

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

Viện Công nghệ thông tin và Truyền thông

Tài liệu mô tả thiết kế phần mềm  
(Software Design Description)

Ecobike Rental

Môn: Thiết kế xây dựng phần mềm

Nhóm 13		
Phạm Hồng Phúc	:	20173303
Ngô Minh Quang	:	20173326
Trần Minh Quang	:	20173329
Sư Hữu Vũ Quang	:	20167972

Hà Nội, ngày 22 tháng 12 năm 2020

*<All notations inside the angle bracket are not part of this document, for its purpose is for extra instruction. When using this document, please erase all these notations and/or replace them with corresponding content as instructed>*

*<This document, written by Asst. Prof. NGUYEN Thi Thu Trang, is used as a case study for student with related courses. Any modifications and/or utilization without the consent of the author is strictly forbidden>*

## Table of Contents

Table of Contents .....	1
1 Giới thiệu .....	5
1.1 Mục đích .....	5
1.2 Phạm vi .....	5
1.3 Từ điển thuật ngữ .....	6
1.4 Tham khảo .....	6
2 Tổng quan .....	7
2.1 Tổng quan chung.....	7
2.2 Giả định/ Ràng buộc/ Rủi ro .....	7
2.2.1 Giả định .....	7
2.2.2 Ràng buộc .....	7
2.2.3 Rủi ro .....	8
3 Kiến trúc hệ thống và thiết kế kiến trúc .....	9
3.1 Mẫu kiến trúc .....	9
3.2 Biểu đồ tương tác .....	10
3.2.1 Biểu đồ tương tác cho UC001 – Xem thông tin của bãi xe .....	10
3.2.2 Biểu đồ tương tác cho UC002 – Xem thông tin chi tiết của xe .....	10
3.2.3 Biểu đồ tương tác cho UC003 – Thuê xe .....	11
3.2.4 Biểu đồ tương tác cho UC004 – trả xe .....	11
3.3 Biểu đồ lớp phân tích.....	12
3.3.1 Thiết kế lớp “View Bike And Station” .....	12
3.3.2 Thiết kế lớp “Rent Bike” .....	13
3.3.3 Thiết kế lớp “Return Bike” .....	13
3.4 Biểu đồ lớp phân tích gộp .....	14
3.5 Kiến trúc bảo mật phần mềm .....	14

4	Thiết kế chi tiết .....	15
4.1	Thiết kế giao diện người dùng .....	15
4.1.1	Chuẩn hóa cấu hình màn hình.....	15
4.1.2	Biểu đồ dịch chuyển màn hình .....	15
4.1.3	Đặc tả màn hình .....	18
4.2	Data Modeling .....	28
4.2.1	Conceptual Data Modeling .....	28
4.2.2	Thiết kế cơ sở dữ liệu .....	28
4.3	Non-Database Management System Files .....	34
4.4	Thiết kế lớp .....	34
4.4.1	Biểu đồ lớp chung.....	34
4.4.2	Biểu đồ lớp.....	35
4.4.3	Thiết kế lớp chi tiết.....	41
5	Design Considerations .....	88
5.1	Goals and Guidelines .....	88
5.2	Architectural Strategies.....	88
5.3	Coupling and Cohesion.....	88
5.3.1	Coupling .....	88
5.3.2	Cohesion .....	89
5.4	Design Principles .....	90
5.4.1	Single responsibility principle .....	90
5.4.2	Open/Closed principle .....	90
5.4.3	Liskov Substitution Principle .....	91
5.4.4	Interface Segregation .....	91
5.4.5	Dependency inversion principle .....	91
5.5	Design Patterns .....	92
5.5.1	Singleton pattern .....	92

## **List of Figures**

No table of figures entries found.

## **List of Tables**

No table of figures entries found.

# 1 Giới thiệu

## 1.1 Mục đích

Tài liệu này đưa ra mô tả chi tiết các chức năng cho hệ thống cho thuê xe đạp theo giờ ở khu đô thị Ecopark. Tài liệu mô tả mục đích và các tính năng của hệ thống, các giao diện, ràng buộc của hệ thống cần thực hiện để phản ứng tới các kích thích bên ngoài.

Tài liệu còn đưa ra các biểu đồ để mô tả các trường hợp xảy ra để phần mềm có thể bao quát các trường hợp, giúp phần mềm đáp ứng được yêu cầu người dùng.

Tài liệu dành cho các bên liên quan (stakeholder) và các nhà phát triển phần mềm.

## 1.2 Phạm vi

Trong thực tế, bất kỳ phần mềm nào cũng cần có các tính năng quản lý người dùng, nhóm người dùng, và cần phân quyền sử dụng các chức năng trong hệ thống một cách linh động. Mục đích của phần mềm nhằm tạo ra phân hệ quản lý người dùng (user), vai trò của người dùng (role) và các chức năng (function) mà người dùng / vai trò người dùng có thể sử dụng tại thời điểm chạy. Người dùng có thể đăng ký để tạo ra tài khoản cho mình, xác thực thông tin, thiết lập quyền truy cập ứng dụng và thiết lập phương thức thanh toán để trả phí thuê xe, sau đó người dùng có thể đăng nhập để sử dụng các chức năng của hệ thống. Sau khi đăng nhập vào hệ thống, hệ thống sẽ cung cấp ba chức năng chính cho người dùng là xem thông tin bãi xe, xem thông tin chi tiết của xe, chức năng thuê xe và trả xe. Người dùng có thể bấm chọn một bãi xe trong danh sách các bãi xe hoặc tìm kiếm bãi xe để xem thông tin chi tiết của bãi xe, đồng thời khách hàng có thể xem thông tin chi tiết của xe trong bãi Khi

người dùng muốn thuê xe, họ cần nhập mã vạch của xe muốn thuê. Sau đó người dùng sẽ được yêu cầu chọn phương thức giao dịch, xác nhận giao dịch, sau khi xác nhận giao dịch thành công thì khóa xe sẽ tự động mở, bắt đầu quá trình thuê xe Khi muốn trả xe, khách hàng chọn bãi xe muốn trả xe, đưa xe vào vị trí trống bất kỳ trong bãi bất kỳ (thông thường là bãi xe gần nhất dựa vào vị trí thực tế) và đóng khoá xe lại.

### **1.3 Từ điển thuật ngữ**

STT	Thuật ngữ	Giải thích	Ví dụ	Ghi chú
1	SecurityKey	Một phần dữ liệu được tạo ở phía server ra chứa thông tin về tài khoản ngân hàng của người dùng và mã token. Token được sử dụng để xác thực người dùng khi muốn thực hiện giao dịch với token đã được cung cấp mà không phải sử dụng trực tiếp tài khoản và mật khẩu	JSON Web Token (JWT)	Token được thiết kế dưới dạng mã hóa, nhỏ gọn, an toàn
2	PK	Primary key(khóa chính)	bikeId	
3	FK	Foreign key(khóa ngoại)		

### **1.4 Tham khảo**

- Template “Tài liệu mô tả thiết kế phần mềm” – GV Nguyễn Thị Thu Trang
- Tài liệu thực hành môn Thiết kế xây dựng phần mềm – GV Nguyễn Thị Thu Trang

## 2 Tổng quan

- Nguyên tắc thiết kế
- Chiến lược thiết kế

### 2.1 Tổng quan chung

- Bối cảnh hệ thống: Người dùng muốn thuê xe đạp
  - o Người dùng đăng nhập vào hệ thống
  - o Người dùng tìm bãi xe mình muốn mượn (Bãi xe bất kì hoặc bãi xe gần nhất)
  - o Người dùng xem thông tin chi tiết các xe có trong bãi xe đó
  - o Người dùng chọn xe cần thuê, nhập thông tin thẻ ngân hàng để thuê
  - o Người dùng xác nhận thuê xe
  - o Người dùng chọn bãi xe muốn trả khi muốn trả xe và tiến hành thanh toán, nhận lại tiền thừa hoặc trả thêm tiền

### 2.2 Giả định/ Ràng buộc/ Rủi ro

#### 2.2.1 Giả định

- Phần mềm: Người dùng có thiết bị điện tử có cài đặt phần mềm, có hỗ trợ check mã code của xe (một dãy mã gồm 9 ký tự)
- Yêu tố con người:
  - o Là người muốn thuê xe
  - o Là người có thẻ sử dụng được phần mềm mà không gặp phải khó khăn nào
  - o Người dùng có thẻ ngân hàng để thanh toán sau khi trả xe
- Trường hợp ngoại lệ:
  - o Thiết bị không có kết nối internet để sử dụng phần mềm
  - o Thông tin thẻ không hợp lệ
  - o Thẻ bị khóa hoặc bị lỗi không thanh toán được
  - o Số dư trong thẻ không đủ để thực hiện thanh toán
  - o Người dùng đang sử dụng phần mềm không phải với phiên bản mới nhất
  - o Những trường hợp ngoại lệ khác

#### 2.2.2 Ràng buộc

- Người dùng phải sử dụng thiết bị đã cài đặt phần mềm

- Giao diện phần mềm phải đảm bảo được các nguyên tắc thiết kế, đáp ứng được như cầu người dùng
- Phần mềm phải đảm bảo được tính bảo mật thông tin cho người dùng
- Phần mềm phải trả về chính xác thông tin xe và bãi xe
- Phần mềm phải trả về chính xác số xe của bãi xe trước và sau khi thuê xe và trả xe
- Phần mềm phải liên kết với ngân hàng để có thể kiểm tra độ tin cậy của thông tin thẻ ngân hàng và kiểm tra số dư trong thẻ khi thực hiện thanh toán

### 2.2.3 Rủi ro

- Phần mềm không tương thích với thiết bị của người dùng
- Phần mềm chưa được cập nhật lên phiên bản mới nhất
- Phần mềm không quét được mã code 9 kí tự của xe
- Giao diện không mang lại sự hài lòng cho người dùng, không đảm bảo tính dùng được
- Phần mềm không cập nhật lại số xe sau khi thuê xe và trả xe
- Người dùng không đủ số dư để thực hiện thanh toán khi thuê xe và trả xe
- Phần mềm không tính chính xác được số tiền thuê xe dựa trên thời gian
- Phần mềm không được liên kết với ngân hàng
- Ngân hàng bảo trì khiến người dùng không thanh toán được

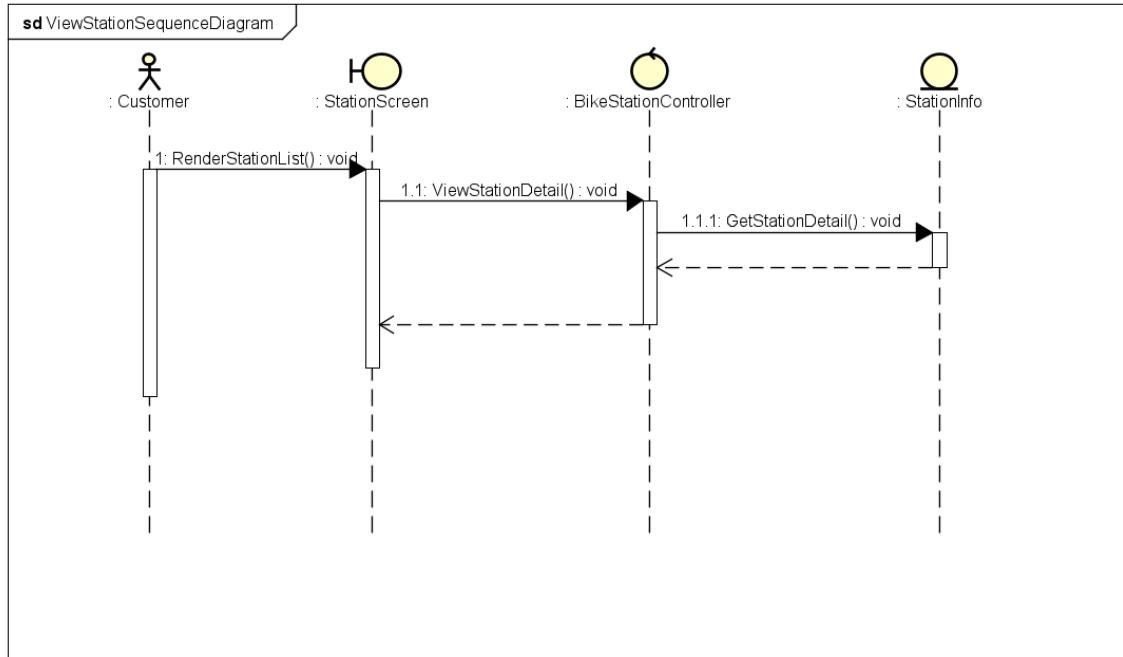
### 3 Kiến trúc hệ thống và thiết kế kiến trúc

#### 3.1 Mẫu kiến trúc

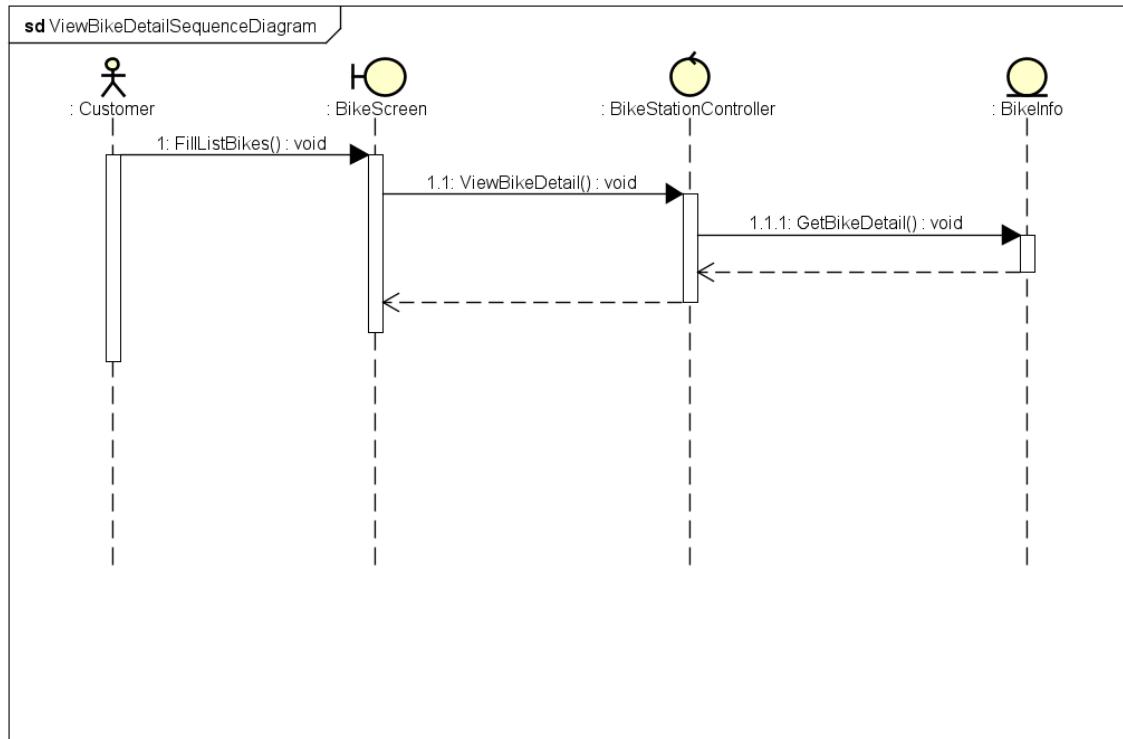
- Mô hình 3 layer
  - o Mô hình 3 lớp có tác dụng phân chia các thành phần trong hệ thống, các thành phần có cùng chức năng sẽ được nhóm lại với nhau và phân chia công việc cho từng nhóm để dữ liệu không bị chồng chéo và chạy lộn xộn
- Thành phần:
  - o Presentation Layer (GUI): Lớp này có nhiệm vụ chính là giao tiếp với người dùng. Nó gồm các thành phần giao diện và thực hiện các công việc như nhập liệu, hiển thị dữ liệu, kiểm tra tính đúng đắn trước khi gọi lớp Business Logic Layer
  - o Business Logic Layer (BLL):
    - Đây là nơi đáp ứng các yêu cầu thao tác dữ liệu của GUI layer, xử lý chính nguồn dữ liệu từ Presentation Layer trước khi truyền xuống Data Access Layer và lưu xuống hệ quản trị CSDL
    - Đây còn là nơi kiểm tra các ràng buộc, tính toàn vẹn và hợp lệ dữ liệu, thực hiện tính toán và xử lý các yêu cầu nghiệp vụ, trước khi trả kết quả về Presentation Layer
  - o Data Access Layer (DAL):
    - Lớp này có chức năng giao tiếp với hệ quản trị CSDL như thực hiện các công việc liên quan đến lưu trữ và truy vấn dữ liệu (tìm kiếm, thêm, xóa, sửa,...)
- Ưu điểm:
  - o Phân loại rõ ràng các lớp có các nhiệm vụ khác nhau. Từ đó ta có thể quản lý và maintain project tốt hơn.
  - o Dễ dàng phân loại các hành động tại Business.
  - o Dễ dàng phân loại các hàm truy xuất tại Database, phân loại hàm theo table,...
  - o Ứng dụng được cho các project lớn ở bên ngoài.

## 3.2 Biểu đồ tương tác

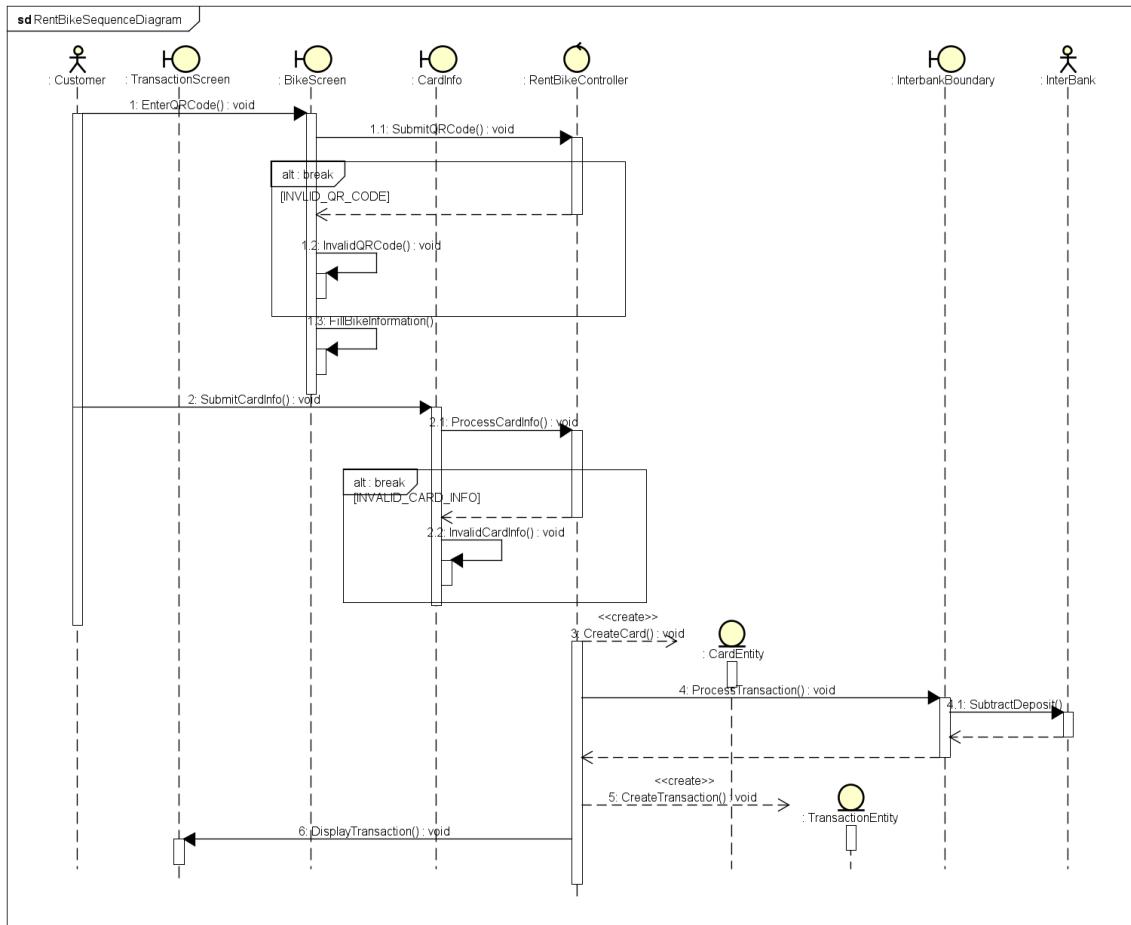
### 3.2.1 Biểu đồ tương tác cho UC001 – Xem thông tin của bãi xe



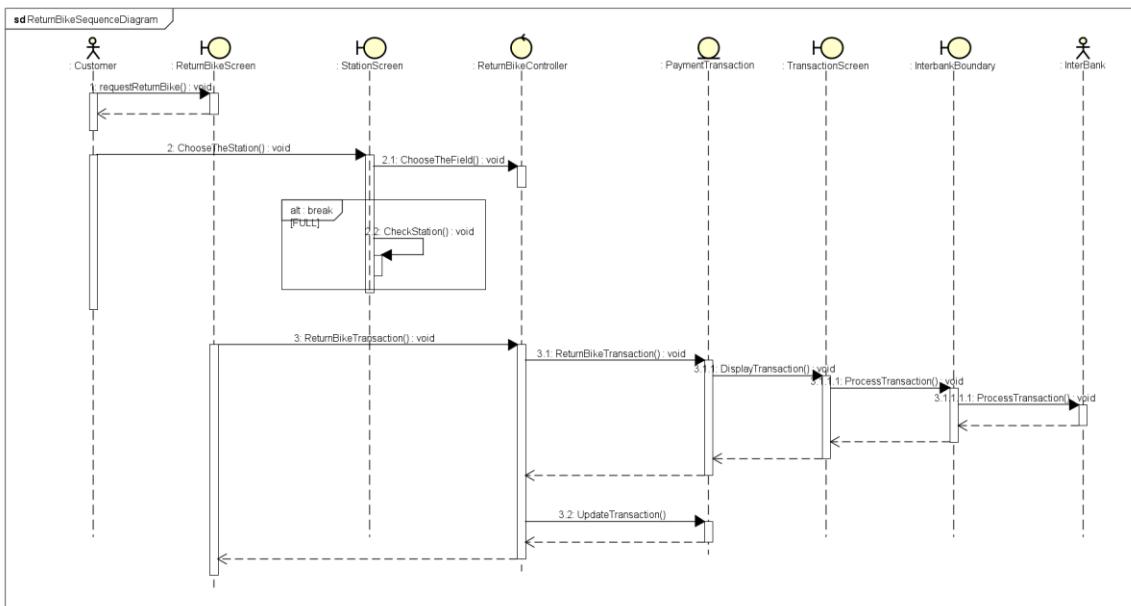
### 3.2.2 Biểu đồ tương tác cho UC002 – Xem thông tin chi tiết của xe



### 3.2.3 Biểu đồ tương tác cho UC003 – Thuê xe

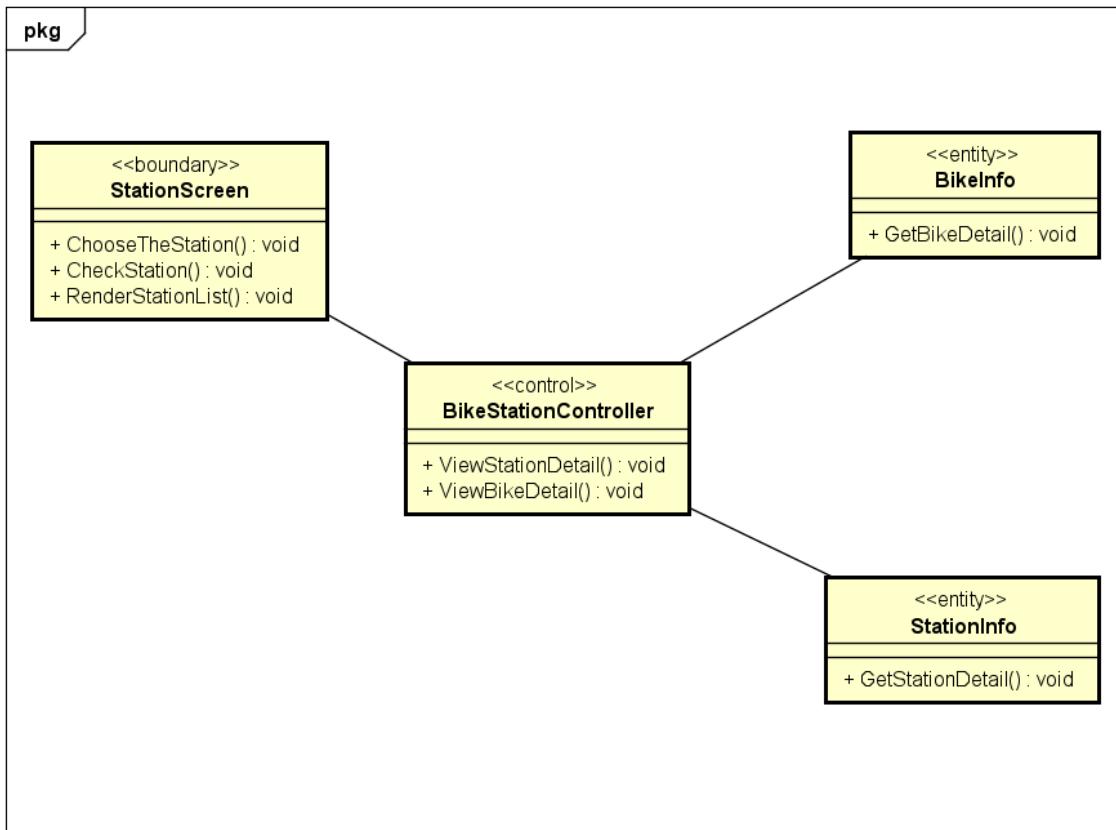


### 3.2.4 Biểu đồ tương tác cho UC004 – trả xe

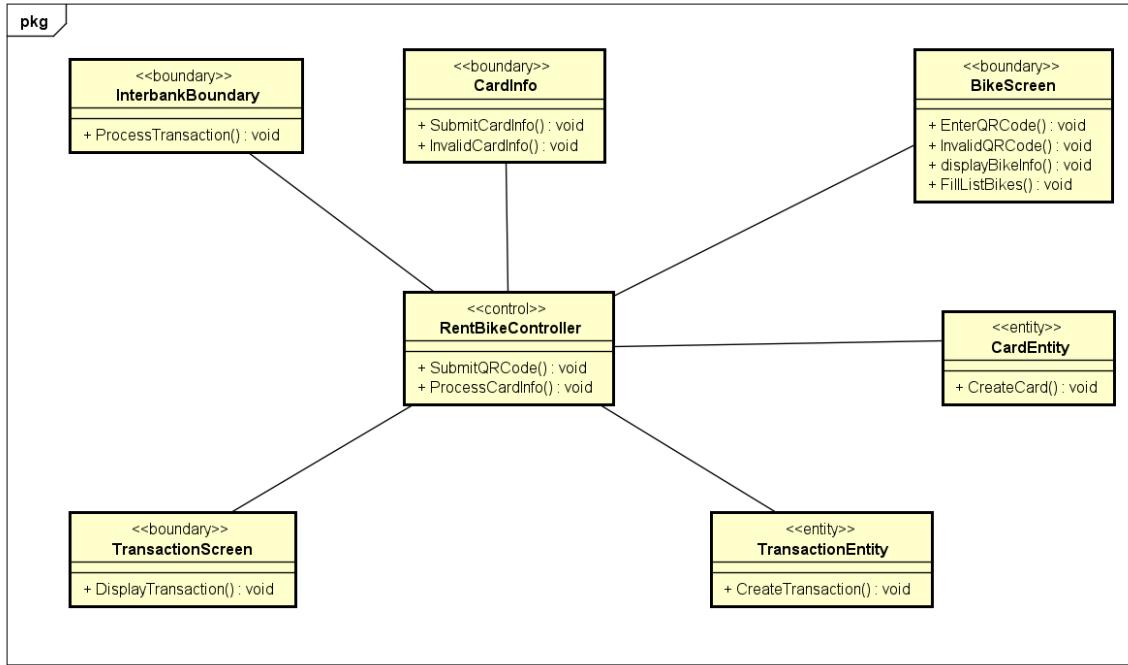


### 3.3 Biểu đồ lớp phân tích

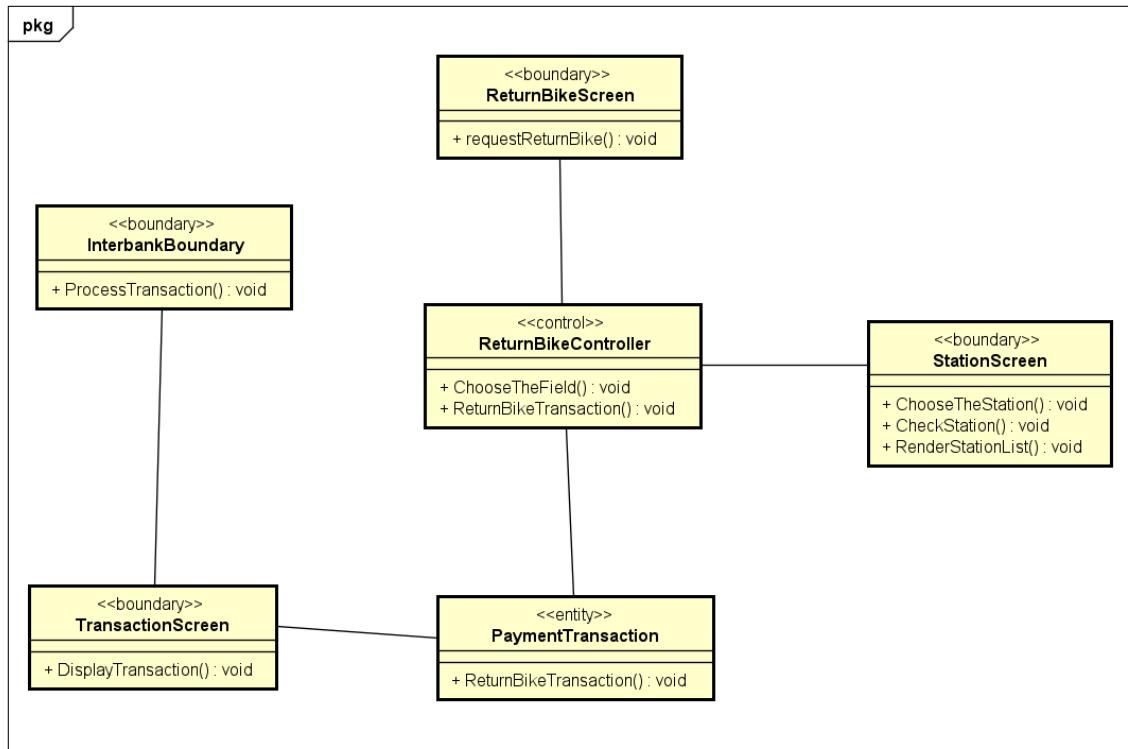
#### 3.3.1 Thiết kế lớp “View Bike And Station”



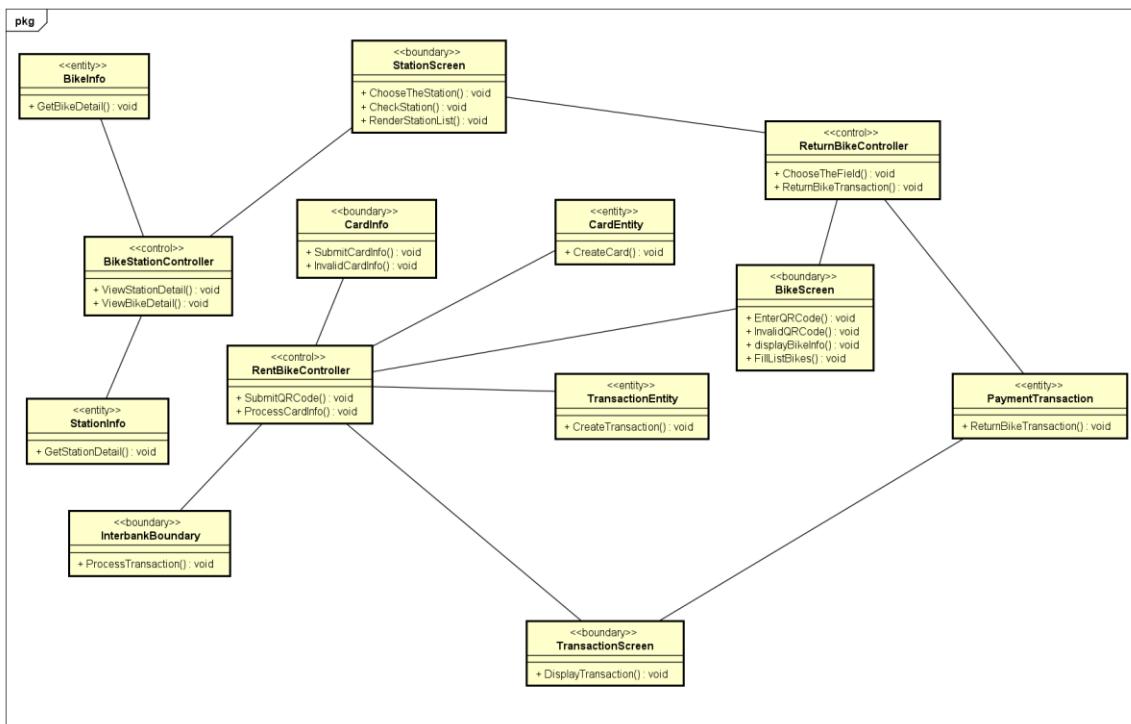
### 3.3.2 Thiết kế lớp “Rent Bike”



### 3.3.3 Thiết kế lớp “Return Bike”



### 3.4 Biểu đồ lớp phân tích gộp



### 3.5 Kiến trúc bảo mật phần mềm

- Các thao tác với thẻ tín dụng được thực hiện thông qua một token bí mật.

## 4 Thiết kế chi tiết

### 4.1 Thiết kế giao diện người dùng

#### 4.1.1 Chuẩn hóa cấu hình màn hình

##### a. Hiển thị

- Số lượng màu được hỗ trợ: 16,777,216 màu
- Độ phân giải: 800 x 500 pixels

##### b. Màn hình

- 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.

##### c. Các control

- Kích thước text: medium size (12px). Font: Arial. Color: #d4e3fc
- 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.

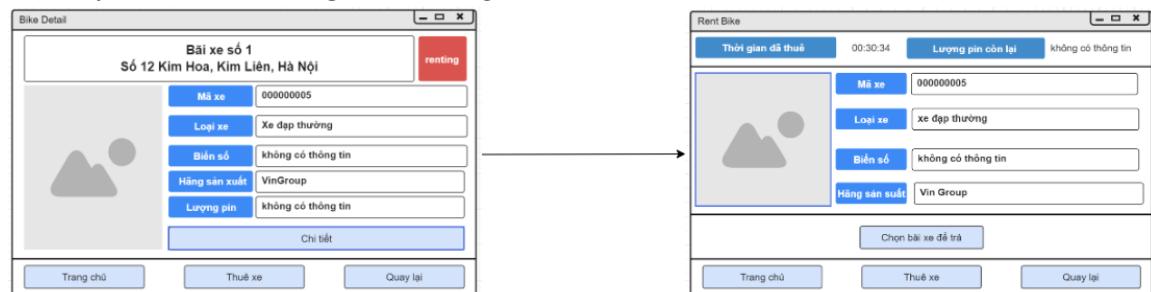
#### 4.1.2 Biểu đồ dịch chuyển màn hình

- a. Biểu đồ dịch chuyển màn hình cho luồng sự kiện chính của chương trình

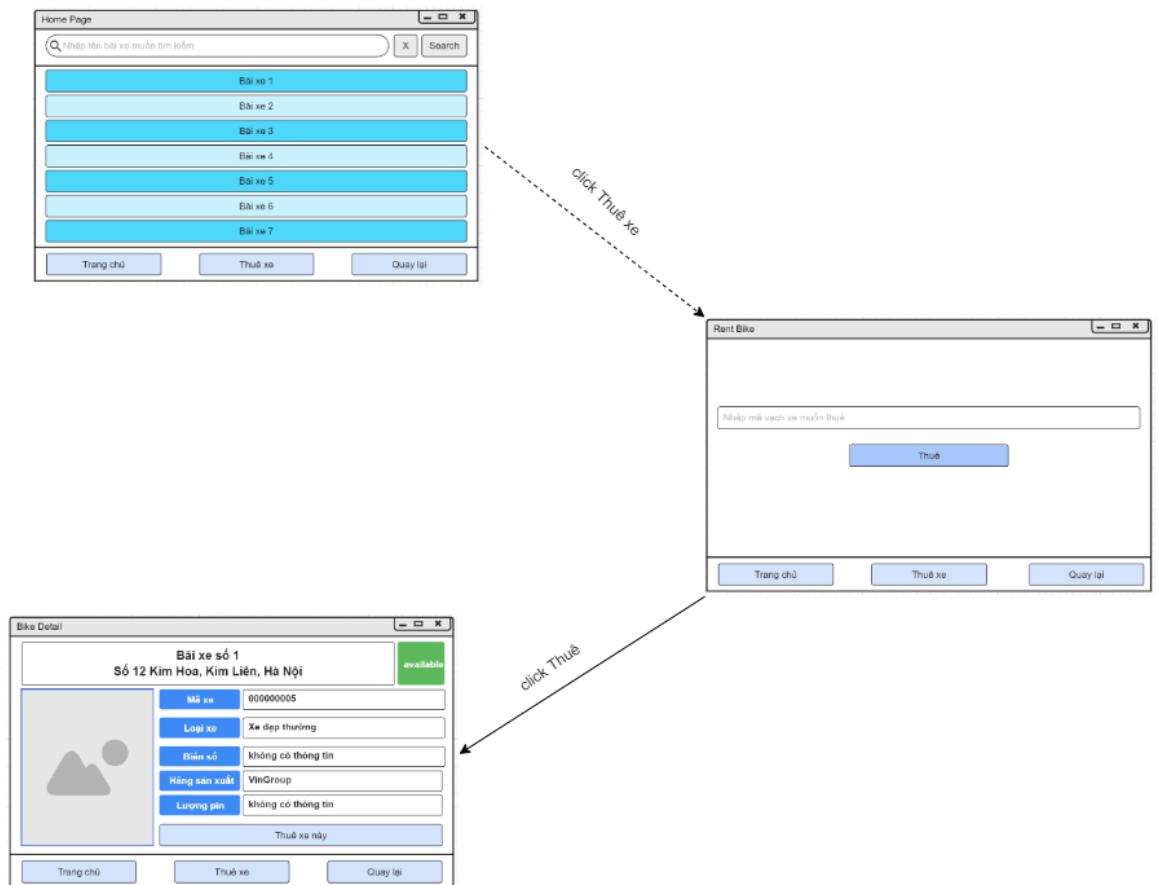


b. Các biểu đồ dịch chuyển màn hình khác

- Khi xe đã được thuê, từ trang thông tin xe chi tiết khi bấm vào nút chi tiết sẽ chuyển luôn tới trang xem thông tin thuê xe



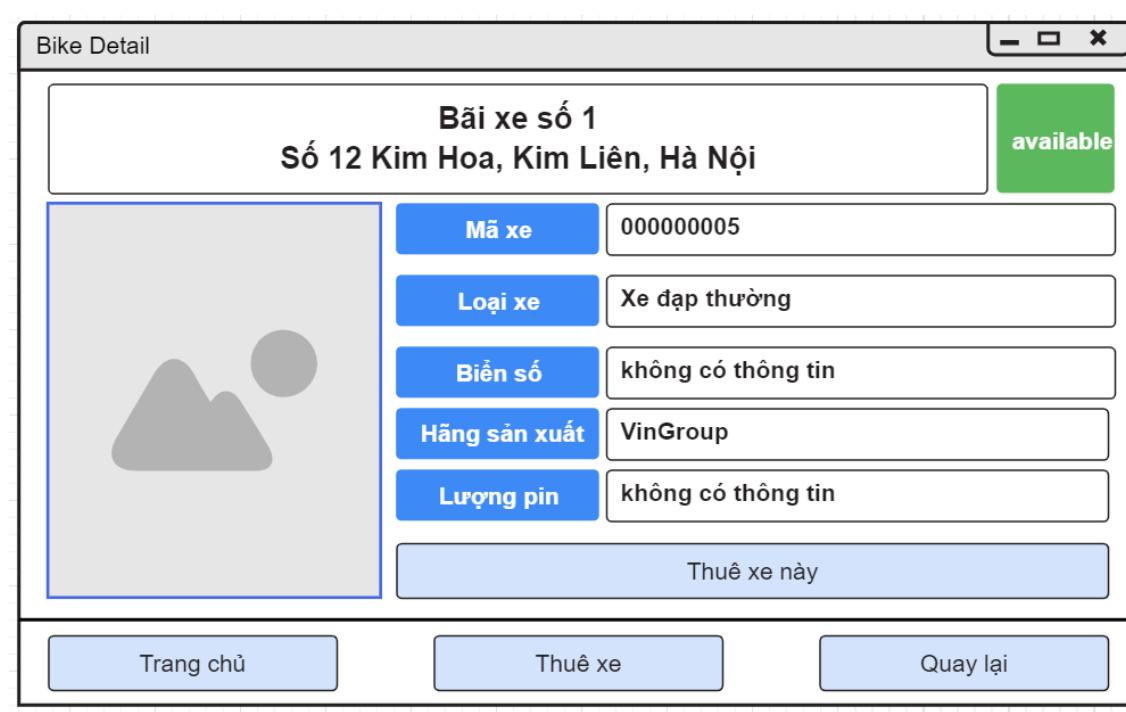
- Ở bất kì trang nào, khi bấm vào nút Thuê xe, màn hình nhập mã qrcode hiện ra, nếu nhập đúng sẽ chuyển đến trang xem thông tin chi tiết xe, nếu qrcode sai thì báo lỗi



- Khi bấm vào nút Quay lại thì sẽ chuyển đến màn hình trước đó, nếu không có màn hình trước đó thì ở lại màn hình cũ

#### 4.1.3 Đặc tả màn hình

##### a. Bike Detail Screen



The screenshot shows the 'Bike Detail' window. At the top, it displays 'Bãi xe số 1' and 'Số 12 Kim Hoa, Kim Liên, Hà Nội'. A green button on the right says 'available'. Below this, there's a large placeholder image for a bike. To its right are five data fields: 'Mã xe' (00000005), 'Loại xe' (Xe đạp thường), 'Biển số' (không có thông tin), 'Hãng sản xuất' (VinGroup), and 'Lượng pin' (không có thông tin). At the bottom, a blue button says 'Thuê xe này'. Along the bottom edge are three buttons: 'Trang chủ', 'Thuê xe', and 'Quay lại'.

Control	Operation	Function
textbox	initial	Display name and address of station contain this bike
Status panel	initial	Display bike status, when bike is being renting, status label is renting and when bike is available, status label is available
Picture box	display	Display preview image of bike
Rent this bike button	click	Display rent bike info screen to confirm rent this bike
Home page button	click	Display home page screen
Rent bike button	click	Display rent bike screen or renting bike screen
Previous form button	click	Display previous form and hide current form

##### b. Home Page Screen

Home Page

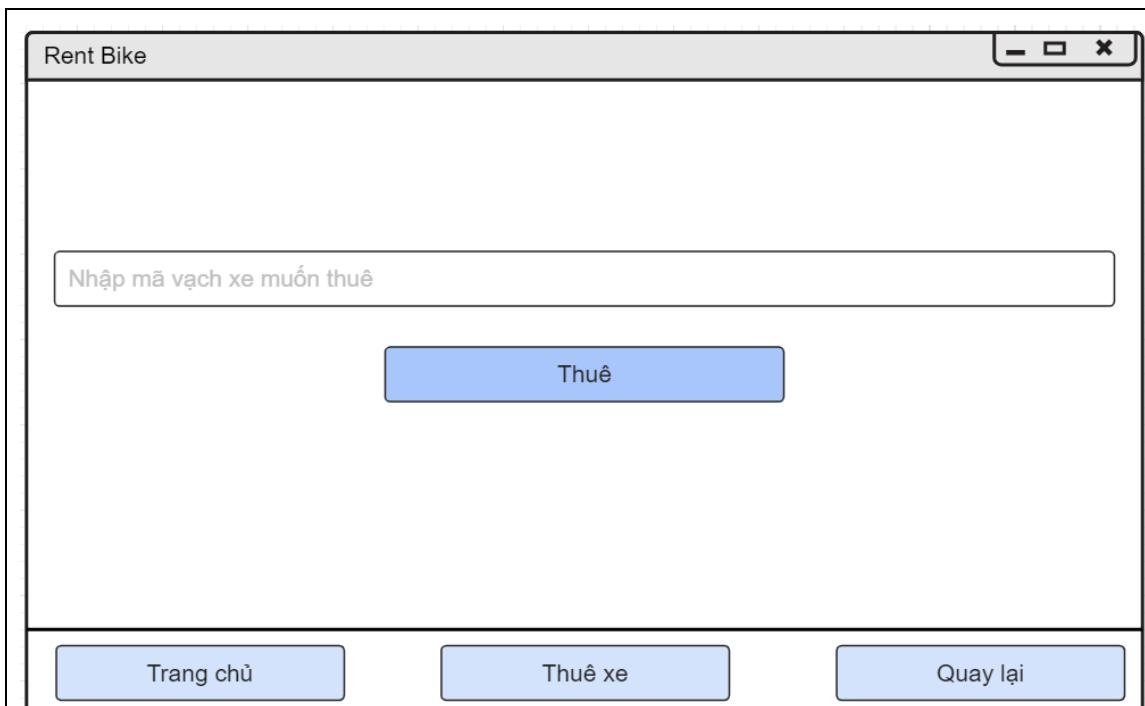
<b>Control</b>	<b>Operation</b>	<b>Function</b>
Search textbox	search	Enter station's name you want to find
Search button	click	Search station base on station's name in search input. If found, display the result. If not, notify
Cancel search button	click	Cancel search result
Station button	click	Display detail information of specified station
Home page button	click	Display home page screen
Rent bike button	click	Display rent bike screen or renting bike screen
Previous form button	click	Display previous form and hide current form

### c. List Bike Screen

The screenshot shows a window titled "List Bike". At the top is a search bar with placeholder text "Nhập mã vạch xe muốn tìm kiếm" and a "Search" button. Below the search bar, there's a section for "Bãi xe số 1 số 12, Kim Hoa, Kim Liên, Hà Nội" and a status box for "Xe đạp thường Còn lại 5 xe". The main area lists five bike entries: "xe số 1: 00000001 - available", "xe số 2: 00000002 - available", "xe số 3: 00000003 - available", "xe số 4: 00000004 - available", and "xe số 5: 00000005 - renting". At the bottom are three buttons: "Trang chủ", "Thuê xe", and "Quay lại".

Control	Operation	Function
Search textbox	search	Enter bike qrcode to find bike
Search button	click	Search bike base on qrcode
Bike detail button	click	Display detail information of bike
RichTextBox station	Initial	Display the station name and station address
RichTextBox bike information	Initial	Display the category bike and remain number of bike in station
Home page button	click	Display home page screen
Rent bike button	click	Display rent bike screen or renting bike screen
Previous form button	click	Display previous form and hide current form

#### d. Rent Bike Screen



The mockup shows a window titled "Rent Bike". Inside, there is a text input field labeled "Nhập mã vạch xe muốn thuê" (Enter QR code of bike you want to rent) and a blue button labeled "Thuê" (Rent). At the bottom, there are three buttons: "Trang chủ" (Home), "Thuê xe" (Rent bike), and "Quay lại" (Back).

Control	Operation	Function
Qrcode textbox	search	Enter qr code of bike you want to rent
Rental button	click	Search bike has this qrcode, if find, display rent bike info screen, if not, notify
Home page button	click	Display home page screen
Rent bike button	click	Display rent bike screen or renting bike screen
Previous form button	click	Display previous form and hide current form

#### e. Card Information Screen

Carf information

Tên chủ thẻ	Nhập tên chủ thẻ
Mã thẻ	Nhập mã thẻ
Ngân hàng phát hành	Chọn ngân hàng phát hành
Mã bảo mật	Nhập mã bảo mật
Ngày hết hạn	Nhập ngày hết hạn
	Hủy
	Xác nhận
Trang chủ	Thuê xe
	Quay lại

Control	Operation	Function
Information textbox	enter	Enter the card information
Combobox	Select	Select bank
Cancel button	click	Cancel card information and return rent bike info screen
Permit button	click	Permit card information and display transaction information screen
Home page button	click	Display home page screen
Rent bike button	click	Display rent bike screen or renting bike screen
Previous form button	click	Display previous form and hide current form

### f. Rent Bike Info Screen

The screenshot shows a window titled "Rent Bike" with the following details:

- Mã xe:** 000000005
- Loại xe:** Xe đạp thường
- Biển số:** không có thông tin
- Tiền cọc:** 100.000

Buttons at the bottom include "Xem thông tin chi tiết của xe này" and "Thuê xe này". Navigation buttons at the bottom are "Trang chủ", "Thuê xe", and "Quay lại".

Control	Operation	Function
Information textbox	initial	Display information of bike you want to rent
See detail information bike button	click	Display bike detail screen
Rent this bike button	click	Display Card information screen to prepare rent this bike
Home page button	click	Display home page screen
Rent bike button	click	Display rent bike screen or renting bike screen
Previous form button	click	Display previous form and hide current form

### g. Renting Bike

The screenshot shows the 'Rent Bike' application window. At the top, it displays the rental time as 'Thời gian đã thuê: 00:30:34' and the remaining power as 'Lượng pin còn lại: không có thông tin'. Below this, there is a preview image of a bicycle. To the right of the image are four data entries: 'Mã xe: 000000005', 'Loại xe: xe đạp thường', 'Biển số: không có thông tin', and 'Hãng sản xuất: Vin Group'. A button labeled 'Chọn bến xe để trả' (Select return station) is located below these entries. At the bottom of the window are three buttons: 'Trang chủ' (Home), 'Thuê xe' (Rent bike), and 'Quay lại' (Back).

Control	Operation	Function
Information textbox	initial	Display information of renting bike
Rented time label	initial	Display the rented time
Remain power label	initial	Display the remain power if electric bike
Choose return station button	click	Display return bike screen
Picture box	display	Display preview image of renting bike
Home page button	click	Display home page screen
Rent bike button	click	Display rent bike screen or renting bike screen
Previous form button	click	Display previous form and hide current form

### h. Return Bike Screen

<b>Control</b>	<b>Operation</b>	<b>Function</b>
Search textbox	enter	Enter the name of station you want to return bike
Search button	click	Search station base on station name in search textbox
Cancel search button	click	Cancel search result
Station button	click	Choose station to return bike
Cancel return bike button	click	Cancel return bike, return Renting Bike Screen
Home page button	click	Display home page screen
Rent bike button	click	Display rent bike screen or renting bike screen
Previous form button	click	Display previous form and hide current form

### i. Station Detail Screen

StationDetail		
Tên	Bãi xe 1	
Địa chỉ	Số 55, Kim hoa, Kim Liên, Hà Nội	
Diện tích	1000 m2	
Số xe	1000	
Khoảng cách	1000 km	
Thời gian	10h	
	Xe đạp đơn	Xe đạp đôi
	Xe đạp điện	
	Trang chủ	Thuê xe
	Quay lại	
Control	Operation	Function
Information textbox	initial	Display information of the station
Bike button	click	Display bikes list in the station
Tandem button	click	Display tandems list in the station
Electric bike button	click	Display electric bike in the station
Home page button	click	Display home page screen
Rent bike button	click	Display rent bike screen or renting bike screen
Previous form button	click	Display previous form and hide current form

### j. Transaction Information Screen

The screenshot shows a window titled "Transaction Information". It contains four input fields with labels in blue boxes:

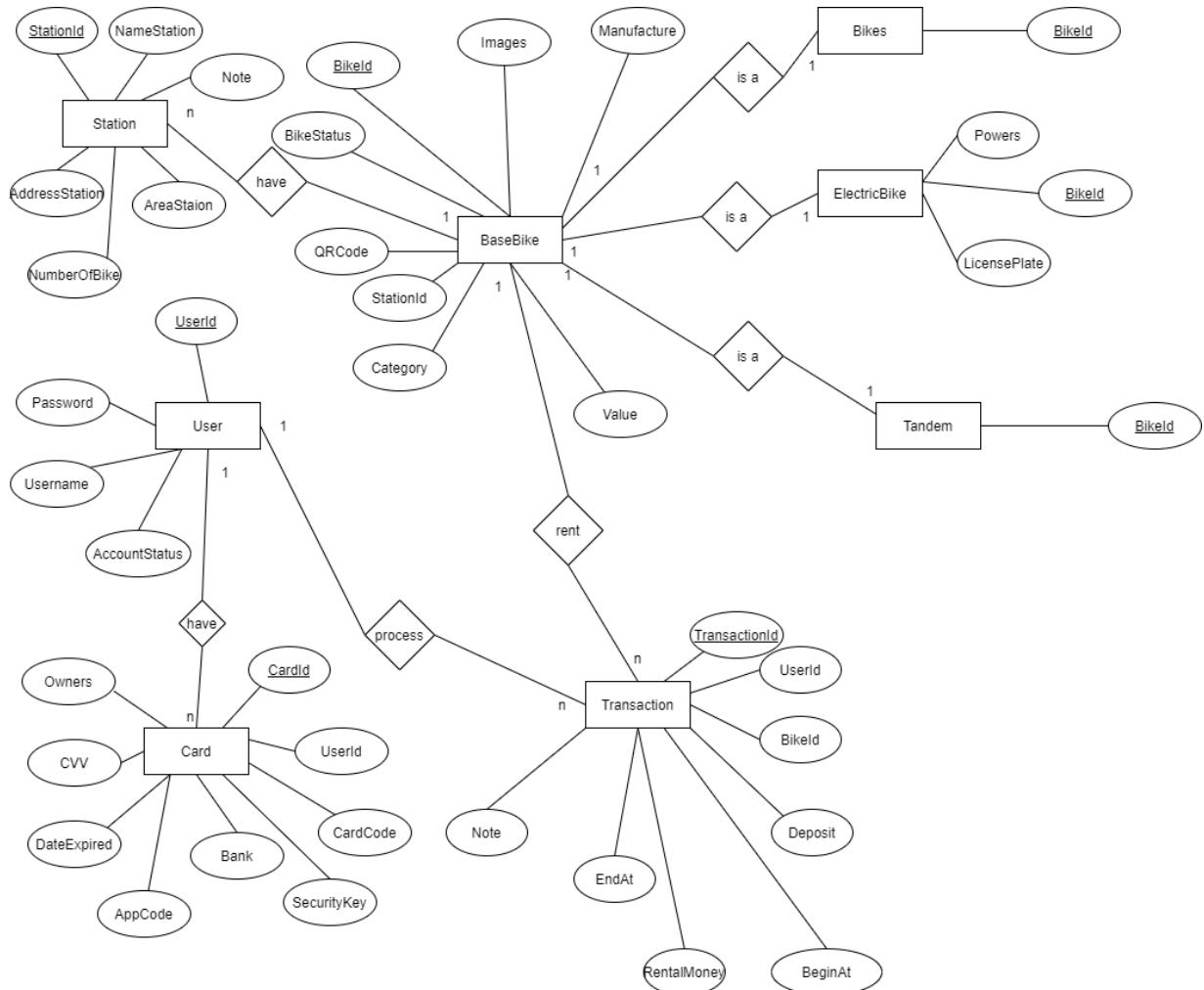
- Tiền cọc: 10.000.000
- Tiền thuê xe: 10.000.000
- Số tiền còn lại: 10.000.000
- Ngày giao dịch: 10/10/2020

Below the input fields are two buttons: "Hủy" (Cancel) and "Xác nhận" (Confirm). At the bottom of the window are three buttons: "Trang chủ" (Home), "Thuê xe" (Rent bike), and "Quay lại" (Back).

Control	Operation	Function
Information textbox	initial	Display the information of transaction
Permit button	click	Permit transaction
Cancel button	click	Cancel transaction
Home page button	click	Display home page screen
Rent bike button	click	Display rent bike screen or renting bike screen
Previous form button	click	Display previous form and hide current form

## 4.2 Data Modeling

### 4.2.1 Conceptual Data Modeling



### 4.2.2 Thiết kế cơ sở dữ liệu

#### 4.2.2.1 Hệ quản trị cơ sở dữ liệu

Nhóm sử dụng hệ quản trị cơ sở dữ liệu SQL server. Hệ quản trị này được nghiên cứu và phát triển bởi Microsoft Inc

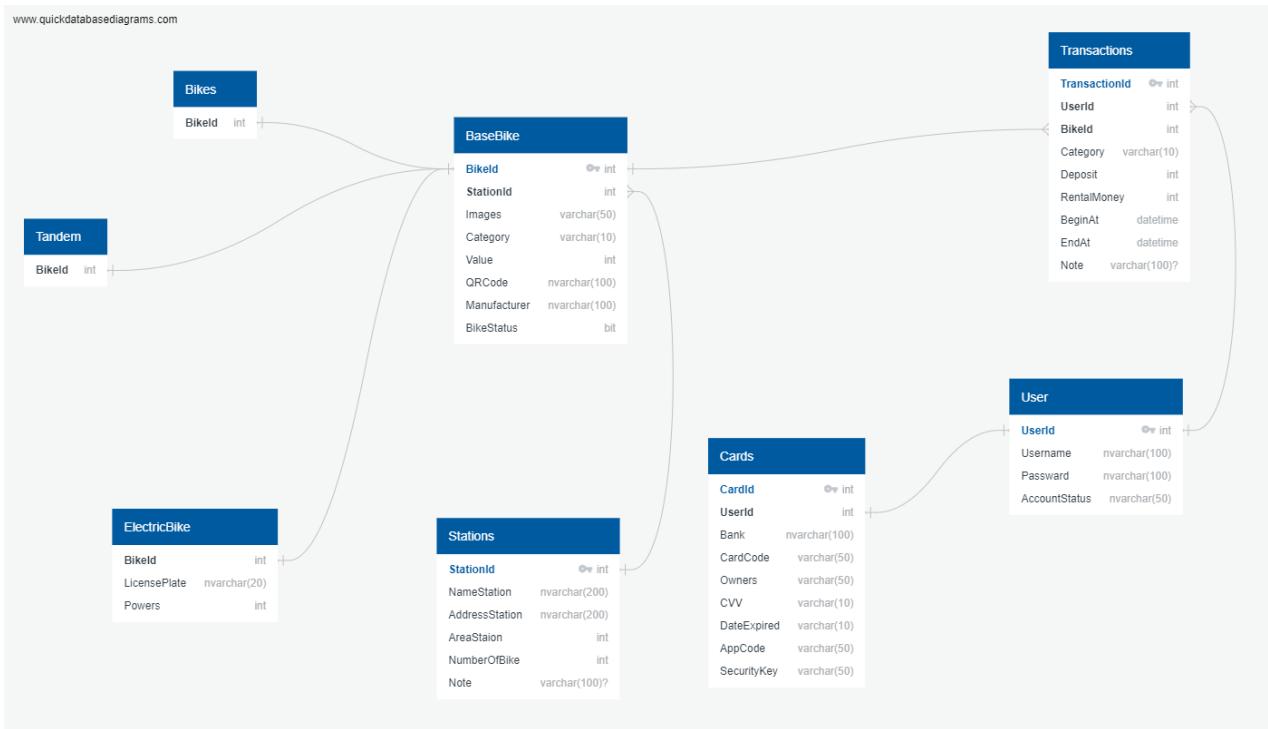
Vào năm 1989, MS SQL Server chính thức được ra mắt. Ngôn ngữ truy vấn đầu tiên của hệ quản trị cơ sở dữ liệu này là T-SQL hay ANSI SQL. Sau nhiều năm, MS SQL Server vẫn luôn nằm trong top các hệ quản trị cơ sở dữ liệu được sử dụng phổ biến nhất.

Một số tính năng của hệ quản trị cơ sở dữ liệu MS SQL Server là:

- Có thể hoạt động tốt trên hệ điều hành Windows và Linux

- Có thể tương tác với Oracle
- Cung cấp và quản lý khôi phục công việc hiệu quả
- Cho phép cùng lúc nhiều người dùng chung một cơ sở dữ liệu

#### 4.2.2.2 Mô hình dữ liệu mức logical



#### 4.2.2.3 Mô hình dữ liệu mức vật lý

##### a. Thành phần Users

#	PK	FK	Column name	Data type	Default value	Mandatory	Description
1	x		UserId	int		Not null	Id của user
2			Username	nvarchar(100)		Not null	Tên đăng nhập của user
3			Password	nvarchar(100)		Not null	Mật khẩu đăng nhập của user

4			AccountStatus	Nvarchar(50)	enable	Not null	Trạng thái của tài khoản
---	--	--	---------------	--------------	--------	----------	--------------------------

b. Thành phần Card

#	PK	FK	Column name	Data type	Default value	Mandatory	Description
1	x		CardId	int		Not null	Id của thẻ
2		x	UserId	int		Not null	Id của user
3			Bank	Nvarchar(100)		Not null	Ngân hàng phát hành thẻ
4			CardCode	Nvarchar(50)		Not null	Mã thẻ
5			Owners	Nvarchar(50)		Not null	Tên chủ sở hữu
6			DateExpired	Nvarchar(10)		Not null	Ngày hết hạn của thẻ
7			AppCode	Varchar(50)		Not null	Mã xác nhận
8			Securitycode	Varchar(50)		Not null	Mã an ninh

c. Thành phần Station

#	PK	FK	Column name	Data type	Default value	Mandatory	Description
1	x		StationId	int		Not null	Id của bãi xe
2			NameStation	Nvarchar(200)		Not null	Tên của bãi xe
3			AddressStation	Nvarchar(200)		Not null	Địa chỉ của bãi xe
4			AreaStation	int		Not null	Diện tích của bãi xe

5			NumberOfBike	int		Not null	Số lượng xe tối đa bãi xe có thể chứa
6			Note	Nvarchar(100)			Ghi chú về bãi xe

d. Thành phần BaseBike

#	PK	FK	Column name	Data type	Default value	Mandatory	Description
1	x		BikeId	int		Not null	Id của xe đạp thường
2		x	StationId	int		Not null	Id của bãi xe chứa xe
3			Images	Varchar(50)		Not null	Ảnh xem trước của xe
4			Category	Varchar(10)		Not null	Loại của xe Bike: xe đạp thường Tandem: xe đạp đôi Electric: xe đạp điện
5			Value	int		Not null	Giá tiền của xe
6			QRCode	Nvarchar(100)		Not null	QR code của xe
7			Manufacture	Nvarchar(100)		Not null	Hãng sản xuất
8			BikeStatus	bit	false	Not null	Trạng thái xe có đang được thuê hay không

							True: đang được thuê  False: chưa được thuê
--	--	--	--	--	--	--	---

e. Thành phần Bikes

#	PK	FK	Column name	Data type	Default value	Mandatory	Description
1	x	x	BikeId	int		Not null	Id của xe đạp thường

f. Thành phần ElectricBike

#	PK	FK	Column name	Data type	Default value	Mandatory	Description
1	x	x	BikeId	int		Not null	Id của xe đạp điện
3			Power	int		Not null	Lượng pin còn lại của xe
4			LicensePlate	Varchar(30)		Not null	Biển số xe

g. Thành phần Tandem

#	PK	FK	Column name	Data type	Default value	Mandatory	Description
1	x	x	BikeId	int		Not null	Id của xe đạp đôi

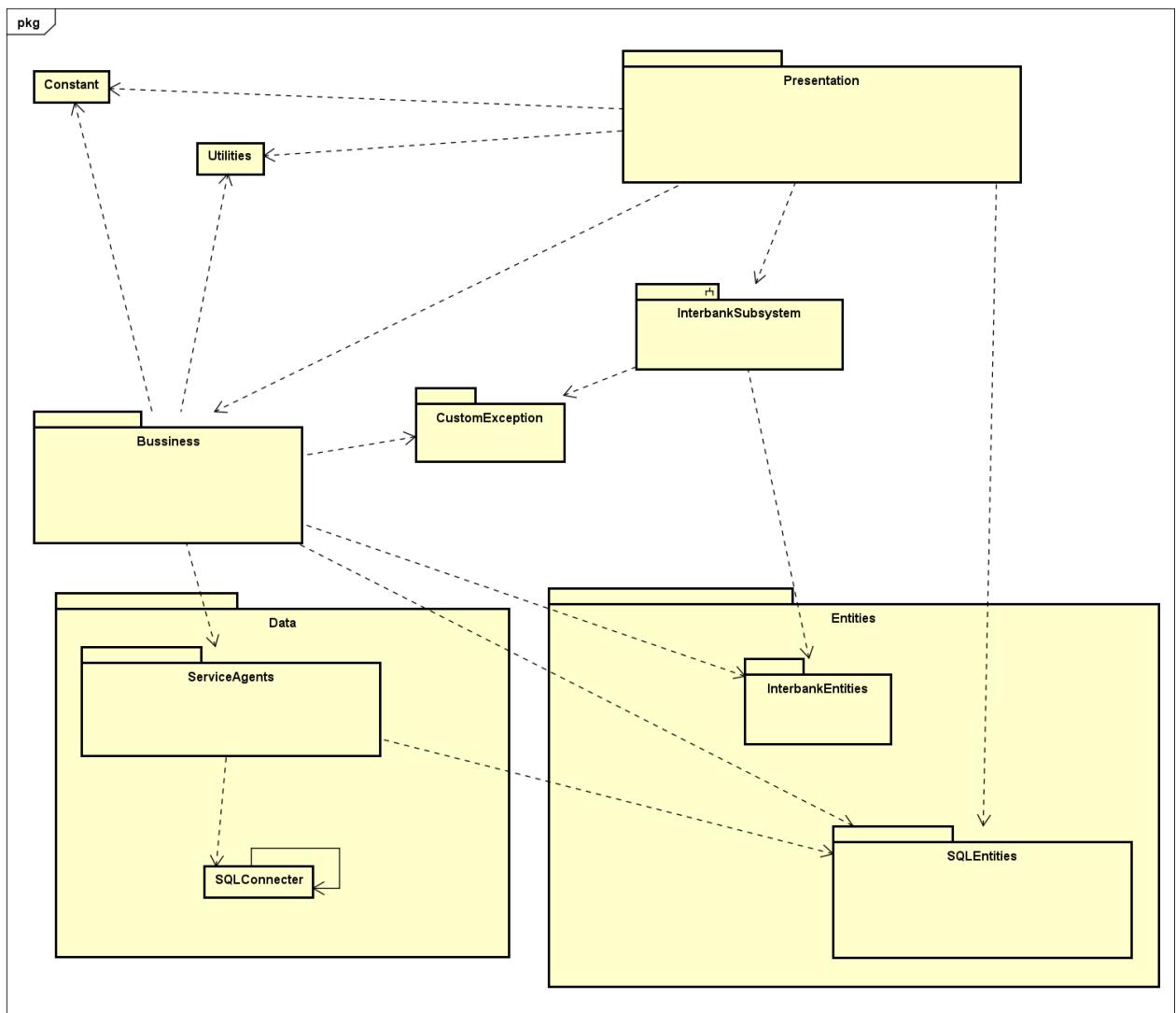
h. Thành phần Transaction

#	PK	FK	Column name	Data type	Default value	Mandatory	Description
1	x		TransactionId	int		Not null	Id của transaction
2		x	UserId	int		Not null	Id của user
3		x	BikeId	Varchar(100)		Not null	Id của xe được thuê
4			Desposit	int		Not null	Tiền cọc của giao dịch
5			RentalMoney	int		Not null	Tiền thuê xe
6			BeginAt	datetime		Not null	Thời gian tạo giao dịch, bắt đầu thuê xe
7			EndAt	datetime		Not null	Thời gian trả xe thành công, kết thúc giao dịch
8			Note	Nvarchar(100)			Ghi chú về giao dịch

### 4.3 Non-Database Management System Files

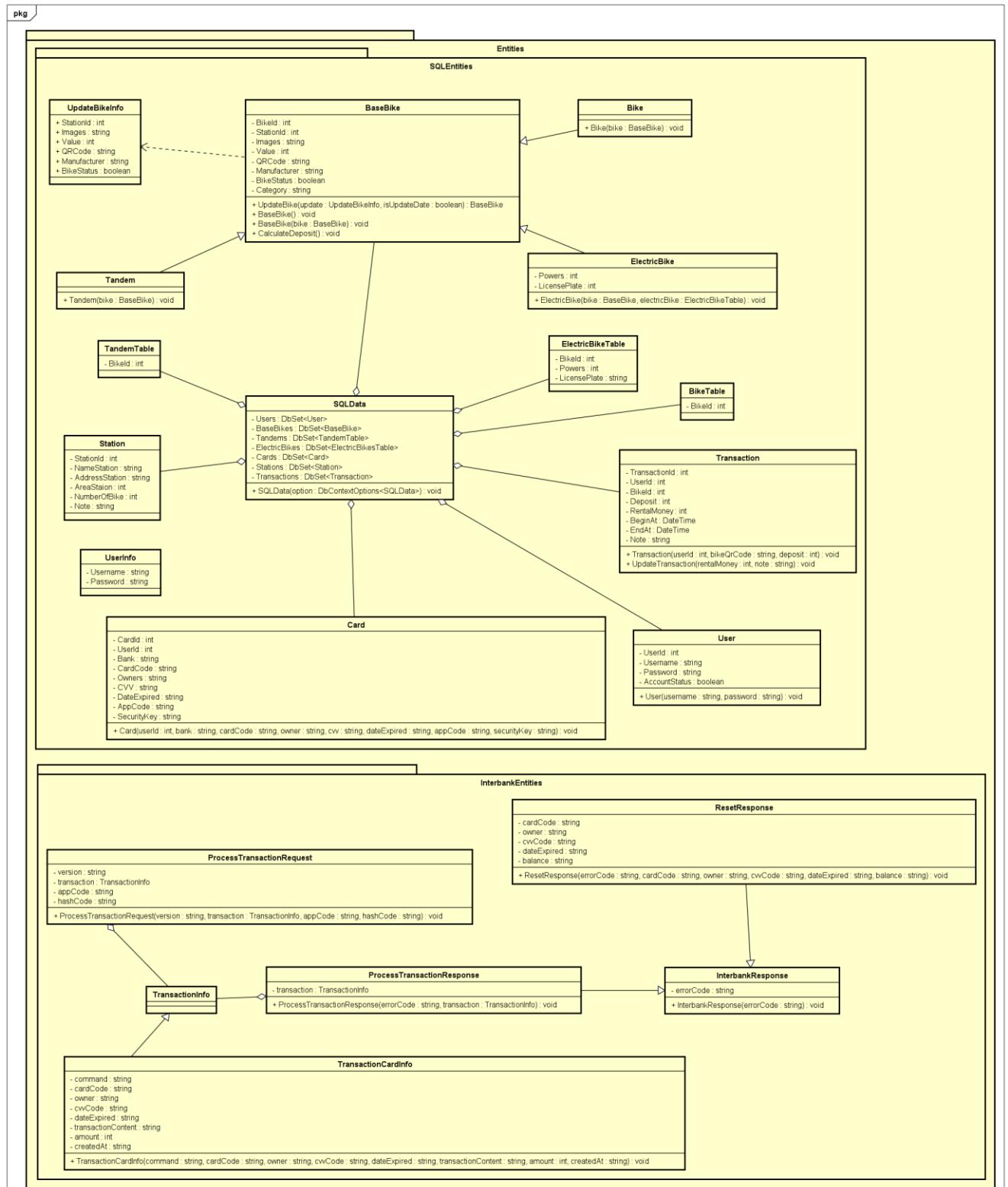
### 4.4 Thiết kế lớp

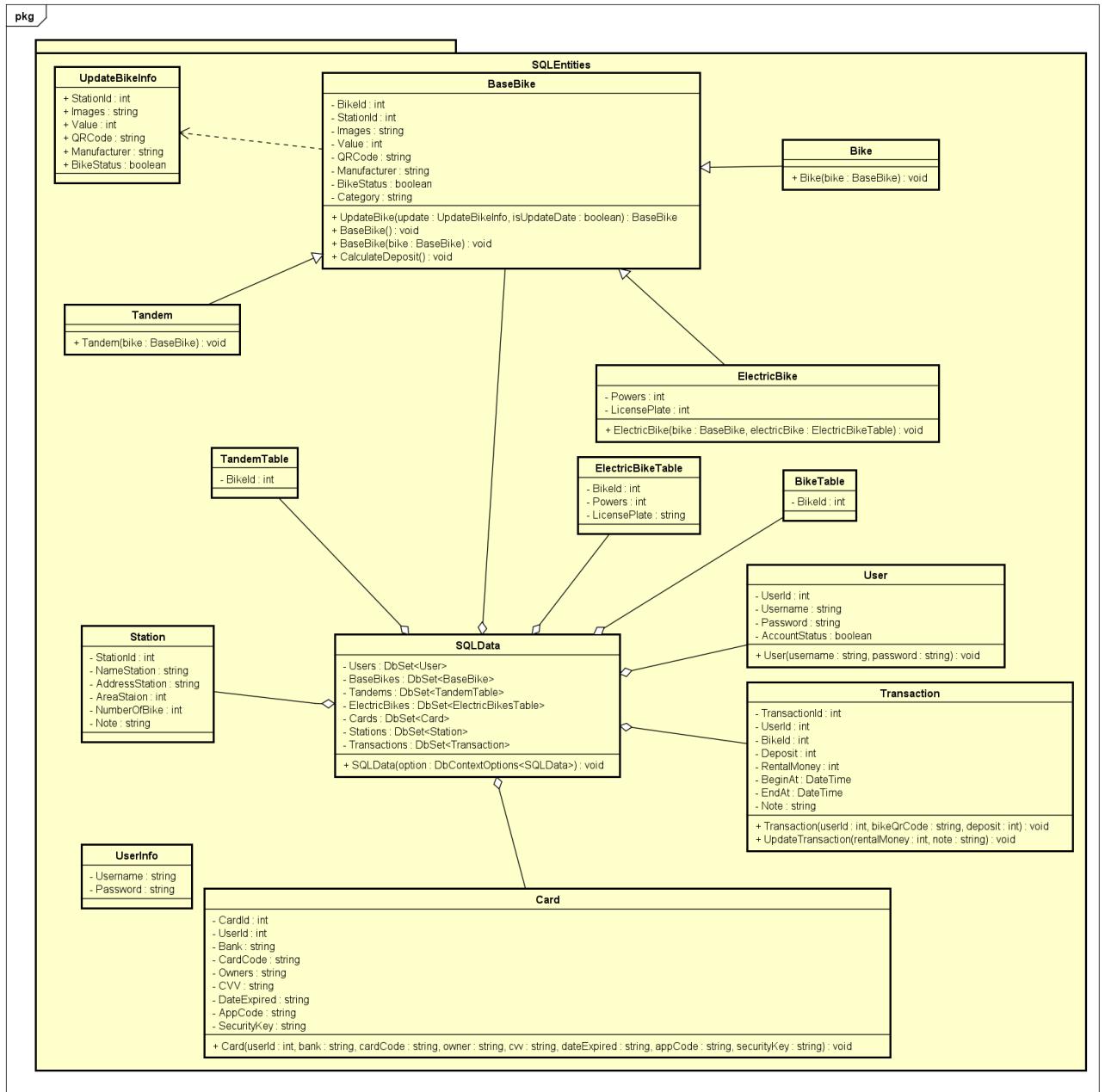
#### 4.4.1 Biểu đồ lớp chung

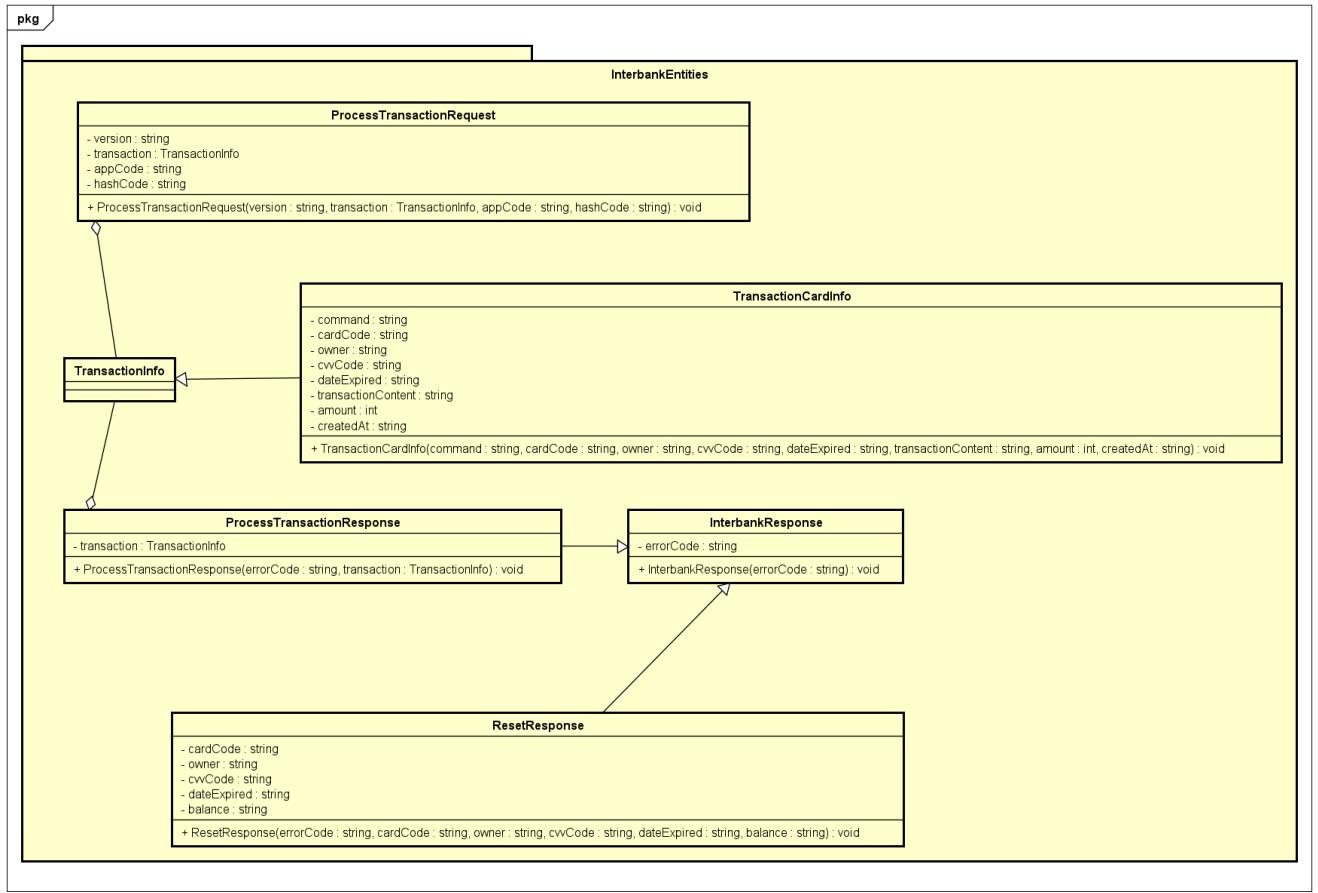


## 4.4.2 Biểu đồ lớp

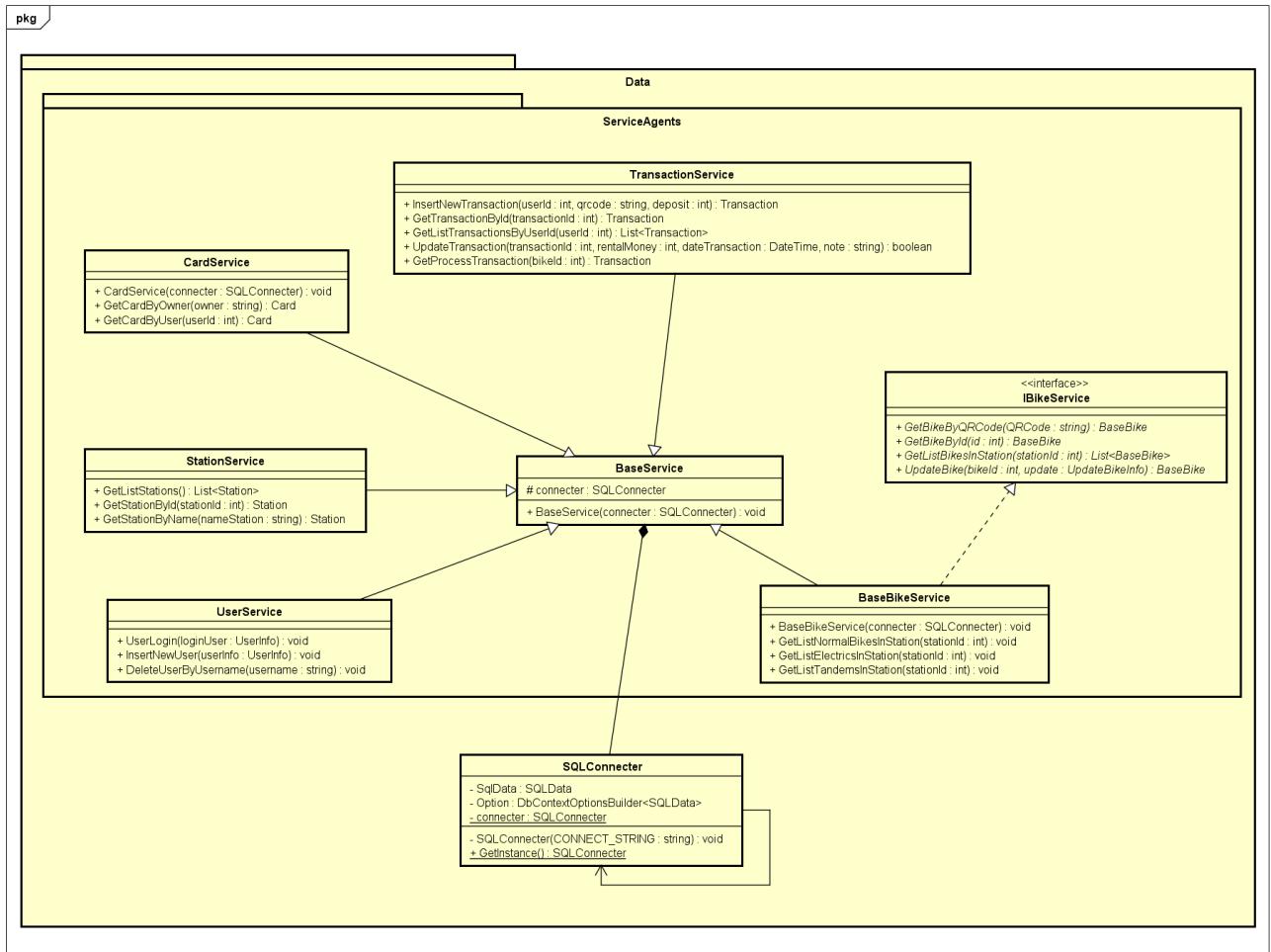
### 4.4.2.1 Biểu đồ lớp của package Entities



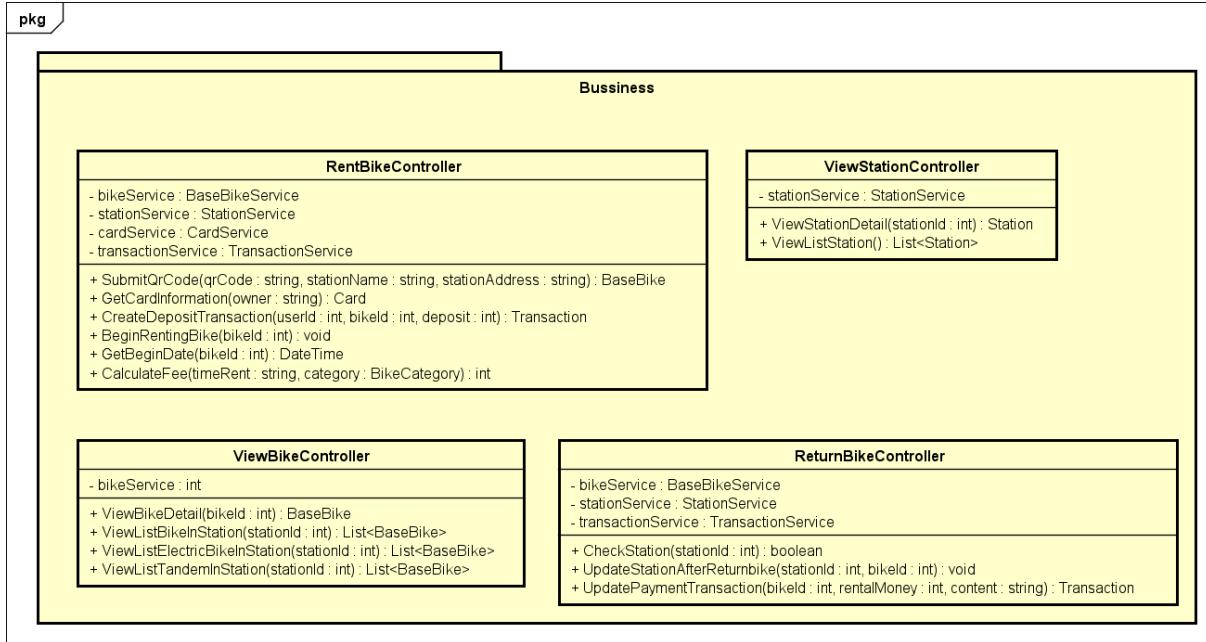




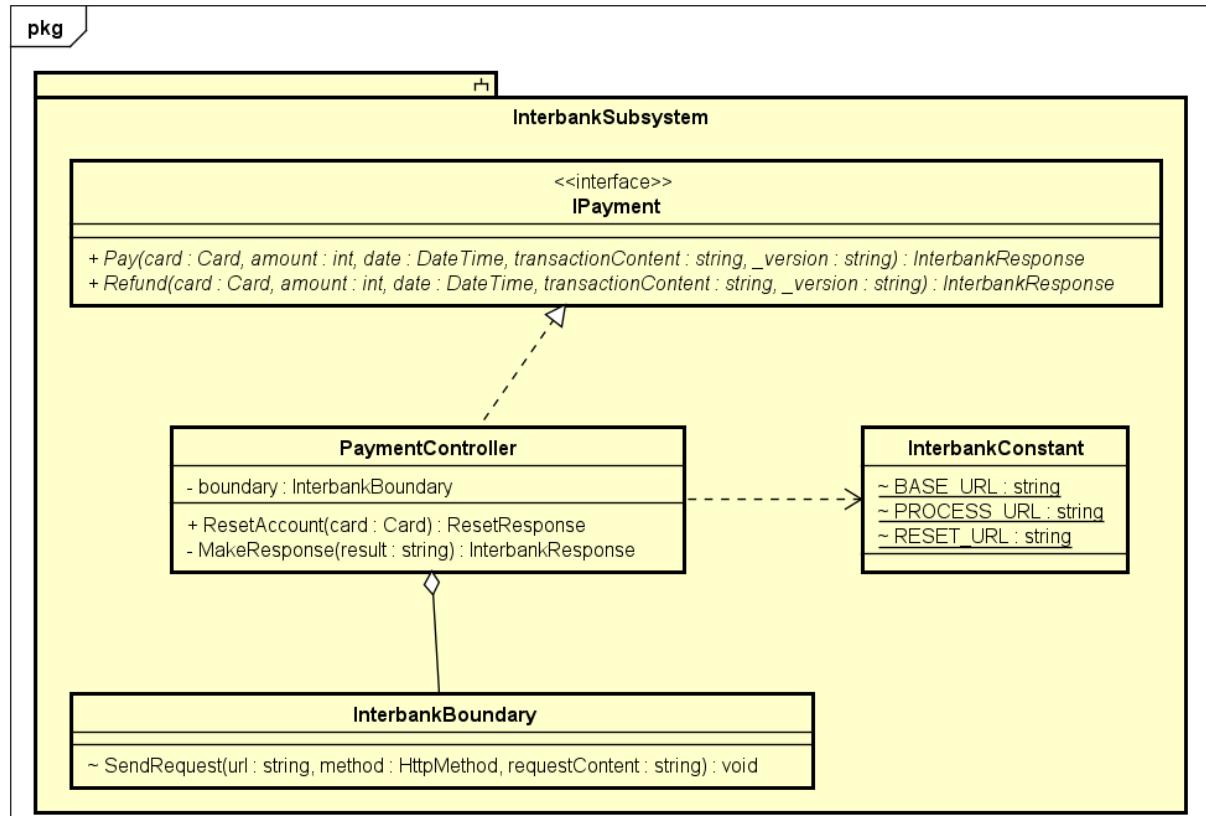
#### 4.4.2.2 Biểu đồ lớp của package Data



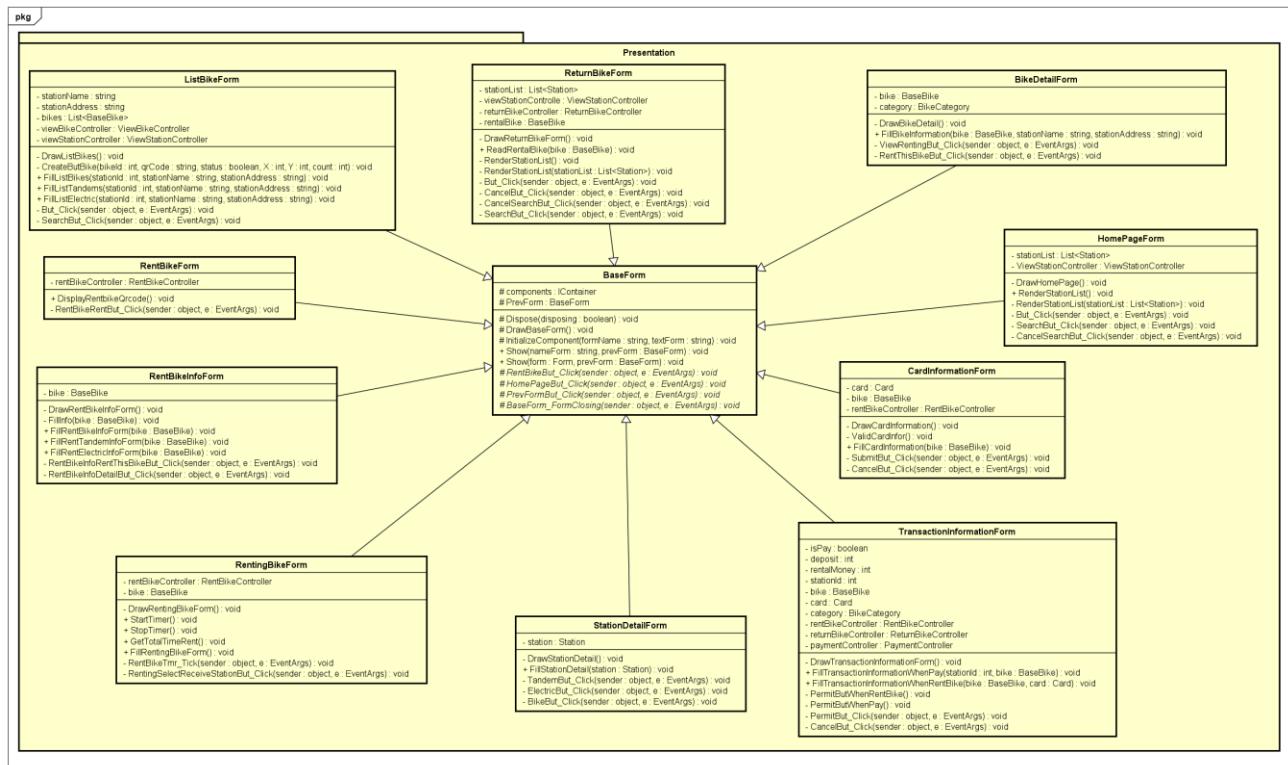
#### 4.4.2.3 Biểu đồ lớp của package Business



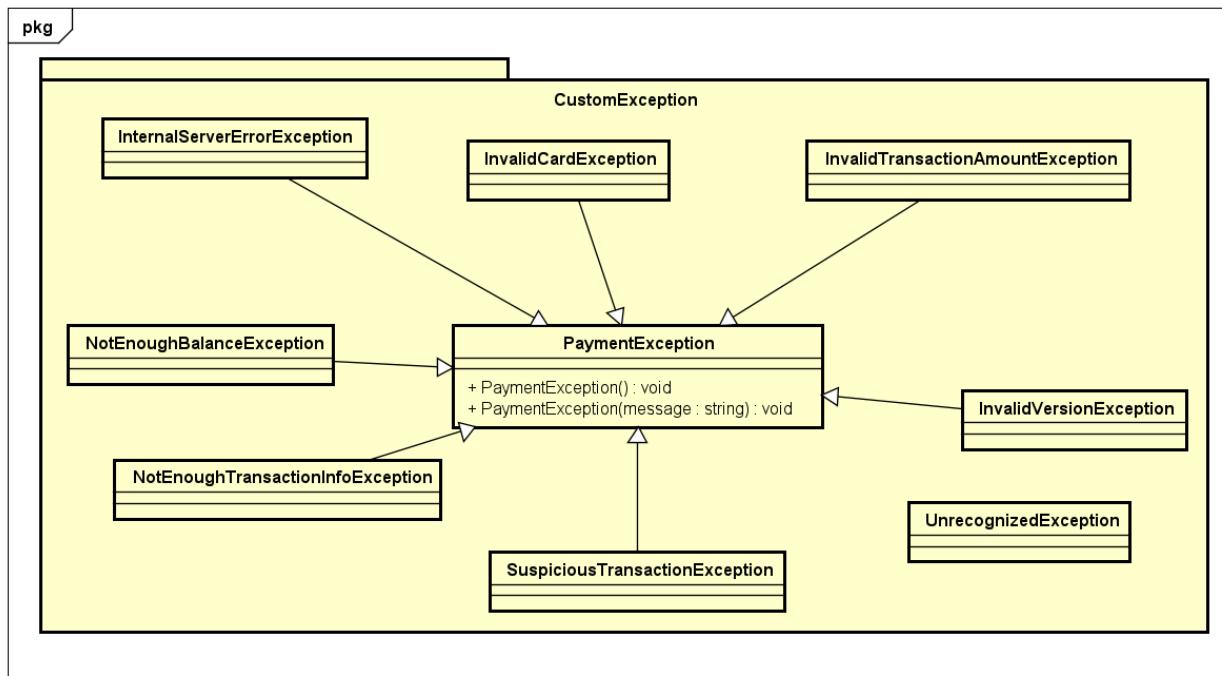
#### 4.4.2.4 Biểu đồ lớp của InterbankSubsystem



#### 4.4.2.5 Biểu đồ lớp của package Presentation



#### 4.4.2.5. Biểu đồ lớp của package CustomException



#### 4.4.3 Thiết kế lớp chi tiết

##### 4.4.3.1 Package Entities

a. Package SQLEntities

###### 1. UpdateBikeInfo



###### Attribute

#	Tên	Kiểu dữ liệu	Giá trị mặc định	Miêu tả
1	StationId	int		Id của bãi xe
2	Images	string		Đường dẫn ảnh xem trước của xe
3	Value	int		Giá tiền của xe
4	QRCode	string		QR code của xe
5	Manufacturer	string		Hãng sản xuất
6	BikeStatus	boolean		Trạng thái xe có đang được thuê hay không True: đang được thuê False: chưa được thuê

**Operation:** Không

**Parameter:** Không

**Exception:** Không

**Method:** Không

## 2. BaseBike

BaseBike	
- BikeId : int	
- StationId : int	
- Images : string	
- Value : int	
- QRCode : string	
- Manufacturer : string	
- BikeStatus : boolean	
- Category : string	
+ UpdateBike(update : UpdateBikeInfo, isUpdateDate : boolean) : BaseBike	
+ BaseBike() : void	
+ BaseBike(bike : BaseBike) : void	
+ CalculateDeposit() : void	

### Attribute

#	Tên	Kiểu dữ liệu	Giá trị mặc định	Miêu tả
1	BikeId	int		Id của xe đạp
2	StationId	int		Id của bãi xe chứa xe
3	Images	string		Ảnh xem trước của xe
4	Value	int		Giá tiền của xe
5	QRCode	string		QR code của xe
6	Manufacturer	string		Hãng sản xuất
7	BikeStatus	boolean	false	Trạng thái xe có đang được thuê hay không True: đang được thuê False: chưa được thuê

8	Category	string		Loại của xe bike: xe đạp thường tandem: xe đạp đôi electric: xe đạp điện
---	----------	--------	--	---

### Operation

#	Tên	Kiểu dữ liệu trả về	Mô tả (mục đích)
1	UpdateBike	BaseBike	Cập nhật thông tin xe
2	BaseBike	void	Lấy thông tin của xe
3	CalculateDeposit	void	Tính toán tiền cọc

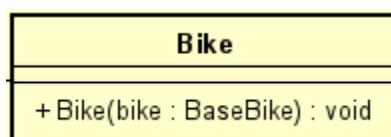
#### Parameter:

- update: thông tin xe cần cập nhật
- isUpdateDate: có cho cập nhật thời gian không
- bike: thông tin cơ bản của xe

#### Exception: Không

#### Method: Không

### 3. Bike



#### Attribute: Không

### Operation

#	Tên	Kiểu dữ liệu trả về	Mô tả (mục đích)
1	Bike	void	lấy thông tin xe đạp thường

**Parameter:**

- bike: thông tin cơ bản của xe

**Exception:** Không

**Method:** Không

#### 4. Tandem



**Attribute:** Không

**Operation**

#	Tên	Kiểu dữ liệu trả về	Mô tả (mục đích)
1	Tandem	void	Lấy thông tin xe đạp đôi

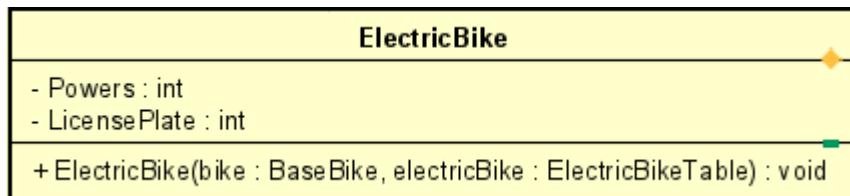
**Parameter:**

- bike: thông tin cơ bản của xe

**Exception:** Không

**Method:** Không

#### 5. ElectricBike



**Attribute**

#	Tên	Kiểu dữ liệu	Giá trị mặc định	Miêu tả
1	Powers	int		Lượng pin còn lại của xe

2	LicensePlate	int		Biển số xe
---	--------------	-----	--	------------

### Operation

#	Tên	Kiểu dữ liệu trả về	Mô tả (mục đích)
1	ElectricBike	void	Lấy thông tin xe đạp điện

**Parameter:**

- bike: thông tin cơ bản của xe
- electricBike: thông tin của xe đạp điện

**Exception:** Không

**Method:** Không

### 6. TandemTable



#### Attribute

#	Tên	Kiểu dữ liệu	Giá trị mặc định	Miêu tả
1	BikeId	int		Id của xe đạp đôi

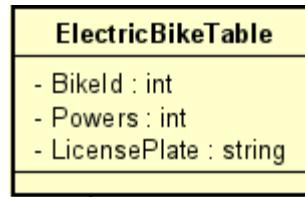
**Operation:** Không

**Parameter:** Không

**Exception:** Không

**Method:** Không

### 7. ElectricBikeTable



### Attribute

#	Tên	Kiểu dữ liệu	Giá trị mặc định	Miêu tả
1	BikeId	int		Id của xe đạp điện
2	Powers	int		Lượng pin còn lại của xe
3	LicensePlate	string		Biển số xe

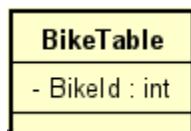
**Operation:** Không

**Parameter:** Không

**Exception:** Không

**Method:** Không

### 8. BikeTable



### Attribute

#	Tên	Kiểu dữ liệu	Giá trị mặc định	Miêu tả
1	BikeId	int		Id của xe đạp thường

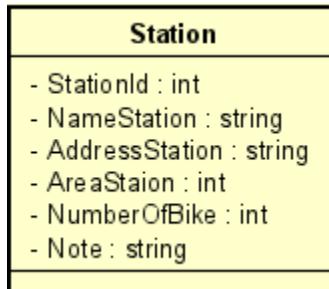
**Operation:** Không

**Parameter:** Không

**Exception:** Không

**Method:** Không

## 9. Station



### Attribute

#	Tên	Kiểu dữ liệu	Giá trị mặc định	Miêu tả
1	StationId	int		Id của bãi xe
2	NameStation	string		Tên của bãi xe
3	AddressStation	string		Địa chỉ của bãi xe
4	AreaStation	int		Diện tích của bãi xe
5	NumberOfBike	int		Số lượng xe tối đa bãi xe có thể chứa
6	Note	string		Ghi chú về bãi xe

**Operation:** Không

**Parameter:** Không

**Exception:** Không

**Method:** Không

## 10.SQLData

SQLData
<ul style="list-style-type: none"> <li>- Users : DbSet&lt;User&gt;</li> <li>- BaseBikes : DbSet&lt;BaseBike&gt;</li> <li>- Tandems : DbSet&lt;TandemTable&gt;</li> <li>- ElectricBikes : DbSet&lt;ElectricBikesTable&gt;</li> <li>- Cards : DbSet&lt;Card&gt;</li> <li>- Stations : DbSet&lt;Station&gt;</li> <li>- Transactions : DbSet&lt;Transaction&gt;</li> </ul>
+ SQLData(option : DbContextOptions<SQLData>) : void

## Attribute

#	Tên	Kiểu dữ liệu	Giá trị mặc định	Miêu tả
1	Users	DbSet<User>		Thông tin bảng User
2	BaseBikes	DbSet<BaseBike>		Thông tin bảng BaseBike
3	Tandems	DbSet<TandemBike>		Thông tin bảng Tandem
4	ElectricBikes	DbSet<ElectricBike>		Thông tin bảng ElectricBike
5	Cards	DbSet<Card>		Thông tin bảng Card
6	Stations	DbSet<Station>		Thông tin bảng Station
7	Transactions	DbSet<Transaction>		Thông tin bảng Transaction

## Operation

#	Tên	Kiểu dữ liệu trả về	Mô tả (mục đích)
1	SQLData	void	Contructor

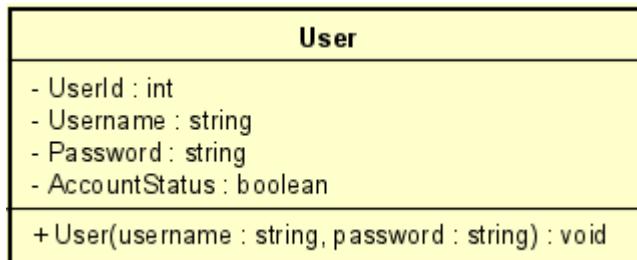
## Parameter:

- option: bảng được lựa chọn để tương tác

**Exception:** Không

**Method:** Không

## 11. User



### Attribute

#	Tên	Kiểu dữ liệu	Giá trị mặc định	Miêu tả
1	UserId	int		Id của user
2	Username	string		Tên đăng nhập của user
3	Password	string		Mật khẩu đăng nhập của user
4	AccountStatus	boolean		Trạng thái của tài khoản

### Operation

#	Tên	Kiểu dữ liệu trả về	Mô tả (mục đích)
1	User	void	Contructor

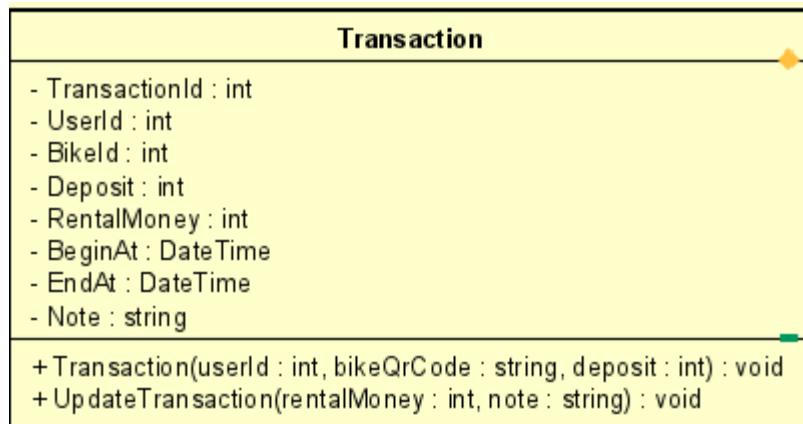
**Parameter:**

- username: tên đăng nhập
- Password: mật khẩu

**Exception:** Không

**Method:** Không

## 12. Transaction



### Attribute

#	Tên	Kiểu dữ liệu	Giá trị mặc định	Miêu tả
1	TransactionId	string		Id của transaction
2	UserId			Id của user
3	BikeId			Id của xe được thuê
4	Deposit			Tiền cọc của giao dịch
5	RentalMoney			Tiền thuê xe
6	BeginAt			Thời gian tạo giao dịch, bắt đầu thuê xe
7	EndAt			Thời gian trả xe thành công, kết thúc giao dịch
8	Note			Ghi chú về giao dịch

### Operation

#	Tên	Kiểu dữ liệu trả về	Mô tả (mục đích)
1	Transaction	void	Contructor
2	UpdateTransaction	void	Cập nhật giao dịch cuối cùng của người dùng sau khi thuê xe đạp

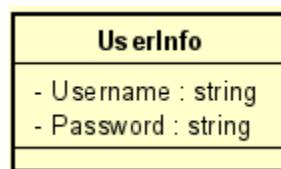
**Parameter:**

- userId: id của user
- bikeQrCode: qrcode của xe
- deposit: tiền cọc
- rentalMoney: tiền thuê
- note: ghi chú

**Exception:** Không

**Method:** Không

### 13. UserInfo



**Attribute**

#	Tên	Kiểu dữ liệu	Giá trị mặc định	Miêu tả
1	Username	string		Tên đăng nhập của user
2	Password	string		Mật khẩu đăng nhập của user

**Operation:** Không

**Parameter:** Không

**Exception:** Không

**Method:** Không

### 14. Card

Card
<ul style="list-style-type: none"> <li>- CardId : int</li> <li>- UserId : int</li> <li>- Bank : string</li> <li>- CardCode : string</li> <li>- Owners : string</li> <li>- CVV : string</li> <li>- DateExpired : string</li> <li>- AppCode : string</li> <li>- SecurityKey : string</li> </ul>
+ Card(userId : int, bank : string, cardCode : string, owner : string, cvv : string, dateExpired : string, appCode : string, securityKey : string) : void

### Attribute

#	Tên	Kiểu dữ liệu	Giá trị mặc định	Miêu tả
1	CardId	int		Id của thẻ
2	UserId	int		Id của user
3	Bank	string		Ngân hàng phát hành thẻ
4	CardCode	string		Mã thẻ
5	Owners	string		Tên chủ sở hữu
6	CVV	string		Mã dùng xác minh thẻ
7	DateExpired	string		Ngày hết hạn của thẻ
8	AppCode	string		Mã xác nhận
9	Securitycode	string		Mã an ninh

### Operation

#	Tên	Kiểu dữ liệu trả về	Mô tả (mục đích)
1	Card	void	Contructor

### Parameter:

- userId: id của user
- bank: tên ngân hàng
- cardCode: mã thẻ
- owner: tên chủ thẻ

- cvv: mã xác minh thẻ
- dateExpired: ngày hết hạn thẻ
- appCode: mã xác nhận
- securityKey: mã an ninh

**Exception:** Không

**Method:** Không

## b. Package InterbankEntities

### 1. *InterbankResponse*



#### Attribute

#	Tên	Kiểu dữ liệu	Giá trị mặc định	Miêu tả
1	errorCode	string		Mã lỗi trả về của response

#### Operation

#	Tên	Kiểu dữ liệu trả về	Mô tả (mục đích)
1	InterbankResponse	void	Contructor

**Parameter:**

- errorCode: mã lỗi

**Exception:** Không

**Method:** Không

## 2. ProcessTransactionRequest

ProcessTransactionRequest
- version : string - transaction : TransactionInfo - appCode : string - hashCode : string
+ ProcessTransactionRequest(version : string, transaction : TransactionInfo, appCode : string, hashCode : string) : void

### Attribute

#	Tên	Kiểu dữ liệu	Giá trị mặc định	Miêu tả
1	version	string		Phiên bản của request
2	transaction	TransactionInfo		Thông tin của request được gửi đi
3	appCode	string		App code của card
4	hashCode	string		Mã hash truyền kèm theo để xác thực

### Operation

#	Tên	Kiểu dữ liệu trả về	Mô tả (mục đích)
1	ProcessTransactionRequest	void	Contructor

#### Parameter:

- version: phiên bản của request
- transaction: thông tin của reuquest gửi đi
- appCode: app code của card
- hashCode: mã hash truyền kèm để xác thực

#### Exception: Không

#### Method: Không

## 3. ProcessTransactionResponse

<b>ProcessTransactionResponse</b>
- transaction : TransactionInfo
+ ProcessTransactionResponse(errorCode : string, transaction : TransactionInfo) : void

## Attribute

#	Tên	Kiểu dữ liệu	Giá trị mặc định	Miêu tả
1	transaction	TransactionInfo		Thông tin transaction được response trả về

## Operation

#	Tên	Kiểu dữ liệu trả về	Mô tả (mục đích)
1	ProcessTransactionResponse	void	Contructor

### Parameter:

- errorCode: mã lỗi
- transaction: thông tin transaction được trả về

### Exception: Không

### Method: Không

## 4. ResetResponse

<b>ResetResponse</b>
- cardCode : string - owner : string - cvvCode : string - dateExpired : string - balance : string
+ ResetResponse(errorCode : string, cardCode : string, owner : string, cvvCode : string, dateExpired : string, balance : string) : void

## Attribute

#	Tên	Kiểu dữ liệu	Giá trị mặc định	Miêu tả
1	cardCode	string		Mã code của thẻ
2	owner	string		Tên người sở hữu thẻ

3	cvvCode	string		Cvv code của thẻ
4	dateExpired	string		Ngày hết hạn của thẻ
5	balance	string		Số tiền còn lại trong thẻ

## Operation

#	Tên	Kiểu dữ liệu trả về	Mô tả (mục đích)
1	ResetResponse	void	Contructor

### Parameter:

- errorCode: mã lỗi
- cardCode: mã code của thẻ
- owner: tên người sở hữu thẻ
- cvvCode: cvv code của thẻ
- dateExpired: ngày hết hạn của thẻ
- balance: số tiền còn lại trong thẻ

### Exception: Không

### Method: Không

## 5. TransactionCardInfo

TransactionCardInfo	
- command : string	
- cardCode : string	
- owner : string	
- cvvCode : string	
- dateExpired : string	
- transactionContent : string	
- amount : int	
- createdAt : string	
+ TransactionCardInfo(command : string, cardCode : string, owner : string, cvvCode : string, dateExpired : string, transactionContent : string, amount : int, createdAt : string) : void	

## Attribute

#	Tên	Kiểu dữ liệu	Giá trị mặc định	Miêu tả
1	command	string		Phương thức của giao dịch
2	cardCode	string		Mã của thẻ
3	owner	string		Chủ sở hữu của thẻ

4	cvvCode	string		Cvv code của thẻ
5	dateExpired	string		Ngày hết hạn của thẻ
6	transactionContent	string		Miêu tả về giao dịch
7	amount	int		Số tiền trong giao dịch
8	createAt	string		Ngày thực hiện giao dịch

### Operation

#	Tên	Kiểu dữ liệu trả về	Mô tả (mục đích)
1	TransactionInfo	void	Contructor

#### Parameter:

- command: phương thức thanh toán
- cardCode: mã code của thẻ
- owner: tên người sở hữu thẻ
- cvvCode: cvv code của thẻ
- dateExpired: ngày hết hạn của thẻ
- transactionContent: miêu tả về giao dịch
- amount: số tiền của giao dịch
- createAt: ngày thực hiện giao dịch

#### Exception: Không

### 6. TransactionInfo



#### 4.4.3.2 package Data

##### a. Package ServiceAgents

### 1. TransactionService

TransactionService
+ InsertNewTransaction(userId : int, qrcode : string, deposit : int) : Transaction + GetTransactionById(transactionId : int) : Transaction + GetListTransactionsByUserId(userId : int) : List<Transaction> + UpdateTransaction(transactionId : int, rentalMoney : int, dateTransaction : DateTime, note : string) : boolean + GetProcessTransaction(bikeId : int) : Transaction

### Attribute:Không

### Operation

#	Tên	Kiểu dữ liệu trả về	Mô tả (mục đích)
1	InsertNewTransaction	Transaction	Thêm mới một giao dịch
2	GetTransactionById	Transaction	Lấy thông tin giao dịch theo id
3	GetListTransactionsByUserId	List<Transaction>	Lấy tất cả thông tin giao dịch của một người dùng theo userid
4	UpdateTransaction	boolean	Cập nhật lại giao dịch
5	GetProcessTransaction	Transaction	Nhận thông tin quá trình giao dịch

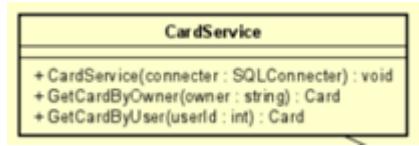
### Parameter:

- userId: thông tin thẻ
- qrcode: mã qr của xe
- deposit: tiền gửi
- transactionId: id của transaction
- rentalMoney: tiền thuê xe
- dateTransaction: Ngày giao dịch
- note: ghi chú
- bikeId: id của xe

### Exception:

Không

## 2.CardService



**Attribute:Không**

### Operation

#	Tên	Kiểu dữ liệu trả về	Mô tả (mục đích)
1	CardService	void	Gửi request connect đến data
2	GetCardByOwner	Card	Lấy thông thẻ theo chủ sở hữu
3	GetCardByUser	Card	Lấy tất cả thẻ theo userid

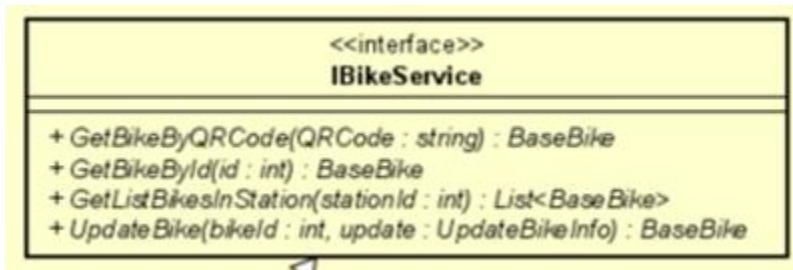
### Parameter:

- owner: chủ sở hữu
- connector: SQLConnecter
- userID:id của user

### Exception:

Không

## 3.IBikeService



**Attribute:Không**

### Operation

#	Tên	Kiểu dữ liệu trả về	Mô tả (mục đích)
1	GetBikeByQRCode	BaseBike	Lấy thông tin bike theo QRcode
2	GetBikeById	BaseBike	Lấy thông tin bike theo id
3	GetListBikesInStation	List<BaseBike>	Lấy thông tin các bike theo bãi gửi xe
4	UpdateBike	BaseBike	Cập nhật thông tin bike

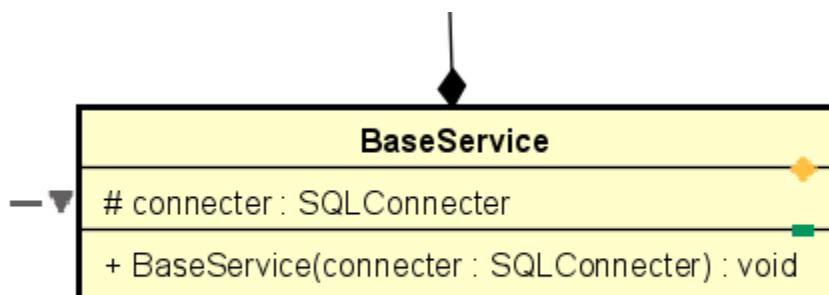
**Parameter:**

- QRCode: mã code của xe
- id: id của xe
- stationId:id của bãi xe
- bikeid:id của xe
- update:UpdateBikeInfo

**Exception:**

Không

#### 4. BaseService



#### Attribute

#	Tên	Kiểu dữ liệu	Giá trị mặc định	Miêu tả
1	connecter	SQLConnecter		Connect đến database

### Operation

#	Tên	Kiểu dữ liệu trả về	Mô tả (mục đích)
1	BaseService	Void	Connect đến Databasea

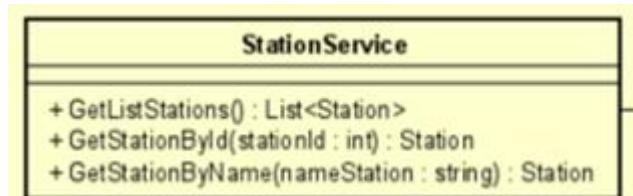
**Parameter:**

- connecter:SQLConnecter

**Exception:**

- Không

### 5.StationService



### Attribute

### Operation

#	Tên	Kiểu dữ liệu trả về	Mô tả (mục đích)
1	GetListStations	List<Station>	Lấy ra thông tin list bãi xe
2	GetStationById	Station	Lấy ra thông tin bãi xe theo id
3	GetStationByName	Station	Lấy ra thông tin bãi xe theo tên bãi xe

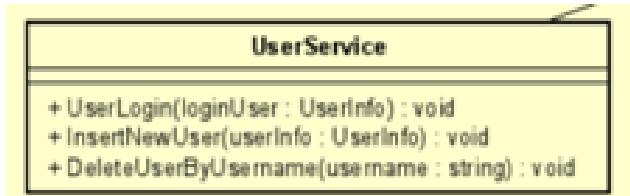
**Parameter:**

- stationID:id của bãi xe
- nameStation:tên của bãi xe

***Exception:***

- Không

## 6.UserService



### Attribute

### Operation

#	Tên	Kiểu dữ liệu trả về	Mô tả (mục đích)
1	UserLogin	void	Login user
2	InsertNewUser	void	Thêm user mới
3	DeleteUserByUsername	void	Xóa user theo tên user

***Parameter:***

- loginUser: thông tin User UserInfo
- userInfo:UserInfo thông tin user
- username:Tên user

***Exception:***

- Không

## 7.BaseBikeService



## Attribute

### Operation

#	Tên	Kiểu dữ liệu trả về	Mô tả (mục đích)
1	BaseBikeService	void	Connect đến database lấy dữ liệu
2	GetListNormalBikesInStation	void	Lấy list xe thường trong bãi xe
3	GetListElectricsInStation	void	Lấy list xe điện trong bãi xe
4	GetListTandemsInStation	void	Lấy list xe đôi trong bãi xe

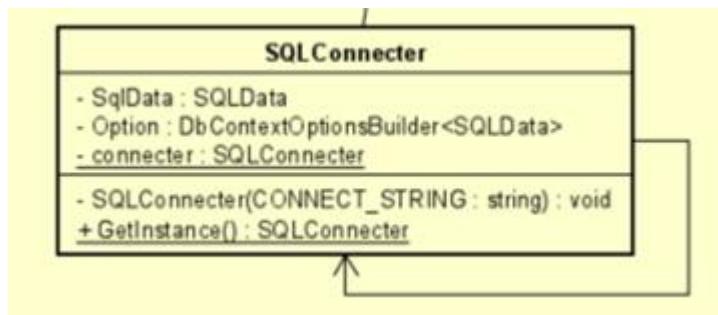
### Parameter:

- stationId: id của bãi xe

### Exception:

- Không

## b.SQLConnecter



### Attribute

#	Tên	Kiểu dữ liệu	Giá trị mặc định	Miêu tả
1	SqlData	SQLData		
2	Option	DbContextOptionsBuilder<SQLData>		Chọn kiểu
3	connecter	SQLConnecter		Connect đến database

### Operation

#	Tên	Kiểu dữ liệu trả về	Mô tả (mục đích)
1	SQLConnecter	Void	Connect đến Databasea
2	GetInstance	SQLConnecter	Lấy Instance

**Parameter:**

- CONNECT\_STRING: hằng số truyền vào để connectDatabase

**Exception:**

- Không

### 4.4.3.3 package Bussiness

#### 1.RentBikeController

RentBikeController	
- bikeService : BaseBikeService	
- stationService : StationService	
- cardService : CardService	
- transactionService : TransactionService	
+ SubmitQrCode(qrCode : string, stationName : string, stationAddress : string) : BaseBike	
+ GetCardInformation(owner : string) : Card	
+ CreateDepositTransaction(userId : int, bikeld : int, deposit : int) : Transaction	
+ BeginRentingBike(bikeld : int) : void	
+ GetBeginDate(bikeld : int) : DateTime	
+ CalculateFee(timeRent : string, category : BikeCategory) : int	

## Attribute

#	Tên	Kiểu dữ liệu	Giá trị mặc định	Miêu tả
1	bikeService	BaseBikeService		
2	stationService	StationService		
3	cardService	CardService		
4	transactionService	TransactionService		

## Operation

#	Tên	Kiểu dữ liệu trả về	Mô tả (mục đích)
1	SubmitQrCode	BaseBike	Submit mã qrcode
2	GetCardInformation	Card	Lấy thông tin card
3	CreateDepositTransaction	Transaction	Theo giao dịch mới
4	BeginRentingBike	void	Gửi yêu cầu bắt đầu thuê
5	GetBeginDate	DateTime	Lấy thông tin ngày bắt đầu
6	CalculateFee	int	Tính tiền

**Parameter:**

- qrCode: mã qrcode của xe
- stationName:tên của bãi xe
- stationAddress:địa chỉ của bãi xe
- owner:Tên chủ sở hữu card
- userid:id của user
- bikeid:id của xe
- deposit:tiền gửi
- timeRent:thời gian thuê
- category: loại xe

**Exception:**

- Không

## 2.ViewBikeController

ViewBikeController	
- bikeService : int	
+ ViewBikeDetail(bikeId : int) : BaseBike	
+ ViewListBikeInStation(stationId : int) : List<BaseBike>	
+ ViewListElectricBikeInStation(stationId : int) : List<BaseBike>	
+ ViewListTandemInStation(stationId : int) : List<BaseBike>	

### Attribute

#	Tên	Kiểu dữ liệu	Giá trị mặc định	Miêu tả
1	bikeService	int		

### Operation

#	Tên	Kiểu dữ liệu trả về	Mô tả (mục đích)
1	ViewBikeDetail	BaseBike	Xem chi tiết xe

2	ViewListBikeInStation	List<BaseBike>	Xem danh sách xe thường trong bãi xe
3	ViewListElectricBikeInStation	List<BaseBike>	Xem danh sách xe điện trong bãi xe
4	ViewListTandemInStation	List<BaseBike>	Xem danh sách xe đôi trong bãi xe

**Parameter:**

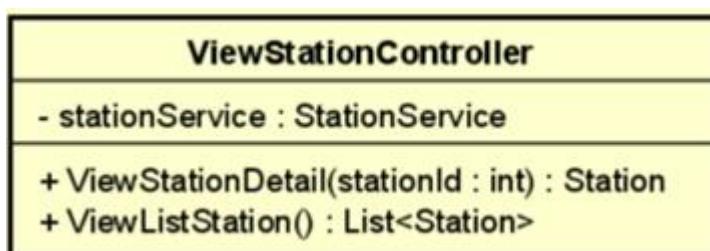
-stationId:id của bãi xe

-bikeid:id của xe

**Exception:**

- Không

### 3. ViewStationController



**Attribute**

#	Tên	Kiểu dữ liệu	Giá trị mặc định	Miêu tả
1	stationService	StationService		

**Operation**

#	Tên	Kiểu dữ liệu trả về	Mô tả (mục đích)
1	ViewStationDetail	Station	Xem chi tiết bãi xe

2	ViewListStation	List< Station >	Xem danh sách bãi xe
---	-----------------	-----------------	----------------------

**Parameter:**

-stationId:id của bãi xe

**Exception:**

- Không

#### 4.ReturnBikeController

ReturnBikeController	
- bikeService : BaseBikeService	
- stationService : StationService	
- transactionService : TransactionService	
+ CheckStation(stationId : int) : boolean	
+ UpdateStationAfterReturnbike(stationId : int, bikeId : int) : void	
+ UpdatePaymentTransaction(bikeId : int, rentalMoney : int, content : string) : Transaction	

#### Attribute

#	Tên	Kiểu dữ liệu	Giá trị mặc định	Miêu tả
1	bikeService	BaseBikeService		
2	stationService	StationService		
3	transactionService	TransactionService		

#### Operation

#	Tên	Kiểu dữ liệu trả về	Mô tả (mục đích)
1	CheckStation	boolean	Xem trạng thái bãi xe

2	UpdateStationAfterReturnbike	void	Update lại thông tin bãi xe
3	UpdatePaymentTransaction	Transaction	Update lại thông tin giao dịch

**Parameter:**

- stationId:id của bãi xe
- bikeid:id của xe
- rentalMoney:Tiền cho thuê
- content:nội dung

**Exception:**

- Không

#### 4.4.3.4 subsystem InterbankSubsystem

##### 1. InterbankBoundary

InterbankBoundary	
+ SendRequest(url : string, method : HttpMethod, requestContent : string) : string	

**Attribute: không**

**Operation**

#	Tên	Kiểu dữ liệu trả về	Mô tả (mục đích)
1	SendRequest	string	Gửi request và nhận lại response

**Parameter:**

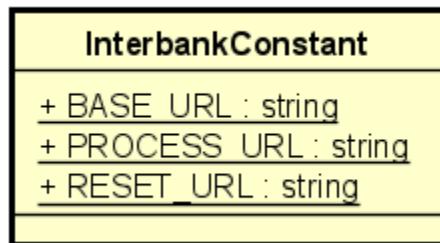
- url: url
- httpMethod: phương thức của API
- requestContent: body của request

**Exception**

- UnrecognizedException

**Method:** Không

## 2. InterbankConstant



**Attribute**

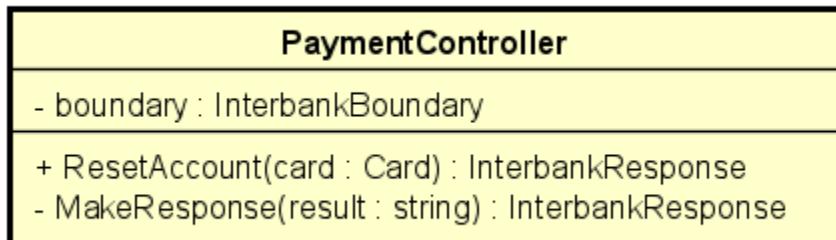
#	Tên	Kiểu dữ liệu	Giá trị mặc định	Miêu tả
1	BASE_URL	string(const)	https://ecopark-system-api.herokuapp.com	Base url của API
2	PROCESS_URL	string(const)	/api/card/processTransaction	url của API thanh toán
3	RESET_URL	string(const)	/api/card/reset-balance	url của API reset thẻ

**Operation:** không

**Parameter:** không

**Exception:** Không

## 3. PaymentController



**Attribute**

#	Tên	Kiểu dữ liệu	Giá trị mặc định	Miêu tả

1	boundary	InterbankBoundary		Giúp gửi request đến server
---	----------	-------------------	--	-----------------------------

## Operation

#	Tên	Kiểu dữ liệu trả về	Mô tả (mục đích)
1	ResetAccount	InterbankResponse	Reset số tiền trong tài khoản
2	MakeResponse	InterbankResponse	Gửi trả response nếu giao dịch thành công, hoặc trả lại exception

### Parameter:

- card: thông tin thẻ
- result: response dưới dạng string

### Exception:

- InvalidCardException
- NotEnoughBalanceException
- InternalServerErrorException
- SuspiciousTransactionException
- NotEnoughTransactionInfoException
- InvalidVersionException
- InvalidTransactionAmountException
- UnrecognizedException

## 4. IPayment

<pre>&lt;&lt;interface&gt;&gt; IPayment</pre>	
<pre>+ Pay(card : Card, amount : int, date : DateTime, transactionContent : string, _version : string) : InterbankResponse + Refund(card : Card, amount : int, date : DateTime, transactionContent : string, _version : string) : InterbankResponse</pre>	

### Attribute: không

## Operation

#	Tên	Kiểu dữ liệu trả về	Mô tả (mục đích)
1	Pay	InterbankResponse	Phương thức thực hiện thanh toán

2	Refund	InterbankResponse	Phương thức để trả lại tiền cho khách hàng
---	--------	-------------------	--

**Parameter:**

- amount: số tiền của giao dịch
- card: thông tin thẻ
- date: thời gian thực hiện giao dịch
- transactionContent: thông tin của giao dịch
- \_version: phiên bản của request gửi đi

**Exception: Không**

#### 4.4.3.5 package Presentation

##### 1. BaseForm

BaseForm	
# components : IContainer	
# PrevForm : BaseForm	
# Dispose(disposing : boolean) : void	
# DrawBaseForm() : void	
# InitializeComponent(formName : string, textForm : string) : void	
+ Show(nameForm : string, prevForm : BaseForm) : void	
+ Show(form : Form, prevForm : BaseForm) : void	
# RentBikeBut_Click(sender : object, e : EventArgs) : void	
# HomePageBut_Click(sender : object, e : EventArgs) : void	
# PrevFormBut_Click(sender : object, e : EventArgs) : void	
# BaseForm_FormClosing(sender : object, e : EventArgs) : void	

##### Attribute

- Không

##### Operation

#	Tên	Kiểu dữ liệu trả về	Mô tả (mục đích)
1	Show	void	Hiển thị giao diện BaseForm

**Parameter:**

- nameForm: tên giao diện
- form: giao diện

- prevForm: giao diện trước

**Exception:**

- không

## 2. ListBikeForm

ListBikeForm	
- stationName : string	
- stationAddress : string	
- bikes : List<BaseBike>	
- viewBikeController : ViewBikeController	
- viewStationController : ViewStationController	
- DrawListBikes() : void	
- CreateButBike(bikeld : int, qrCode : string, status : boolean, X : int, Y : int, count : int) : void	
+ FillListBikes(stationId : int, stationName : string, stationAddress : string) : void	
+ FillListTandems(stationId : int, stationName : string, stationAddress : string) : void	
+ FillListElectric(stationId : int, stationName : string, stationAddress : string) : void	
- But_Click(sender : object, e : EventArgs) : void	
- SearchBut_Click(sender : object, e : EventArgs) : void	

### Attribute

#	Tên	Kiểu dữ liệu	Giá trị mặc định	Miêu tả
1	stationName	String		Tên của bãi xe
2	stationAddress	String		Địa chỉ của bãi xe
3	Bikes	List<BaseBike>		Danh sách xe hiển thị lên giao diện
4	viewBikeController	ViewBikeController		Xử lý thông tin request, trả về thông tin xe
5	viewStationController	ViewStationController		Xử lý thông tin request, trả về thông tin bãi xe

### Operation

#	Tên	Kiểu dữ liệu trả về	Mô tả (mục đích)
1	DrawListBike	Void	Hiển thị danh sách xe
2	FillListBikes	Void	Hiển thị danh sách xe đạp đơn
3	FillListTandems	Void	Hiển thị danh sách xe đạp đôi

4	FillListElectric	Void	Hiển thị danh sách xe đạp điện
---	------------------	------	--------------------------------

**Parameter:**

- stationId: mã bãi xe
- stationName: tên bãi xe
- stationAddress: địa chỉ bãi xe

**Exception:**

- Không

### 3. RentBikeForm

RentBikeForm
- rentBikeController : RentBikeController
+ DisplayRentbikeQrcode() : void - RentBikeRentBut_Click(sender : object, e : EventArgs) : void

**Attribute**

#	Tên	Kiểu dữ liệu	Giá trị mặc định	Miêu tả
1	rentBikeController	RentBikeController		Cung cấp chức năng và cách thức để người dùng tương tác với giao diện Thuê xe

**Operation**

#	Tên	Kiểu dữ liệu trả về	Mô tả (mục đích)
1	DisplayRentBikeQrcode	Void	Hiển thị giao diện nhập QR code

**Parameter:**

- Không

**Exception:**

- Không

#### 4. RentBikeInfoForm

RentBikeInfoForm	
- bike : BaseBike	
- DrawRentBikeInfoForm() : void	
- FillInfo(bike : BaseBike) : void	
+ FillRentBikeInfoForm(bike : BaseBike) : void	
+ FillRentTandemInfoForm(bike : BaseBike) : void	
+ FillRentElectricInfoForm(bike : BaseBike) : void	
- RentBikeInfoRentThisBikeBut_Click(sender : object, e : EventArgs) : void	
- RentBikeInfoDetailBut_Click(sender : object, e : EventArgs) : void	

#### Attribute

#	Tên	Kiểu dữ liệu	Giá trị mặc định	Miêu tả
1	Bike	BaseBike		Thông tin xe được thuê

#### Operation

#	Tên	Kiểu dữ liệu trả về	Mô tả (mục đích)
1	DrawRentBikeInfo	Void	Hiển thị giao diện thông tin thuê xe
2	FillInfo	Void	Hiển thị thông tin của các trường trong giao diện

#### Parameter:

- bike

#### Exception:

- Không

#### 5. RentingBikeForm

RentingBikeForm	
- rentBikeController : RentBikeController	
- bike : BaseBike	
- DrawRentingBikeForm() : void	
+ StartTimer() : void	
+ StopTimer() : void	
+ GetTotalTimeRent() : void	
+ FillRentingBikeForm() : void	
- RentBikeTmr_Tick(sender : object, e : EventArgs) : void	
- RentingSelectReceiveStationBut_Click(sender : object, e : EventArgs) : void	

## Attribute

#	Tên	Kiểu dữ liệu	Giá trị mặc định	Miêu tả
1	rentBikeController	RentBikeController		Khởi tạo, cung cấp cách thức để người dùng tương tác với giao diện Đang thuê xe
2	Bike	BaseBike		Thông tin xe được thuê

## Operation

#	Tên	Kiểu dữ liệu trả về	Mô tả (mục đích)
1	DrawRentingBikeForm	Void	Hiển thị giao diện đang thuê xe
2	StartTime	Void	Bắt đầu tính thời gian thuê xe
3	StopTime	Void	Dừng thời gian thuê xe
4	GetTotalTimeRent	Void	Tính tổng thời gian thuê xe
5	FillRentingBikeForm	Void	Hiển thị thông tin của giao diện đang thuê xe

### Parameter:

- Không

### Exception:

- Không

## 6. StationDetailForm

<b>StationDetailForm</b>
- station : Station
- DrawStationDetail() : void
+ FillStationDetail(station : Station) : void
- TandemBut_Click(sender : object, e : EventArgs) : void
- ElectricBut_Click(sender : object, e : EventArgs) : void
- BikeBut_Click(sender : object, e : EventArgs) : void

### Attribute

#	Tên	Kiểu dữ liệu	Giá trị mặc định	Miêu tả
1	Station	Station		Đại diện cho bãi xe

### Operation

#	Tên	Kiểu dữ liệu trả về	Mô tả (mục đích)
1	DrawStationDetail	Void	Hiển thị giao diện thông tin bãi xe
2	FillStationDetail	Void	Hiển thị thông tin trên giao diện bãi xe

#### Parameter:

- Station: đại diện cho bãi xe

#### Exception:

- Không

## 7. TransactionInformationForm

TransactionInformationForm	
- isPay : boolean	
- deposit : int	
- rentalMoney : int	
- stationId : int	
- bike : BaseBike	
- card : Card	
- category : BikeCategory	
- rentBikeController : RentBikeController	
- returnBikeController : ReturnBikeController	
- paymentController : PaymentController	
- DrawTransactionInformationForm() : void	
+ FillTransactionInformationWhenPay(stationId : int, bike : BaseBike) : void	
+ FillTransactionInformationWhenRentBike(bike : BaseBike, card : Card) : void	
- PermitButWhenRentBike() : void	
- PermitButWhenPay() : void	
- PermitBut_Click(sender : object, e : EventArgs) : void	
- CancelBut_Click(sender : object, e : EventArgs) : void	

### Attribute

#	Tên	Kiểu dữ liệu	Giá trị mặc định	Miêu tả
1	isPay	Boolean		
2	Deposit	Int		
3	rentalMoney	Int		
4	stationId	Int		
5	Bike	BaseBike		
6	card	Card		
7	category	BikeCategory		
8	rentBikeController	RentBikeController		
9	returnBikeController	ReturnBikeController		
10	paymentController	PaymenrController		

### Operation

#	Tên	Kiểu dữ liệu trả về	Mô tả (mục đích)
1	DrawTransactionInfomationForm	Void	Hiển thị giao diện thông tin thanh toán

2	FillTransactionInfomationWhenPay	Void	Hiển thị thông tin của các trường của giao diện thanh toán
3	FillTransactionInfomationWhenRentBike	Void	Hiển thị thông tin của các trường của giao diện thanh toán

**Parameter:**

- stationId: mã bã xe
- bike: xe
- card: thông tin thẻ ngân hàng

**Exception:**

- Không

## 8. CardInformationForm

CardInformationForm	
<ul style="list-style-type: none"> <li>- card : Card</li> <li>- bike : BaseBike</li> <li>- rentBikeController : RentBikeController</li> </ul>	
<ul style="list-style-type: none"> <li>- DrawCardInformation() : void</li> <li>- ValidCardInfor() : void</li> <li>+ FillCardInformation(bike : BaseBike) : void</li> <li>- SubmitBut_Click(sender : object, e : EventArgs) : void</li> <li>- CancelBut_Click(sender : object, e : EventArgs) : void</li> </ul>	

### Attribute

#	Tên	Kiểu dữ liệu	Giá trị mặc định	Miêu tả
1	card	Card		
2	bike	BaseBike		
3	rentBikeController	RentBikeController		

### Operation

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

1	DrawCardInfomation	Void	Hiển thị giao diện thẻ ngân hàng
2	ValidCardInfo	Void	Kiểm tra tính đúng đắn của thông tin thẻ ngân hàng
3	FillCardInfomation	Void	Hiển thị thông tin các trường của thẻ ngân hàng

**Parameter:**

- bike: xe

**Exception:**

- InvalidCardException
- InternalServerErrorException
- SuspiciousTransactionException
- NotEnoughTransactionInfoException
- InvalidVersionException

## 9. HomePageForm

HomePageForm	
- stationList : List<Station>	- ViewStationController : ViewStationController
<ul style="list-style-type: none"> <li>- DrawHomePage() : void</li> <li>+ RenderStationList() : void</li> <li>- RenderStationList(stationList : List&lt;Station&gt;) : void</li> <li>- But_Click(sender : object, e : EventArgs) : void</li> <li>- SearchBut_Click(sender : object, e : EventArgs) : void</li> <li>- CancelSearchBut_Click(sender : object, e : EventArgs) : void</li> </ul>	

**Attribute**

#	Tên	Kiểu dữ liệu	Giá trị mặc định	Miêu tả
1	stationList	List<Station>		Danh sách bãi xe
2	viewStationController	ViewStationController		

**Operation**

#	Tên	Kiểu dữ liệu trả về	Mô tả (mục đích)
1	DrawHomePage	Void	Hiển thị giao diện Trang chủ
2	RenderStationList	Void	Hiển thị danh sách bãi xe

**Parameter:**

- không

**Exception:**

- không

## 10. BikeDetailForm

BikeDetailForm
- bike : BaseBike - category : BikeCategory
- DrawBikeDetail() : void + FillBikeInformation(bike : BaseBike, stationName : string, stationAddress : string) : void - ViewRentingBut_Click(sender : object, e : EventArgs) : void - RentThisBikeBut_Click(sender : object, e : EventArgs) : void

### Attribute

#	Tên	Kiểu dữ liệu	Giá trị mặc định	Miêu tả
1	bike	BaseBike		
2	category	BikeCategory		Danh mục loại xe

### Operation

#	Tên	Kiểu dữ liệu trả về	Mô tả (mục đích)
1	DrawBikeDetail	Void	Hiển thị giao diện Thông tin xe
2	FillBikeInfomation	Void	Hiển thị thông tin các trường của giao diện Thông tin xe

**Parameter:**

- bike: xe
- stationName: Tên bãi xe

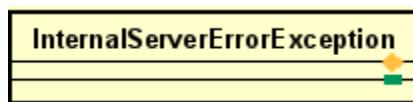
- stationAddress: Địa chỉ bến xe

*Exception:*

- không

#### 4.4.3.6 package CustomException

##### 1. InternalServerErrorException



**Attribute:** Không

**Operation:** Không

**Parameter:** Không

**Exception:** Không

**Method:** Không

##### 2. InvalidCardException



**Attribute:** Không

**Operation:** Không

**Parameter:** Không

**Exception:** Không

**Method:** Không

##### 3. InvalidTransactionAmountException



**Attribute:** Không

**Operation:** Không

**Parameter:** Không

**Exception:** Không

**Method:** Không

#### 4. NotEnoughBalanceException



**Attribute:** Không

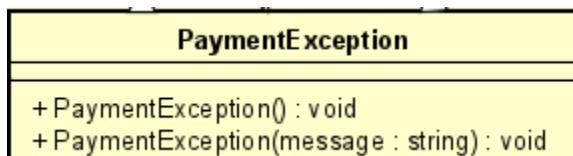
**Operation:** Không

**Parameter:** Không

**Exception:** Không

**Method:** Không

#### 5. PaymentException



**Attribute:** Không

**Operation**

#	Tên	Kiểu dữ liệu trả về	Mô tả (mục đích)
1	PaymentException	void	Thanh toán ngoại lệ

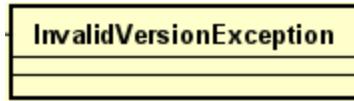
**Parameter:**

- Message: thông báo

**Exception:** Không

**Method:** Không

#### 6. InvalidVersionException



**Attribute:** Không

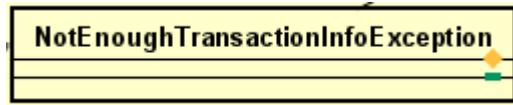
**Operation:** Không

**Parameter:** Không

**Exception:** Không

**Method:** Không

## 7. NotEnoughTransactionInfoException



**Attribute:** Không

**Operation:** Không

**Parameter:** Không

**Exception:** Không

**Method:** Không

## 8. SuspiciousTransactionException



**Attribute:** Không

**Operation:** Không

**Parameter:** Không

**Exception:** Không

**Method:** Không

## 9. UnrecognizedException



**Attribute:** Không

**Operation:** Không

**Parameter:** Không

**Exception:** Không

**Method:** Không

#### 4.4.3.7 Class Utilities

<b>Utilities</b>
+ ConvertDateToString(year : int, mounth : int, day : int, hour : int, minute : int, second : int) : string + ConvertDateToString(date : DateTime) : string + SubtractDate(date1 : DateTime, date2 : DateTime) : string + MD5Hash(input : string) : string + SetComboBoxHeight(comboBoxHandle : IntPtr, comboBoxDesire dHeight : Int32) : void + InvlidString(regex : string, value : string) : boolean

**Attribute:** Không

**Operation**

#	Tên	Kiểu dữ liệu trả về	Mô tả (mục đích)
1	ConvertDateToString	string	Chuyển từ kiểu Date sang String
2	SubtractDate	string	Trừ ngày
3	MD5Hash	string	Băm chuỗi được chỉ định với mã băm MD5
4	SetComboBoxHeight	void	Đặt chiều cao của comboBox
5	InvlidString	boolean	Hợp lệ chuỗi với regex được chỉ định

**Parameter:**

- year: năm
- month: tháng
- day: ngày
- hour: giờ
- minute: phút
- second: giây
- date: đại diện cho thời gian muốn chuyển đổi
- input: chuỗi cần hash
- regex: regex được chỉ định
- value: chuỗi
- comboBoxHandle: comboBox Xử lý
- comboBoxDesiredHeight: comboBox Chiều cao mong muốn

**Exception:** Không

**Method:** Không

#### 4.4.3.8 Class Constant

Constant
+ SQL_CONNECT_STRING : string + BikeCategory : enum + QRValid : string

#### Attribute

#	Tên	Kiểu dữ liệu	Giá trị mặc định	Miêu tả
1	SQL_CONNECT_STRING	string		Chuỗi dùng để kết nối với database
2	BikeCategory	enum(kiểu liệt kê)		

3	QRValid	string		dùng để valid mã qr do người dung nhập vào
---	---------	--------	--	--

**Operation:** Không

**Parameter:** Không

**Exception:** Không

**Method:** Không

## 5 Design Considerations

### 5.1 Goals and Guidelines

- Mục tiêu:
  - o Phần mềm hoạt động tốt trên mọi nền tảng
  - o Các chức năng hoạt động bình thường
  - o Người dùng thuê được xe và trả được xe sau thời gian thuê xe
  - o Người dùng thanh toán thành công
  - o Các trường hợp ngoại lệ được thông báo tới người dùng kịp thời
  - o Hạn chế tối đa các trường hợp rủi ro
- Mục tiêu:
  - o Trong thời gian tới sẽ phát triển thêm chức năng định vị bãi gửi xe trên bản đồ
  - o Dữ liệu sát thực tế hơn

### 5.2 Architectural Strategies

#### a. Ngôn ngữ lập trình

Ngôn ngữ lập trình: C# (Winform Core)

Các thư viện đã sử dụng trong chương trình

- Newtonsoft.Json(version: 12.0.3) sử dụng để hỗ trợ tương tác với json trong C#
- Microsoft.EntityFrameworkCore.SqlServer(version: 5.0.0) sử dụng để kết nối với SQL server
- Nunit(version: 3.12.0), NUnit3TestAdapter(version: 3.15.1), Microsoft.NET.Test.Sdk(version: 16.4.0) sử dụng để viết unit test

#### b. Database

Hệ quản trị cơ sở dữ liệu SQL Server

### 5.3 Coupling and Cohesion

#### 5.3.1 Coupling

##### 5.3.1.1 Content Coupling

Không có

##### 5.3.1.2 Common Coupling

Không có

### 5.3.1.3 Control Coupling

#### a. RentBikeController

```
public int CalculateFee(string timeRent, BikeCategory category)
{
    string[] times = timeRent.Split(':');
    double hour = Int64.Parse(times[0]);
    double minute = Int64.Parse(times[1]);
    double second = Int64.Parse(times[2]);
    double timeMinutes = 60 * hour + minute + Math.Abs(second / 60) - 10;
    if (category != BikeCategory.BIKE) timeMinutes = 1.5 * timeMinutes;
    if (timeMinutes <= 0) return 0;
    timeMinutes -= 30;
    if (timeMinutes <= 0) return 10000;
    return (int)(10000 + Math.Abs(timeMinutes / 15));
}
```

Vấn đề: hàm CalculateFee cần truyền tham số BikeCategory, tùy vào cách truyền thì hàm CalculateFee sẽ có cách tính phí khác nhau

#### b. BaseBikeService

```
public BaseBike GetBikeById(int id)
{
    BaseBike baseBike = connecter.SqlData.BaseBikes.Find(id);
    BaseBike result = null;
    if (baseBike.Category == "bike") result = new Bike(baseBike);
    else if (baseBike.Category == "tandem") result = new Tandem(baseBike);
    else if (baseBike.Category == "electric")
    {
```

Vấn đề: hàm GetBikeById và GetBikeByQrcode cần dựa vào Category để có cách thực hiện khác nhau

### 5.3.1.4 Stamp Coupling

Không có

## 5.3.2 Cohesion

### 5.3.2.1 Coincidental cohesion

#### a. Lớp Constant và InterbankConstant

Vấn đề: hai lớp này lưu các hằng số phục vụ cho chương trình

#### b. Lớp Utilities

Utilities
+ ConvertDateToString(year : int, mounth : int, day : int, hour : int, minute : int, second : int) : string
+ ConvertDateToString(date : DateTime) : string
+ SubtractDate(date1 : DateTime, date2 : DateTime) : string
+ MD5Hash(input : string) : string
+ SetComboBoxHeight(comboBoxHandle : IntPtr, comboBoxDesiredHeight : Int32) : void
+ IsValidString(regex : string, value : string) : boolean

Vấn đề: chứa các hàm tiện ích xử lý những việc không thuộc phạm vi của các package trong chương trình

Cải thiện: có thể tách ba hàm đầu tiên thành một package DateHandle để xử lý về ngày tháng. Tách hàm MD5Hash thành package dùng để mã hóa

### 5.3.2.2 *Logical cohesion*

#### a. *Lớp PaymentController*

Hàm MakeResponse chỉ liên quan về mặt logic với các hàm còn lại trong class này

## 5.4 *Design Principles*

### 5.4.1 Single responsibility principle

*Chưa tuân thủ*

- Ở lớp PaymentController, Class này thực hiện hai nhiệm vụ là thực hiện giao dịch và chuyển đổi dữ liệu trả về thành kiểu đối tượng

=> cách cải thiện: tách chưa năng chuyển đổi dữ liệu thành một lớp mới

### 5.4.2 Open/Closed principle

*Tuân thủ*

Khi có thêm một loại xe mới, ví dụ xe đạp điện đôi, thay vì phải sửa đổi lớp BaseBike để phù hợp với loại xe mới thì ta chỉ cần tạo một lớp mới kế thừa lớp BaseBike đã có

Khi có thêm một Interbank mới hoặc giao thức API thay đổi, thay vì chỉnh sửa lớp PaymentController đã có thì ta sẽ viết một lớp mới Implement interface Ipayment

*Không tuân thủ*

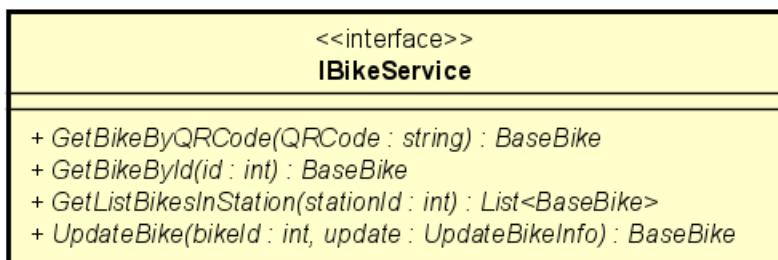
Lớp BaseBikeService khi có thêm một loại xe mới thì hai hàm GetBikeById và GetBikeByQRCode sẽ phải chỉnh sửa code để phù hợp với yêu cầu mới

### 5.4.3 Liskov Substitution Principle

Tuân thủ

Hệ thống phân cấp kế thừa ở class BaseBike đã tuân thủ theo nguyên tắc này

### 5.4.4 Interface Segregation

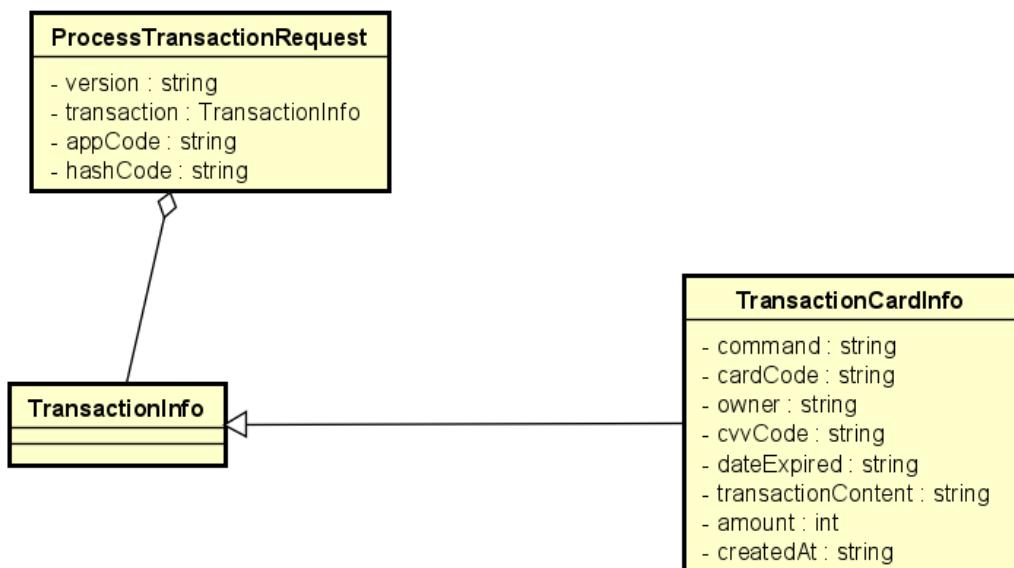


Có thể không tuân thủ

Interface **IBikeService** có hàm **UpdateBike** có thể chỉ dùng cho admin và các hàm **Get** còn lại dùng cho cả user và admin

Cải thiện: tách thành các interface với chức năng **Get bike** và **Update bike** riêng

### 5.4.5 Dependency inversion principle

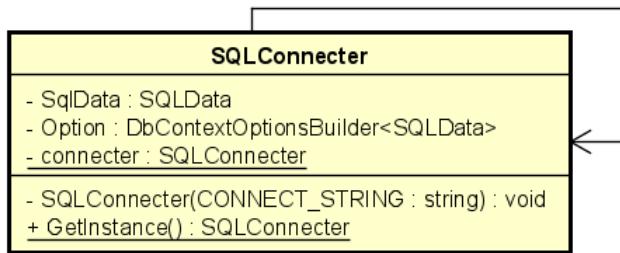


Tuân thủ

Lớp ProcessTransactionRequest thay vì phụ thuộc vào TransactionCardInfo là class cụ thể thì nó chỉ phụ thuộc vào TransactionInfo là class trừu tượng hơn. Khi có một cách thanh toán mới, chỉ cần viết thêm một lớp mới kế thừa lớp TransactionInfo

## 5.5 Design Patterns

### 5.5.1 Singleton pattern



Class SQLConnecter dùng để kết nối đến cơ sở dữ liệu. Thiết kế singleton được áp dụng trong class này để chỉ có nhiều nhất một liên kết tới cơ sở dữ liệu