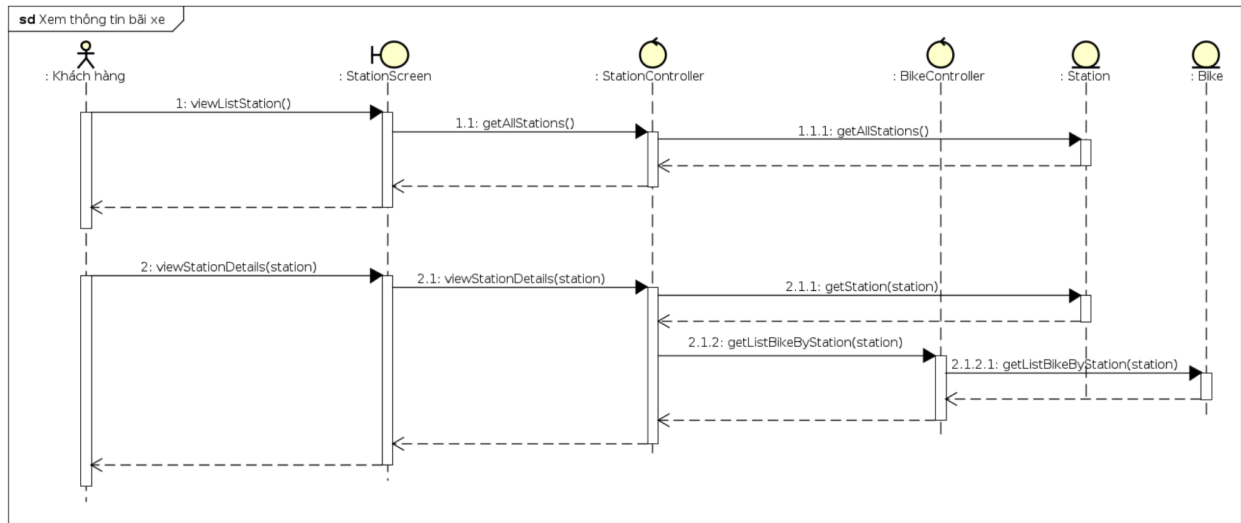
Nhóm chọn thiết kế theo kiến trúc 3 tầng Model - View - Controller

### 2. Sequence Diagrams

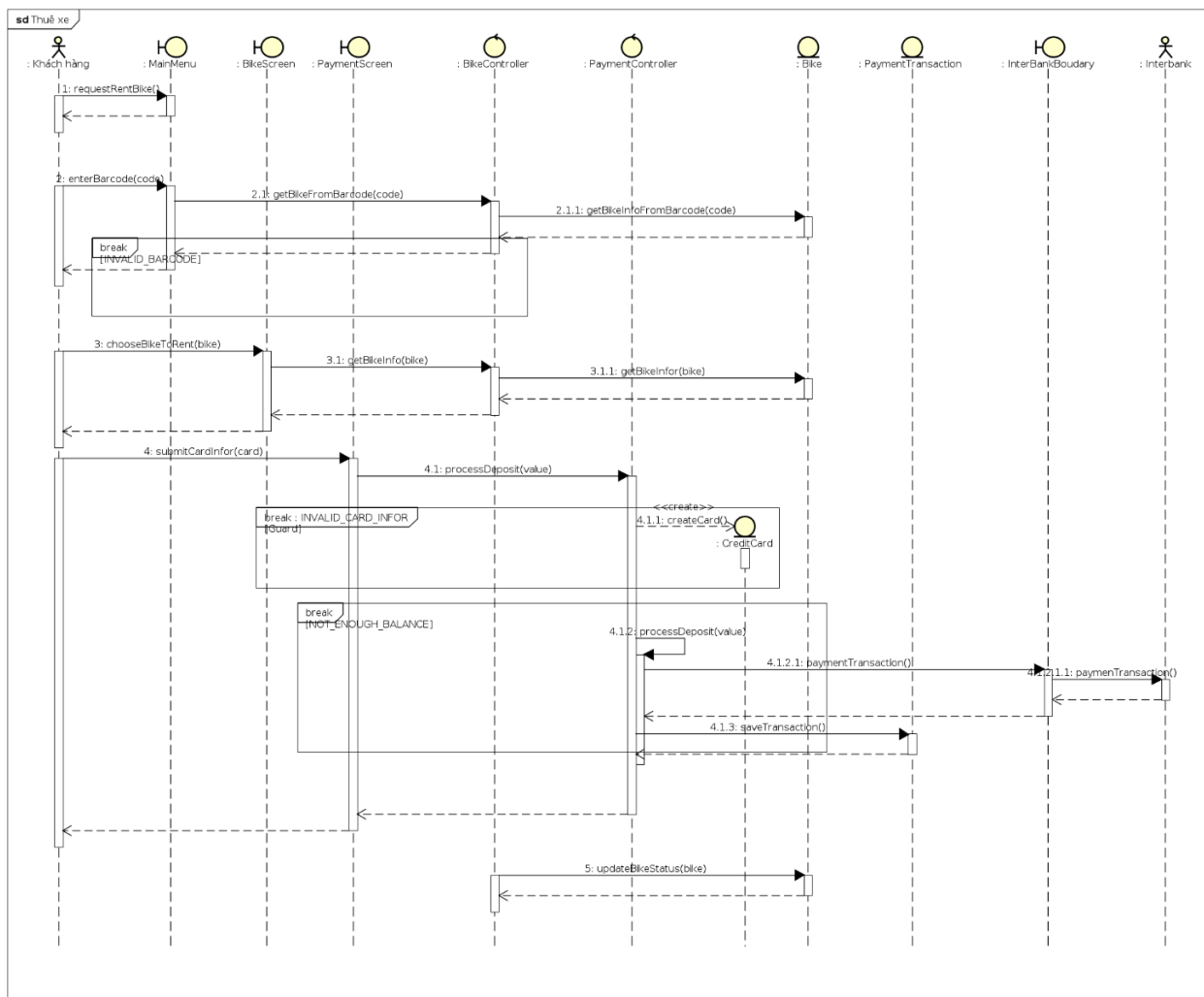


Hình 1: Tìm kiếm xe

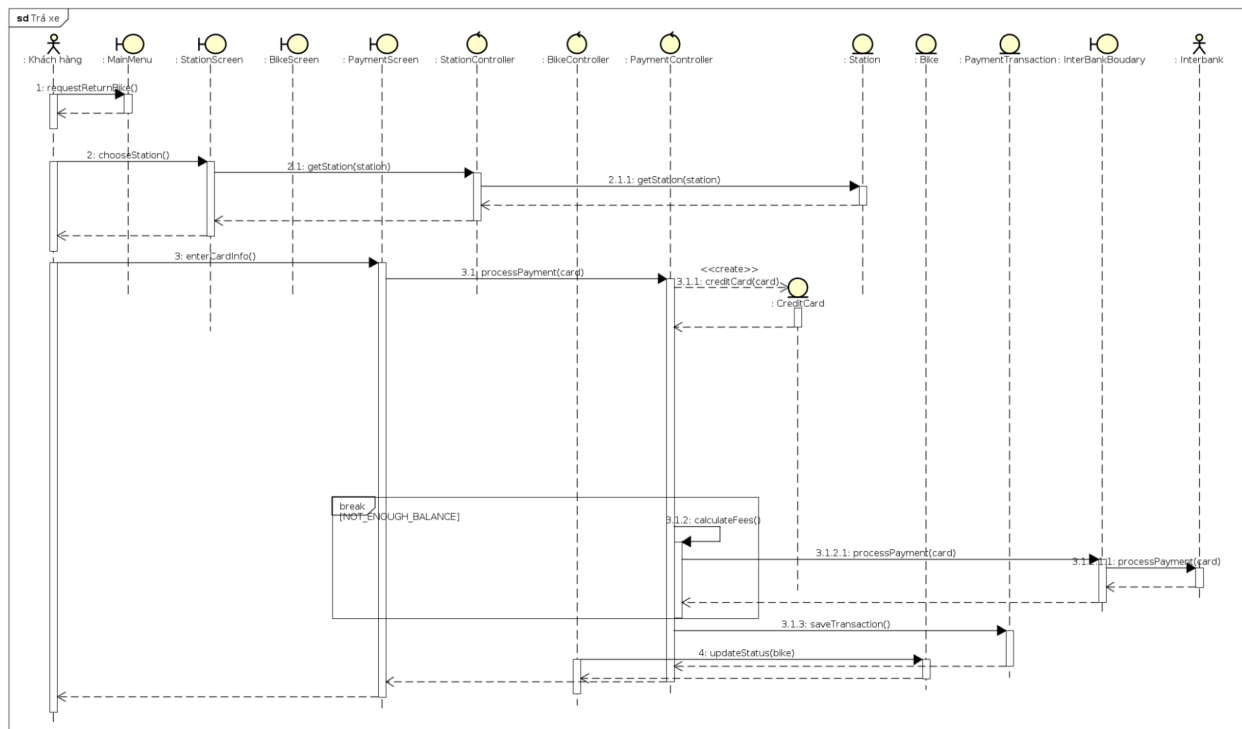




Hình 2: Xem thông tin bãi xe

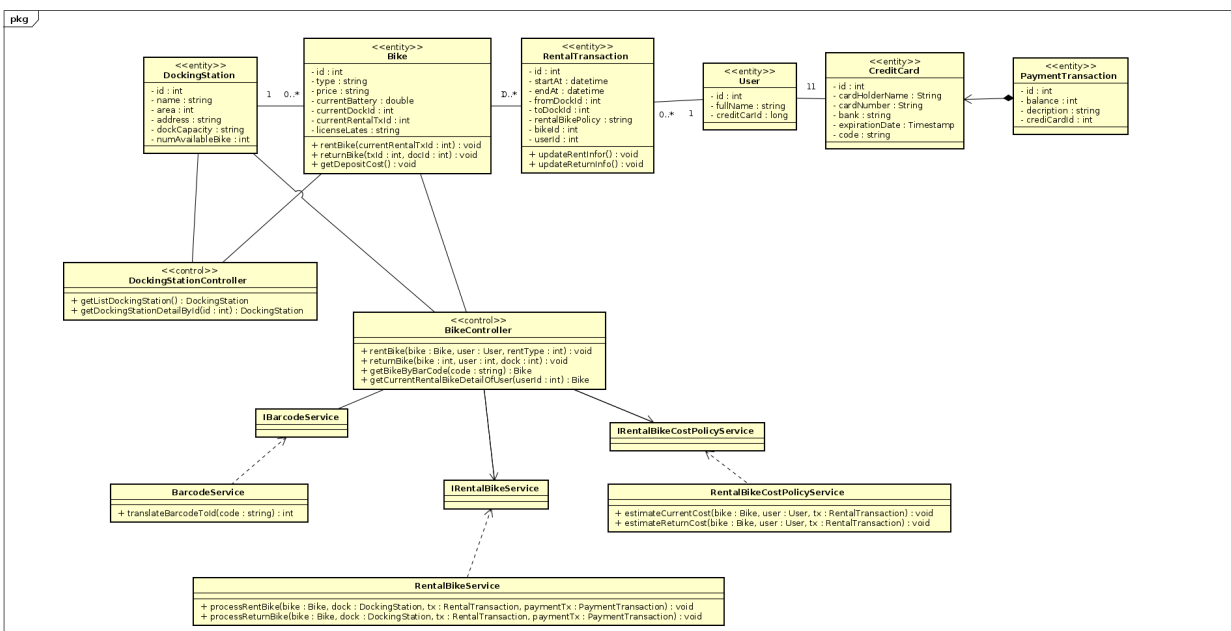


Hình 3: Thuê xe



*Hình 4: Trả xe*

### 3. Analysis Class Diagrams



## **V. Detailed Design**

### **1. User Interface Design**

#### **- Display**

Số lượng màu được hỗ trợ: 16,777,216 màu

Độ phân giải: 1366 x 768 pixels

#### **- Screen**

Vị trí của cửa button: Ở giữa (theo chiều dọc) và ở bên phải (theo chiều ngang) của khung.

Vị trí của message: Ở giữa trung tâm khung màn hình

Vị trí của screen title: Title đặt ở góc trên bên trái của màn hình.

Sự nhất quán trong hiển thị chữ số: dấu phẩy để phân cách hàng nghìn và chuỗi chỉ bao gồm các ký tự, chữ số, dấu phẩy, dấu chấm, dấu cách, dấu gạch dưới và ký hiệu gạch nối.

#### **- Control**

Kích thước text: medium size (24px).

Font: Segoe UI.

Color: 000000

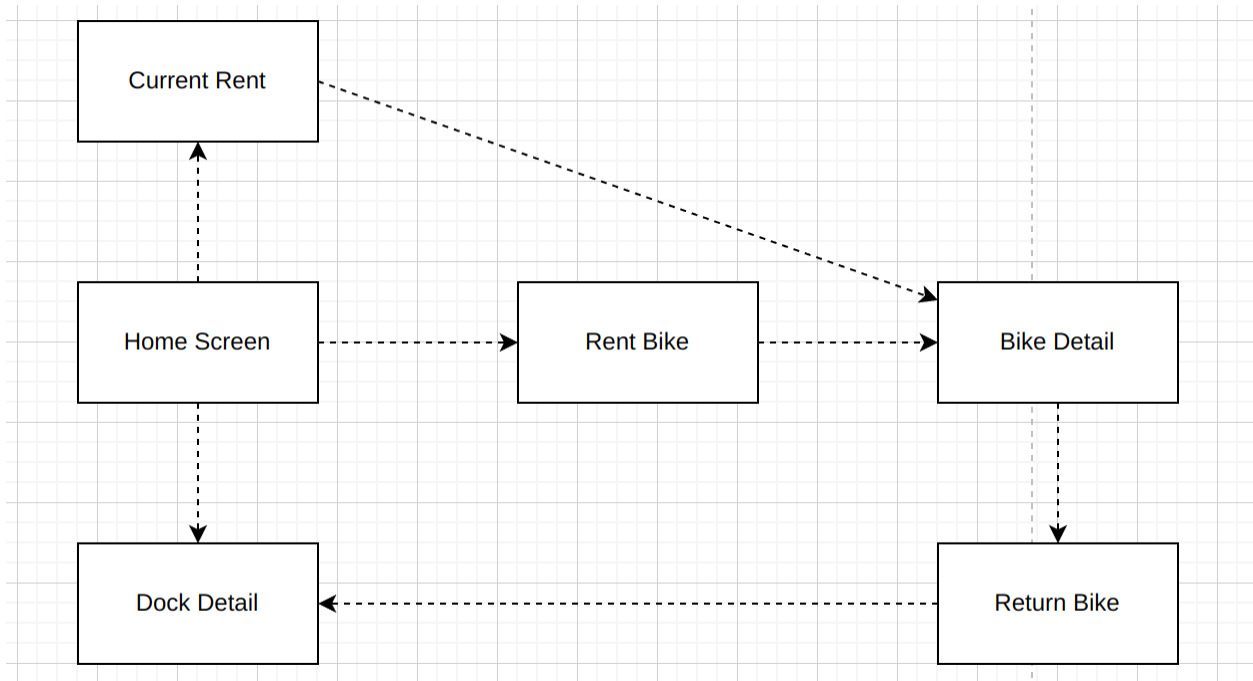
Xử lý check input: Nên kiểm tra xem input có empty hay không. Tiếp theo, kiểm tra xem input có đúng format hay không.

Dịch chuyển màn hình: Không có các khung chồng lên nhau. Các màn hình được tách biệt.

#### **- Error**

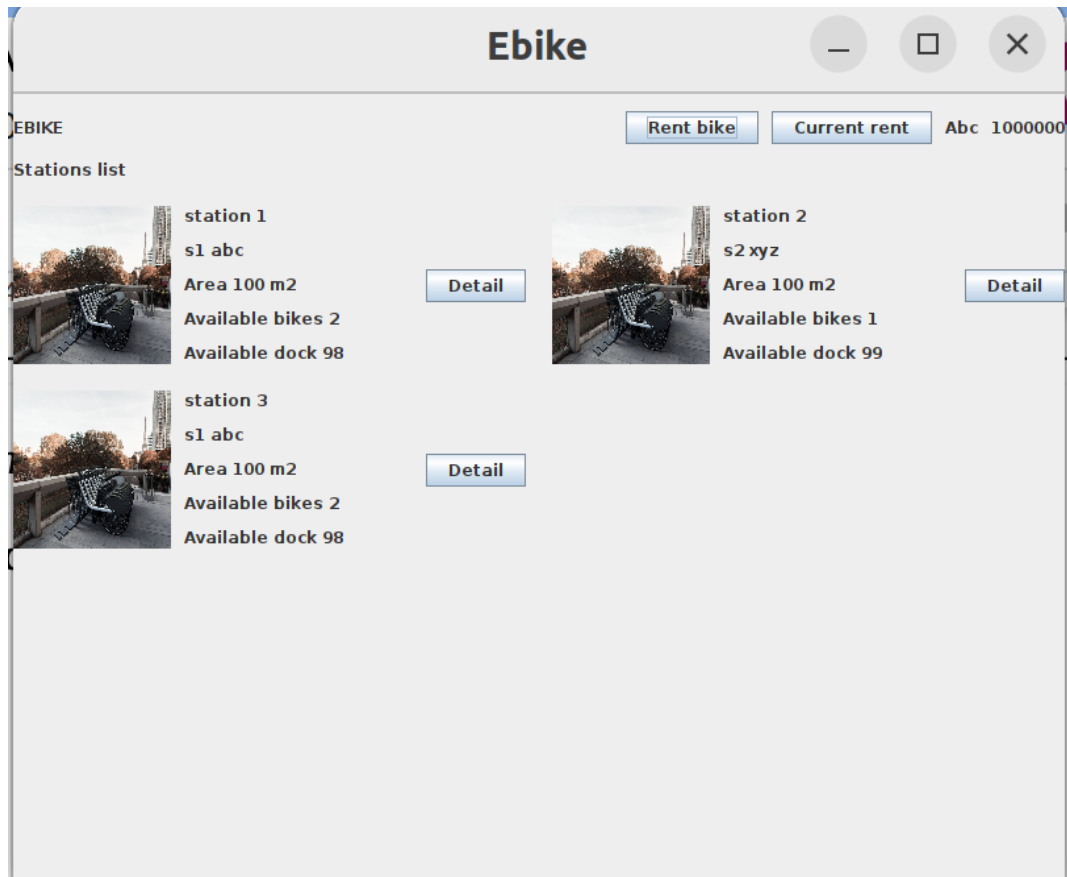
Một thông điệp sẽ được hiện lên để thông báo cho người dùng biết vấn đề đang gặp phải là gì.

## 2. Screen Transition Diagrams



*Sơ đồ chuyển màn hình*

### 3. Screen Specifications



*Home Screen*

Ebike


EBIKE

Rent bike

Current rent

Abc 1000000

Back




station 1

s1 abc

Area 100 m2

Available bikes 2

Available dock 98




SINGLE\_BIKE

Plate 0123

Battery 50

Deposit cost 400



ELETRIC\_SIGNLE\_BIKE

Plate 0124

Battery 60

Deposit cost 800

*Dock Detail*

Ebike

—

□

×

EBIKE

Rent bike

Current rent

Abc 1000000

Back

Code

Rent

*Rent Bike*

Ebike

—

□

×

EBIKE

Rent bike

Current rent

Abc 1000000




Plate 0123

Battery 50

SINGLE\_BIKE

Deposit cost 400

Confirm

*Bike Detail*



Ebike

EBIKE

Rent bike

Current rent

Abc 999600




Plate 0123

Battery 50

SINGLE\_BIKE

From station: station 1

Start at: February 7, 2023 at 11:59:02 PM ICT

Deposit cost: 400

Estimate current cost: 0.0

Return bike

*Current Rent*

Ebike


EBIKE

Rent bike

Current rent

Abc 999600

Select station to return



station 1


s1 abc

Area 100 m2

Available bikes 1

Available dock 99

Return



station 2


s2 xyz

Area 100 m2

Available bikes 1

Available dock 99

Return



station 3

s1 abc

Area 100 m2

Available bikes 2

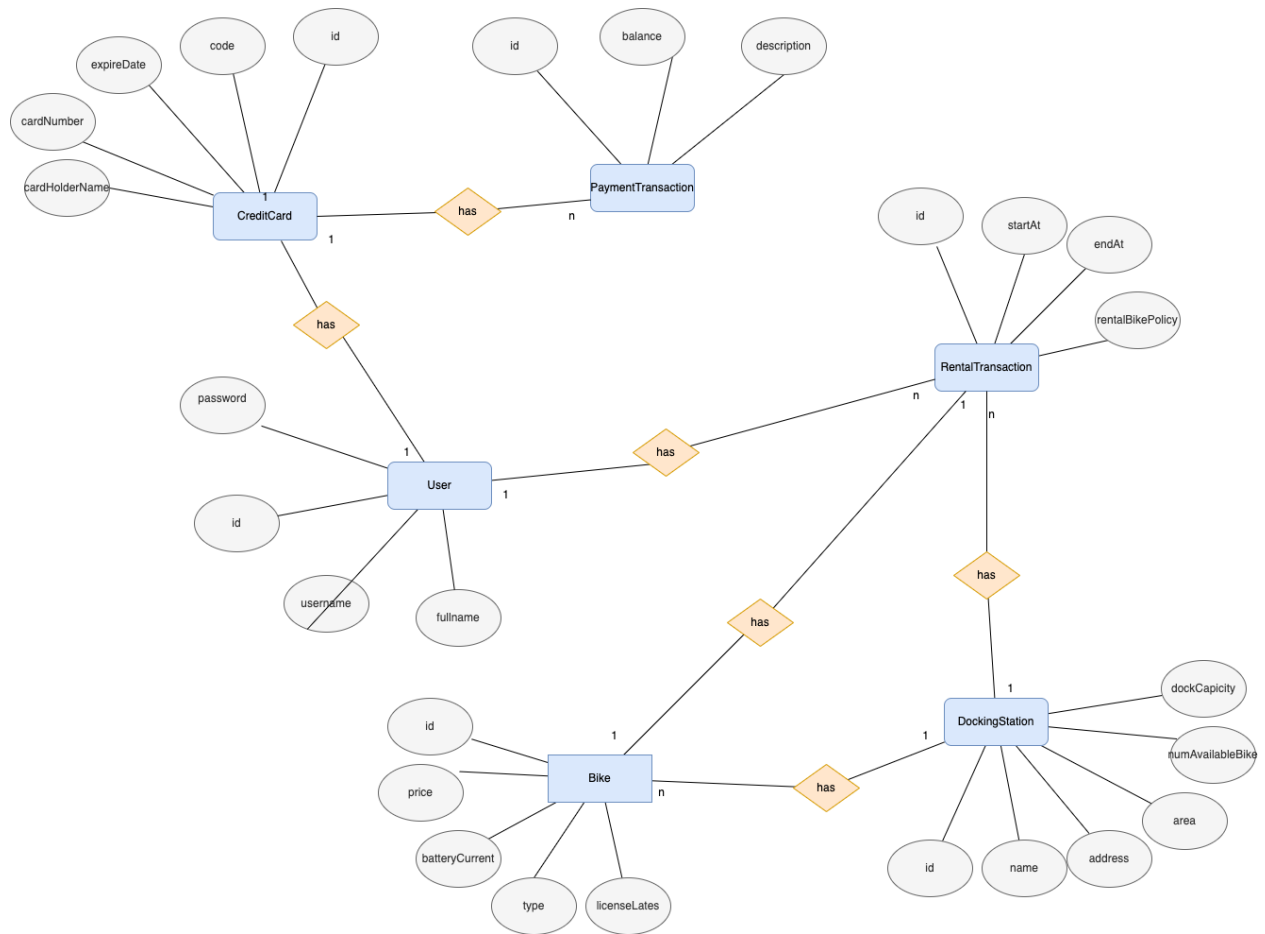
Available dock 98

Return

*Return Bike*

## V. Data Modeling

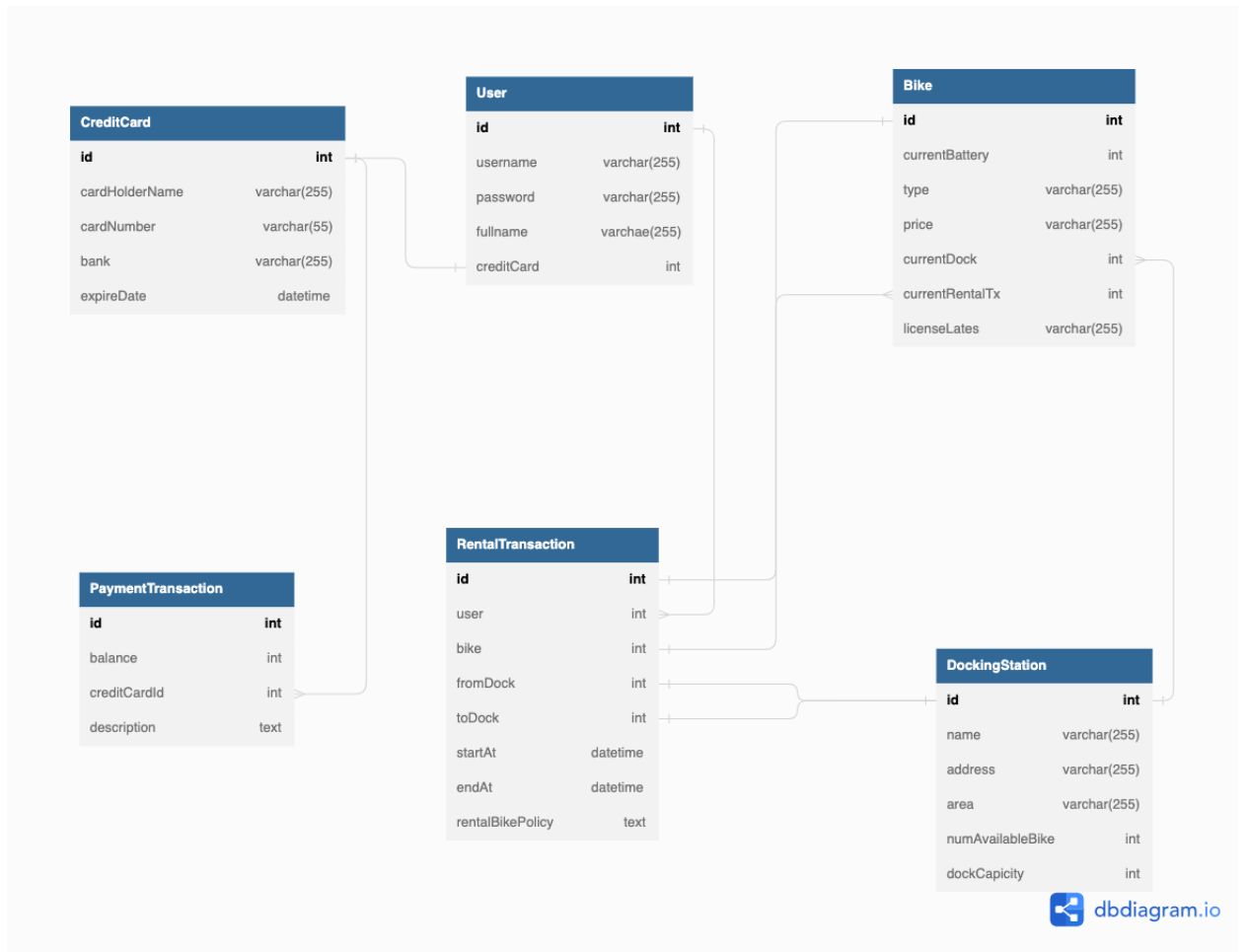
### 1. Conceptual Data Modeling



*Hình 25 ER diagram*

## 2. Database Design

### a) Mô hình hoá dữ liệu (Logical data model)



Hình 26: LDM Diagram

### b) Mô hình hoá dữ liệu vật lý (Phsyscal data model)

- User

#	PK	FK	Column name	Data type	Mandatory	Description
1	x		id	Integer	Yes	ID, auto increase
2			username	VARCHAR(255)	Yes	Username

3			password	VARCHAR(255)	Yes	Password
4			fullname	VARCHAR(255)	Yes	Name
5		x	ccreditCardId	Integer	Yes	Credit Card

- Bike

#	PK	FK	Column name	Data type	Mandatory	Description
1	x		id	Integer	Yes	ID, auto increase
2		x	currentRentalTx	Integer	Yes	Current Rental Transaction
3			currentBattery	Integer	Yes	Current battery
4			type	VARCHAR(255)	Yes	Type of bike
5		x	currentDock	int	Yes	Current Dock
6			licenseLates	VARCHAR(255)	Yes	License lates

- Docking Station

#	PK	FK	Column name	Data type	Mandatory	Description
1	x		id	Integer	Yes	ID, auto increase

2			name	VARCHAR(255)	Yes	Dock Station Name
3			address	VARCHAR(255)	Yes	Dock Station Address
4			Area	VARCHAR(255)	Yes	Area
5			dockCapacity	VARCHAR(255)	Yes	Dock Capacity
6		x	availableBikes	Integer[]	Yes	Available Bikes

- CreditCard

#	PK	FK	Column name	Data type	Mandatory	Description
1	x		id	Integer	Yes	ID, auto increase
2			cardHolderName	VARCHAR(255)	Yes	Card Holder Name
3			cardNumber	VARCHAR(55)	Yes	Card number
4			bank	VARCHAR(255)	Yes	Bank name
6			expirationDate	DATETIME	Yes	Expire date

- RentalTransaction

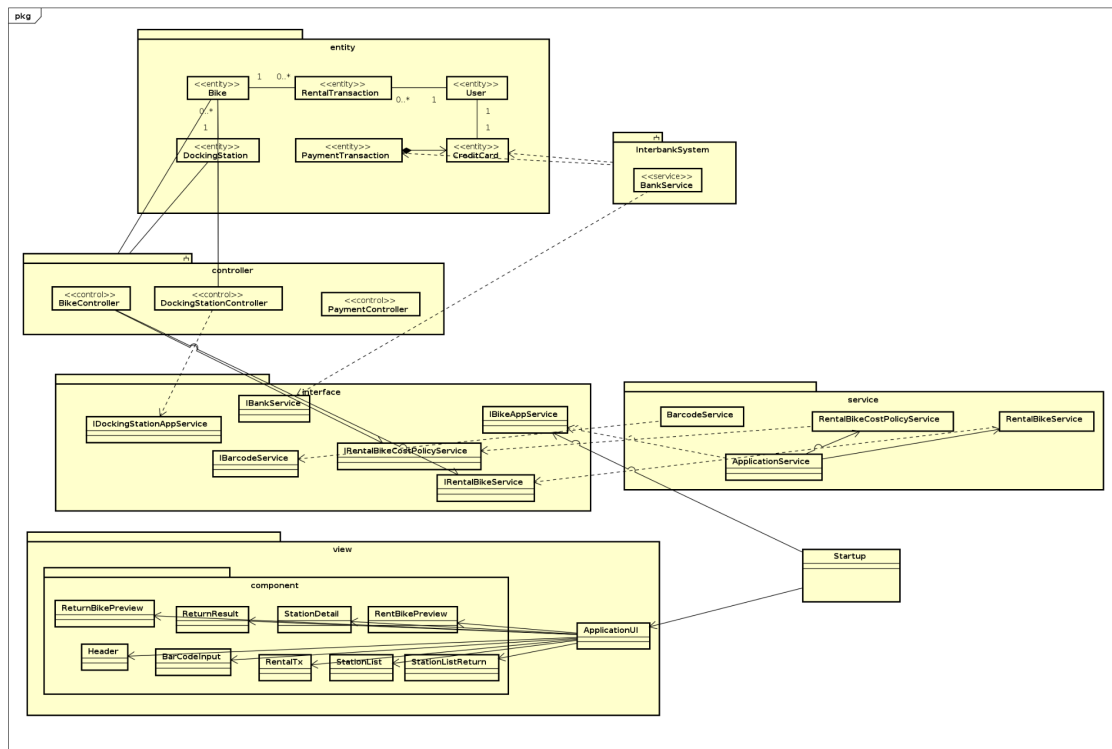
#	PK	FK	Column name	Data type	Mandatory	Description
1	x		id	Integer	Yes	ID, auto increase
2		x	userID	Integer	Yes	User id
3		x	bikeID	Integer	Yes	Bike id
4		x	fromDock	Integer	Yes	From Docking Station
5		x	toDock	Integer	Yes	To Docking Station
6			startAt	DATETIME	Yes	Started time
7			endAt	DATETIME	Yes	Ended time

- PaymentTransaction

#	PK	FK	Column name	Data type	Mandatory	Description
1	x		id	Integer	Yes	ID, auto increase
2			balance	Integer	Yes	Balance
3		x	creditCardId	Integer	Yes	Credit Card
5			Description	VARCHAR(255)	Yes	Description

### 3. Class Design

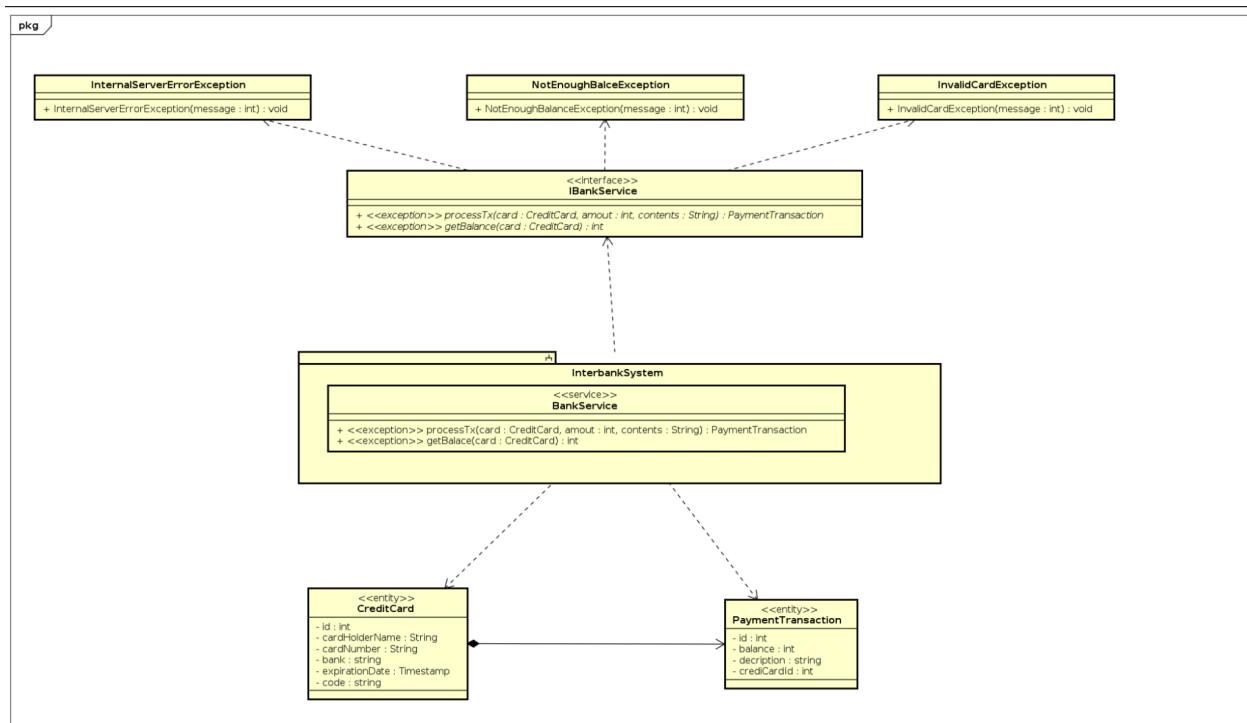
#### Class Diagram



*Class Diagram*

### Bank System

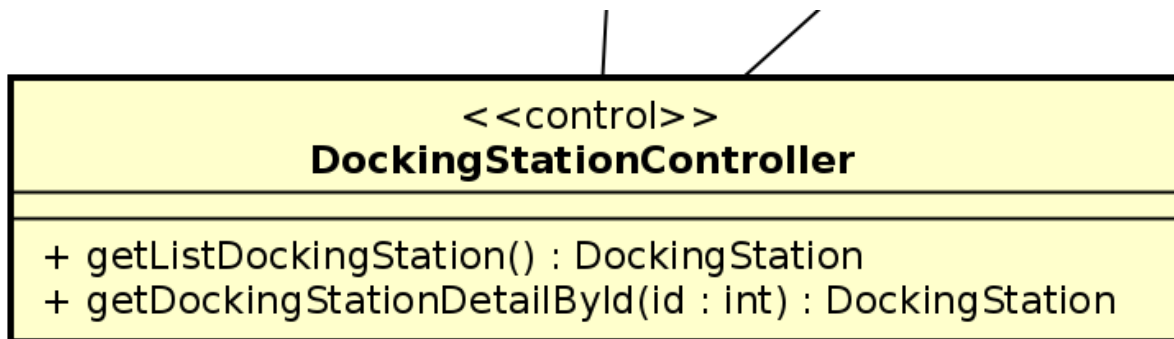




*Bank System*

## 4. Class Specification

Class “DockingStationController”



*Attribute:* Không có

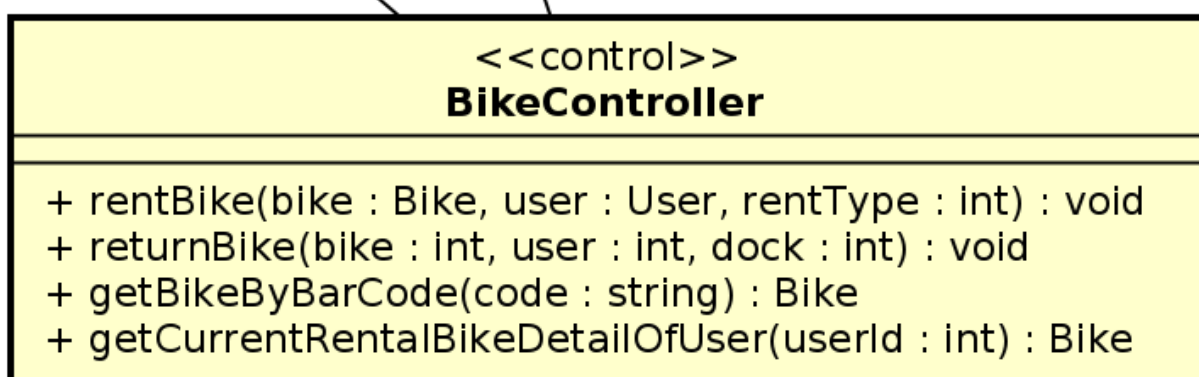
*Operation:*

ST T	Tên	Kiểu dữ liệu trả về	Mô tả (mục đích)
1	getListDockingStation()	DockingStation	Lấy danh sách docking station
2	getDockingStationDetailById(id)	DockingStation	Lấy thông tin docking station cụ thể

*Parameter:*

id: int, id của xe

### Class “BikeController”



*Attribute:* Không có

*Operation:*

ST T	Tên	Kiểu dữ liệu trả về	Mô tả (mục đích)
1	rentBike()	void	Thuê xe
2	returnBike()	void	Trả xe
3	getBikeByBarCode()	Bike	Tìm kiếm thông tin xe bằng bar code
4	getCurrentRentalBikeDetailOfUser()	Bike	Tìm kiếm thông tin xe người dùng đã thuê

*Parameter:*

bike: Bike, thông tin xe

user: User, người dùng

rentType: int, kiểu thuê

code: string, barcode

userId: int, id của user

### Class “BarcodeService”

BarcodeService	
+ translateBarcodeTold(code : string) : int	

*Attribute:* Không có

*Operation:*

ST T	Tên	Kiểu dữ liệu trả về	Mô tả (mục đích)
1	translateBarcode Told()	int	Đọc barcode

*Parameter:*

code: string, barcode

### Class “RentalBikeService”

RentalBikeService	
+ processRentBike(bike : Bike, dock : DockingStation, tx : RentalTransaction, paymentTx : PaymentTransaction) : void + processReturnBike(bike : Bike, dock : DockingStation, tx : RentalTransaction, paymentTx : PaymentTransaction) : void	

*Attribute:* Không có

*Operation:*

ST T	Tên	Kiểu dữ liệu trả về	Mô tả (mục đích)
1	processRentBike()	void	thực hiện thao tác thuê xe
2	processReturnBike()	void	thực hiện thao tác trả xe

*Parameter:*

bike: Bike, thông tin xe

dock: DockingStation, thông tin của trạm

paymentTx: PaymentTransaction, thông tin của PaymentTransaction

### Class “RentalBikeCostPolicyService”

RentalBikeCostPolicyService	
+ estimateCurrentCost(bike : Bike, user : User, tx : RentalTransaction) : void + estimateReturnCost(bike : Bike, user : User, tx : RentalTransaction) : void	

*Attribute:* Không có

*Operation:*

ST T	Tên	Kiểu dữ liệu trả về	Mô tả (mục đích)
1	estimateCurrentCost()	void	Ước lượng giá thuê xe
2	estimateReturnBike()	void	Ước lượng giá trả xe

*Parameter:*

bike: Bike, thông tin xe

user: User, thông tin người dùng

tx: RentalTransaction, thông tin của RentalTransaction

### Class “BankService”

<<service>> BankService	
+ <<exception>> processTx(card : CreditCard, amout : Int, contents : String) : PaymentTransaction + <<exception>> getBalace(card : CreditCard) : Int	

*Attribute:* Không có

*Operation:*

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

1	processTx()	PaymentTransaction	Thanh toán
2	getBalance()	int	Trả về số tiền trong tài khoản

*Parameter:*

card: CreditCard, thông tin thẻ

amount: int, số tiền thanh toán

contents: string, nội dung thanh toán

## **VI. Design Considerations**

### **1. Goals and Guidelines**

- Mục tiêu:

Phần mềm mô phỏng quá trình thuê và trả xe trong khu đô thị, với ứng dụng mô phỏng có thể chạy được đa nền tảng, thuận tiện cho người dùng trong việc tìm kiếm xe và bãi xe.

- Cách dùng:

Thuê xe trong thời gian ngắn vì số tiền ảo không đáp ứng được. Chỉ thuê 1 xe trong 1 thời điểm

### **2. Architectural Strategies**

- Hệ thống sử dụng ngôn ngữ lập trình Java, cùng thư viện JavaFx hỗ trợ đa nền tảng, có thể chạy trên nhiều hệ điều hành khác nhau
- Hệ thống sử dụng SQLite để làm Cơ sở dữ liệu