HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

School of Information and communications technology

Software Design Document

Version 1.3

EcobikeRental

Môn: Thiết kế xây dựng phần mềm

Nhóm 05

Nguyễn Bá Đức MSSV: 20183499

Trần Quốc Chung MSSV: 20183489

Nguyễn Năng Đức Anh MSSV: 20183475

*Hà Nội, ngày 5 tháng 1 năm 2022*

*<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 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 Introduction 3

1.1 Objective 3

1.2 Scope 3

1.3 Glossary 3

1.4 References 3

2 Overall Description 4

2.1 General Overview 4

2.2 Assumptions/Constraints/Risks 8

2.2.1 Assumptions 8

2.2.2 Constraints 9

2.2.3 Risks 9

3 System Architecture and Architecture Design 10

3.1 Architectural Patterns 10

3.2 Interaction Diagrams 10

3.3 Analysis Class Diagrams 13

3.4 Unified Analysis Class Diagram 15

3.5 Security Software Architecture 15

4 Detailed Design 16

4.1 User Interface Design 16

4.1.1 Screen Configuration Standardization 16

4.1.2 Screen Transition Diagrams 16

4.1.3 Screen Specifications 17

4.2 Data Modeling 28

4.2.1 Conceptual Data Modeling 28

4.2.2 Database Design 28

4.3 Non-Database Management System Files 37

4.4 Class Design 38

4.4.1 General Class Diagram 38

4.4.2 Class Diagrams 38

4.4.3 Class Design 41

5 Design Considerations 75

5.1 Goals and Guidelines 75

5.2 Architectural Strategies 75

5.3 Coupling and Cohesion 75

5.4 Design Principles 77

5.5 Design Patterns 78

# Introduction

## Objective

Tài liệu mô tả SDD này mô tả bản thiết kế cho ứng dụng EcoBikeRental . Mô tả chi tiết thiết kế các lớp , cơ sở dữ liệu dùng trong ứng dụng

## Scope

* Phần mềm được xây dựng là : Eco Bike Rental
* Phần mềm có chức năng :
  + Cho phép người dùng xem danh sách bãi xe trong hệ thống , xem chi tiết bãi xe , xem thông tin chi tiết từng xe
  + Tiến hành mượn xe bằng cách nhập mã xe
  + Thực hiện thanh toán , đặt cọc bằng thẻ tín dụng
  + Trả xe tại bất kì bãi xe nào trong hệ thống

## Glossary

JavaFX : 1 thư viện Java dùng để xây dựng các ứng dụng với giao diện hiện đại

SQLite : hệ cơ sở dữ liệu được cài đặt mặc định trên nhiều nền tảng

SDD : Software design document – tài liệu mô tả thiết kế tổng quan , chi tiết của hệ thống

## References

# Overall Description

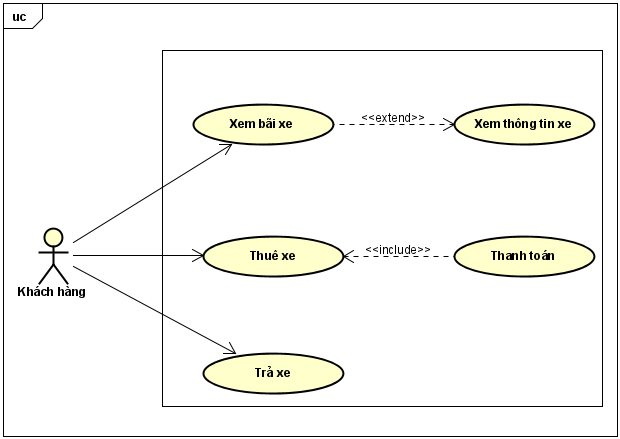
<*This section describes the principles and strategies to be used as guidelines when designing and implementing the system.>*

## General Overview

Hệ thống EcoBikeRental mô phỏng cho dịch vụ cho thuê xe đạp. Phần mềm được chạy trên máy tính , yêu cầu cần có JRE .

Phần mềm này được thiết kế bằng cách sử dụng mô hình thiết kế kiến trúc MVC.

* Biểu đồ usecase tổng quan



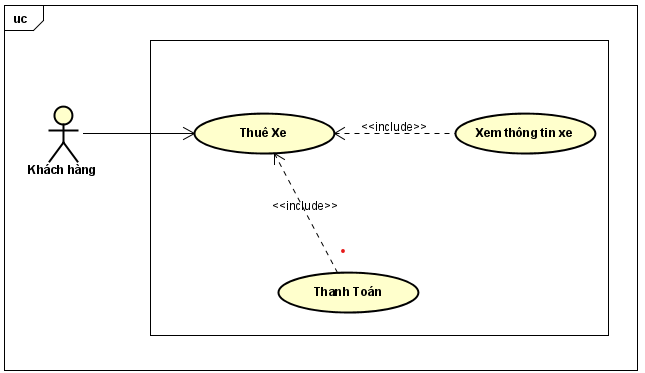
Hệ thống có 1 actor là khách hàng sử dụng hệ thống thuê xe , 3 usecase trong hệ thống là Xem xe trong bãi xe , thuê xe , trả xe .

* Biểu đồ usecase “Xem xe đạp trong bãi xe “

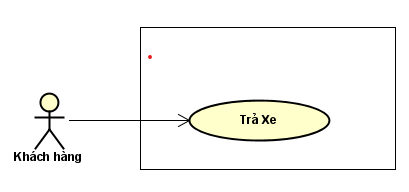
Ảnh có chứa văn bản

Mô tả được tạo tự động

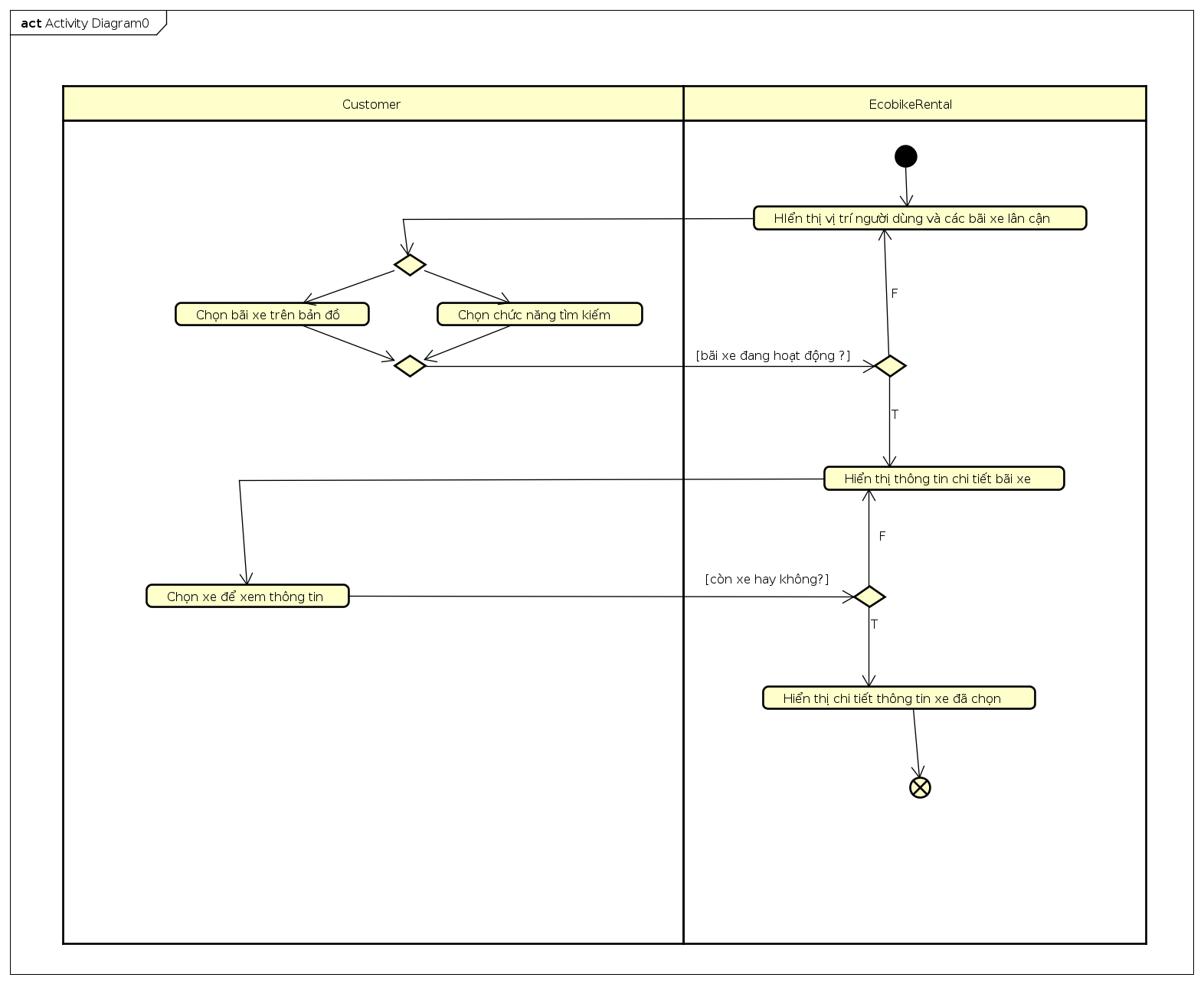
* Biểu đồ usecase “Thuê xe đạp “



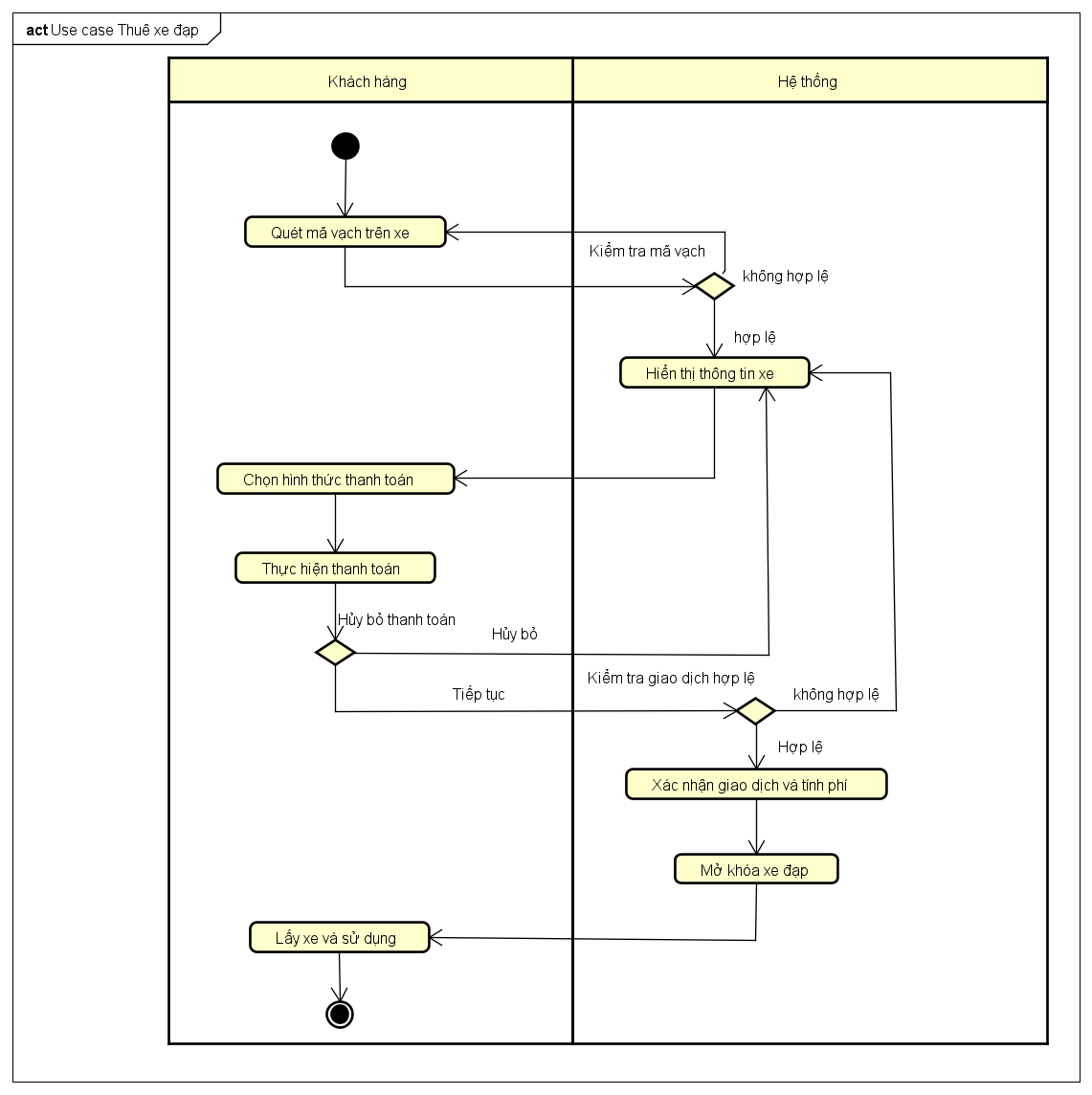
* Biểu đồ usecase “Trả xe đạp”



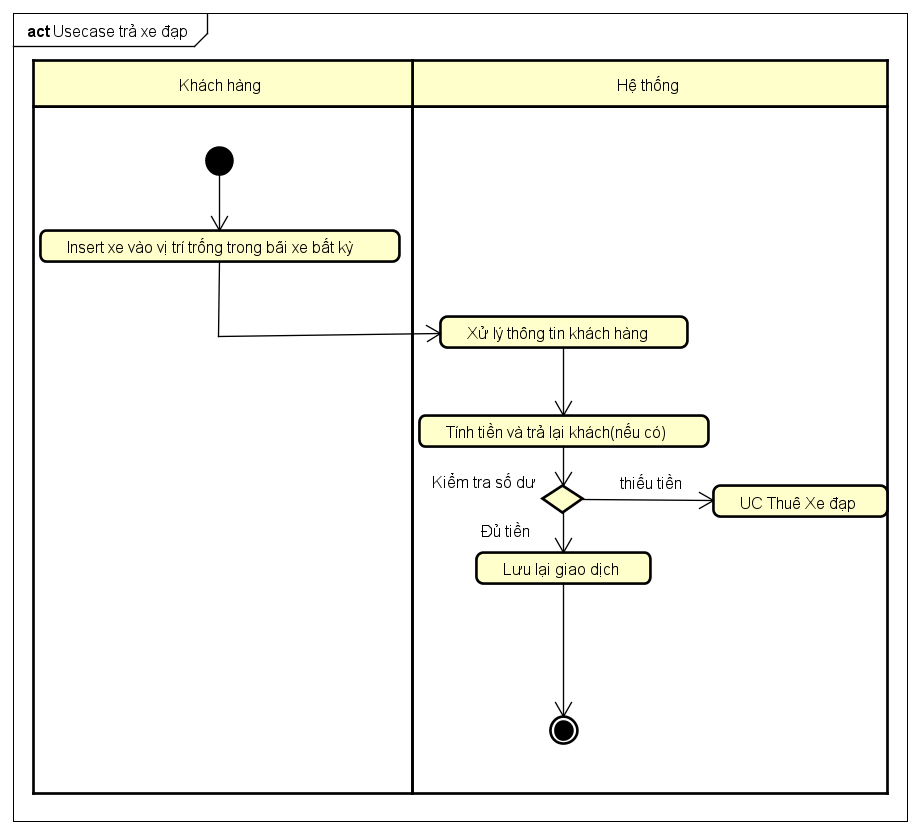
* Biểu đồ hoạt động usecase “ Xem xe đạp “

****

* Biểu đồ hoạt động usecase “ Thuê xe “



* Biểu đồ hoạt động usecase “ Trả xe “



## Assumptions/Constraints/Risks

### Assumptions

Phần mềm được thiết kế dựa trên yêu cầu của môn học , các nghiệp vụ không giống như trong thực tế .

### Constraints

Hệ thống có thể chạy trên nền tảng có cài đặt JRE và SQLite .

Hệ thống sử dụng JavaFx để tạo giao diện và SQLite để tạo database mô phỏng

### Risks

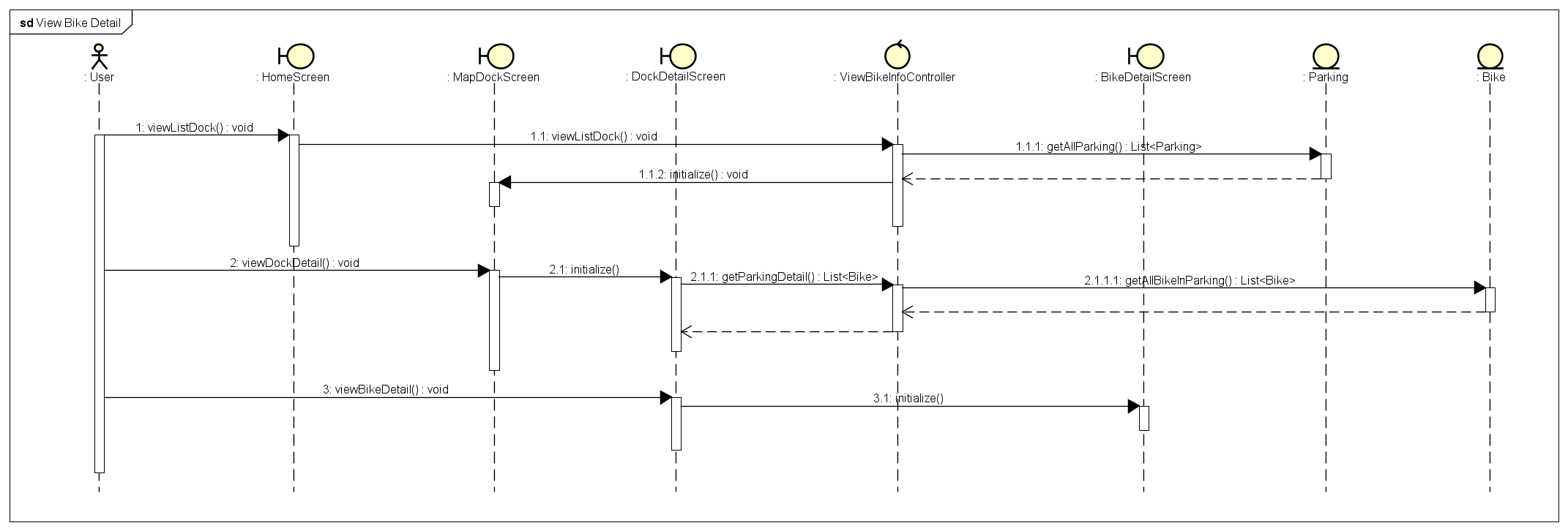
# System Architecture and Architecture Design

## Architectural Patterns

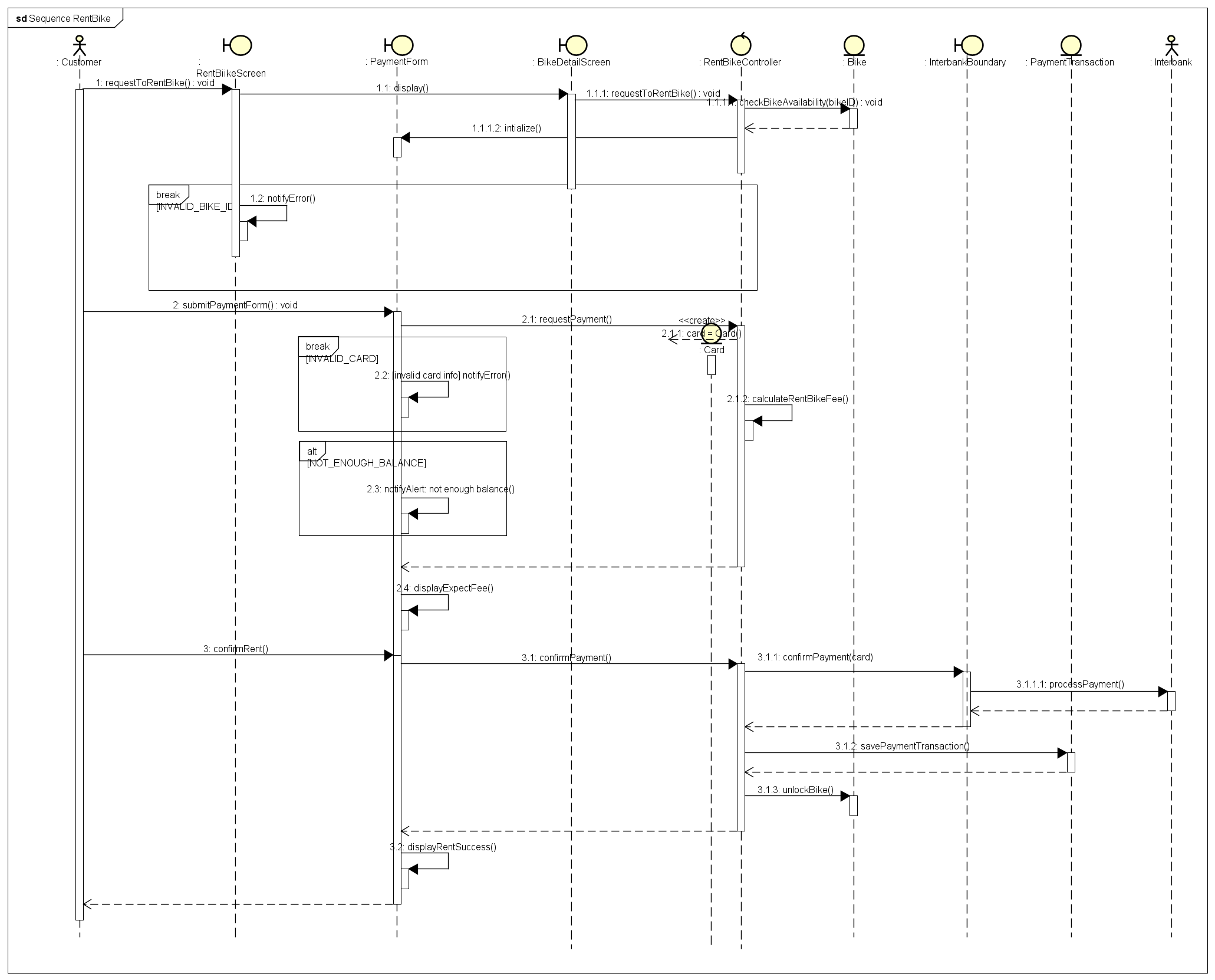
Mô hình kiến trúc được sử dụng để xây dựng ứng dụng này là MVC . MVC là 1 kiến trúc đã có từ lâu , nó có các ưu điểm so với các kiến trúc khác đó là đơn giản , dễ hiểu và cũng dễ cài đặt , thường được sử dụng trong các ứng dụng không quá lớn .

## Interaction Diagrams

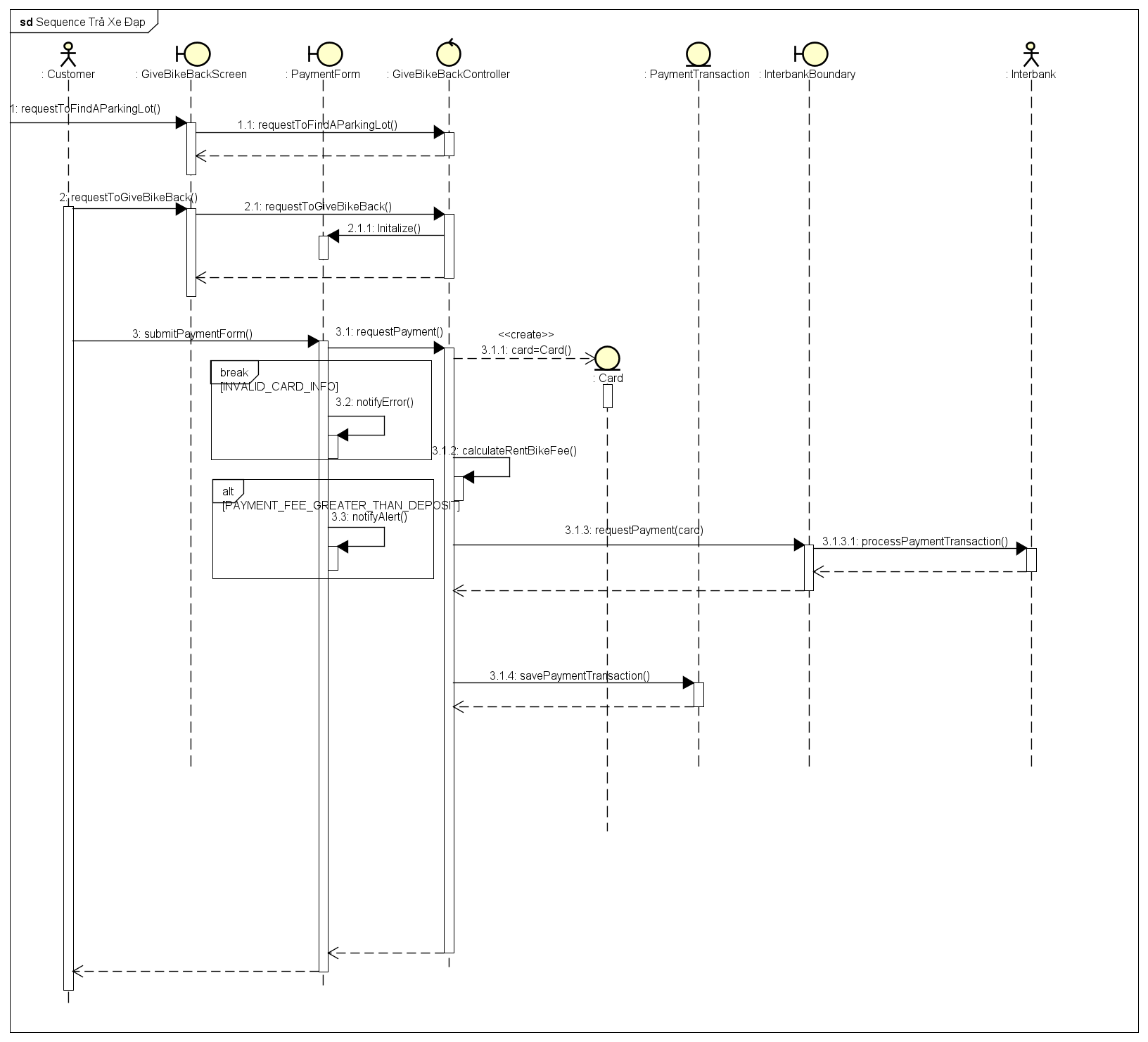
### Usecase Xem xe trong bãi xe



### Usecase Mượn xe

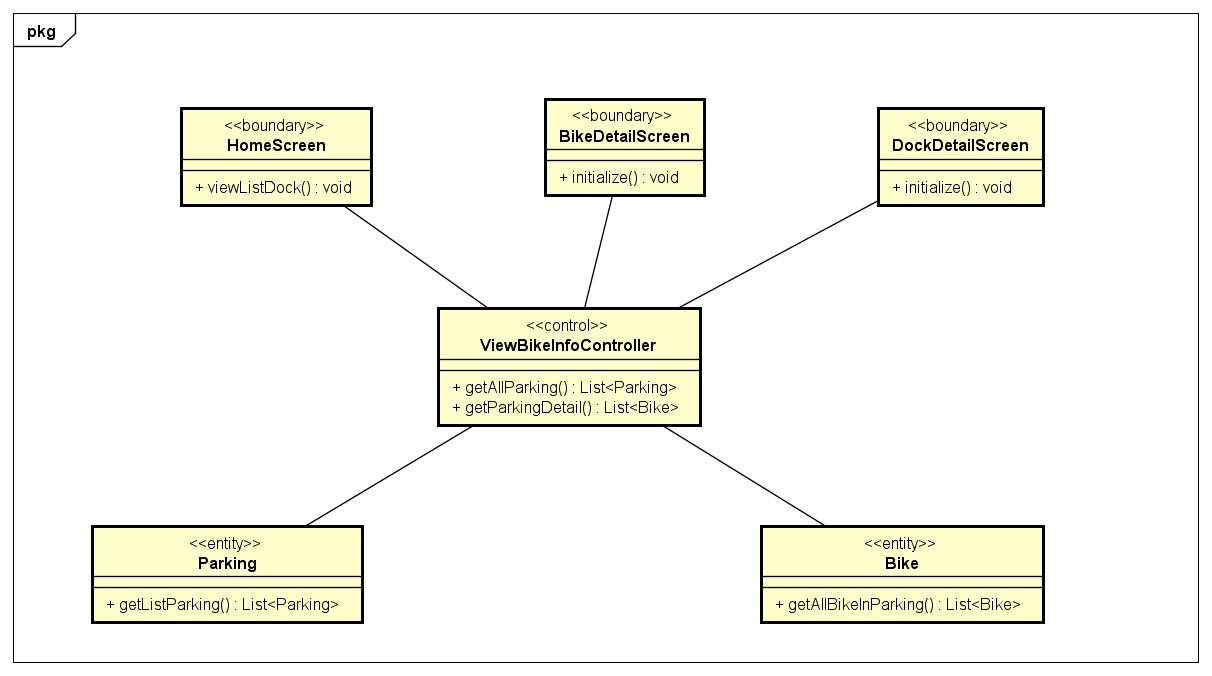


### Usecase Trả xe

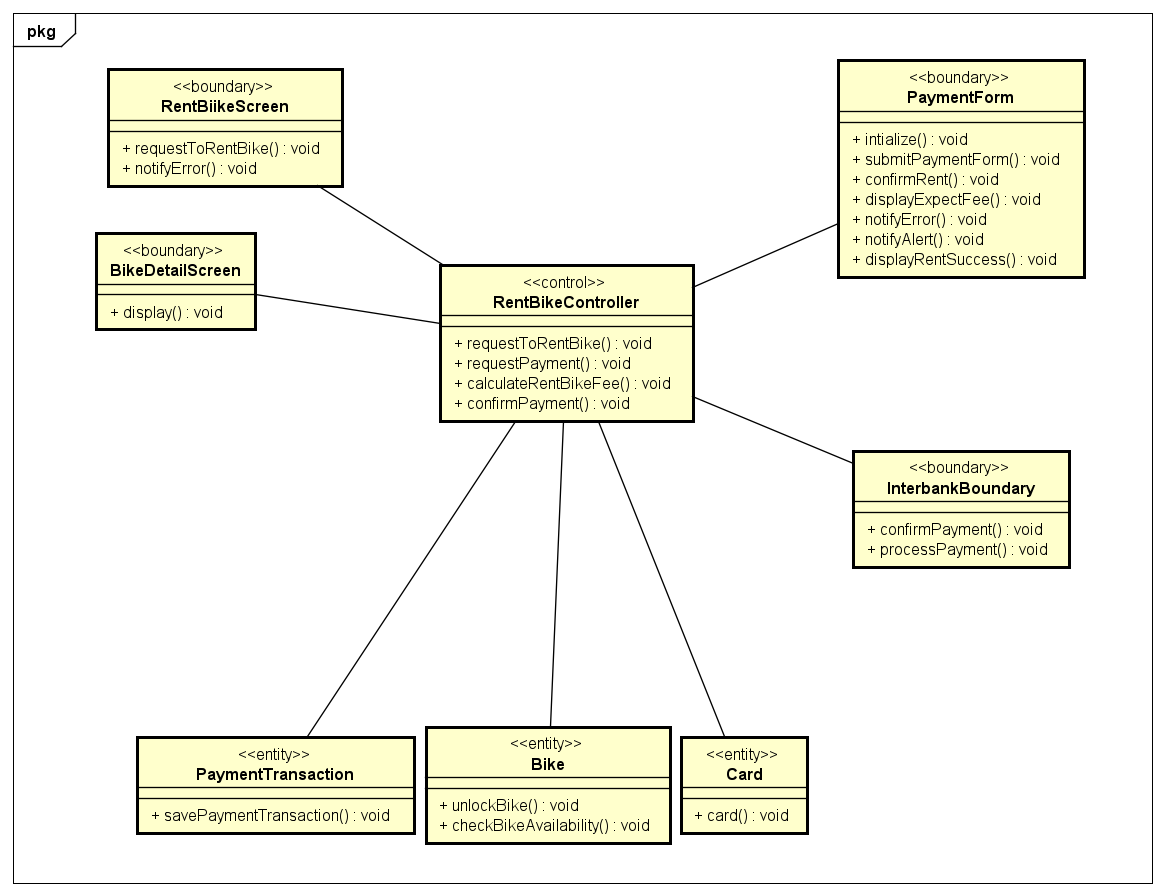


## Analysis Class Diagrams

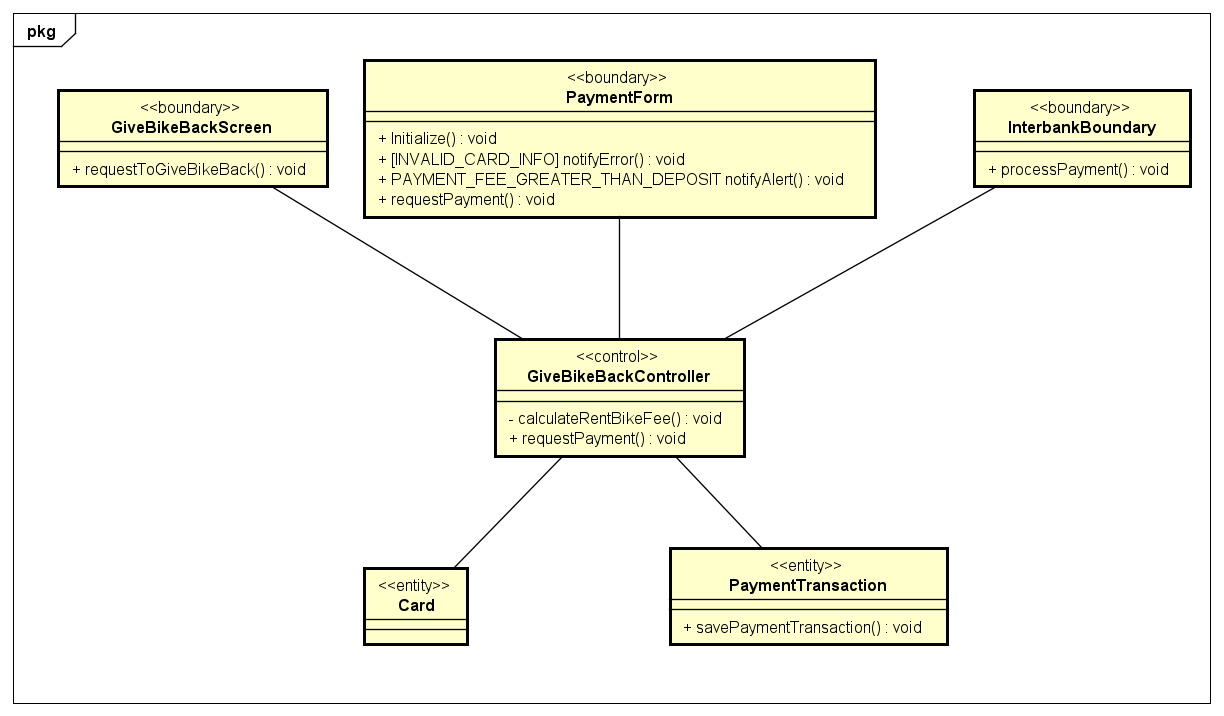
* **View Bike**



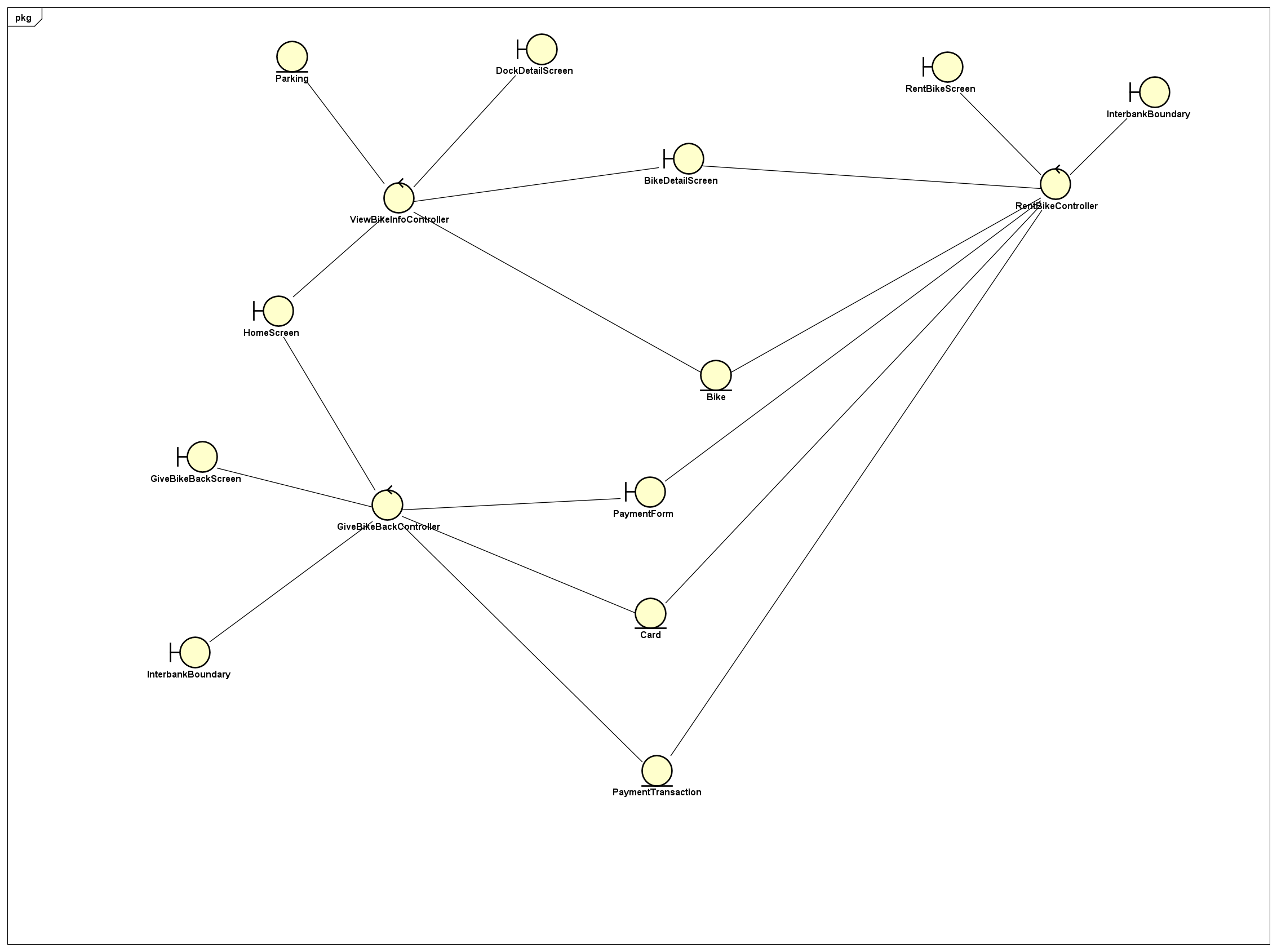
* **Rent Bike**



* **Give Bike Back**



## Unified Analysis Class Diagram



## Security Software Architecture

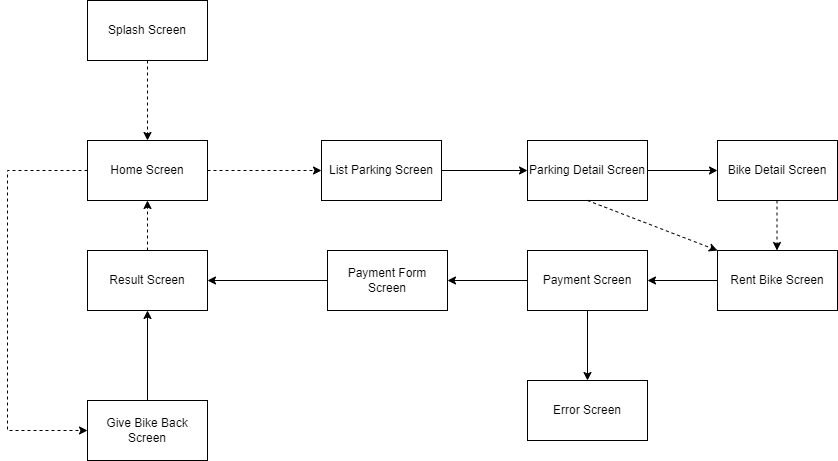
# Detailed Design

## User Interface Design

### Screen Configuration Standardization

* Display : 360\*640 pixel
* Thanh tiêu đề : nằm ở trên cùng
* Logo ứng dụng : góc trên cùng bên trái
* Thứ tự các màn hình trong hệ thống
  + Splash Screen : màn hình chờ ban đầu
  + Home Screen : trang chủ
  + List parking screen : màn hình danh sách bãi xe
  + Parking detail screen : màn hình chi tiết bãi xe
  + Bike Detail Screen : màn hình chi tiết xe
  + Rent bike screen : màn hình nhập xe muốn thuê
  + Payment Screen : màn hình thanh toán
  + Payment Form Screen : form điền thông tin thanh toán
  + Result Screen : màn hình kết quả thanh toán
  + Give Bike Back Screen : màn hình trả xe

### Screen Transition Diagrams



### Screen Specifications

#### Splash screen



#### Home Screen

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **EBR Software** | | **Date of creation** | **Approved by** | **Reviewed by** | **Person in charge** |
| **Screen specification** | Home Screen | 10/12/2021 |  |  |  |
|  | | **Control** | **Operation** | **Function** | |
| “Xem danh sách” Button | Click | Display screen List parking | |
| Area for entering search | Initial | Receive Text for Searching | |
| “Trả xe” Button | Click | Thực hiện trả xe , chuyển đến màn hình Give bike back screen | |

#### List Parking Screen

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **EBR Software** | | **Date of creation** | **Approved by** | **Reviewed by** | **Person in charge** |
| **Screen specification** | List Parking Screen | 10/12/2021 |  |  |  |
| Ảnh có chứa văn bản  Mô tả được tạo tự động | | **Control** | **Operation** | **Function** | |
| “Quay lại” Button | Click | Trở về màn hình Home screen | |
| Item in listview Parking | Click | Chuyển đến màn hình Parking Screen | |

#### Parking Screen

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **EBR Software** | | **Date of creation** | **Approved by** | **Reviewed by** | **Person in charge** |
| **Screen specification** | Parking Screen | 10/12/2021 |  |  |  |
|  | | **Control** | **Operation** | **Function** | |
| “Xem thông tin” Button | Click | Chuyển đến màn hình “Parking Detail Screen” | |

#### Parking Detail Screen

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **EBR Software** | | **Date of creation** | **Approved by** | **Reviewed by** | **Person in charge** |
| **Screen specification** | Parking Detail Screen | 10/12/2021 |  |  |  |
| Ảnh có chứa bàn  Mô tả được tạo tự động | | **Control** | **Operation** | **Function** | |
| “Quay lại” Button | Click | Trở về màn hình Home screen | |
| Item in listview bike | Click | Chuyển đến màn hình Bike Detail Screen | |

#### Bike Detail Screen

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **EBR Software** | | **Date of creation** | **Approved by** | **Reviewed by** | **Person in charge** |
| **Screen specification** | Bike Detail Screen | 10/12/2021 |  |  |  |
|  | | **Control** | **Operation** | **Function** | |
| “OK” Button | Click | Trở lại màn hình trước “Parking Detail Screen” | |
| “Thuê” Button | Click | Thực hiện thuê xe , chuyển đến màn hình Payment Screen | |

#### Rent Bike Screen

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **EBR Software** | | **Date of creation** | **Approved by** | **Reviewed by** | **Person in charge** |
| **Screen specification** | Rent Bike Screen | 10/12/2021 |  |  |  |
|  | | **Control** | **Operation** | **Function** | |
| “Quay lại” Button | Click | Trở lại màn hình Parking Detail Screen | |
| Area for entering bike code | Texting | Lấy mã xe để thực hiện thuê xe | |
| “Thuê xe” Button | Click | Thực hiện thuê xe , chuyển đến màn hình Bike Detail Screen | |

#### Payment Screen

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **EBR Software** | | **Date of creation** | **Approved by** | **Reviewed by** | **Person in charge** |
| **Screen specification** | Payment Screen | 10/12/2021 |  |  |  |
|  | | **Control** | **Operation** | **Function** | |
| “Quay lại” Button | Click | Trở về màn hình Bike Detail Screen | |
| “Thanh toán” screen | Click | Thực hiện thanh toán , chuyển đến màn hình “Payment Form screen” | |

#### Payment Form Screen

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **EBR Software** | | **Date of creation** | **Approved by** | **Reviewed by** | **Person in charge** |
| **Screen specification** | Payment Form Screen | 10/12/2021 |  |  |  |
|  | | **Control** | **Operation** | **Function** | |
| “Xác nhận” Button | Click | Tiên hành thanh toán tiền đặt cọc và chuyển đến màn hình Result Screen | |
|  |  |  | |

#### Result Screen

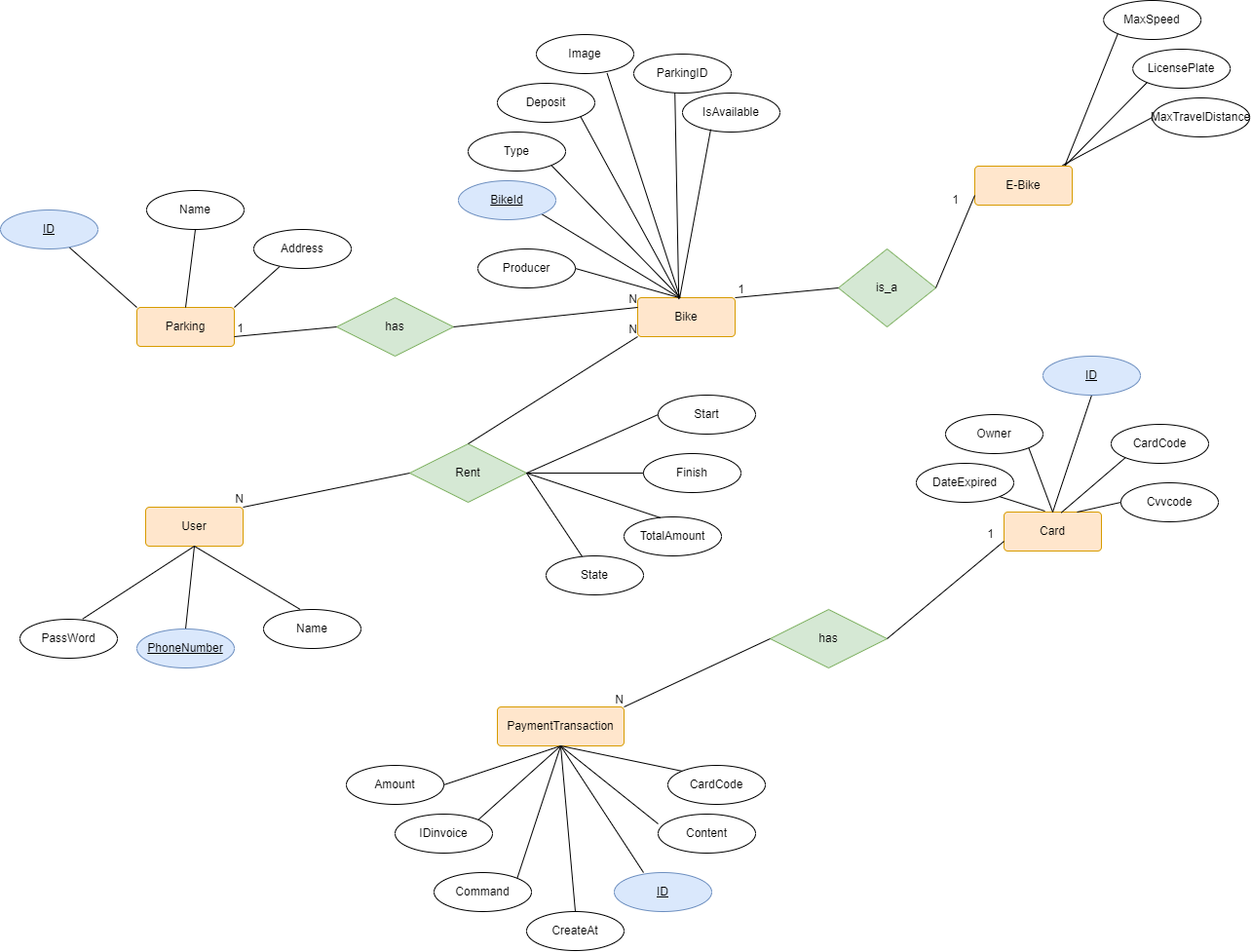
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **EBR Software** | | **Date of creation** | **Approved by** | **Reviewed by** | **Person in charge** |
| **Screen specification** | Result Screen | 10/12/2021 |  |  |  |
|  | | **Control** | **Operation** | **Function** | |
| “Xác nhận” Button | Click | Trở về màn hình “Home Screen” | |

#### Give bike back Screen

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **EBR Software** | | **Date of creation** | **Approved by** | **Reviewed by** | **Person in charge** |
| **Screen specification** | Give Bike Back Screen | 10/12/2021 |  |  |  |
|  | | **Control** | **Operation** | **Function** | |
| “Xác nhận” Button | Click | Xác nhận thanh toán hóa đơn , hoàn tiền về tài khoản . Đi đến màn hình “Result Screen” | |

## Data Modeling

### Conceptual Data Modeling

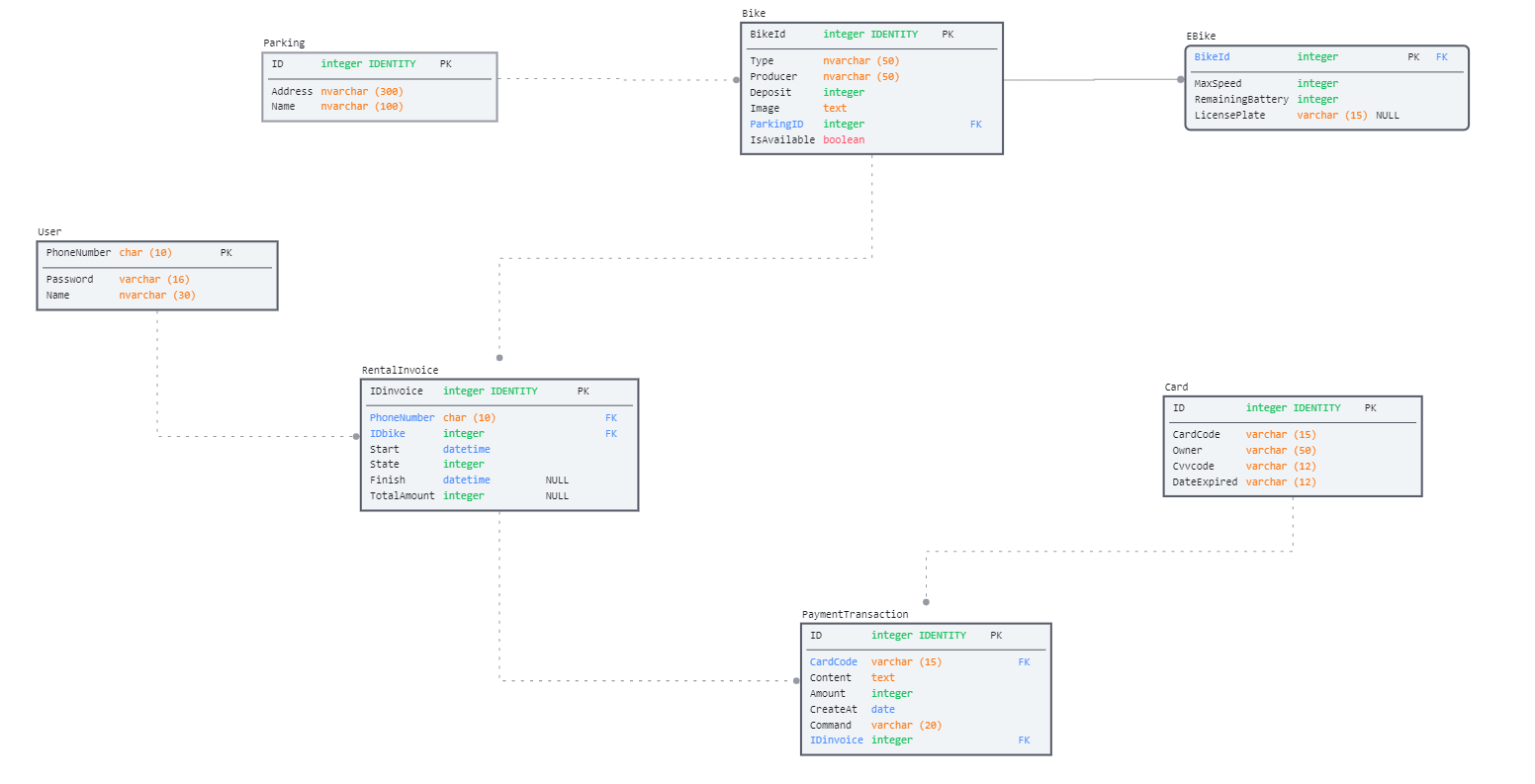


### Database Design

#### Database Management System

Database management system được sử dụng trong hệ thống là SQLite .

#### Database Diagram



#### Database Detail Design

**Parking**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *#* | *PK* | *FK* | *Column name* | *Data type* | *Default value* | *Mandatory* | *Description* |
| 1 | X |  | ID | Integer |  | Yes | Id of Parking , auto increment |
| 2 |  |  | Address | nvarchar(300) |  | Yes | Address of Parking |
| 3 |  |  | Name | Nvarchar(100) |  | Yes | Name of Parking |

**User**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *#* | *PK* | *FK* | *Column name* | *Data type* | *Default value* | *Mandatory* | *Description* |
| 1 | X |  | PhoneNumber | char(10) |  | Yes | Phone number of user |
| 2 |  |  | Password | varchar(16) |  | Yes | Password |
| 3 |  |  | Name | Nvarchar(30) |  | Yes | Name of user |

**Bike**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *#* | *PK* | *FK* | *Column name* | *Data type* | *Default value* | *Mandatory* | *Description* |
| 1 | X |  | BikeId | Integer |  | Yes | Auto increment |
| 2 |  |  | Type | nvarchar(50) |  | Yes | Type of bike |
| 3 |  |  | Producer | nvarchar(50) |  | Yes | Producer of bike |
| 4 |  |  | Deposit | Integer |  | Yes | Deposit of renting |
| 5 |  |  | Image | Text |  | Yes | URL of bike’s image |
| 6 |  | X | ParkingID | Integer |  | Yes | ID of parking |
| 7 |  |  | IsAvailable | Boolean |  | Yes | True if bike hasn’t been rented |

**Ebike**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *#* | *PK* | *FK* | *Column name* | *Data type* | *Default value* | *Mandatory* | *Description* |
| 1 | X | X | BikeId | Integer |  | Yes | Id of bike |
| 2 |  |  | MaxSpeed | Integer |  | Yes | Max speed of bike |
| 3 |  |  | RemainingBattery | Integer |  | Yes | Remaining of battery |
| 4 |  |  | LicensePlate | Varchar(15) |  | Yes | License plate of bike |

**Card**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *#* | *PK* | *FK* | *Column name* | *Data type* | *Default value* | *Mandatory* | *Description* |
| 1 | X |  | ID | Integer |  | Yes | Id of card,auto increment |
| 2 |  |  | CardCode | Varchar(15) |  | Yes | Card code |
| 3 |  |  | Owner | Varchar(15) |  | Yes | Cardholder |
| 4 |  |  | CvvCode | Varchar(12) |  | Yes | CVV code |
| 5 |  |  | DateExpired | Varchar(12) |  | Yes | Expiration date |

**RentalInvoice**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *#* | *PK* | *FK* | *Column name* | *Data type* | *Default value* | *Mandatory* | *Description* |
| 1 | X |  | Idinvoice | Integer |  | Yes | Auto increment |
| 2 |  | X | PhoneNumber | Char(10) |  | Yes | Phone number of user |
| 3 |  | X | IDbike | Integer |  | Yes | Id bike |
| 4 |  |  | Start | Datetime |  | Yes | Rental start time |
| 5 |  |  | Finish | Datetime |  | No | Rental end time |
| 6 |  |  | State | Integer |  | Yes | State of invoice , equals 1 if bike rental is finish |
| 7 |  |  | TotalAmount | Integer |  | No | Total amount of invoice |

**PaymentTransaction**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *#* | *PK* | *FK* | *Column name* | *Data type* | *Default value* | *Mandatory* | *Description* |
| 1 | X |  | ID | Integer |  | Yes | Auto increment |
| 2 |  | X | CardCode | Varchar(15) |  | Yes | Card code |
| 3 |  |  | Content | Text |  | Yes | Transaction content |
| 4 |  |  | CreateAt | Date |  | Yes | Date of creation |
| 5 |  |  | Command | Varchar(20) |  | No | Method of payment |
| 6 |  |  | Amount | Integer |  | Yes | Amount of transaction |
| 7 |  | X | IDinvoice | Integer |  | No | Id of invoice |

**SQL script**

--

-- File generated with SQLiteStudio v3.3.3 on Wed Jan 5 08:09:11 2022

--

-- Text encoding used: UTF-8

--

PRAGMA foreign\_keys = off;

BEGIN TRANSACTION;

-- Table: User

CREATE TABLE User (

PhoneNumber CHAR (10) PRIMARY KEY,

Password VARCHAR (16) NOT NULL,

Name NVARCHAR (30) NOT NULL

);

-- Table: Card

CREATE TABLE Card (

ID INTEGER PRIMARY KEY AUTOINCREMENT,

CardCode VARCHAR (15) NOT NULL,

Owner VARCHAR (50) NOT NULL,

Cvvcode VARCHAR (12) NOT NULL,

DateExpired VARCHAR (12) NOT NULL

);

-- Table: Parking

CREATE TABLE Parking (

ID INTEGER PRIMARY KEY AUTOINCREMENT,

Address NVARCHAR (300) NOT NULL,

Name NVARCHAR (100) NOT NULL

);

INSERT INTO Parking (ID, Address, Name) VALUES (1, 'Hà Nội - BX Giáp Bát', 'Giáp Bát ECB');

INSERT INTO Parking (ID, Address, Name) VALUES (2, 'Hà Nội - Hai Bà Trưng - Bách Khoa', 'Bách Khoa ECB');

INSERT INTO Parking (ID, Address, Name) VALUES (3, 'Hồ Gươm,Hà Nội', 'Bãi xe Hồ Gươm');

-- Table: Bike

CREATE TABLE Bike (

BikeId INTEGER PRIMARY KEY AUTOINCREMENT,

Type NVARCHAR (50) NOT NULL,

Producer NVARCHAR (50) NOT NULL,

Deposit INTEGER NOT NULL,

Image TEXT NOT NULL,

ParkingID INTEGER NOT NULL,

IsAvailable BOOLEAN NOT NULL,

CONSTRAINT fk\_parkingID FOREIGN KEY (

ParkingID

)

REFERENCES Parking (ID)

);

INSERT INTO Bike (BikeId, Type, Producer, Deposit, Image, ParkingID, IsAvailable) VALUES (1, 'standard bike', 'YAMAHA', 500000, 'assets/image/xedapthuong.jpg', 1, 'true');

INSERT INTO Bike (BikeId, Type, Producer, Deposit, Image, ParkingID, IsAvailable) VALUES (2, 'standard ebike', 'YAMAHA', 1000000, 'assets/image/xedapdien.jpg', 2, 'true');

INSERT INTO Bike (BikeId, Type, Producer, Deposit, Image, ParkingID, IsAvailable) VALUES (3, 'twin bike', 'YAMAHA', 7000000, 'assets/image/xedapdoi.jpg', 1, 'true');

INSERT INTO Bike (BikeId, Type, Producer, Deposit, Image, ParkingID, IsAvailable) VALUES (4, 'standard bike', 'YAMAHA', 1000000, 'assets/image/xedapthuong2.jpg', 2, 'true');

INSERT INTO Bike (BikeId, Type, Producer, Deposit, Image, ParkingID, IsAvailable) VALUES (5, 'standard ebike', 'YAMAHA', 1400000, 'assets/image/xedapdien3.jpg', 2, 'true');

INSERT INTO Bike (BikeId, Type, Producer, Deposit, Image, ParkingID, IsAvailable) VALUES (6, 'standard ebike', 'HONDA', 1200000, 'assets/image/xedapdien2.jpg', 3, 'true');

-- Table: EBike

CREATE TABLE EBike (

BikeId INTEGER PRIMARY KEY,

MaxSpeed INTEGER NOT NULL,

RemainingBattery INTEGER NOT NULL,

LicensePlate VARCHAR (15),

CONSTRAINT fk\_bikeID FOREIGN KEY (

BikeId

)

REFERENCES Bike (BikeId)

);

INSERT INTO EBike (BikeId, MaxSpeed, RemainingBattery, LicensePlate) VALUES (2, 40, 70, '30A-4556');

INSERT INTO EBike (BikeId, MaxSpeed, RemainingBattery, LicensePlate) VALUES (5, 35, 80, '29A-1252');

INSERT INTO EBike (BikeId, MaxSpeed, RemainingBattery, LicensePlate) VALUES (6, 40, 100, '30Z-5612');

-- Table: RentalInvoice

CREATE TABLE RentalInvoice (

IDinvoice INTEGER PRIMARY KEY AUTOINCREMENT,

PhoneNumber CHAR (10) NOT NULL,

IDbike INTEGER NOT NULL,

Start DATETIME NOT NULL,

State INTEGER NOT NULL,

Finish DATETIME,

TotalAmount INTEGER,

CONSTRAINT fk\_BikeID FOREIGN KEY (

IDbike

)

REFERENCES Bike (BikeId),

CONSTRAINT fk\_phonenumber FOREIGN KEY (

PhoneNumber

)

REFERENCES User (PhoneNumber)

);

-- Table: PaymentTransaction

CREATE TABLE PaymentTransaction (

ID INTEGER PRIMARY KEY AUTOINCREMENT,

CardCode VARCHAR (15) NOT NULL,

Content TEXT NOT NULL,

Amount INTEGER NOT NULL,

CreateAt DATE NOT NULL,

Command VARCHAR (20) NOT NULL,

IDinvoice INTEGER NOT NULL,

CONSTRAINT fk\_invoice FOREIGN KEY (

IDinvoice

)

REFERENCES RentalInvoice (IDinvoice),

CONSTRAINT fk\_cardcode FOREIGN KEY (

CardCode

)

REFERENCES Card (Cardcode) ON UPDATE CASCADE

);

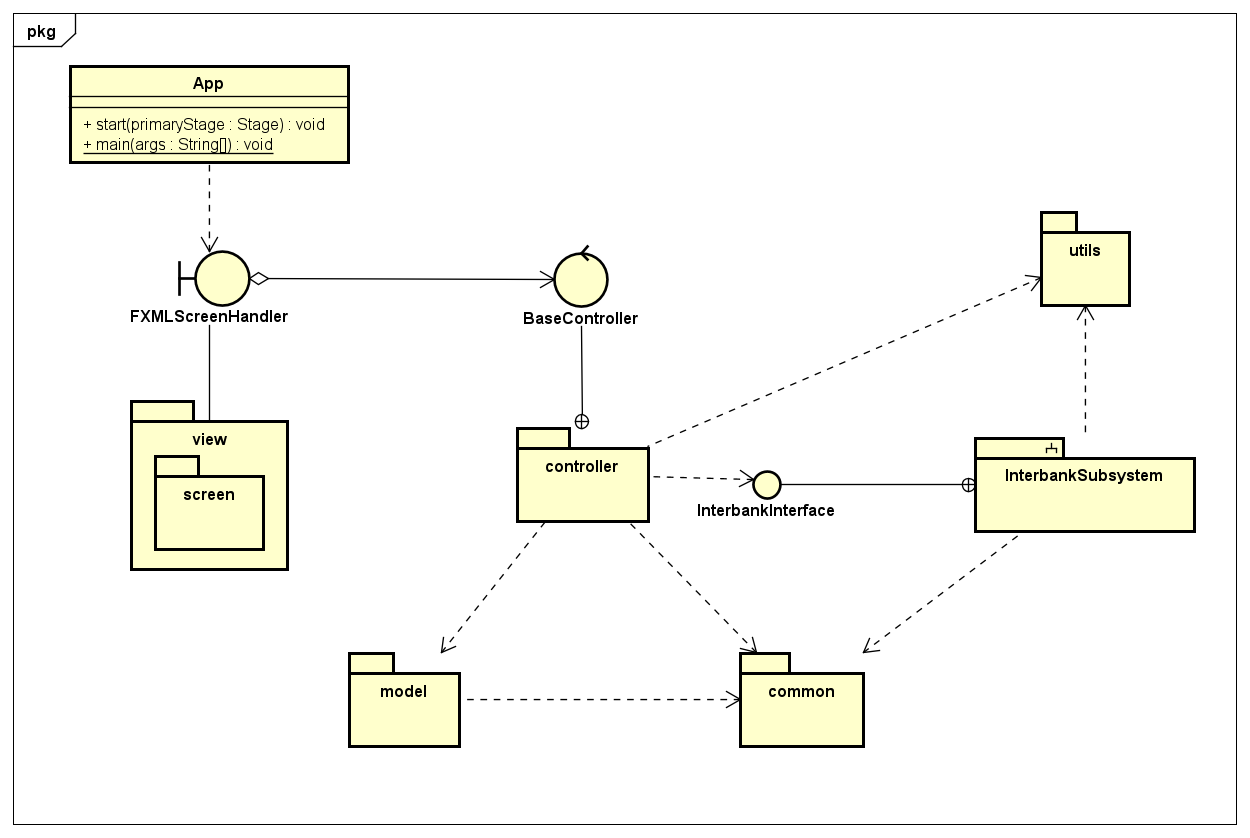
COMMIT TRANSACTION;

PRAGMA foreign\_keys = on;

## Non-Database Management System Files

## Class Design

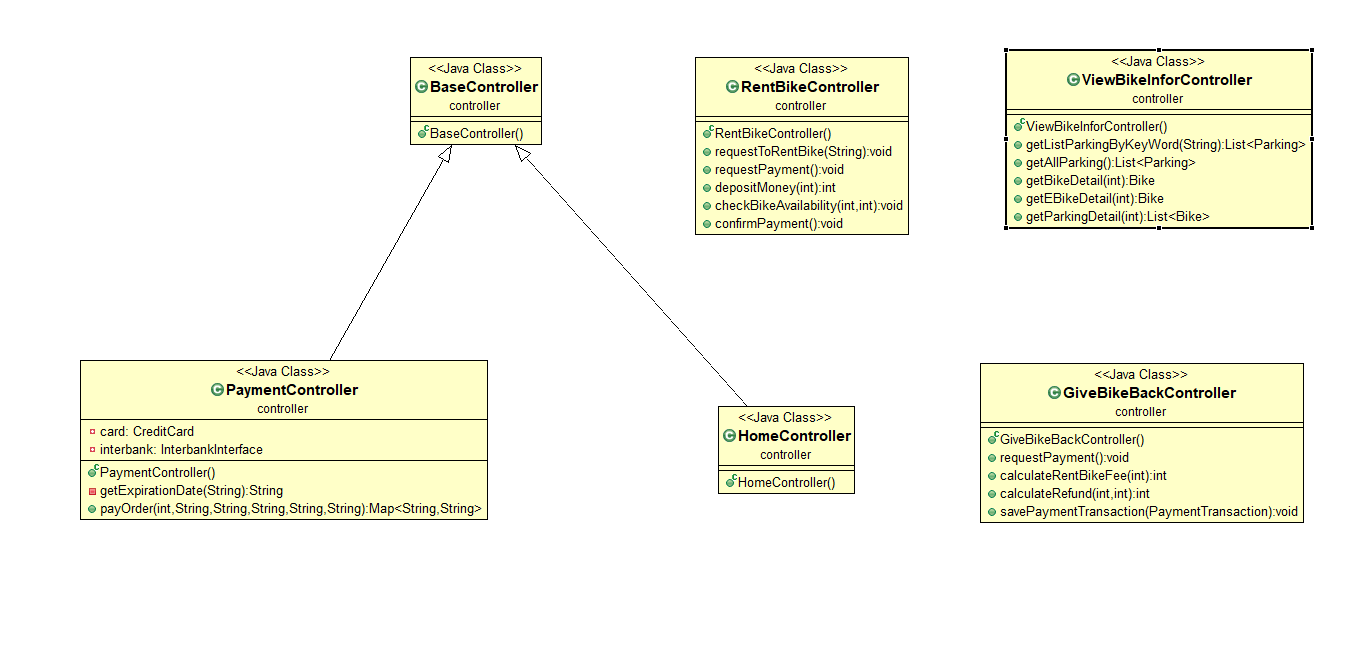
### General Class Diagram



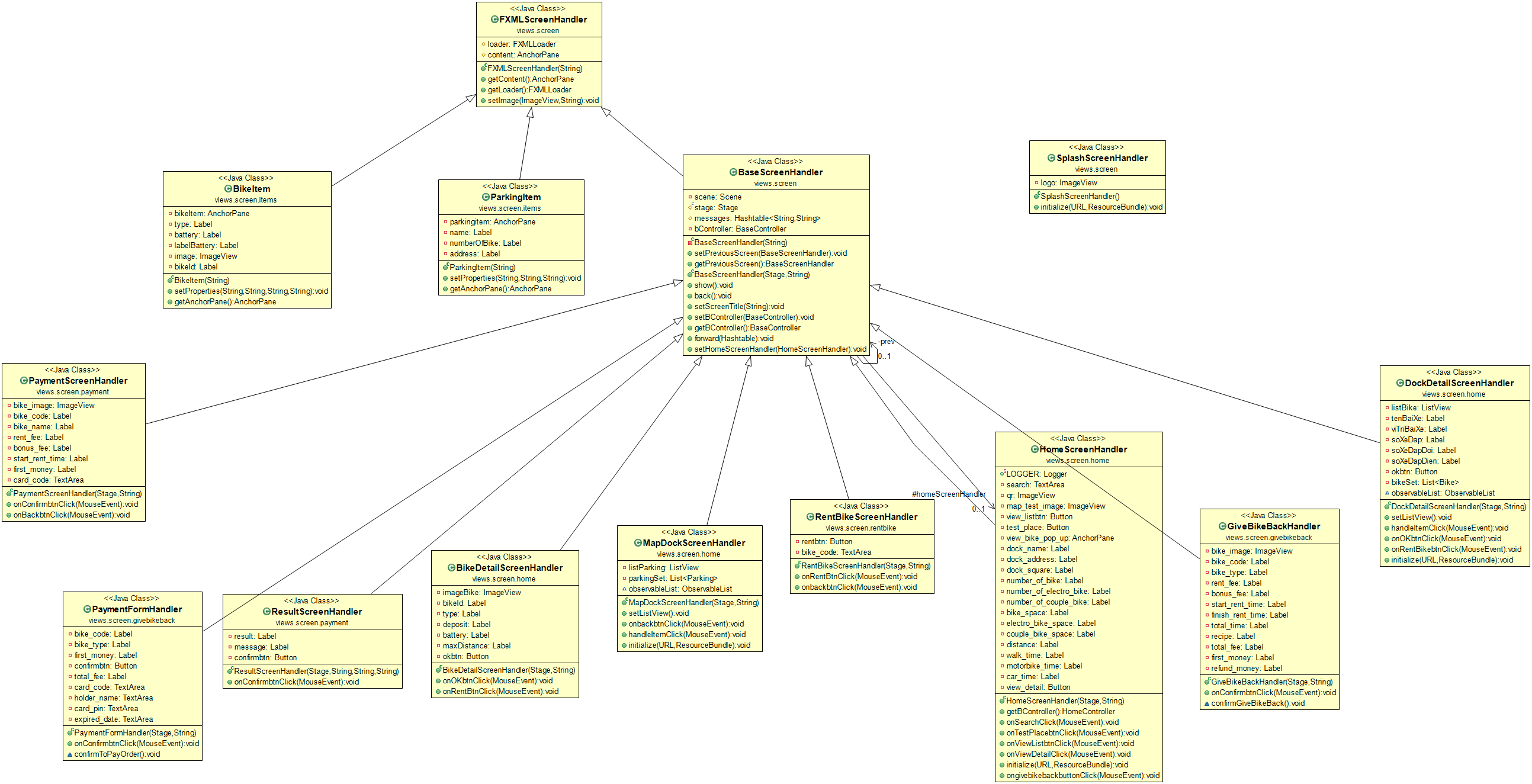
### Class Diagrams

<Detail class diagram with full attributes and operations>

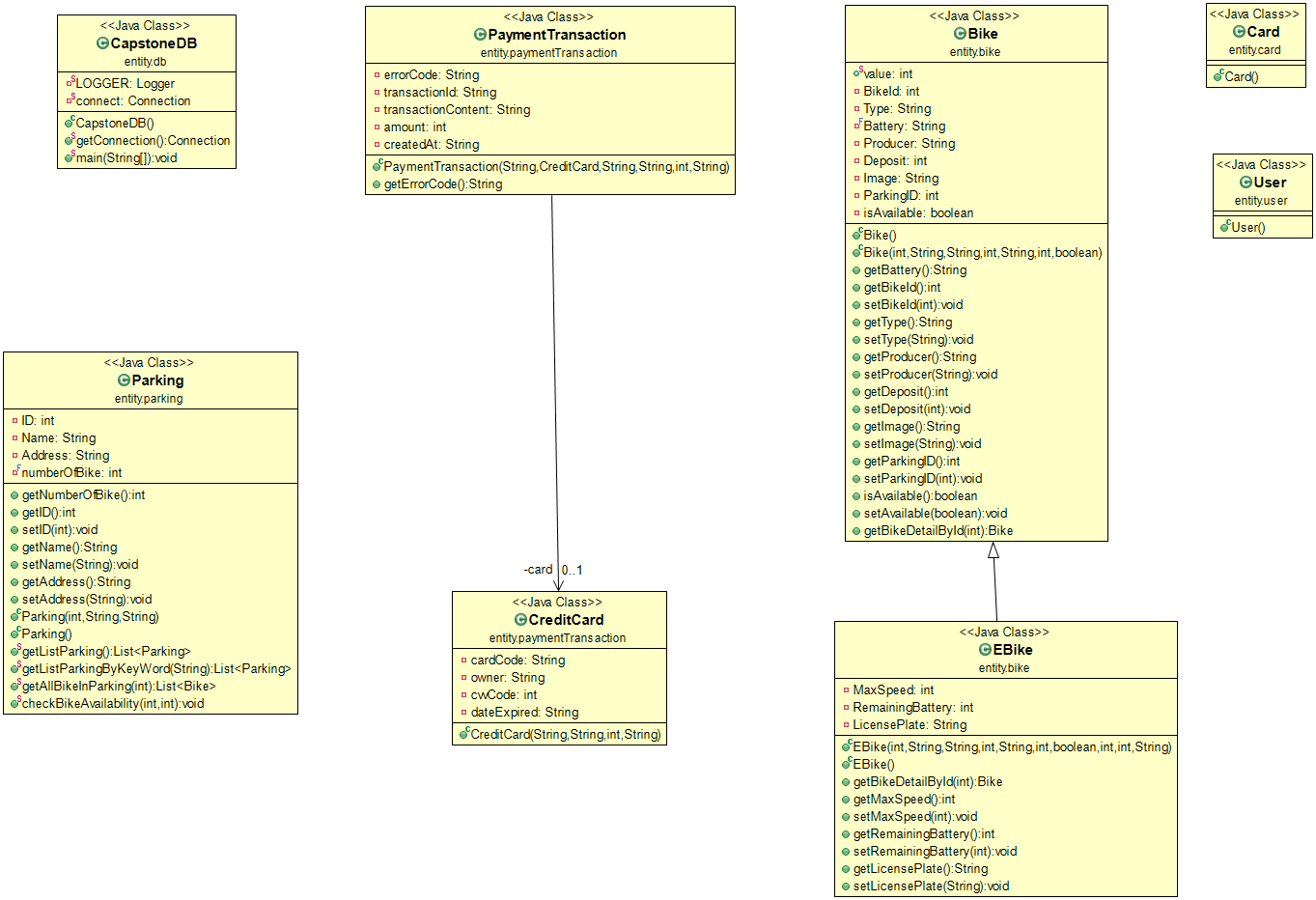
#### Class Diagram for Package Controller



#### Class Diagram for package View



#### Class Diagram for package entity

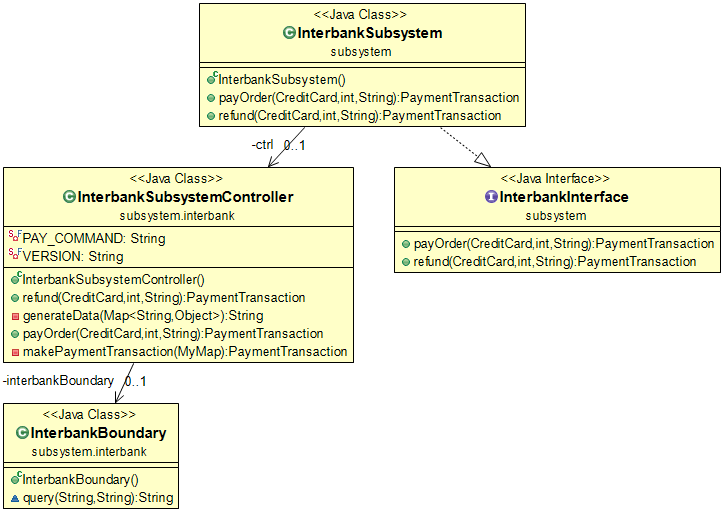


#### Class diagram for package Utils

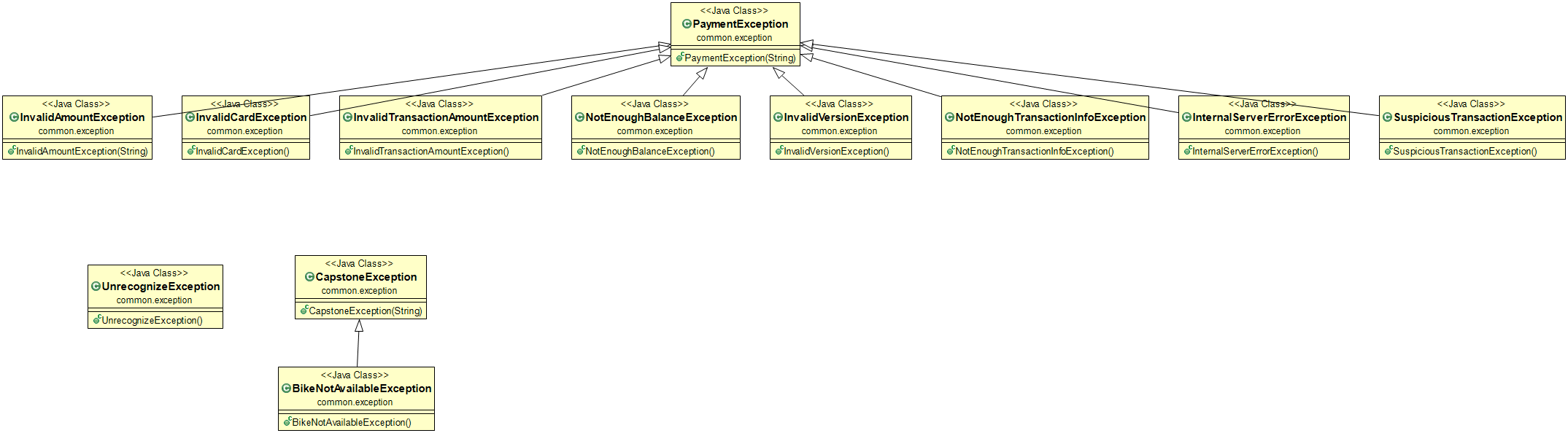
Ảnh có chứa văn bản

Mô tả được tạo tự động

#### Class diagram for subsystem interbank

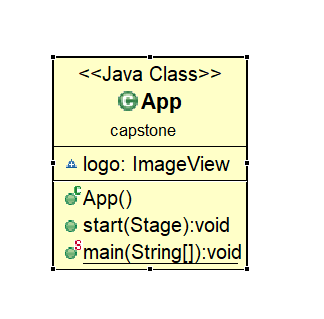


#### Class diagram for package common.exception



### Class Design

#### Class “App”



Attribute

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *#* | *Name* | *Data type* | *Default value* | *Description* |
| 1 | Logo | ImageView |  | Logo của ứng dụng |

Operation

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | start | void | Gọi bởi javafx |
| 2 | main | void | Nơi bắt đầu khởi chạy ứng dụng |

*Parameter*:

* Stage : javafx main stage

*Exception*: None

**Method :**

None

**State**

None

#### Class “ViewBikeInfoController”

Ảnh có chứa văn bản

Mô tả được tạo tự độngAttribute

None

Operation

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | getBikeDetail | Bike | Lấy thông tin chi tiết xe |
| 2 | getListParkingByKeyWord | List<Parking> | Lấy danh sách bãi xe theo từ khóa |
| 3 | getAllParking | List<Parking> | Lấy danh sách tất cả bãi xe |
| 4 | getParkingDetail | List<Bike> | Lấy danh sách tất cả xe trong bãi |
| 5 | getEBikeDetail | Bike | Lấy thông tin xe |

*Parameter*:

* getBikeDetail
  + bikeID : id của xe
* getEBikeDetail
  + bikeID : id của xe
* getListParkingByKeyWord
  + keyword : từ khóa người dùng nhập vào
* getParkingDetail
  + parkingID : id của bãi xe

*Exception*:

* SQLException

**Method**

None

**State**

None

#### Class “RentBikeController”

Ảnh có chứa bàn

Mô tả được tạo tự động

**Attribute**

None

Operation

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | requestToRentBike | void | Request thuê xe |
| 2 | depositMoney | Int | Tính tiền cọc |
| 3 | checkBikeAvailability | void | Kiểm tra tình trạng sãn sàng của xe |

*Parameter*:

* bikeID : id của xe
* bike\_cost : giá tiền xe
* idparking : id của bãi xe

*Exception*:

* BikeNotAvailableException
* SQLEception

**Method**

None

**State**

None

#### Class “GiveBikeBackController”

Ảnh có chứa văn bản

Mô tả được tạo tự động

Attribute

None

**Operation**

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | requestToPayment | void | Request thanh toán |
| 2 | calculateRentBikeFee | Int | Tính tổng tiền thuê xe |
| 3 | calculateRefund | Int | Tính tiền thừa |

*Parameter*:

* time : thời gian thuê xe , đơn vị phút
* rentfee : phí thuê xe
* bike\_cost : giá xe

*Exception*:

None

**Method**

None

**State**

None

#### Class “PaymentController”

Ảnh có chứa bàn

Mô tả được tạo tự độngAttribute

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *#* | *Name* | *Data type* | *Default value* | *Description* |
| 1 | card | CreditCard |  | Thẻ thanh toán của người dùng |
| 2 | Interbank | InterbankInterface |  | Interface interbank |

Operation

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | payOrder | Map<String,String> | Thực hiện thanh toán hóa đơn |

*Parameter*:

* amount : số tiền cần thanh toán
* contents : nội dung thanh toán
* cardNumber : số thẻ
* cardHolderName : tên chủ thẻ
* expirationDate : ngày hết hạn
* securityCode : mã bảo mật

*Exception*:

None

**Method**

getExpirationDate : lấy thông tin ngày hết hạn của thẻ

**State**

None

#### Class “BaseScreenHandler”

Ảnh có chứa bàn

Mô tả được tạo tự độngAttribute

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *#* | *Name* | *Data type* | *Default value* | *Description* |
| 1 | scene | Scene |  | Màn hình để render |
| 2 | prev | BaseSceneHandler |  | Previous scene |
| 3 | stage | Stage |  | Javafx stage |
| 4 | homeScreenHandler | HomeScreenHandler |  | Home screen |
| 5 | bController | BaseController |  | Controller cho screen này |

Operation

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | show | void | Hiển thị màn hình được chọn |
| 2 | back | void | Trở về màn hình trước đó |

*Parameter*:

None

*Exception*:

None

**Method**

None

**State**

None

#### Class “FXMLScreenHandler”

Ảnh có chứa bàn

Mô tả được tạo tự động

Attribute

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *#* | *Name* | *Data type* | *Default value* | *Description* |
| 1 | loader | FXMLLoader |  | Javafx loader |
| 2 | content | AnchorPane |  |  |

Table 1. Example of operation design

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | setImage | void | Set Image cho 1 imageview |

*Parameter*:

* imv : image view
* path : path tới image

*Exception*:

None

**Method**

None

**State**

None

#### Class “SplashScreenHandler”

Ảnh có chứa bàn

Mô tả được tạo tự độngAttribute

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *#* | *Name* | *Data type* | *Default value* | *Description* |
| 1 | logo | ImageView |  | Imageview hiển thị logo ứng dụng |

Operation

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | initialize | void | Khởi tạo splash screen |

*Parameter*:

* location : Url tới ảnh logo
* resources

*Exception*:

None

**Method**

None

**State**

None

#### Class “HomeScreenHandler”

Ảnh có chứa văn bản

Mô tả được tạo tự độngAttribute

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *#* | *Name* | *Data type* | *Default value* | *Description* |
| 1 | search | TextArea |  | Search bar |
| 2 | qr | ImageView |  | Image QR code |
| 3 | map\_test\_image | ImageView |  | Image map |
| 4 | view\_listbtn | Button |  | Button hiển thị danh sách parking |
| 5 | test\_place | Button |  |  |
| 6 | view\_bike\_pop\_up | AnchorPane |  | Popup view bike |
| 7 | dock\_name | Label |  | Label parking name |
| 8 | dock\_address | Label |  | Label parking address |
| 9 | dock\_square | Label |  | Label parking square |
| 10 | number\_of\_bike | Label |  | Label number of bike |
| 11 | number\_of\_electro\_bike | Label |  | Label number of e-bike |
| 12 | number\_of\_couple\_bike | Label |  | Label number of couple bike |
| 13 | bike\_space | Label |  | Bike space |
| 14 | electro\_bike\_space | Label |  | e-bike space |
| 15 | couple\_bike\_space | Label |  | Couple bike space |
| 16 | distance | Label |  | Distance |
| 17 | walk\_time | Label |  | Thời gian đi bộ |
| 18 | motorbike\_time | Label |  | Thời gian đi xe máy |
| 19 | car\_time | Label |  | Thời gian đi ô tô |
| 20 | view\_detail | Button |  | Button view detail parking |

Operation

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | onViewListbtnClick | void | hiển thị danh sách bãi xe |
| 2 | initialize | void | Khởi tạo màn hình |
| 3 | ongivebikebackbuttonClick | void | Thực hiện trả xe |
| 4 | onSearchClick | void | Tìm kiếm bãi xe |
| 5 | onTestPlacebtnClick | void |  |

*Parameter*:

*Exception*:

None

**Method**

None

**State**

None

#### Class “MapDockScreenHandler”

Ảnh có chứa văn bản

Mô tả được tạo tự động

Attribute

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *#* | *Name* | *Data type* | *Default value* | *Description* |
| 1 | listParking | ListView |  | ListView hiển thị danh sách bãi xe |
| 2 | parkingSet | List<Parking> |  | List danh sách thông tin bãi xe |
| 3 | observableList | ObservableList |  | Observable list |

Operation

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | setListView | void | Lấy danh sách bãi xe , hiển thị vào listview |
| 2 | onbackbtnClick | void | Trở về screen trước |
| 3 | handleItemClick | void | Xử lý sự kiện chọn bãi xe |
| 4 | initialize | void | Gọi khi khởi tạo |

*Parameter*:

* location
* event : bắt sự kiện click

*Exception*:

* IOException

**Method**

None

**State**

None

#### Class “DockDetailScreenHandler”

Ảnh có chứa văn bản

Mô tả được tạo tự động

Attribute

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *#* | *Name* | *Data type* | *Default value* | *Description* |
| 1 | listBike | ListView |  | Hiển thị danh sách xe |
| 2 | tenBaiXe | Label |  | Hiển thị tên bãi xe |
| 3 | vitriBaiXe | Label |  | Vị trí bãi xe |
| 4 | soXeDap | Label |  | Số xe đạp sẵn sàng |
| 5 | soXeDapDoi | Label |  | Số xe đạp đôi sẵn có |
| 6 | soXeDapDien | Label |  | Số xe đạp điện sẵn có |
| 7 | Okbtn | Button |  | Button |
| 8 | bikeSet | List<Bike> |  | Danh sách xe |
| 9 | observableList | ObservableList |  | Observable list |

Operation

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | setListView | void | Lấy danh sách xe , hiển thị vào listview |
| 2 | handlerItemClick | void | Xử lý sự kiện chọn xe |
|  | onOkbtnClick | void | Sự kiện click okbtn |
|  | onRentBikebtnClick | void | Chuyển sang màn thuê xe |
|  | initialize | void | Gọi khi khởi tạo |

*Parameter*:

* location
* resources
* event :mouse event

*Exception*:

None

**Method**

None

**State**

None

#### Class “BikeDetailScreenHandler”

Ảnh có chứa văn bản

Mô tả được tạo tự động Attribute

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *#* | *Name* | *Data type* | *Default value* | *Description* |
| 1 | imageBike | ImageView |  | Image of bike |
| 2 | bikeId | Lable |  | Id bike |
| 3 | type | Lable |  | Loại xe |
| 4 | deposit | Lable |  | Tiền cọc |
| 5 | battery | Lable |  | Pin còn lại |
| 6 | maxDistance | Lable |  | Khoảng cách tối đa đi được |

Operation

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | onOKbtnClick | void | Trở về màn hình trước |
| 2 | onRentBtnClick | void | Tiến hành chuyển sang màn hình thanh toán |

*Parameter*:

event :mouse event

*Exception*:

None

**Method**

None

**State**

None

#### Class “GiveBikeBackHandler”

Ảnh có chứa văn bản

Mô tả được tạo tự động

Attribute

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *#* | *Name* | *Data type* | *Default value* | *Description* |
| 1 | bike\_image | ImageView |  | Ảnh xe |
| 2 | bike\_code | Label |  | Mã xe |
|  | bike\_type | Label |  | Loại xe |
|  | rent\_fee | Label |  | Phí thuê |
|  | bonus\_fee | Label |  | Phí thêm |
|  | start\_rent\_time | Label |  | Thời gian bắt đầu thuê |
|  | finish\_rent\_time | Label |  | Thời gian kết thúc thuê |
|  | total\_time | Label |  | Tổng thời gian |
|  | recipe | Label |  | Công thức tính |
|  | total\_fee | Label |  | Tổng phí |
|  | first\_money | Label |  | Tiền ban đầu |
|  | refund\_money | Label |  | Tiền hoàn lại |

Operation

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | confirmGiveBikeBack | void | Thực hiện trả xe |

*Parameter*:

event :mouse event

*Exception*:

None

**Method**

None

**State**

None

#### Class “PaymentFormHandler”

Ảnh có chứa văn bản, bàn

Mô tả được tạo tự động

Attribute

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *#* | *Name* | *Data type* | *Default value* | *Description* |
| 1 | bike\_code | Label |  | Mã xe |
| 2 | bike\_type | Label |  | Loại xe |
|  | first\_money | Label |  | Tiền trả ban đầu |
|  | confirmbtn | Button |  | Button |
|  | total\_fee | Label |  | Tổng phí |
|  | card\_code | TextArea |  | Mã thẻ |
|  | holder\_name | TextArea |  | Chủ thẻ |
|  | card\_pin | TextArea |  | Mã pin thẻ |
|  | expired\_date | TextArea |  | Ngày hết hạn thẻ |

Operation

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | confirmToPayOrder | void | Thanh toán hóa đơn thuê xe |

*Parameter*:

event :mouse event

*Exception*:

None

**Method**

None

**State**

None

#### Class “RentBikeScreenHandler”

Ảnh có chứa văn bản

Mô tả được tạo tự động

Attribute

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *#* | *Name* | *Data type* | *Default value* | *Description* |
| 1 | rentbtn | Button |  | Button |
| 2 | bike\_code | TextArea |  | Text nhập mã xe |

Operation

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | onbackbtnClick | void | Trở về màn hình trước |
| 2 | onRentBtnClick | void | Chuyển sang màn hình mượn xe |

*Parameter*:

event :mouse event

*Exception*:

None

**Method**

None

**State**

None

#### Class “PaymentScreenHandler”

Ảnh có chứa văn bản

Mô tả được tạo tự động Attribute

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *#* | *Name* | *Data type* | *Default value* | *Description* |
| 1 | bike\_image | ImageView |  | Ảnh xe |
| 2 | bike\_code | Label |  | Mã xe |
| 3 | bike\_name | Label |  | Tên xe |
| 4 | rent\_fee | Label |  | Phí thuê |
| 5 | bonus\_fee | Label |  | Phí phụ |
| 6 | start\_rent\_time | Label |  | Thời gian bắt đầu thuê |
| 7 | first\_money | Label |  | Tiền trả ban đầu |
| 8 | card\_code | TextArea |  | Mã thẻ |

Operation

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | onBackbtnClick |  | Trở về màn hình trước |
| 2 | onConfirmbtnClick |  | Thanh toán hóa đơn |

*Parameter*:

* Event : mouse event

*Exception*:

None

**Method**

None

**State**

None

#### Class “ResultScreenHandler”

Ảnh có chứa văn bản

Mô tả được tạo tự động

Attribute

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *#* | *Name* | *Data type* | *Default value* | *Description* |
| 1 | Result | Label |  | Label |
| 2 | message | Label |  | Tin nhắn thông báo |
| 3 | confirmbtn | Button |  | Button |

Operation

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | onConfirmbtnClick | void | Trở về màn hình home |

*Parameter*:

* event : mouse event

*Exception*:

None

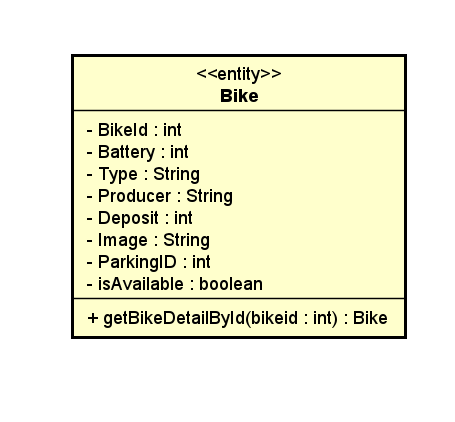
**Method**

None

**State**

None

#### Class “Bike”



Attribute

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *#* | *Name* | *Data type* | *Default value* | *Description* |
| 1 | BikeId | Int |  | Id xe |
| 2 | Battery | Int |  | % pin còn lại |
| 3 | Type | String |  | Loại xe |
| 4 | Producer | String |  | Nhà sản xuất |
| 5 | Deposit | Int |  | Tiền cọc |
| 6 | Image | String |  | ảnh xe |
| 7 | ParkingID | Int |  | Id bãi xe |
| 8 | isAvailable | Boolean |  | Tình trạng thuê của xe |

Operation

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | getBikeDetailById | Bike | Lấy thông tin chi tiết xe |

*Parameter*:

* Bikeid : id của xe

*Exception*:

* SQLException

**Method**

None

**State**

None

#### Class “Parking”

Ảnh có chứa bàn

Mô tả được tạo tự độngAttribute

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *#* | *Name* | *Data type* | *Default value* | *Description* |
| 1 | ID | Int |  | Id của bãi xe |
| 2 | Name | String |  | Tên bãi xe |
| 3 | Address | String |  | Địa chỉ bãi xe |

Operation

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | getListParking | List<Parking> | Lấy danh sách tất các bãi xe |
| 2 | getListParkingByKeyWord | List<Parking> | Lấy danh sách tất các bãi xe theo từ khóa |
| 3 | getAllBikeInParking | List<Bike> | Lấy danh sách xe |
| 4 | checkBikeAvailability | void | Kiểm tra tình trạng của xe |

*Parameter*:

* Keyword : từ khóa tìm kiếm
* idparking : id bãi xe
* BikeID : id xe

*Exception*:

* SQLException

**Method**

None

**State**

None

#### Class “PaymentTransation”

Ảnh có chứa văn bản

Mô tả được tạo tự độngAttribute

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *#* | *Name* | *Data type* | *Default value* | *Description* |
| 1 | errorCode | String |  | Mã lỗi |
| 2 | card | CreditCard |  | Thẻ thanh toán |
| 3 | transactionId | String |  | Id giao dịch |
| 4 | transactionContent | String |  | Nội dung giao dịch |
| 5 | amount | Int |  | Số tiền thanh toán |
| 6 | createAt | String |  | Thời điểm thanh toán |

Table 1. Example of operation design

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | getErrorCode | String | Lấy mã lỗi trả về |

*Parameter*:

None

*Exception*:

None

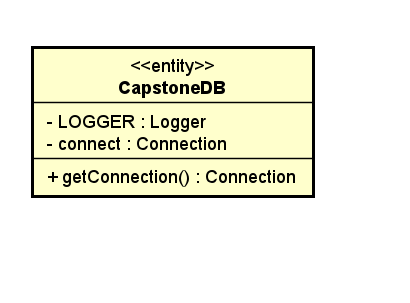
**Method**

None

**State**

None

#### Class “CapstoneDB”



Atribute

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *#* | *Name* | *Data type* | *Default value* | *Description* |
| 1 | LOGGER | Logger |  | Log của ứng ụng |
| 2 | connect | Connection |  | Đối tượng connect tới DB |

Operation

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | getConnectiom | Connection | Lấy đối tượng connect tới DB |

*Parameter*:

None

*Exception*:

None

**Method**

None

**State**

None

#### Class “InterbankSubsystem”

Ảnh có chứa văn bản

Mô tả được tạo tự động Attribute

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *#* | *Name* | *Data type* | *Default value* | *Description* |
| 1 | Ctrl | InterbankSubsystemController |  | Controller interbank |

Operation

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | payOrder | PaymentTransaction | Thanh toán hóa đơn |
| 2 | Refund | PaymentTransaction | Hoàn tiền |

*Parameter*:

* card : CreditCard
* amount : số tiền cần thanh toán
* content : nội dung thanh toán

*Exception*:

* PaymentException

**Method**

None

**State**

None

#### Class “InterbankBoundary”

Ảnh có chứa văn bản

Mô tả được tạo tự động

Attribute

None

Operation

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | query | String | Thực hiện query |

*Parameter*:

* url : đường dẫn cần query
* data : dữ liệu gửi lên

*Exception*:

None

**Method**

None

**State**

None

#### Class “InterbankSubsystemController”

Ảnh có chứa văn bản

Mô tả được tạo tự động Attribute

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *#* | *Name* | *Data type* | *Default value* | *Description* |
| 1 | PAY\_COMMAND | String |  |  |
| 2 | VERSION | String |  | Version interbank |
| 3 | interbankBoundary | InterbankBoundary |  | Interbank boundary |

Operation

|  |  |  |  |
| --- | --- | --- | --- |
| *#* | *Name* | *Return type* | *Description (purpose)* |
| 1 | refund | PaymentTransaction | Hoàn tiền |
| 2 | payOrder | PaymentTransaction | Thanh toán |

*Parameter*:

* card : CreditCard
* amount : số tiền cần thanh toán
* content : nội dung thanh toán

*Exception*:

None

**Method**

makePaymentTransaction()

**State**

None

# Design Considerations

## Goals and Guidelines

* Goals
  + Thiết kế các lớp sao cho thỏa mãn được các nguyên lý thiết kế SOLID , có thể dễ dàng bảo trì và phát triển trong tương lai
  + Tốc độ phản hồi của ứng dụng nhanh chóng
  + Giao diện dễ sử dụng
* Guidelines
  + Tuân thủ coding convention trong Java

## Architectural Strategies

* Design architecture: MVC
* Design principles: SOLID principles
* Database design: SQLite
* Framework: JavaFX.

## Coupling and Cohesion

### Coupling

#### Content coupling

|  |  |  |
| --- | --- | --- |
| Related modules | Description | Improvement |
| RentBike  ViewBikeInfor  GiveBikeBack  Payment | Không vi phạm  Truy xuất data qua các getter và setter |  |

## Common coupling

|  |  |  |
| --- | --- | --- |
| Related modules | Description | Improvement |
| RentBike  ViewBikeInfor  GiveBikeBack  Payment | Không vi phạm  Sử dụng ngôn ngữ Java không có biến global |  |

## Control coupling

|  |  |  |
| --- | --- | --- |
| Related modules | Description | Improvement |
| RentBike  ViewBikeInfor  GiveBikeBack  Payment | Không vi phạm  Các funtion không có tham số điều khiển |  |

## Stamp coupling

|  |  |  |
| --- | --- | --- |
| Related modules | Description | Improvement |
| Payment | Có thể vi phạm  Các tham số trong funtion là Map<string, string> có thể bao gồm những dữ liệu thừa trong tham số truyền vào | Có thể chấp nhận được |

## Data coupling

Chưa đạt đến mức này

### Cohesion

### Coincidental cohesion

Không vi phạm

## Logical cohesion

Không vi phạm, RentBike và GiveBikeBack liên quan đến nhau về logic chứ không về chức năng, ban đầu chúng cùng tính tiền với method calculateFee() tuy nhiên sau đó thay đổi, RentBike chỉ dùng depositFee() và GiveBikeBack có calculateFee()

## Temporal cohesion

Không vi phạm

## Procedural cohesion

Không vi phạm

## Communicational cohesion

Không vi phạm

## Sequential Cohesion

Không vi phạm

## Functional cohesion

## Design Principles

**Single Responsibility Principle**

Không vi phạm, các class đều chỉ có 1 trách nhiệm

**Open/Closed Principle**

Không vi phạm, khi muốn thay đổi các giá trị đã có class configs, khi muốn thêm chức năng thì chỉ viết class mới

**Liskov Substitution Principe**

Không vi phạm

**Interface Segregation Principle**

Không vi phạm

**Dependency Inversion Principle**

|  |  |  |
| --- | --- | --- |
| Related modules | Description | Improvement |
| Payment | Vi phạm  Vì không implement bất cứ interface nào nên nếu class khác muốn sử dụng các funtion class này sẽ bị phụ thuộc trực tiếp vào class này. | Có thể implement 1 interface khác |

## Design Patterns