HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

School of Information and communications technology

Software Design Document

Version 1.0

Eco Bike Rental Software

Subject: ITSS Software Development

Group 18

Nguyễn Mạnh Cường – 20183874

Hoàng Hải Đăng – 20183877

Nguyễn Văn Chiến – 20183870

Nguyễn Đào Duy Kiên - 20183935

*Hanoi,* *January, 2022*

**Table of Contents**

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 4

2.2.1 Assumptions 4

2.2.2 Constraints 4

2.2.3 Risks 4

3 System Architecture and Architecture Design 5

3.1 Architectural Patterns 5

3.2 Interaction Diagrams 5

3.2.1 View Dock and Bike Info 5

3.2.2 Rent Bike 7

3.2.3 Return Bike 10

3.3 Analysis Class Diagrams 12

3.3.1 Docks and Bikes 12

3.3.2 Rent Bike 12

3.3.3 Return Bike 13

3.4 Unified Analysis Class Diagram 13

3.5 Security Software Architecture 13

4 Detailed Design 14

4.1 User Interface Design 14

4.1.1 Screen Configuration Standardization 14

4.1.2 Screen Transition Diagrams 15

4.1.3 Screen Specifications 15

4.2 Data Modeling 27

4.2.1 Conceptual Data Modeling 27

4.2.2 Database Design 28

4.3 Non-Database Management System Files 39

4.4 Class Design 40

4.4.1 General Class Diagram 40

4.4.2 Class Diagrams 40

4.4.3 Class Design 42

5 Design Considerations 108

5.1 Goals and Guidelines 108

5.2 Architectural Strategies 108

5.3 Coupling and Cohesion 108

5.4 Design Principles 108

# Introduction

## Objective

Mục đích của tài liệu này là trình bày mô tả chi tiết về các thiết kế của Eco Bike Rental. Tài liệu này dành cho nhóm 18, sử dụng các thiết kế làm hướng dẫn để thực hiện dự án. Mặt khác, tài liệu này được TS.Nguyễn Thị Thu Trang và trợ giảng xem xét như một phần của khóa học.

## Scope

Mục đích của phần mềm Eco Bike Rental là cung cấp dịch vụ cho thuê xe đạp cho khách hàng.

Trong phần mềm này, người dùng sẽ được cung cấp khả năng tìm kiếm các bãi xe và xem thông tin chi tiết của nó. Tại mỗi bãi xe, người dùng có thể xem thông tin về những chiếc xe đạp đang đỗ tại đó. Cho thuê và trả xe đạp là những chức năng chính của hệ thống này.

Liên ngân hàng sẽ có khả năng thực hiện các giao dịch thanh toán. Liên ngân hàng sẽ diễn ra quá trình xác nhận, thêm tiền và khấu trừ tiền.

Tài liệu này mô tả chi tiết về kiến trúc phần mềm của hệ thống kiểm kê. Nó chỉ định cấu trúc và thiết kế của một số mô-đun được thảo luận trong SRS. Nó cũng cung cấp các sơ đồ tuần tự và hoạt động cho một số trường hợp sử dụng. Biểu đồ lớp cho thấy cách nhóm lập trình sẽ triển khai mô-đun cụ thể.

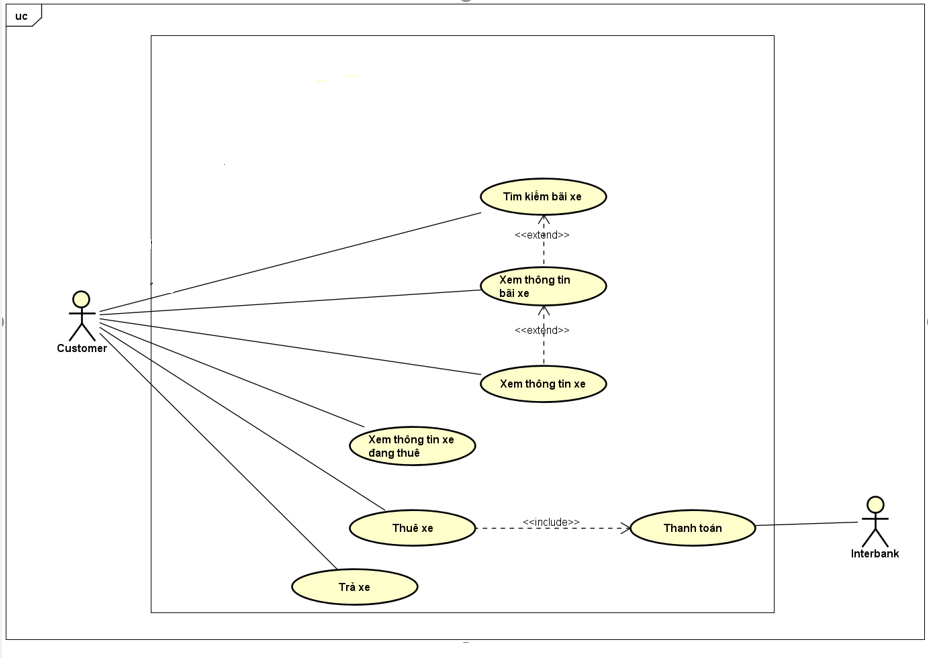
## Glossary

## References

# Overall Description

## General Overview

Hệ thống EBR là một mô phỏng cho dịch vụ cho thuê xe đạp. Đây là một phần mềm máy tính để bàn chạy trên Java Runtime Environment. Phần mềm này được thiết kế bằng cách sử dụng mô hình thiết kế kiến trúc MVC.



## Assumptions/Constraints/Risks

### Assumptions

Thiết kế được mô tả trong tài liệu này dựa trên các yêu cầu mô phỏng thay vì các yêu cầu thực tế.

Phần mềm này được thiết kế để chạy trên bất kỳ hệ điều hành nào có JRE.

### Constraints

Hệ thống được xây dựng chỉ có thể truy cập thông qua máy tính cá nhân. Hệ thống được triển khai bằng Java và sử dụng JavaFX để tạo giao diện người dùng.

Để lưu trữ dữ liệu, sử dụng PostgreSQL.

### Risks

# System Architecture and Architecture Design

## Architectural Patterns

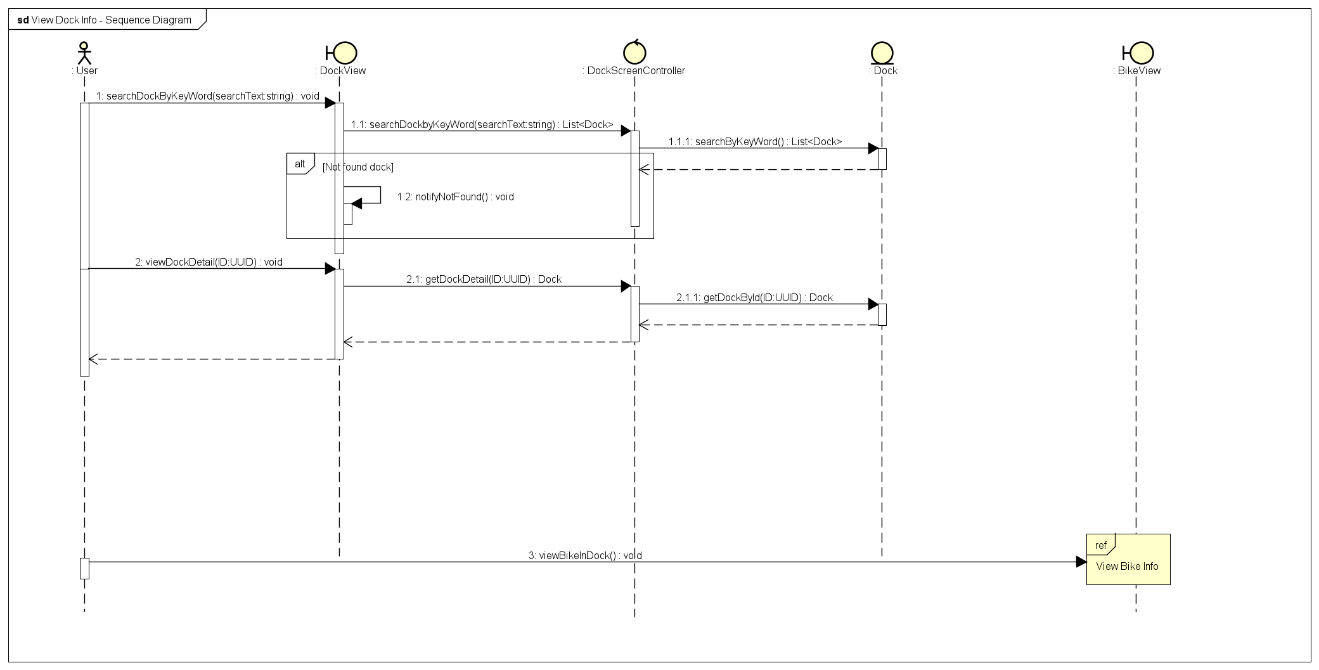
Model-View-Controller (MVC) là mẫu thiết kế được chọn.

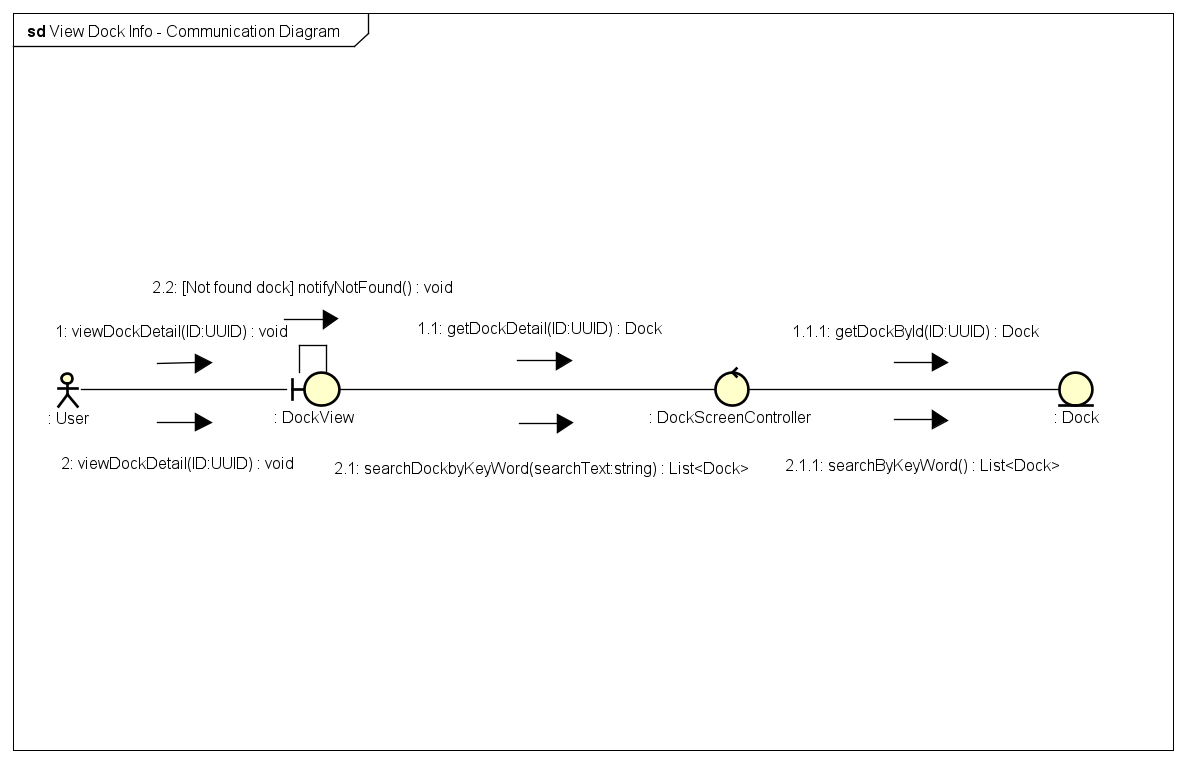
Các lợi ích của MVC:

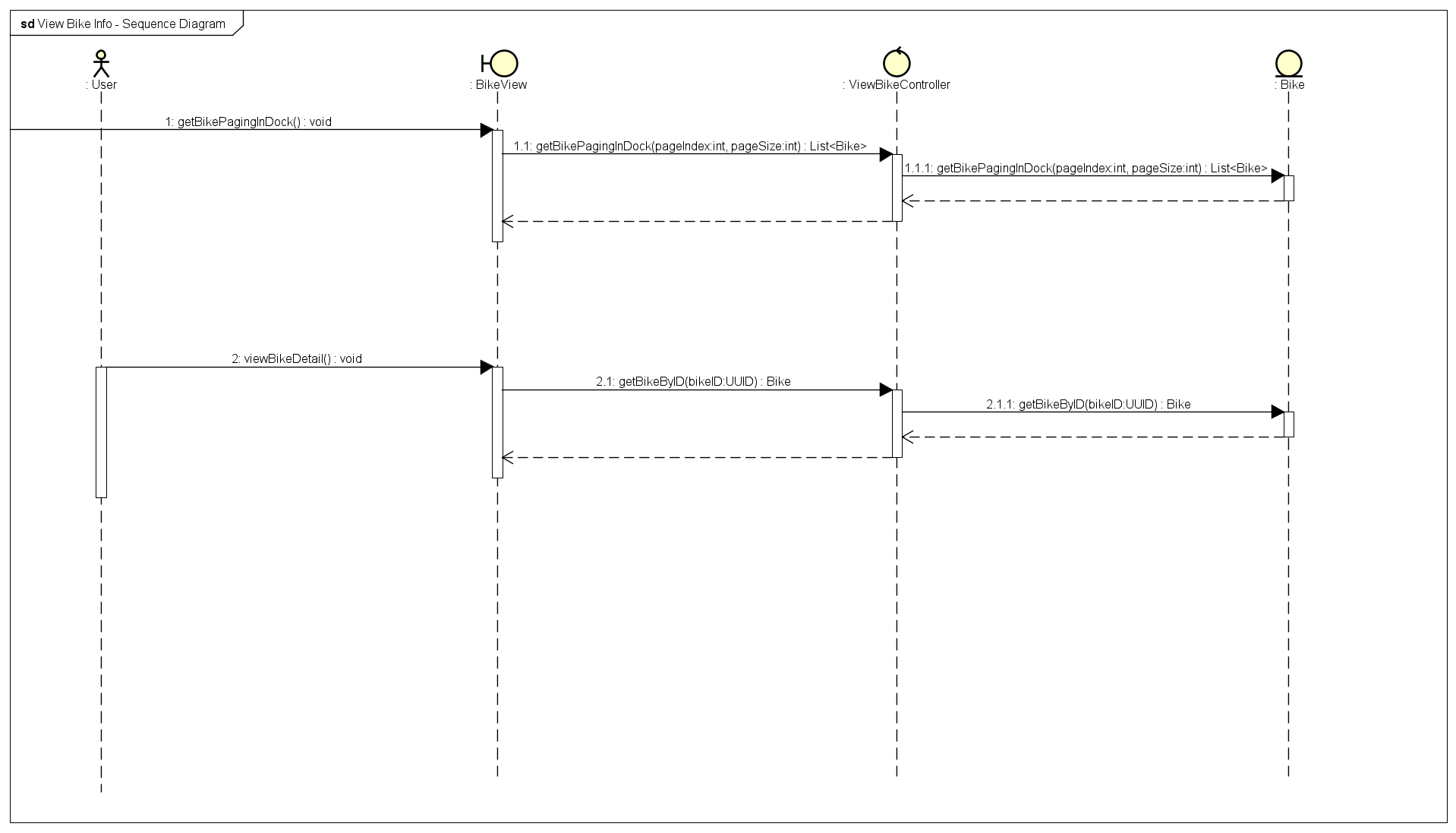
* Phát triển đồng thời
* [High cohesion](https://en.wikipedia.org/wiki/Cohesion_(computer_science))
* [Low coupling](https://en.wikipedia.org/wiki/Loose_coupling)
* Dễ dàng sửa đổi, phát triển trong tương lai

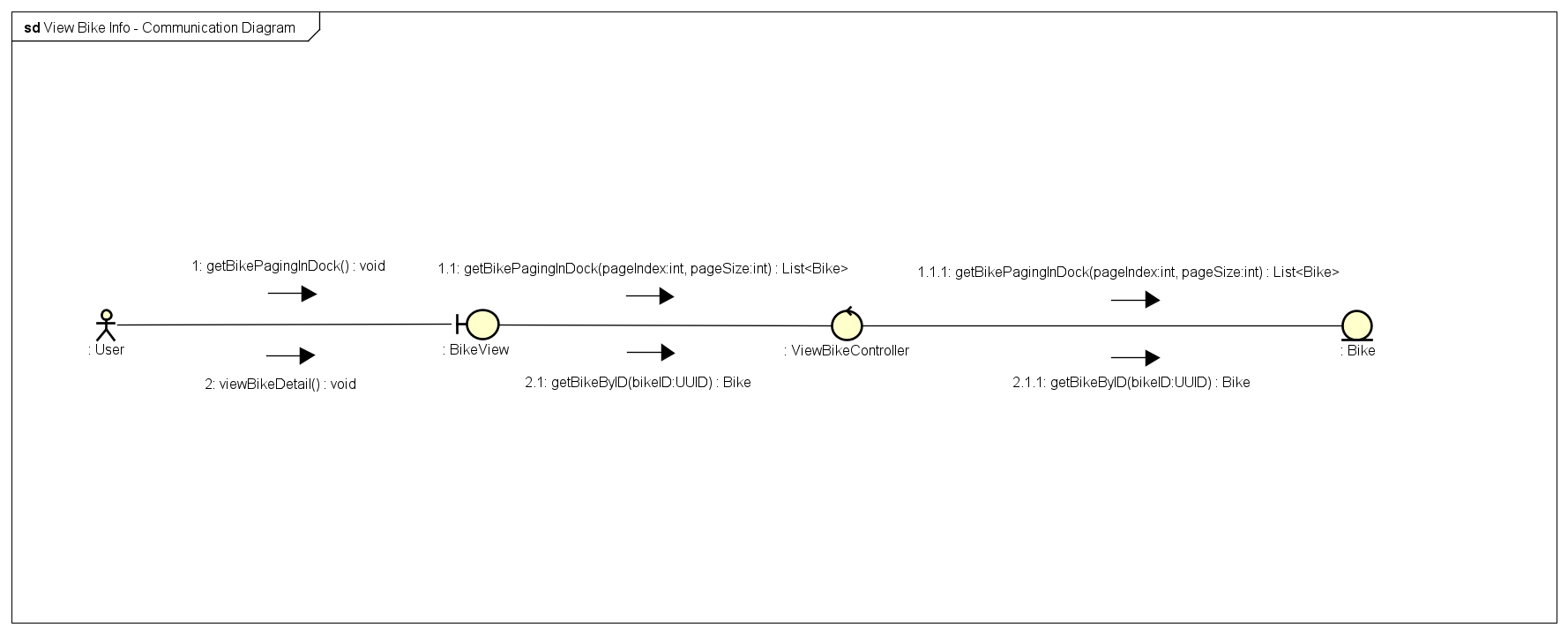
## Interaction Diagrams

### View Dock and Bike Info

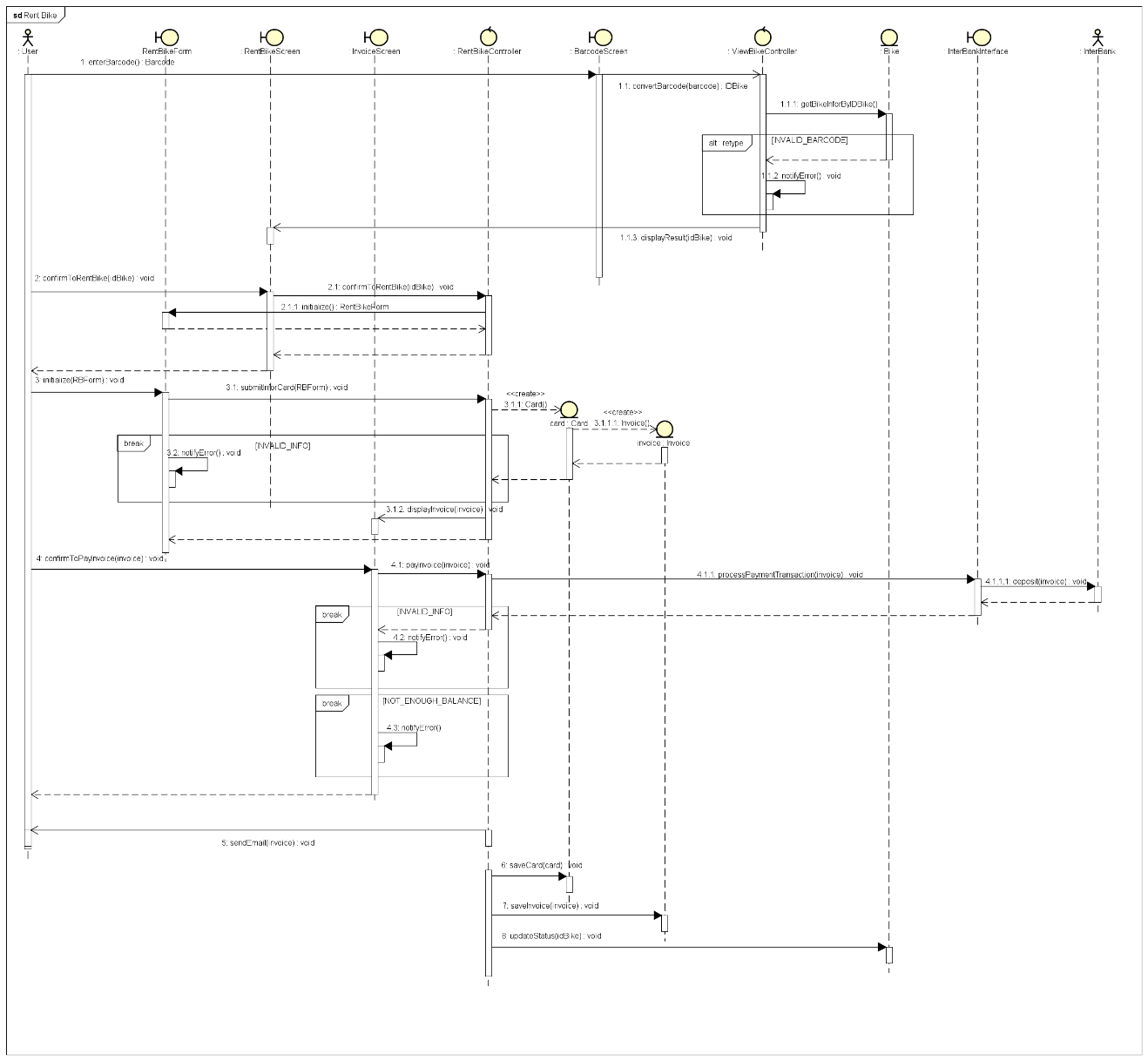


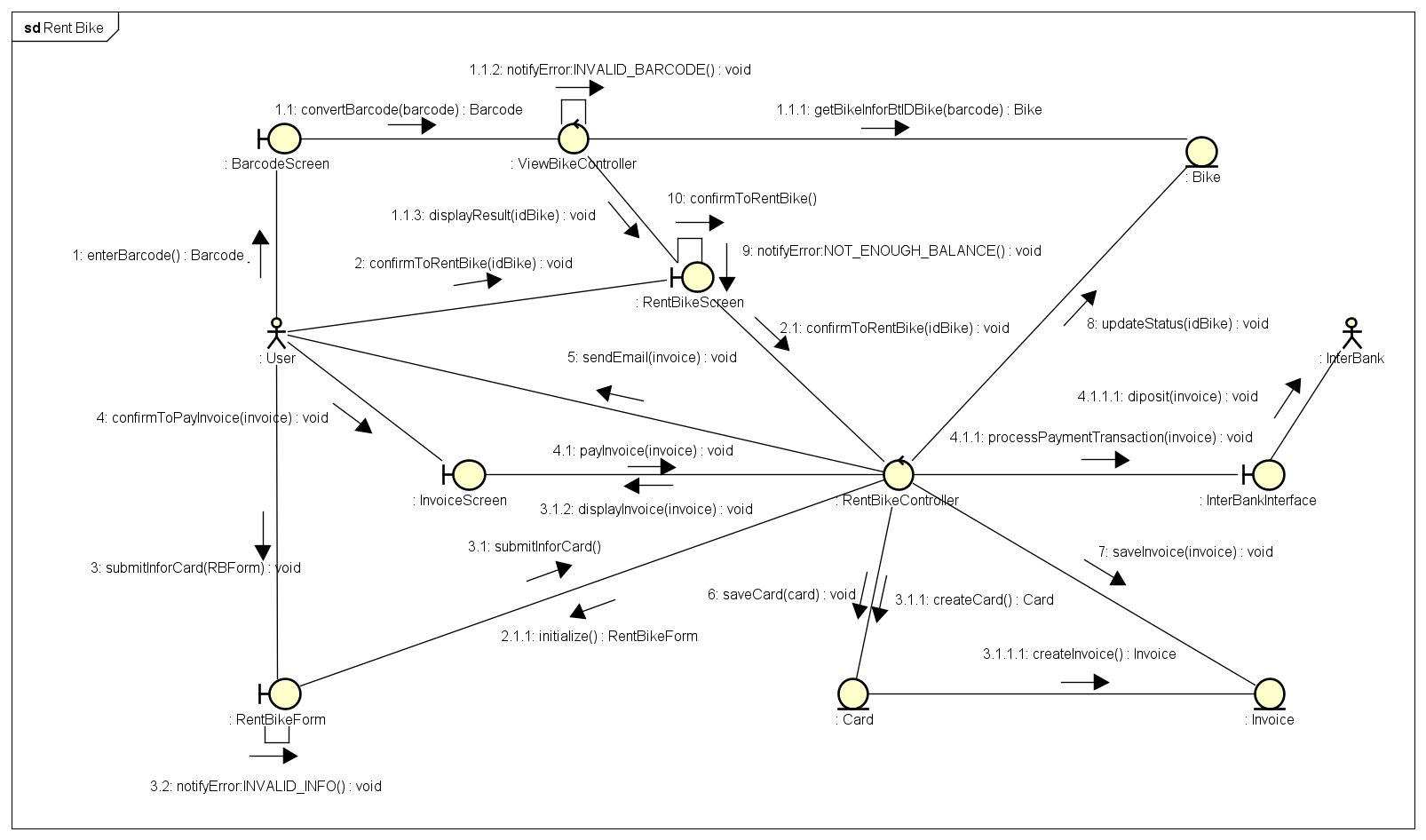




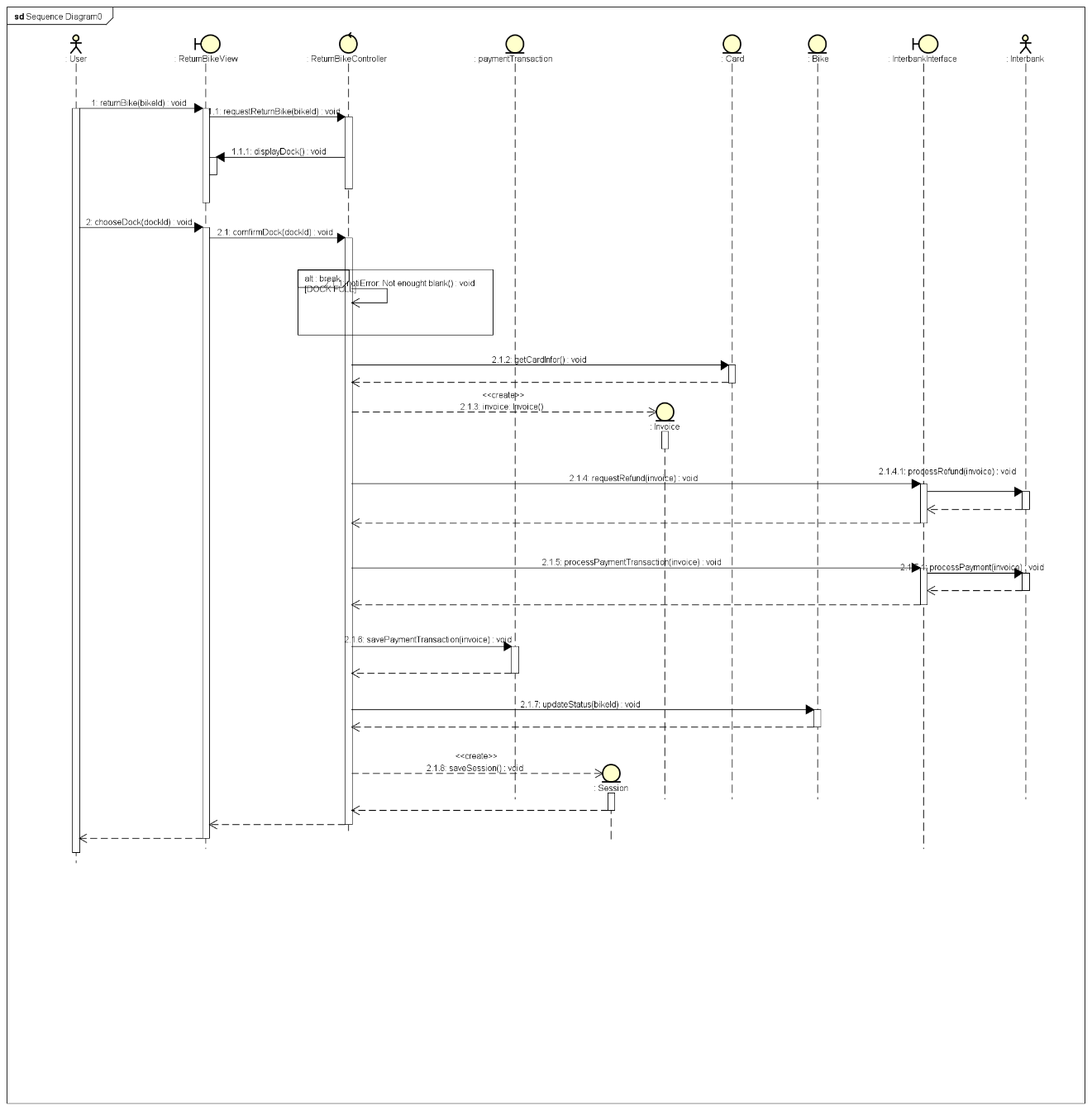


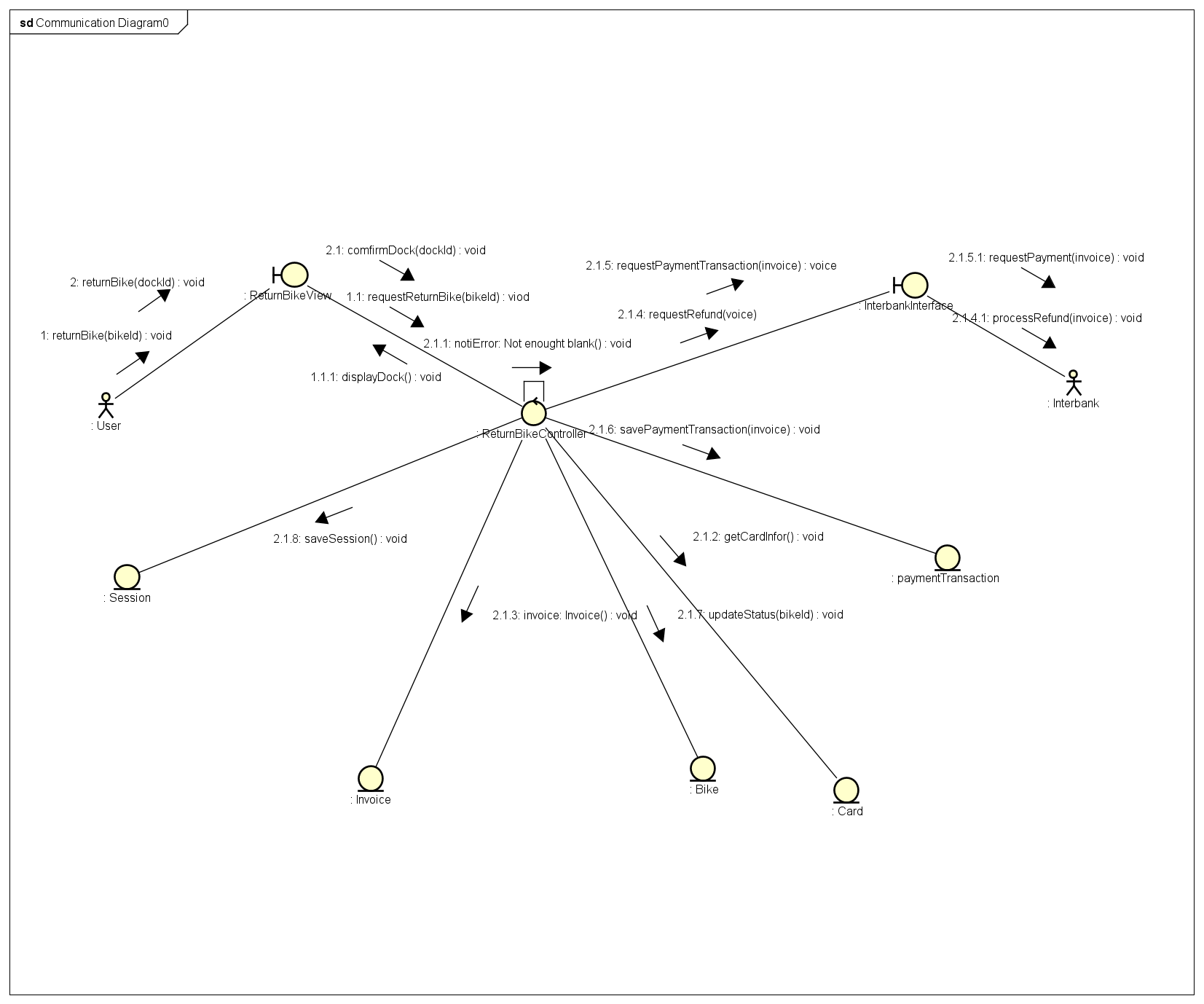
### Rent Bike





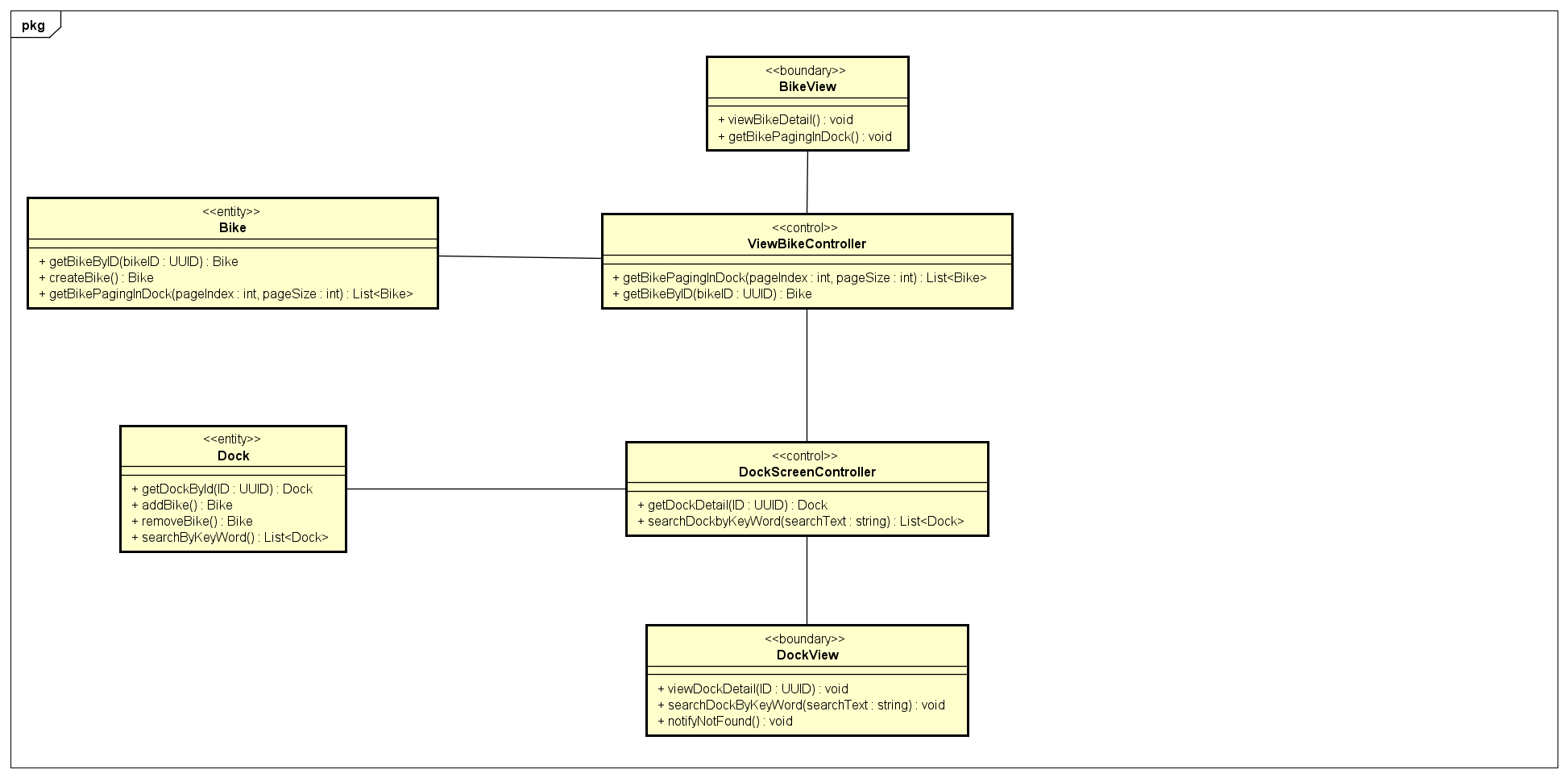
### Return Bike



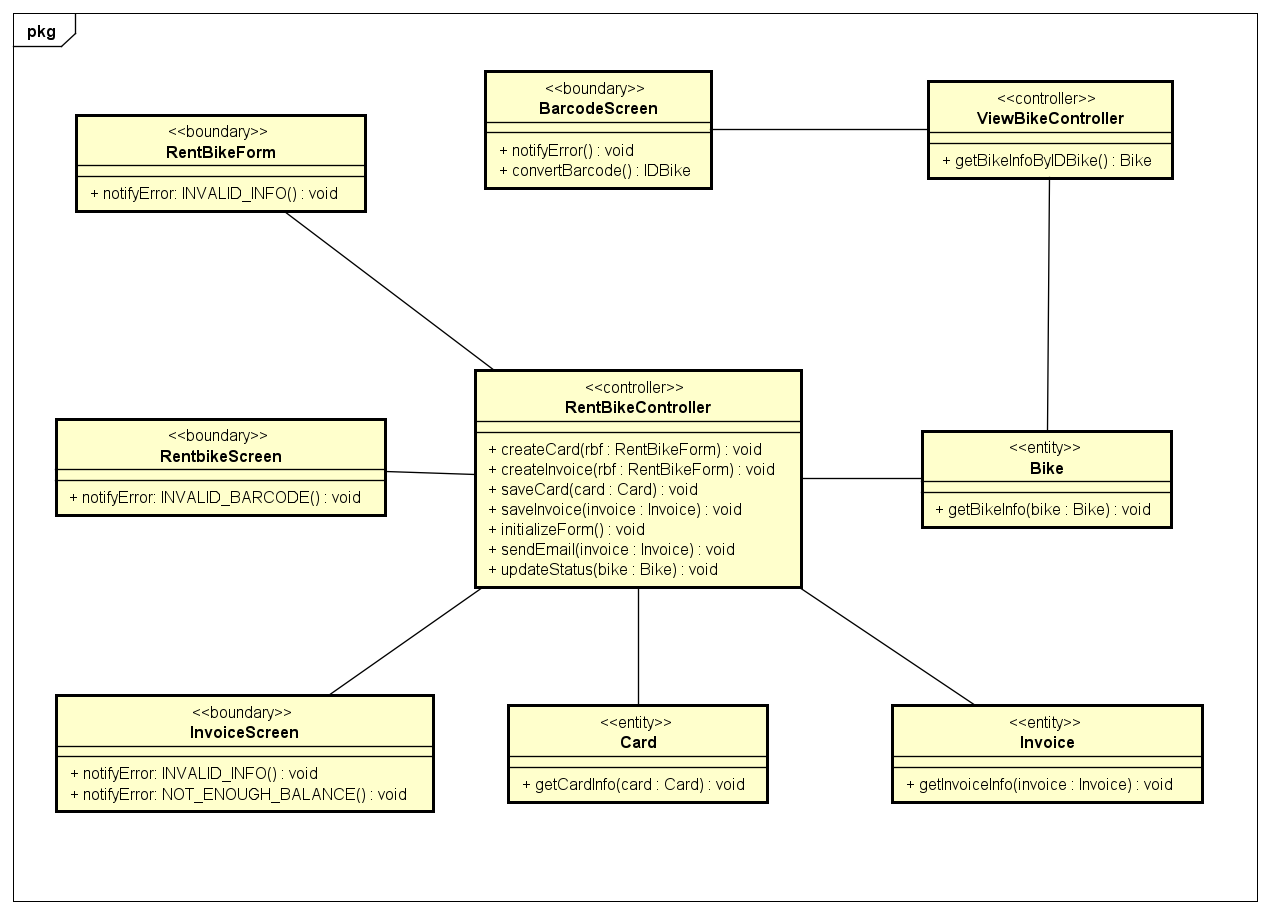


## Analysis Class Diagrams

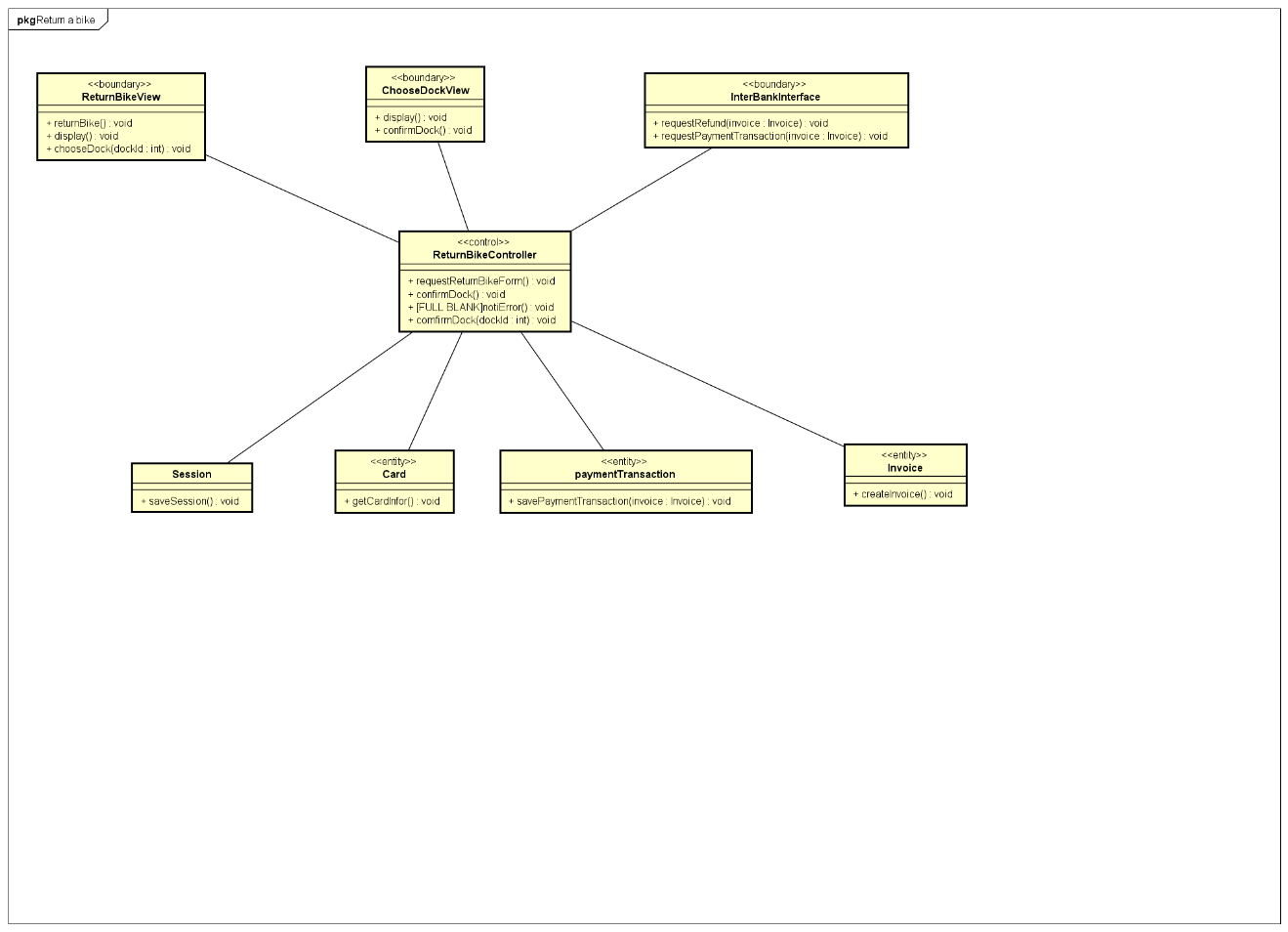
### Docks and Bikes



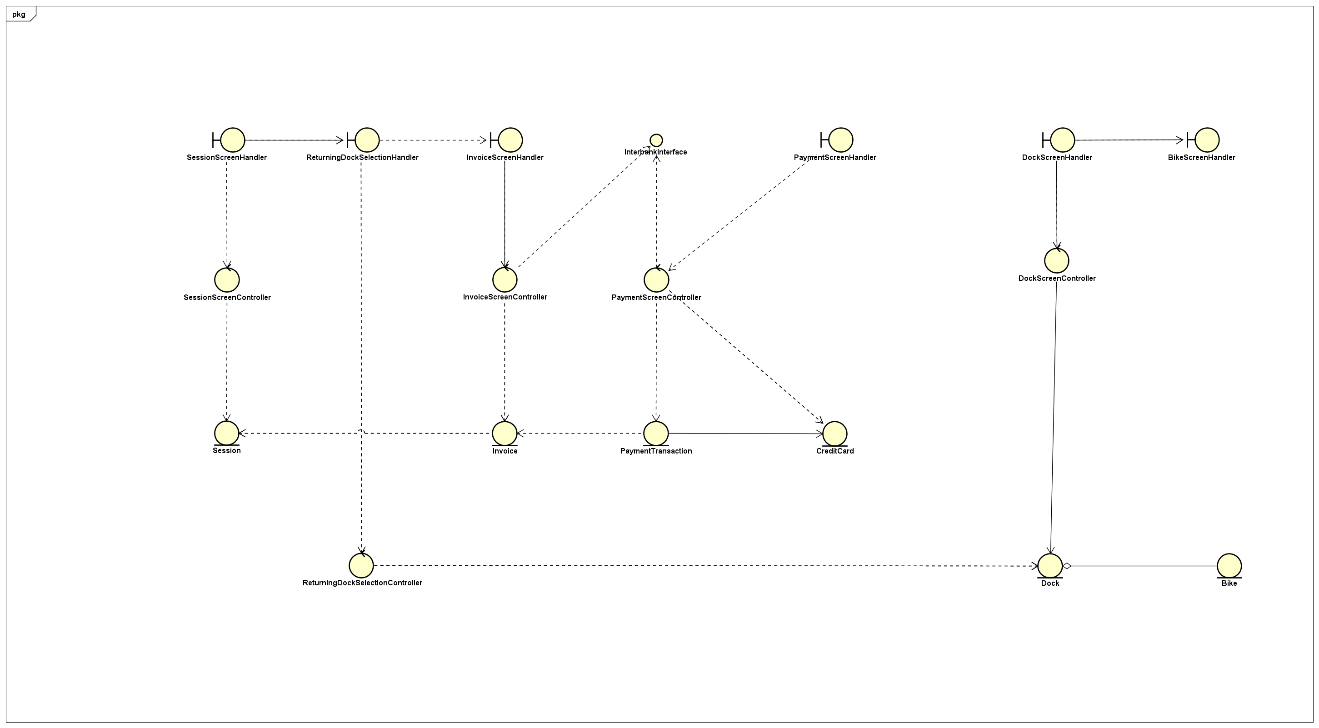
### Rent Bike



### Return Bike



## Unified Analysis Class Diagram



## Security Software Architecture

# Detailed Design

## User Interface Design

### Screen Configuration Standardization

***Display***

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

Độ phân giải: 1366 × 768 pixels

***Screen***

*Vị trí của của button*: Ở dưới cùng (theo chiều dọc) và ở giữa (theo chiều ngang) của khung.

*Vị trí thông điệp:* Ở giữa trung tâm màn hình

*Vị trí Tiêu đề màn hình:* Đặt góc bên trái màn hình

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

***Control***

*Kích thước chữ:*

* Title and header text: font: System, style: bold, size: 48px, color: #36C0AA

(medium-dark purple)

* Text in button: font: System, style: black, size: 24px, color: #FFFFFF
* Other text: font: System, style: regular, size: 14px, color: #000000

*Xử lí kiểm tra đầu vào*: 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. Tuy nhiên, hướng dẫn sử dụng được xem như là 1 popup message vì màn hình chính ở dưới sẽ không thể thao tác trong khi màn hình hướng dẫn sử dụng đang được hiển thị. Ban đầu khi app khởi chạy thì màn hình splash screen (màn hình chớp) sẽ được hiện lên và sau đó màn hình đầu tiên(Home Screen) sẽ xuất hiện.

*Thứ tự các màn hình trong hệ thống:*

1. Splash screen (first screen)
2. Home screen – Hiện danh sách bãi xe
3. Barcode Popup – Nhập Barcode
4. Dock View Screen – Xem thông tin bãi xe vào danh sách xe trong bãi
5. Bike View Screen – Xem thông tin chi tiết xe
6. Payment Screen – Điền thông tin thẻ
7. Transaction Result Popup – Hiện kết quả giao dịch
8. Renting Session – Thông tin xe đang thuê
9. Invoice screen – Xem chi tiết hóa đơn
10. Dock Selection Screen – Hiển thị để chọn bãi xe cho việc trả xe

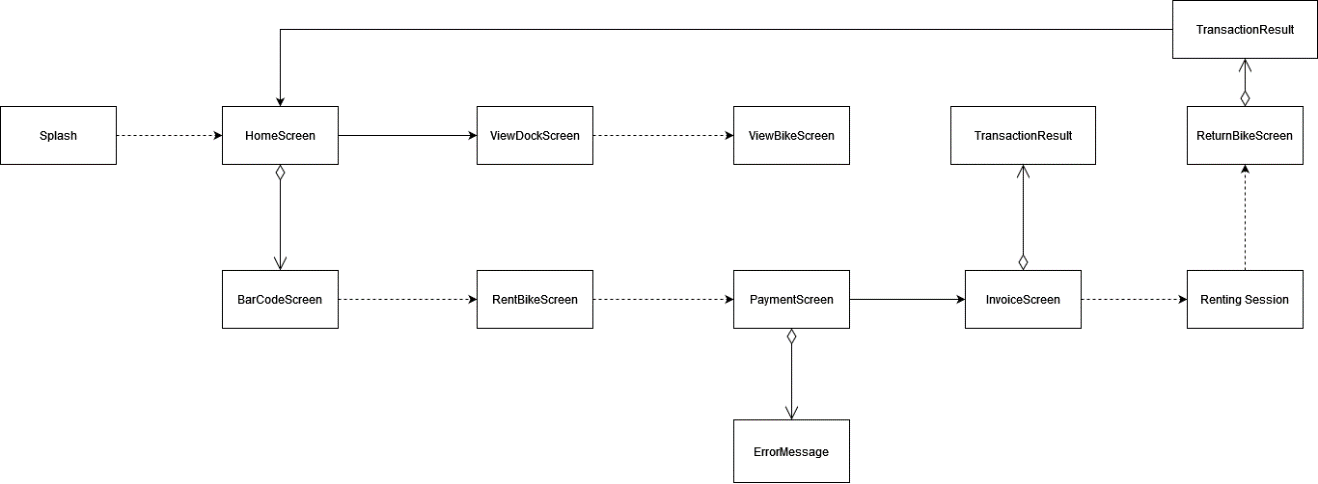
***Nhập đầu vào từ bàn phím***

Sẽ không có phím tắt. Có các button quay lại để quay lại các màn hình trước đó. Ngoài ra button “X” nằm ở thanh tiêu đề bên phải để đóng screen

***Error***

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

### Screen Transition Diagrams



### Screen Specifications

#### Home Screen

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **EBR Software** | | **Date of creation** | **Approved by** | **Reviewed by** | **Person in charge** |
| **Screen specification** | Home Screen | 17/11/2021 |  |  | Nguyễn Mạnh Cường |
|  | | **Control** | **Operation** | **Function** | |
| “Enter Barcode” Button | Click | Display Barcode Popup Screen | |
| Area for entering search | Initial | Receive Text for Searching | |
| “Search” Button | Click | Searching Dock | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Screen name | Home Screen | | | |
| Item name | Number of digits (bytes) | Type | Field attribute | Remarks |
| Search field | 50 | String | Black | Left justified |
| Dock’s name | 50 | String | Green | Left justified |
| Dock’s address | 50 | String | Black | Left justified |
| Available | 10 | String | Black | Left justified |

#### Dock View Screen

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| EBR Software | | Date of creation | Approved by | Reviewed by | Person in charge |
| Screen specification | Dock View Screen | 18/11/2021 |  |  | Hoàng Hải Đăng |
|  | | Control | Operation | Function | |
| Enter Barcode Button | Click | Display Barcode Popup Screen | |
| Back Button | Click | Back to Home Screen | |
| Rent Bike Button | Click | Display Bike View Screen | |
| EBR Home Screen Button | Click | Display Home Screen | |
| Area for Displaying Number of Spots and Bikes | Initial | Displaying Number of Spots and Bikes | |
| Area for Displaying Bike in the Dock | Initial | Displaying Bike in the Dock | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Screen name | Dock View Screen | | | |
| Item name | Number of digits (bytes) | Type | Field attribute | Remarks |
| Number of Spot | 3 | Numeral | Black | Right justified |
| Number of Standard Bike | 3 | Numeral | Black | Right justified |
| Number of Twin Bike | 3 | Numeral | Black | Right justified |
| Number of E-Bike | 3 | Numeral | Black | Right justified |
| Bike Barcode | 10 | String | Black | Center justified |
| Battery Left | 4 | String | Green | Center justified |
| Bike Type | 6 | Numeral | Black | Center justified |

#### View Bike Screen

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1. EBR Software | | Date of creation | Approved by | Reviewed by | Person in charge |
| Screen specification | View Bike Screen | 18/11/2021 |  |  | Hoàng Hải Đăng |
| Graphical user interface  Description automatically generated | | Control | Operation | Function | |
| “x” Button | Click | Display Home Screen | |
| “Return Bike” Button | Click | Display Bike Screen | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Screen name | Invoice Screen | | | |
| Item name | Number of digits (bytes) | Type | Field attribute | Remarks |
| Bike code | 20 | String | Black | Left justified |
| License Plate | 8 | String | Black | Left justified |
| Location | 30 | String | Black | Left justified |
| Battery | 4 | String | Black | Left justified |
| Deposit | 20 | String | Black | Left justified |
| Charge | 30 | String | Black | Left justified |
| Pedals | 1 | Numeric | Black | Left justified |
| Saddle | 1 | Numeric | Black | Left justified |
| Rear Seat | 1 | Numeric | Black | Left justified |

#### Payment Screen

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| EBR Software | | Date of creation | Approved by | Reviewed by | Person in charge |
| Screen specification | Payment Screen | 11/18/2021 |  |  | Nguyễn Văn Chiến |
| Graphical user interface, website  Description automatically generated | | Control | Operation | Function | |
| Area for display Payment Information | Initial | Display the Card Holder Name, Card Number, Expiration Date, Security Number, Transaction Contents | |
| “Continue” Button | Click | Display Invoice Screen | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Screen name | Payment Screen | | | |
| Item name | Number of digits (bytes) | Type | Field attribute | Remarks |
| Card Number | 20 | String | Black | Left justified |
| Card Holder Name | 20 | String | Black | Left justified |
| Expiration Date | 5 | String | Black | Left justified |
| Security Code | 4 | Numeric | Black | Left justified |
| Transaction Contents | 100 | String | Black | Left justified |

#### Renting Session Screen

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| EBR Software | | Date of creation | Approved by | Reviewed by | Person in charge |
| Screen specification | Session Screen | 11/18/2021 |  |  | Nguyễn Văn Chiến |
|  | | Control | Operation | Function | |
| “Close” Button | Click | Display Home Screen | |
| “Return Bike” Button | Click | Display Bike Screen | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Screen name | Invoice Screen | | | |
| Item name | Number of digits (bytes) | Type | Field attribute | Remarks |
| Bike code | 20 | String | Black | Left justified |
| License Plate | 8 | String | Black | Left justified |
| Start time | 30 | String | Black | Left justified |
| Battery | 4 | String | Black | Left justified |
| Total Fees | 30 | String | Black | Left justified |
| Charge | 30 | String | Black | Left justified |
| Barcode | 30 | String | Black | Left justified |
| Session Time | 8 | String | Black | Left justified |

#### Invoice Screen

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| EBR Software | | Date of creation | Approved by | Reviewed by | Person in charge |
| Screen specification | Invoice Screen | 11/18/2021 |  |  | Nguyễn Văn Chiến |
|  | | Control | Operation | Function | |
| “Confirm Payment” Button | Click | Display Transaction Result Screen | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Screen name | Invoice Screen | | | |
| Item name | Number of digits (bytes) | Type | Field attribute | Remarks |
| Card Number | 20 | String | Black | Left justified |
| Card Holder Name | 20 | String | Black | Left justified |
| Expiration Date | 5 | String | Black | Left justified |
| Security Code | 4 | Numeric | Black | Left justified |
| Bike Type | 30 | String | Black | Left justified |
| Amount | 30 | String | Black | Left justified |
| Charge | 30 | String | Black | Left justified |
| Deposit | 30 | String | Black | Left justified |
| Barcode | 30 | String | Black | Left justified |
| License plate | 8 | String | Black | Left justified |
| Transaction Contents | 100 | String | Black | Left justified |

#### Barcode Screen

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **EBR Software** | | **Date of creation** | **Approved by** | **Reviewed by** | **Person in charge** |
| **Screen specification** | Barcode Screen | 17/11/2021 |  |  | Nguyễn Mạnh Cường |
|  | | **Control** | **Operation** | **Function** | |
| Area for entering Barcode | Initial | Receive barcode for renting bike | |
| “Enter” Button | Click | Display Renting Bike Screen | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Screen name | Barcode Screen | | | |
| Item name | Number of digits (bytes) | Type | Field attribute | Remarks |
| Barcode | 8 | String | Black | Left justified |

#### Select Dock Screen

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **EBR Software** | | **Date of creation** | **Approved by** | **Reviewed by** | **Person in charge** |
| **Screen specification** | Dock selection | 18/11/2021 |  |  | Nguyễn Đào Duy Kiên |
|  | | **Control** | **Operation** | **Function** | |
| “Return Bike” Button | Click | Display Payment Screen | |
| Area for entering search | Initial | Receive Text for Searching | |
| “Search” Button | Click | Searching Dock | |
| “Item Dock” | Click | Display item dock in left dock information | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Screen name | Home Screen | | | |
| Item name | Number of digits (bytes) | Type | Field attribute | Remarks |
| Search field | 50 | String | Black | Left justified |
| Dock’s name | 50 | String | Green | Left justified |
| Dock’s address | 50 | String | Black | Left justified |
| Available | 10 | String | Black | Left justified |

#### Transaction Result Screen

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **EBR Software** | | **Date of creation** | **Approved by** | **Reviewed by** | **Person in charge** |
| **Screen specification** | Transaction result | 18/11/2021 |  |  | Nguyễn Đào Duy Kiên |
| Không có mô tả. | | **Control** | **Operation** | **Function** | |
| Close Button | Click | Close popup | |
| Area for display transaction Information | Initial | Display transaction result and transaction info | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Screen name | Transaction result | | | |
| Item name | Number of digits (bytes) | Type | Field attribute | Remarks |
| Transaction details | 50 | number | Black | Left justified |
| Card number | 50 | number | Black | Left justified |
| Amout | 50 | number | Black | Left justified |
| Payment method | 50 | String | Black | Left justified |

#### Error Message Screen

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **EBR Software** | | **Date of creation** | **Approved by** | **Reviewed by** | **Person in charge** |
| **Screen specification** | Error Message Screen | 17/11/2021 |  |  | Nguyễn Mạnh Cường |
|  | | **Control** | **Operation** | **Function** | |
| Area for displaying error message | Initial | Display error message | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Screen name | Error Message Screen | | | |
| Item name | Number of digits (bytes) | Type | Field attribute | Remarks |
| Error Message | 100 | String | Black | Center |

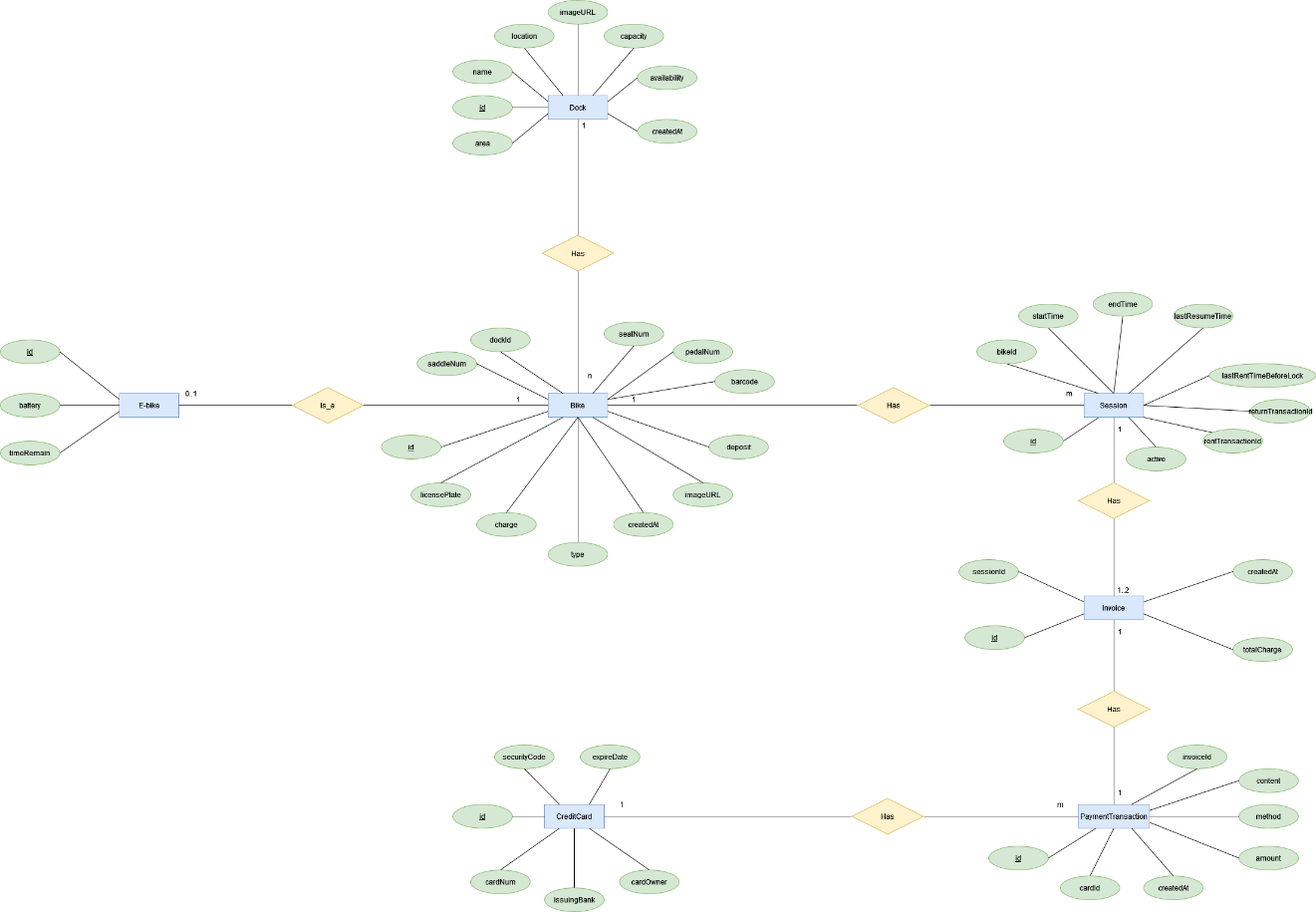
#### Rent Bike Screen

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| EBR Software | | Date of creation | Approved by | Reviewed by | Person in charge |
| Screen specification | Rent Bike Screen | 11/18/2021 |  |  | Nguyễn Văn Chiến |
|  | | Control | Operation | Function | |
| Enter “Rent Now” Button | Click | Display Payment Screen | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Screen name | Rent Bike Screen | | | |
| Item name | Number of digits (bytes) | Type | Field attribute | Remarks |
| Bike Code | 8 | String | Black | Left justified |
| License Plate | 8 | String | Black | Left justified |
| Location | 50 | String | Black | Left justified |
| Deposit | 25 | String | Black | Left justified |
| Charge | 15 | String | Black | Left justified |

## Data Modeling

### Conceptual Data Modeling

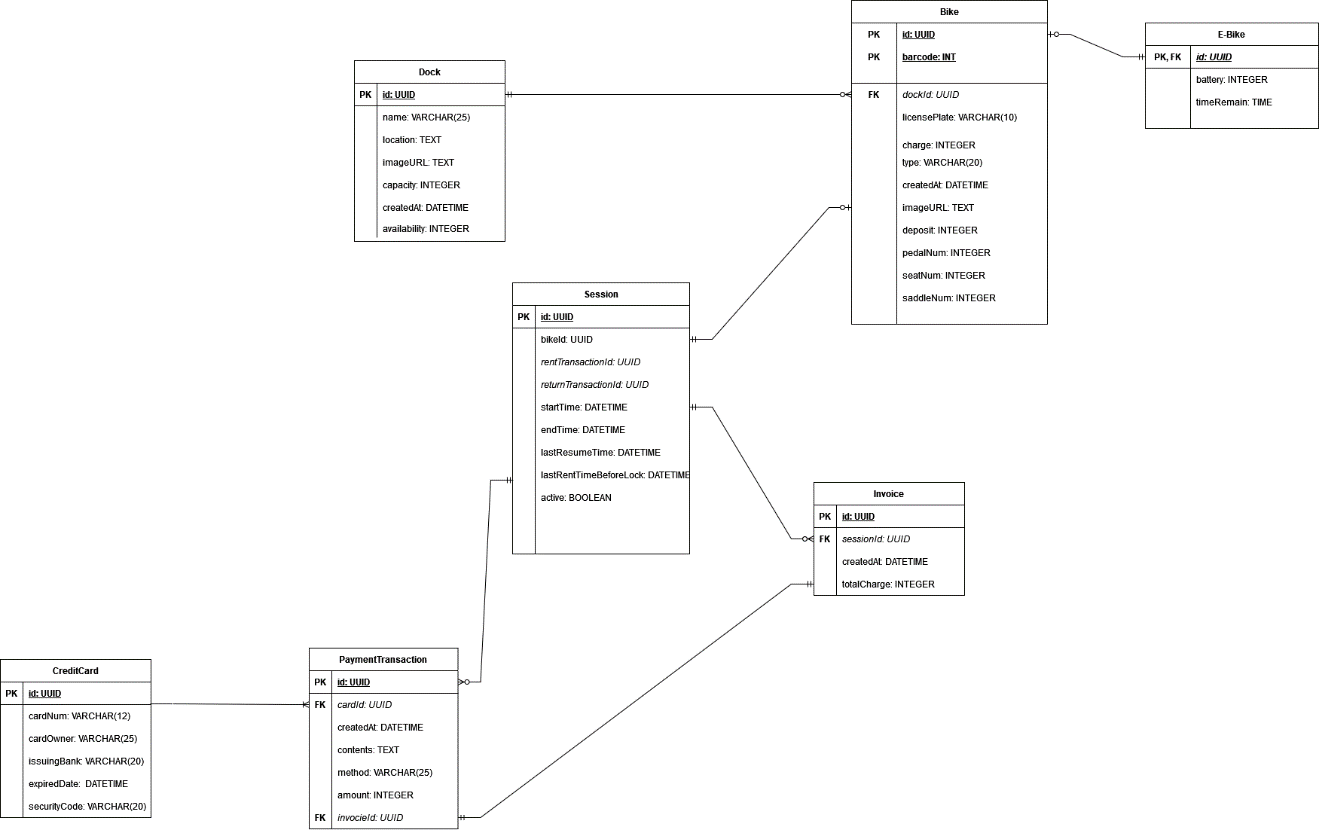


### Database Design

#### Database Management Systems

*Database Management System: PostgreSQL*

#### Logical Data Model



#### Physical Data Model

**Dock**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| # | PK | FK | Column Name | Data Type | Mandatory | Description |
| 1 | X |  | id | UUID | Yes | Dock ID |
| 2 |  |  | name | VARCHAR(25) | Yes | Name of the dock |
| 3 |  |  | location | TEXT | Yes | Address of the dock |
| 4 |  |  | imageURL | TEXT | Yes | URL of image of dock |
| 5 |  |  | capacity | INTEGER | Yes | Number of slots in the dock |
| 6 |  |  | createdAt | DATETIME | Yes | Created Time |
| 7 |  |  | availability | INTEGER | Yes | Number of available bikes in dock |

**Bike**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| # | PK | FK | Column Name | Data Type | Mandatory | Description |
| 1 | X |  | id | UUID | Yes | Bike ID |
| 2 | X |  | barcode | INT | Yes | Barcode for renting |
| 3 |  | X | dockId | UUID | Yes | Dock ID |
| 4 |  |  | licensePlate | VARCHAR(10) | Yes | License Plate |
| 5 |  |  | charge | INTEGER | Yes | Charge to rent bike per hour |
| 6 |  |  | type | VARCHAR(20) | Yes | Bike type |
| 7 |  |  | createdAt | DATETIME | Yes | Created time |
| 8 |  |  | imageURL | TEXT | Yes | URL of image of bike |
| 9 |  |  | deposit | INTEGER | Yes | Deposit of renting |
| 10 |  |  | seatNum | INTEGER | Yes | Number of seats |
| 11 |  |  | pedalNum | INTEGER | Yes | Number of pedals |
| 12 |  |  | saddleNum | INTEGER | Yes | Number of saddles |

**E\_Bike**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| # | PK | FK | Column Name | Data Type | Mandatory | Description |
| 1 | X | X | id | UUID | Yes | Bike ID |
| 2 |  |  | battery | INTEGER | Yes | Battery left percentage |
| 3 |  |  | timeRemain | TIME | Yes | Time remaining |

**Session**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| # | PK | FK | Column Name | Data Type | Mandatory | Description |
| 1 | X |  | id | UUID | Yes | Session ID |
| 2 |  | X | bikeId | UUID | Yes | Bike Id |
| 3 |  | X | rentTransactionId | UUID | Yes | Rent Transaction ID |
| 4 |  | X | returnTransactionId | UUID | Yes | Return Transaction ID |
| 5 |  |  | start\_time | DATETIME | Yes | Start time of session |
| 6 |  |  | endTime | DATETIME | No | End time of session |
| 7 |  |  | active | BOOLEAN | Yes | Check session finished |
| 8 |  |  | lastResumeTime | DATETIME | No | Last Resume Time |
| 9 |  |  | lastRentTimeBeforeLock | DATETIME | No | Last rent time before locking |

**Invoice**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| # | PK | FK | Column Name | Data Type | Mandatory | Description |
| 1 | X |  | id | UUID | Yes | Invoice ID |
| 2 |  | X | sessionId | UUID | Yes | Session ID |
| 3 |  |  | createdAt | DATETIME | Yes | Created time |
|  |  |  | type | VARCHAR(20) | Yes | Type of Invoice |
| 5 |  |  | totalCharge | INTEGER | Yes | Total charge money |

**PaymentTransaction**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| # | PK | FK | Column Name | Data Type | Mandatory | Description |
| 1 | X |  | id | UUID | Yes | Payment Transaction ID |
| 2 |  | X | cardId | UUID | Yes | Card ID |
| 3 |  |  | createAt | DATETIME | Yes | Created transaction time |
| 4 |  |  | contents | TEXT | No | Transaction contents |
| 5 |  |  | method | VARCHAR(25) | Yes | Method of payment |
| 6 |  |  | amount | INTEGER | Yes | Transaction amount |
| 7 |  | X | invoiceId | UUID | Yes | Invoice ID |

**CreditCard**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| # | PK | FK | Column Name | Data Type | Mandatory | Description |
| 1 | X |  | id | UUID | Yes | Card ID |
| 2 |  |  | cardNum | VARCHAR(12) | Yes | Card number |
| 3 |  |  | cardOwner | VARCHAR(25) | Yes | Card holder name |
| 4 |  |  | issuingBank | VARCHAR(20) | Yes | Issuing Bank |
| 5 |  |  | expiredDate | DATETIME | Yes | Expired date |
| 6 |  |  | securityCode | VARCHAR(20) | Yes | Security code |

**SQL Script**

SET statement\_timeout = 0;

SET lock\_timeout = 0;

SET idle\_in\_transaction\_session\_timeout = 0;

SET client\_encoding = 'UTF8';

SET standard\_conforming\_strings = on;

SELECT pg\_catalog.set\_config('search\_path', '', false);

SET check\_function\_bodies = false;

SET xmloption = content;

SET client\_min\_messages = warning;

SET row\_security = off;

CREATE EXTENSION IF NOT EXISTS adminpack WITH SCHEMA pg\_catalog;

COMMENT ON EXTENSION adminpack IS 'administrative functions for PostgreSQL';

CREATE EXTENSION IF NOT EXISTS "uuid-ossp" WITH SCHEMA public;

COMMENT ON EXTENSION "uuid-ossp" IS 'generate universally unique identifiers (UUIDs)';

SET default\_tablespace = '';

SET default\_table\_access\_method = heap;

DROP TABLE IF EXISTS public.invoice;

DROP TABLE IF EXISTS public.session;

DROP TABLE IF EXISTS public.payment\_transaction;

DROP TABLE IF EXISTS public.card;

DROP TABLE IF EXISTS public.e\_bike;

DROP TABLE IF EXISTS public.bike;

DROP TABLE IF EXISTS public.dock;

CREATE TABLE public.bike

(

id uuid DEFAULT public.uuid\_generate\_v4() NOT NULL,

type integer DEFAULT 1 NOT NULL,

barcode integer NOT NULL,

saddle\_num integer DEFAULT 1 NOT NULL,

pedal\_num integer DEFAULT 1 NOT NULL,

rear\_seat\_num integer DEFAULT 1 NOT NULL,

value integer NOT NULL,

rental\_fees integer NOT NULL,

image\_url text DEFAULT ''::text,

created\_at timestamp with time zone DEFAULT CURRENT\_TIMESTAMP,

description text,

dock\_id uuid

);

ALTER TABLE public.bike

OWNER TO postgres;

CREATE TABLE public.card

(

id uuid DEFAULT public.uuid\_generate\_v4() NOT NULL,

card\_num text NOT NULL UNIQUE,

card\_owner text NOT NULL,

security\_code text NOT NULL,

exp\_date text NOT NULL,

created\_at timestamp with time zone DEFAULT CURRENT\_TIMESTAMP

);

ALTER TABLE public.card

OWNER TO postgres;

CREATE TABLE public.dock

(

id uuid DEFAULT public.uuid\_generate\_v4() NOT NULL,

name text NOT NULL,

location text NOT NULL,

taken\_slot integer NOT NULL DEFAULT 0,

capacity integer NOT NULL,

image\_url text DEFAULT ''::text,

created\_at timestamp with time zone DEFAULT CURRENT\_TIMESTAMP

);

ALTER TABLE public.dock

OWNER TO postgres;

CREATE TABLE public.e\_bike

(

bike\_id uuid NOT NULL,

battery numeric(3, 1) NOT NULL,

time\_remain integer NOT NULL,

created\_at timestamp with time zone DEFAULT CURRENT\_TIMESTAMP

);

ALTER TABLE public.e\_bike

OWNER TO postgres;

CREATE TABLE public.invoice

(

id uuid DEFAULT public.uuid\_generate\_v4() NOT NULL,

session\_id uuid NOT NULL,

total\_charge numeric,

created\_at timestamp with time zone DEFAULT CURRENT\_TIMESTAMP

);

ALTER TABLE public.invoice

OWNER TO postgres;

CREATE TABLE public.payment\_transaction

(

id uuid DEFAULT public.uuid\_generate\_v4() NOT NULL,

card\_id uuid NOT NULL,

type text NOT NULL,

amount numeric NOT NULL,

method text,

transaction\_id text NOT NULL,

created\_at timestamp with time zone DEFAULT CURRENT\_TIMESTAMP

);

ALTER TABLE public.payment\_transaction

OWNER TO postgres;

CREATE TABLE public.session

(

id uuid DEFAULT public.uuid\_generate\_v4() NOT NULL,

card\_id uuid NOT NULL,

bike\_id uuid NOT NULL,

rent\_transactionid uuid NOT NULL,

return\_transactionid uuid,

last\_rent\_time\_before\_lock INT DEFAULT 0,

active boolean DEFAULT true,

start\_time Text,

last\_resume\_time Text,

end\_time Text,

created\_at timestamp with time zone DEFAULT CURRENT\_TIMESTAMP

);

ALTER TABLE public.session

OWNER TO postgres;

CREATE OR REPLACE FUNCTION public.bike\_dock\_change()

RETURNS TRIGGER

LANGUAGE PLPGSQL

AS

$$

BEGIN

IF OLD.dock\_id IS NOT NULL AND NEW.dock\_id IS NULL THEN

UPDATE public.dock

SET taken\_slot = taken\_slot - 1

WHERE id = OLD.dock\_id;

END IF;

IF OLD.dock\_id IS NULL AND NEW.dock\_id IS NOT NULL THEN

UPDATE public.dock

SET taken\_slot = taken\_slot + 1

WHERE id = NEW.dock\_id;

END IF;

RETURN NEW;

END;

$$;

CREATE TRIGGER dock\_changes

AFTER UPDATE

ON public.bike

FOR EACH ROW

EXECUTE PROCEDURE public.bike\_dock\_change();

CREATE OR REPLACE FUNCTION public.insert\_new\_bike()

RETURNS TRIGGER

LANGUAGE PLPGSQL

AS

$$

BEGIN

IF NEW.dock\_id IS NOT NULL THEN

UPDATE public.dock

SET taken\_slot = taken\_slot + 1

WHERE id = NEW.dock\_id;

END IF;

RETURN NEW;

END;

$$;

CREATE TRIGGER bike\_insert

AFTER INSERT

ON public.bike

FOR EACH ROW

EXECUTE PROCEDURE public.insert\_new\_bike();

-- DOCK

INSERT INTO public.dock (id, name, location, capacity, image\_url, created\_at)

VALUES ('4a4e10eb-1de2-411f-a35b-fc5be0cc7393', 'ECO Bike Park 1', 'Location 1', 200, '',

'2020-12-03 21:52:19.458054+07');

INSERT INTO public.dock (id, name, location, capacity, image\_url, created\_at)

VALUES ('012b8b33-32a3-4c15-ba0f-ba3204e119b0', 'ECO Bike Park 2', 'Location 2', 200, '',

'2020-12-03 21:52:30.624056+07');

INSERT INTO public.dock (id, name, location, capacity, image\_url, created\_at)

VALUES ('31b2f5d7-dcbc-428a-8be5-3f49fa5a3573', 'ECO Bike Park 3', 'Location 3', 100, '',

'2020-12-03 21:52:36.709013+07');

-- BIKE

INSERT INTO public.bike (id, type, barcode, saddle\_num, pedal\_num, rear\_seat\_num, value, rental\_fees, image\_url,

created\_at, description, dock\_id)

VALUES ('078a3525-4146-49ba-9c4c-9caf1678ddcd', 1, 467295, 1, 1, 1, 1000000, 25000,

'assets\images\bikes\StandardBike.png', '2020-12-03 21:53:34.242179+07', NULL,

'012b8b33-32a3-4c15-ba0f-ba3204e119b0');

INSERT INTO public.bike (id, type, barcode, saddle\_num, pedal\_num, rear\_seat\_num, value, rental\_fees, image\_url,

created\_at, description, dock\_id)

VALUES ('1be8a4e1-5eb6-48f6-b9ff-5f67a6c45f85', 4, 452593, 2, 2, 1, 8000000, 50000,

'assets\images\bikes\TwinElectricalBike.png', '2020-12-03 21:58:17.775008+07', NULL,

'31b2f5d7-dcbc-428a-8be5-3f49fa5a3573');

INSERT INTO public.bike (id, type, barcode, saddle\_num, pedal\_num, rear\_seat\_num, value, rental\_fees, image\_url,

created\_at, description, dock\_id)

VALUES ('31e1ac5c-a755-4946-aa1b-9cf8989ee87f', 1, 784318, 1, 1, 1, 1000000, 25000,

'assets\images\bikes\StandardBike.png', '2020-12-03 21:54:50.621477+07', NULL,

'4a4e10eb-1de2-411f-a35b-fc5be0cc7393');

INSERT INTO public.bike (id, type, barcode, saddle\_num, pedal\_num, rear\_seat\_num, value, rental\_fees, image\_url,

created\_at, description, dock\_id)

VALUES ('524ccbd1-c1d5-47d9-9c28-827f88a9adcd', 3, 697316, 1, 1, 1, 6000000, 50000,

'assets\images\bikes\ElectricalBike.png', '2020-12-03 21:56:37.492976+07', NULL,

'012b8b33-32a3-4c15-ba0f-ba3204e119b0');

INSERT INTO public.bike (id, type, barcode, saddle\_num, pedal\_num, rear\_seat\_num, value, rental\_fees, image\_url,

created\_at, description, dock\_id)

VALUES ('78270ffe-be03-4e83-b466-b9c2e274a976', 4, 228636, 2, 2, 1, 8000000, 50000,

'assets\images\bikes\TwinElectricalBike.png', '2020-12-03 21:57:52.695281+07', NULL,

'012b8b33-32a3-4c15-ba0f-ba3204e119b0');

INSERT INTO public.bike (id, type, barcode, saddle\_num, pedal\_num, rear\_seat\_num, value, rental\_fees, image\_url,

created\_at, description, dock\_id)

VALUES ('9ef7ceb8-38c9-4d56-9fb6-1ddfd0364f46', 1, 216064, 1, 1, 1, 1000000, 25000,

'assets\images\bikes\StandardBike.png', '2020-12-03 21:54:09.924574+07', NULL,

'31b2f5d7-dcbc-428a-8be5-3f49fa5a3573');

INSERT INTO public.bike (id, type, barcode, saddle\_num, pedal\_num, rear\_seat\_num, value, rental\_fees, image\_url,

created\_at, description, dock\_id)

VALUES ('b3af2957-1a4d-4c64-8cef-7b0b393625ce', 2, 527841, 2, 2, 1, 4000000, 50000, 'assets\images\bikes\TwinBike.png',

'2020-12-03 21:55:52.16724+07', NULL, '31b2f5d7-dcbc-428a-8be5-3f49fa5a3573');

INSERT INTO public.bike (id, type, barcode, saddle\_num, pedal\_num, rear\_seat\_num, value, rental\_fees, image\_url,

created\_at, description, dock\_id)

VALUES ('cc2e033a-7505-4a66-a639-4d9cd35f9d6d', 2, 826086, 2, 2, 1, 4000000, 50000, 'assets\images\bikes\TwinBike.png',

'2020-12-03 21:55:25.58271+07', NULL, '4a4e10eb-1de2-411f-a35b-fc5be0cc7393');

INSERT INTO public.bike (id, type, barcode, saddle\_num, pedal\_num, rear\_seat\_num, value, rental\_fees, image\_url,

created\_at, description, dock\_id)

VALUES ('d06ba58c-4759-419d-85c3-07b3f738f500', 1, 533503, 1, 1, 1, 1000000, 25000,

'assets\images\bikes\StandardBike.png', '2020-12-03 21:53:49.299133+07', NULL,

'012b8b33-32a3-4c15-ba0f-ba3204e119b0');

INSERT INTO public.bike (id, type, barcode, saddle\_num, pedal\_num, rear\_seat\_num, value, rental\_fees, image\_url,

created\_at, description, dock\_id)

VALUES ('e50afbf2-e6f7-46c6-a9e3-82505901af21', 3, 188831, 1, 1, 1, 6000000, 50000,

'assets\images\bikes\ElectricalBike.png', '2020-12-03 21:56:55.635597+07', NULL,

'012b8b33-32a3-4c15-ba0f-ba3204e119b0');

-- E-BIKE

INSERT INTO public.e\_bike (bike\_id, battery, time\_remain, created\_at)

VALUES ('1be8a4e1-5eb6-48f6-b9ff-5f67a6c45f85', 99.9, 360, '2020-12-11 13:51:02.69043+07');

INSERT INTO public.e\_bike (bike\_id, battery, time\_remain, created\_at)

VALUES ('524ccbd1-c1d5-47d9-9c28-827f88a9adcd', 20.0, 72, '2020-12-11 13:52:01.710722+07');

INSERT INTO public.e\_bike (bike\_id, battery, time\_remain, created\_at)

VALUES ('78270ffe-be03-4e83-b466-b9c2e274a976', 40.0, 144, '2020-12-11 13:52:01.710722+07');

INSERT INTO public.e\_bike (bike\_id, battery, time\_remain, created\_at)

VALUES ('e50afbf2-e6f7-46c6-a9e3-82505901af21', 80.0, 288, '2020-12-11 13:52:01.710722+07');

ALTER TABLE ONLY public.bike

ADD CONSTRAINT bike\_pk PRIMARY KEY (id);

ALTER TABLE ONLY public.card

ADD CONSTRAINT card\_pk PRIMARY KEY (id);

ALTER TABLE ONLY public.dock

ADD CONSTRAINT dock\_pk PRIMARY KEY (id);

ALTER TABLE ONLY public.e\_bike

ADD CONSTRAINT e\_bike\_pk PRIMARY KEY (bike\_id);

ALTER TABLE ONLY public.invoice

ADD CONSTRAINT invoice\_pk PRIMARY KEY (id);

ALTER TABLE ONLY public.payment\_transaction

ADD CONSTRAINT payment\_transaction\_pk PRIMARY KEY (id);

ALTER TABLE ONLY public.payment\_transaction

ADD CONSTRAINT payment\_transaction\_fk FOREIGN KEY (card\_id) REFERENCES public.card (id) ON DELETE CASCADE;

ALTER TABLE ONLY public.session

ADD CONSTRAINT session\_pk PRIMARY KEY (id);

ALTER TABLE ONLY public.bike

ADD CONSTRAINT bike\_fk FOREIGN KEY (dock\_id) REFERENCES public.dock (id) ON DELETE CASCADE;

ALTER TABLE ONLY public.e\_bike

ADD CONSTRAINT e\_bike\_fk FOREIGN KEY (bike\_id) REFERENCES public.bike (id) ON DELETE CASCADE;

ALTER TABLE ONLY public.invoice

ADD CONSTRAINT invoice\_fk FOREIGN KEY (session\_id) REFERENCES public.session (id);

ALTER TABLE ONLY public.session

ADD CONSTRAINT session\_fk1 FOREIGN KEY (card\_id) REFERENCES public.card (id);

ALTER TABLE ONLY public.session

ADD CONSTRAINT session\_fk2 FOREIGN KEY (bike\_id) REFERENCES public.bike (id);

ALTER TABLE ONLY public.session

ADD CONSTRAINT session\_fk3 FOREIGN KEY (rent\_transactionid) REFERENCES public.payment\_transaction (id);

ALTER TABLE ONLY public.session

ADD CONSTRAINT session\_fk4 FOREIGN KEY (return\_transactionid) REFERENCES public.payment\_transaction (id);

update dock set image\_url = 'assets\images\docks\dockImg1.png' where name = 'ECO Bike Park 1';

update dock set image\_url = 'assets\images\docks\dockImg2.png' where name = 'ECO Bike Park 2';

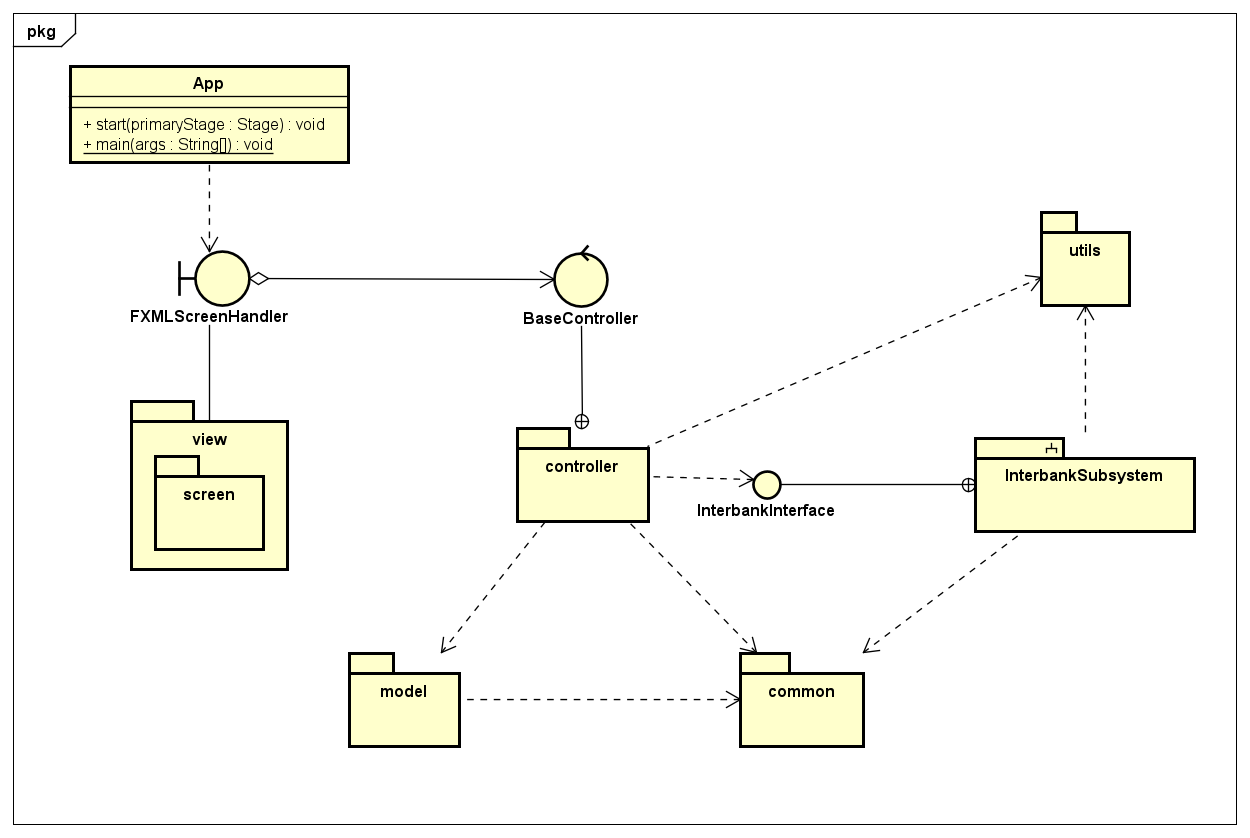
update dock set image\_url = 'assets\images\docks\dockImg3.png' where name = 'ECO Bike Park 3';

ALTER TABLE bike add column license\_plate varchar(255);

## Non-Database Management System Files

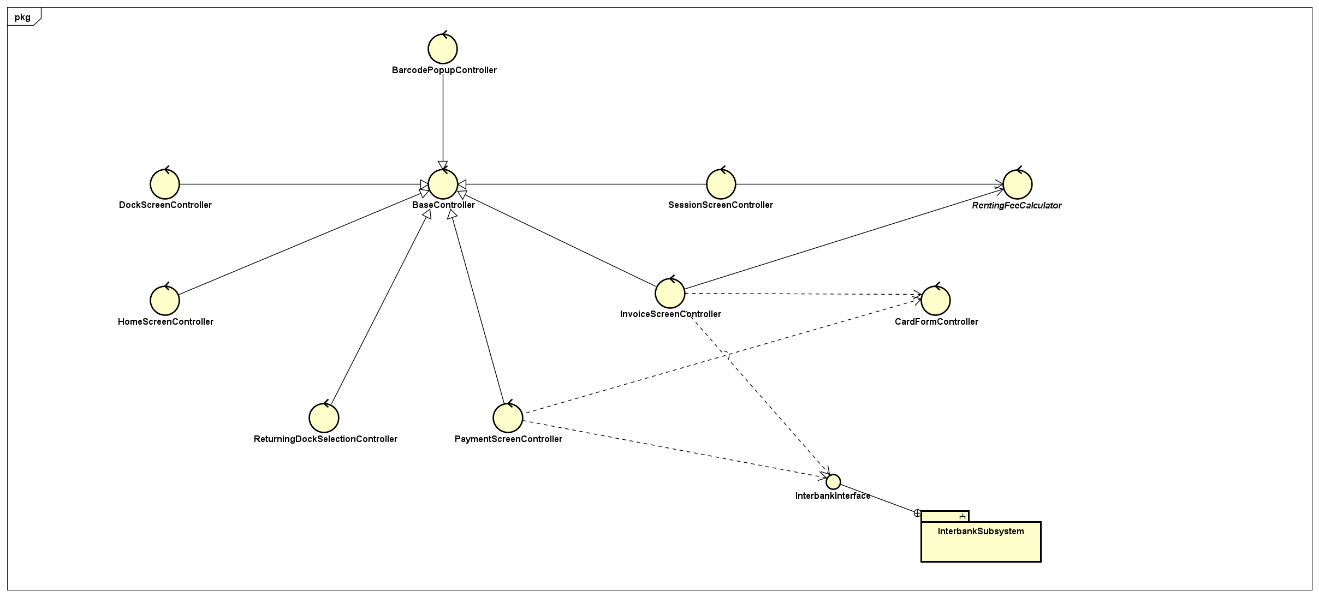
## Class Design

### General Class Diagram

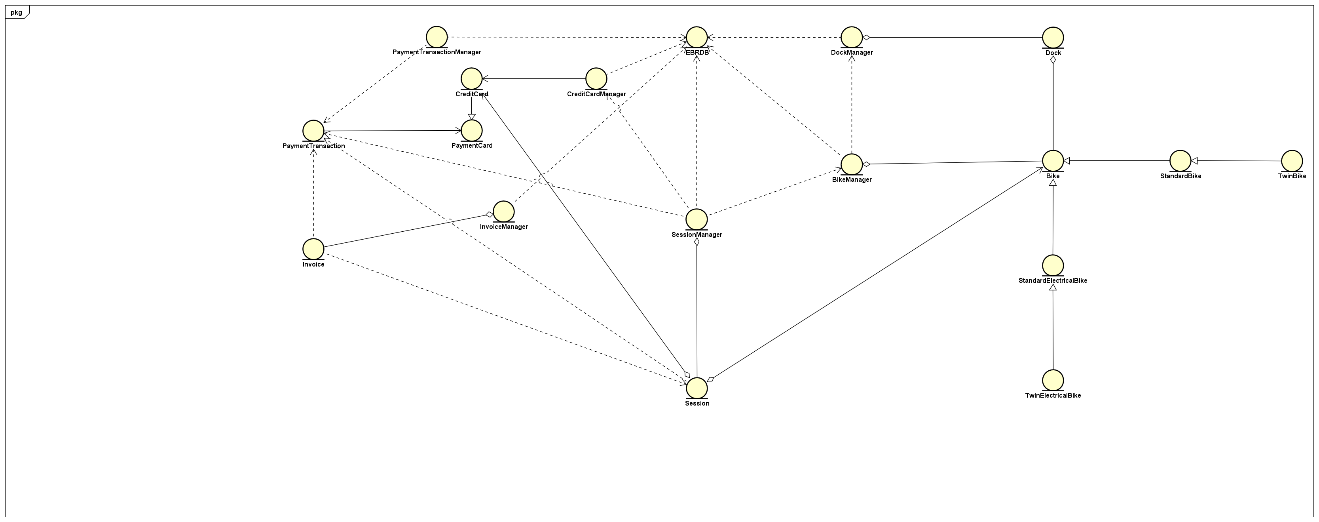


### Class Diagrams

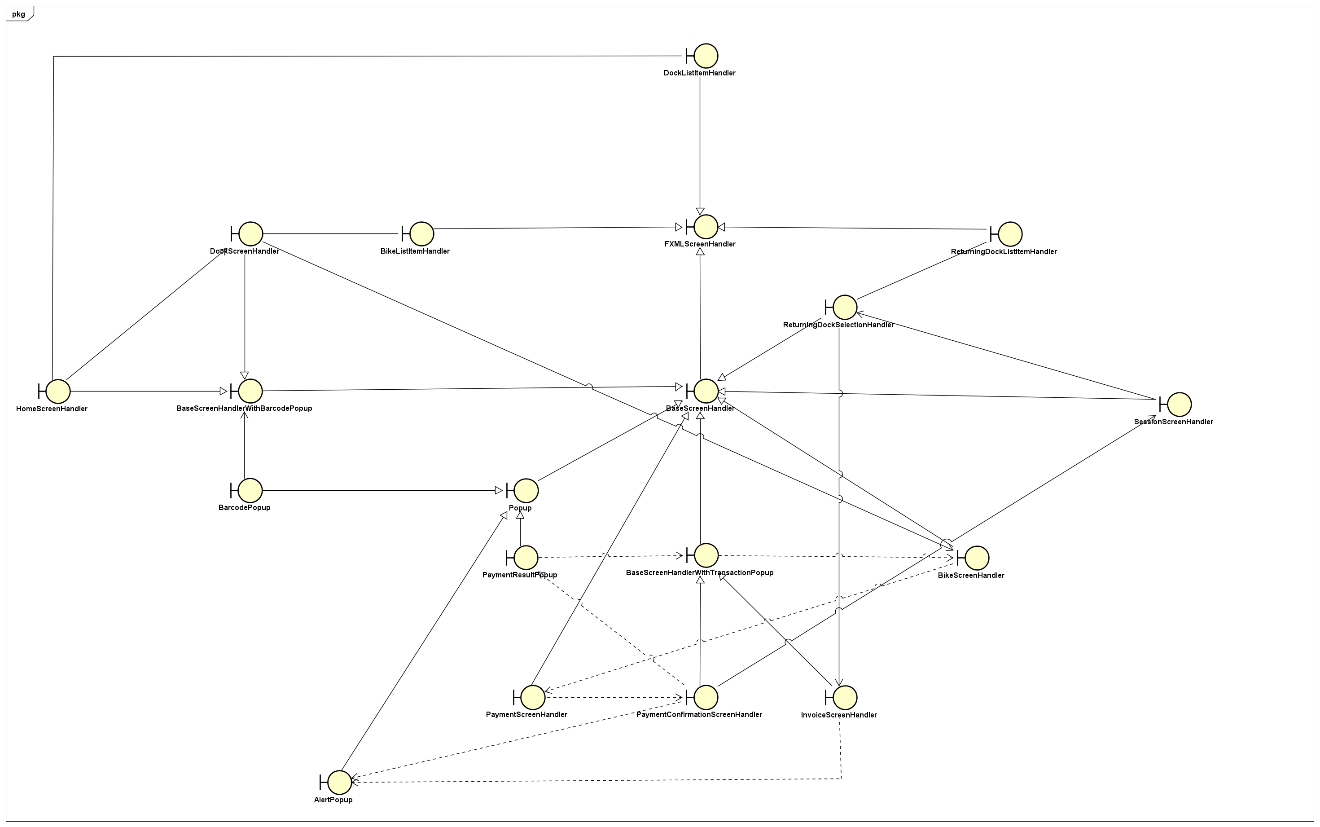
#### Class Diagram for Package “controller”



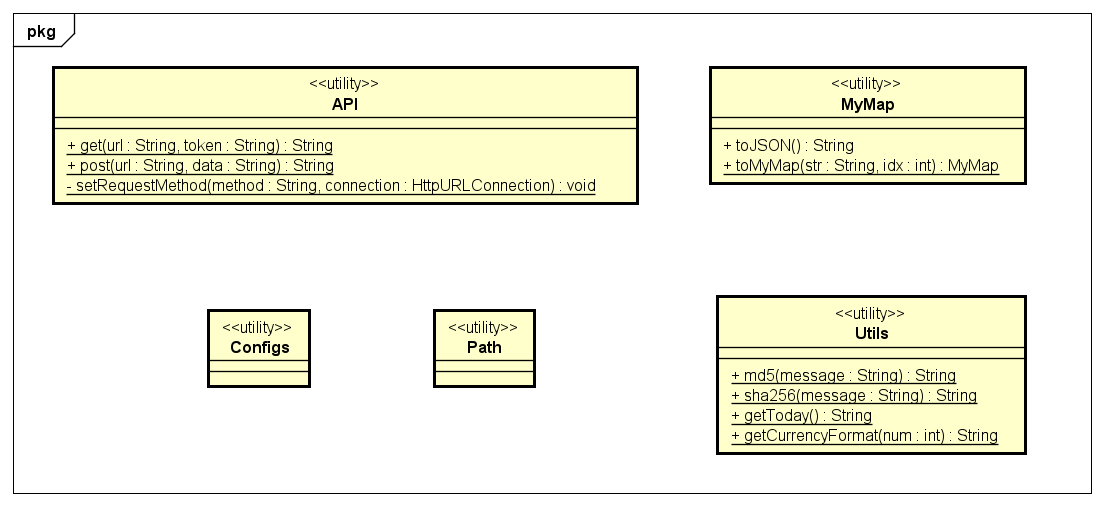
#### Class Diagram for Package “model”



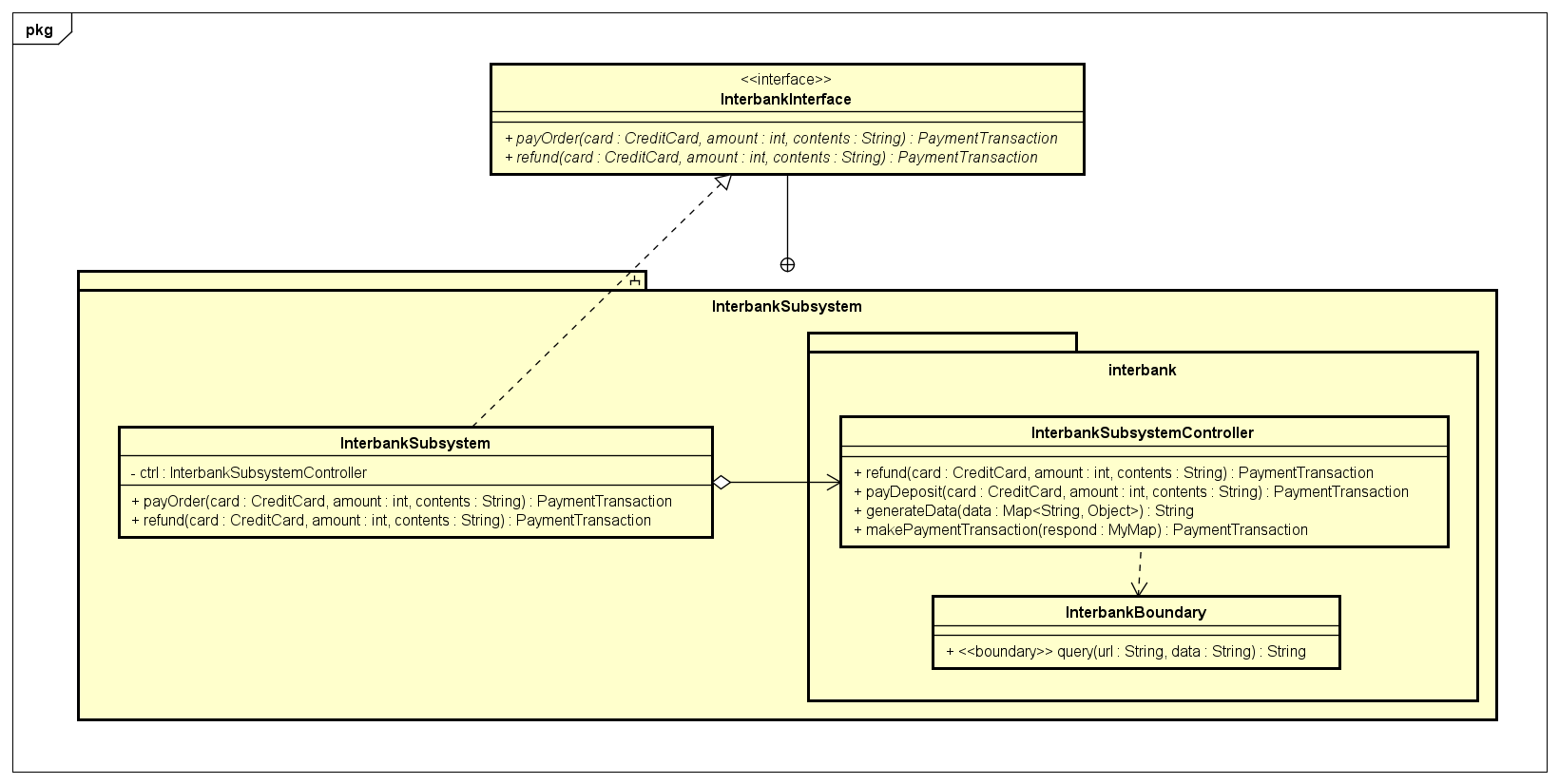
#### Class Diagram for Package “view”



#### Class Diagram for Package “utils”

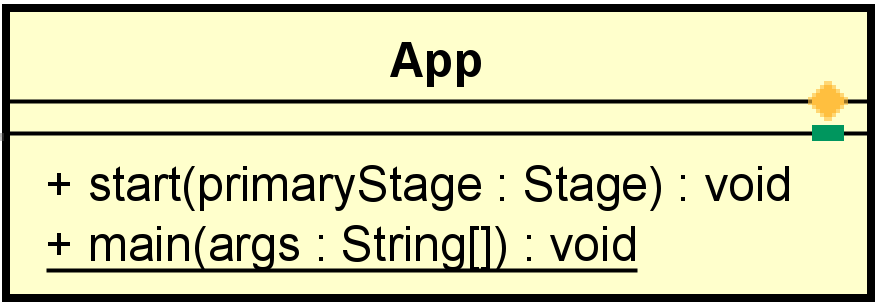


#### Class Diagram for Subsystem “interbank”



### Class Design

* **Class “App”**



**Attribute**

None

**Operation**

|  |  |  |  |
| --- | --- | --- | --- |
| # | Name | Return type | Description |
| 1 | start | void | javaFx framwork will call this |
| 2 | main | void | execution entry |

**Parameter:**

* start()
  + stage: javaFx main stage

**Exception:**

None

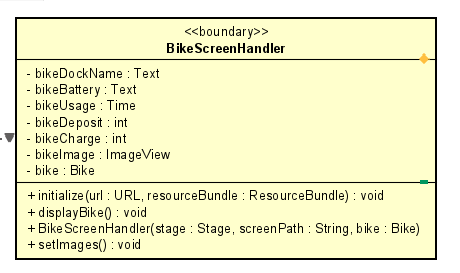
**Method**

None

**State**

None

* **Class “BikeScreenHandler”**



**Attribute**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Name | Data type | Default value | Description |
| 1 | bikeDockName | Text | NULL | Dock name fxml |
| 2 | bikeBattery | int | NULL | Bike battery fxml |
| 3 | bikeUsage | Time | NULL | Bike usage fxml |
| 4 | bikeDeposit | int | NULL | Bike deposit fxml |
| 5 | bikeCharge | int | NULL | Bike charge fxml |
| 6 | bikeImage | ImageView | NULL | Bike image |
| 7 | bike | Bike | NULL | bike |

**Operation**

|  |  |  |  |
| --- | --- | --- | --- |
| # | Name | Return type | Description |
| 1 | initialize | void | initialize |
| 2 | displayBike | void | Display bike info screen |
| 3 | setImages | void | Set bike image |

**Parameter:**

* initialize
  + url - URL
  + resourceBundle - ResourceBundle
* BikeScreenHandle
  + stage - Stage

**Exception:**

None

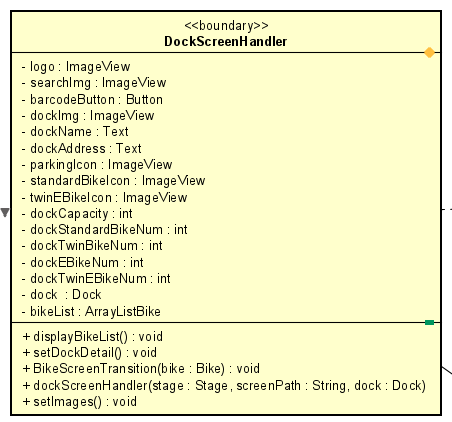
**Method**

None

**State**

None

* **Class “DockScreenHandler”**



**Attribute**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Name | Data type | Default value | Description |
| 1 | logo | ImageView | NULL | Logo image fxml |
| 2 | searchImg | imageView | NULL | Search image fxml |
| 3 | barcodeButton | Button | NULL | BarcodeButton fxml |
| 4 | dockName | Text | NULL | Dock name |
| 5 | dockAddress | Text | NULL | Dock address |
| 6 | dockImg | ImageView | NULL | Dock image fxml |
| 7 | parkingIcon | ImageView | NULL | Parking icon fxml |
| 8 | standardBikeIcon | ImageView | NULL | Standard bike icon fxml |
| 9 | twinEBikeIcon | ImageView | NULL | Twin bike icon fxml |
| 10 | dockStandardBikeNum | Int | NULL | Standard bike number |
| 11 | dockTwinBikeNum | Int | NULL | Twin bike number f |
| 12 | dockEBikeNum | Int | NULL | E-bike number |
| 13 | dockTwinEBikeNum | Int | NULL | Twin e-bikenumber |
| 14 | dock | Dock | NULL | Dock |
| 15 | bikeList | ArrayList<Bike> | NULL | Bike list in dock |

**Operation**

|  |  |  |  |
| --- | --- | --- | --- |
| # | Name | Return type | Description |
| 1 | displayBikeList | void | Display bike list |
| 2 | setDockDetail | void | Set dock detail |
| 3 | BikeScreenTransition | void | Transition to bike screen |
| 4 | setImages | void | Set dock image |

**Parameter:**

* BikeScreenTransition
  + bike
* dockScreenHandle
  + stage – Stage
  + screenPath – String
  + dock - Dock

**Exception:**

None

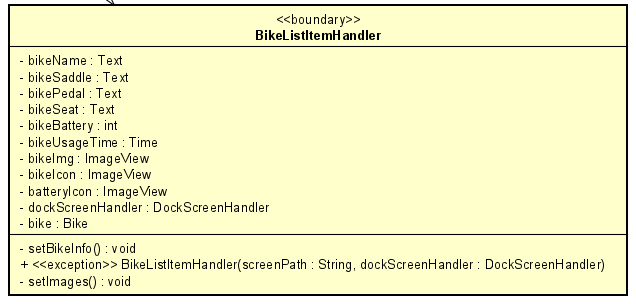
**Method**

None

**State**

None

* **Class “BikeListItemHandler”**



**Attribute**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Name | Data type | Default value | Description |
| 1 | bikeImg | ImageView | NULL | Bike image fxml |
| 2 | bikeIcon | imageView | NULL | Bike image fxml |
| 4 | bikeName | Text | NULL | Bike name |
| 5 | bikeSaddle | Text | NULL | Bike saddle |
| 6 | bikePedal | Text | NULL | Bike pedal |
| 7 | bikeSeat | Text | NULL | Bike seat |
| 8 | dockScreenHandler | DockScreenHandler | NULL | object |
| 9 | bike | Bike | NULL | object |
| 10 | bikeBattery | Int | NULL | Bike Battery |
| 11 | BikeUsageTime | Time | NULL | Bike Usage Time |

**Operation**

|  |  |  |  |
| --- | --- | --- | --- |
| # | Name | Return type | Description |
| 1 | setBikeInfo | void | Set bike info |
| 2 | setImages | void | Set bike image in list |

**Parameter:**

none

**Exception:**

IOException

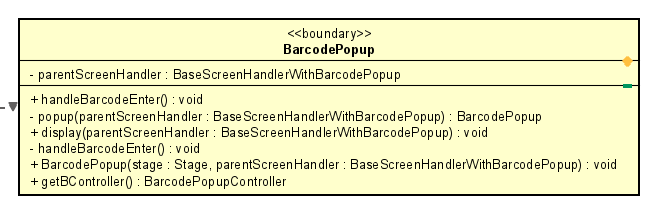
**Method**

None

**State**

None

* **Class “BarcodePopup”**



**Attribute**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Name | Data type | Default value | Description |
| 1 | parentScreenHandler | BaseScreenHandlerWithBarcodePopup | NULL | Parent screen |

**Operation**

|  |  |  |  |
| --- | --- | --- | --- |
| # | Name | Return type | Description |
| 1 | handleBarcodeEnter | void | Handle action “enter barcode” |
| 2 | popup | BaseScreenHandlerWithBarcodePopup | BarcodePopup |
| 3 | display | void | Display popup |
| 4 | handleBarcodeEnter | void | Button enter handler |
| 5 | barcodePopup | void | Popup barcode |
| 6 | getBController | BarcodePopupController | Get barcode |

**Parameter:**

* popup
  + parentScreenHandler
* display
  + parentScreenHandler
* barcodePopup
  + parentScreenHandler
  + stage

**Exception:**

None

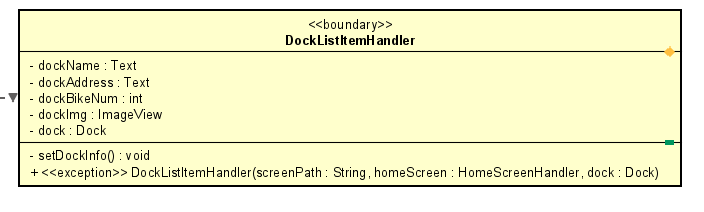
**Method**

None

**State**

None

* **Class “DockListItemHandler”**



**Attribute**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Name | Data type | Default value | Description |
| 1 | dockName | Text | NULL | Dock name |
| 2 | dockAddress | Text | NULL | Dock address |
| 3 | dockBikeNum | int | NULL | Number of bikes in dock |
| 4 | dockImg | ImageView | NULL | Dock image |
| 5 | dock | Dock | NULL | dock |

**Operation**

|  |  |  |  |
| --- | --- | --- | --- |
| # | Name | Return type | Description |
| 1 | setDockInfo | void | Set dock info |

**Parameter:**

None

**Exception:**

IOException

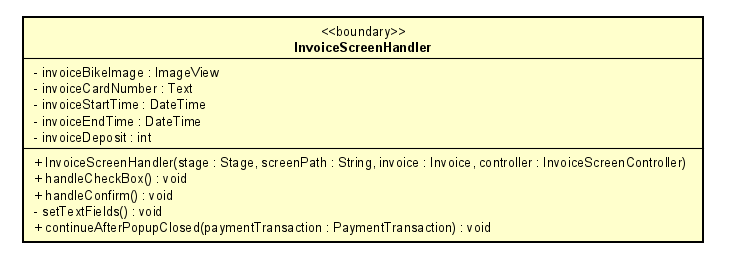
**Method**

None

**State**

None

* **Class “InvoiceScreenHandler”**



**Attribute**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Name | Data type | Default value | Description |
| 2 | invoiceBikeImage | ImageView | NULL | Bike image |
| 3 | invoiceCardNumber | Text | NULL | Invoice card number |
| 4 | invoiceStartTime | DateTime | NULL | Invoice start time |
| 5 | invoiceEndTime | DateTime | NULL | Invoice end time |
| 6 | invoiceDeposit | int | NULL | Invoice deposit |

**Operation**

|  |  |  |  |
| --- | --- | --- | --- |
| # | Name | Return type | Description |
| 1 | handleCheckBox | void | Handle check box |
| 2 | handleConfirmButton | void | Handle confirm button |
| 3 | setTextFields | void | Set text fields |
| 4 | continueAfterPopupClosed | void | Continue program after popup close |

**Parameter:**

* continueAfterPopupClosed
  + paymentTransaction

**Exception:**

None

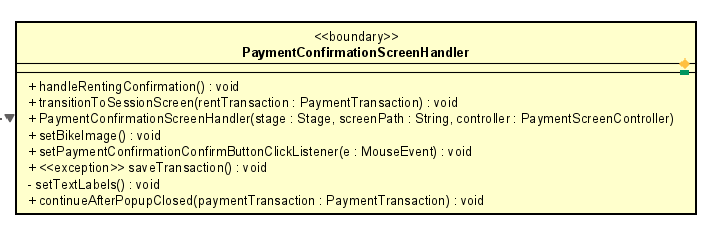
**Method**

None

**State**

None

* **Class “PaymentConfirmationScreenHandler”**



**Attribute**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Name | Data type | Default value | Description |

**Operation**

|  |  |  |  |
| --- | --- | --- | --- |
| # | Name | Return type | Description |
| 1 | handleRentingConfirmation | void | Handle confirmation action |
| 2 | transitionToSessionScreen | void | Transition to session screen |
| 3 | paymentConfirmationScreenHandler | void | Handle confirm payment |
| 4 | setBikeImage | void | Set bike image |
| 5 | setPaymentConfirmationConfirmButtonClickListener | void | Payment Confirmation Confirm Button Click Listener |
| 6 | saveTransaction | void | Save transaction |
| 7 | setTextLabels | void | Set text |
| 8 | continueAfterPopupClosed | void | Continue program after popup close |

**Parameter:**

* transitionToSessionScreen
  + rentTransaction
* paymentConfirmationScreenHandler
  + stage
  + screenPath
  + controller
* setPaymentConfirmationConfirmButtonClickListener
  + e
* continueAfterPopupClosed
  + paymentTransaction

**Exception:**

IOException

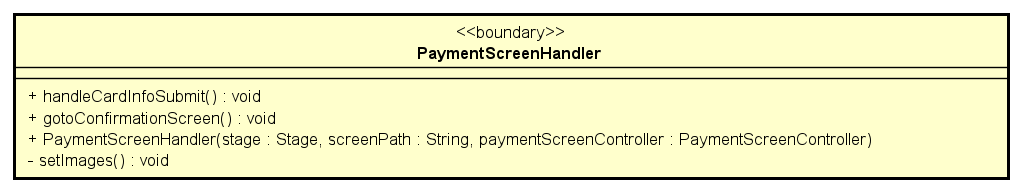
**Method**

None

**State**

None

* **Class “PaymentScreenHandler”**



**Attribute**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Name | Data type | Default value | Description |

**Operation**

|  |  |  |  |
| --- | --- | --- | --- |
| # | Name | Return type | Description |
| 1 | handleCardInfoSubmit | void | Handle action submit card |
| 2 | gotoConfirmationScreen | void | Transition to confirmation screen |
| 3 | setImages | vodi | Set payment image |

**Parameter:**

None

**Exception:**

None

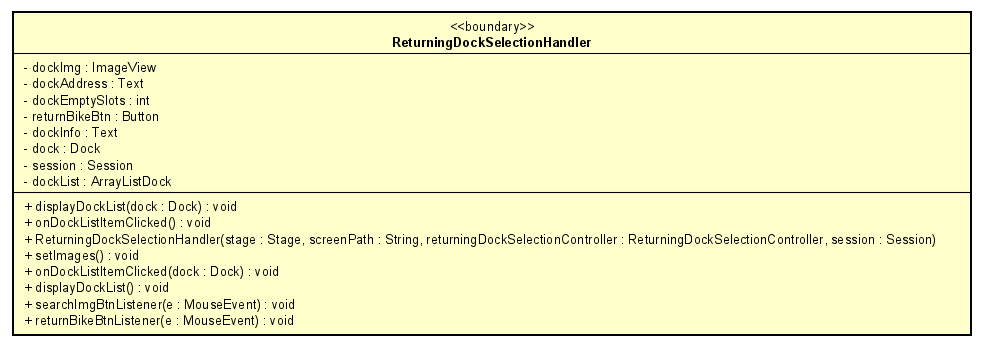
**Method**

None

**State**

None

* **Class “ReturningDockListSelectionHandler”**



**Attribute**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Name | Data type | Default value | Description |
| 1 | returnBikeBtn | Button | NULL | Button to return bike |
| 2 | dockEmptySlots | int | NULL | Dock parking slot |
| 3 | dockImg | ImageView | NULL | Dock image fxml |
| 4 | dockInfo | Text | NULL | Dock information |
| 5 | dockAdress | Text | NULL | Dock address |
| 6 | dock | Dock | NULL | dock |
| 7 | session | Session | NULL | Session |
| 8 | dockList | ArrayListDock | NULL | Dock list |

**Operation**

|  |  |  |  |
| --- | --- | --- | --- |
| # | Name | Return type | Description |
| 1 | setDockInfo | void | Set dock info |
| 2 | onDockListItemClicked | void | Return dock clicked listener |
| 3 | setImages | void | Set return dock image |
| 4 | displayDockList | void | Display return dock list |
| 5 | searchImgBtnListener | void | Search image button listener |
| 6 | returnBikeBtnListener | void | Return bike button listener |

**Parameter:**

None

**Exception:**

None

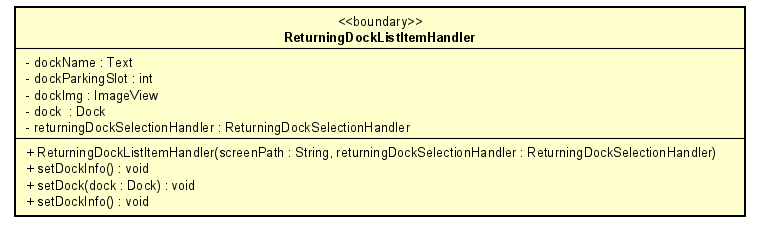
**Method**

None

**State**

None

* **Class “ReturningDockListItemHandler”**



**Attribute**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Name | Data type | Default value | Description |
| 1 | dockName | Text | NULL | Dock name fxml |
| 2 | dockParkingSlot | int | NULL | Dock address fxml |
| 3 | dockImg | ImageView | NULL | Dock image fxml |
| 4 | dock | Dock | NULL | dock |
| 5 | returningDockSelectionHandler | ReturningDockSelectionHandler | NULL | Returning Dock Selection Handler |

**Operation**

|  |  |  |  |
| --- | --- | --- | --- |
| # | Name | Return type | Description |
| 1 | setDockInfo | void | Display dock information |
| 2 | setDock | void | Display dock |

**Parameter:**

* setDock
  + dock

**Exception:**

None

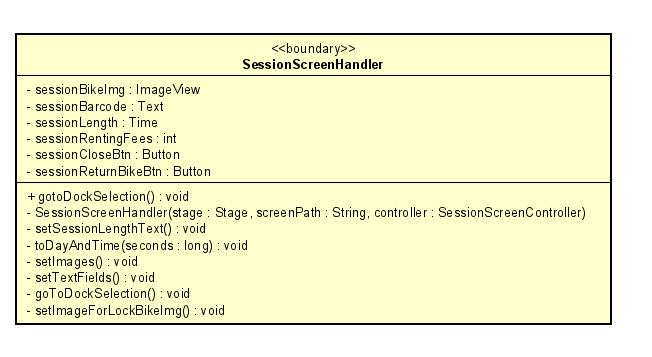
**Method**

None

**State**

None

* **Class “SessionScreenHandler”**



**Attribute**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Name | Data type | Default value | Description |
| 1 | sessionBikeImg | ImageView | NULL | Bike image fxml |
| 2 | sessionBarcode | Text | NULL | Session barcode |
| 3 | sessionLength | Time | NULL | Session length |
| 4 | sessionRentingFees | Int | NULL | Session renting fees |
| 5 | sessionCloseBtn | Button | NULL | Button |
| 6 | sessionReturnBikeBtn | Button | NULL | Button |

**Operation**

|  |  |  |  |
| --- | --- | --- | --- |
| # | Name | Return type | Description |
| 1 | gotoDockSeletion | void | Transition to dock selection |
| 2 | setSessionLengthText | void | Set session’s length text |
| 3 | toDayAndTime | void | Day and time conversion |
| 4 | setImages | void | Set image for session |
| 5 | setTextFields | void | Set text fields for session |
| 6 | goToDockSelection | void | Back to the dock selection |
| 7 | setImageForLockBikeImg | void | Set image for bike |

**Parameter:**

* ToDayAndTime
  + seconds

**Exception:**

None

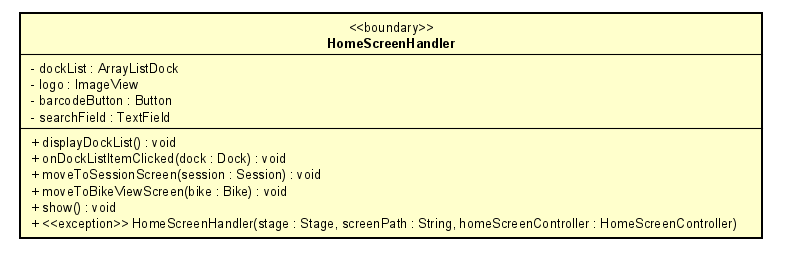
**Method**

None

**State**

None

* **Class “HomeScreenHandler”**



**Attribute**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Name | Data type | Default value | Description |
| 1 | vboxDockList | VBox | NULL | Vbox fxml |
| 2 | dockList | ArrayListDock | NULL | Dock list |
| 3 | logo | ImageView | NULL | Logo fxml |
| 4 | barcodeButton | Button | NULL | Barcode button fxml |
| 5 | searchField | TextField | NULL | Search field fxml |

**Operation**

|  |  |  |  |
| --- | --- | --- | --- |
| # | Name | Return type | Description |
| 1 | displayDockList | void | Display dock list |
| 2 | onDockListItemClicked | void | Dock list item clicked listener |
| 3 | moveToSessionScreen | void | Move to session screen |
| 4 | moveToBikeViewScreen | void | Move to bike view screen |
| 5 | show | void | Show screen |

**Parameter:**

* onDockListItemClicked
  + dock
* moveToSessionScreen
  + session
* moveToBikeViewScreen
  + bike

**Exception:**

IOException

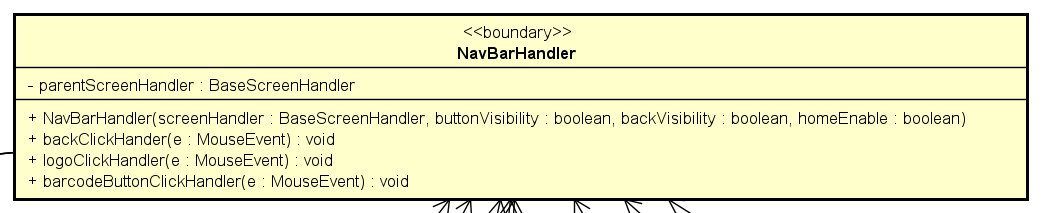
**Method**

None

**State**

None

* **Class “NavBarHandler”**



**Attribute**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Name | Data type | Default value | Description |
| 1 | parentScreenHandler | BaseScreenHandler | NULL | handler |

**Operation**

|  |  |  |  |
| --- | --- | --- | --- |
| # | Name | Return type | Description |
| 1 | backClickHandler | void | go to previous screen |
| 2 | logoClickHandler | void | go to home screen |
| 3 | barcodeButtonClickHandler | void | bring up barcode popup |

**Parameter:**

* backClickHandler
  + e – MouseEvent
* logoClickHandler
  + e - MouseEvent

**Exception:**

None

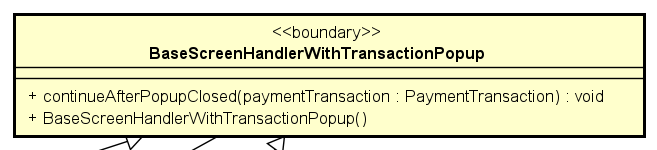
**Method**

None

**State**

None

* **Class “BaseScreenHandlerWithTransactionPopup”**



**Attribute**

None

**Operation**

|  |  |  |  |
| --- | --- | --- | --- |
| # | Name | Return type | Description |
| 1 | continueAfterPopupClosed | void | show then closed popup of payment result |

**Parameter:**

* continueAfterPopupClosed
  + paymentTransaction – transaction to be shown result

**Exception:**

None

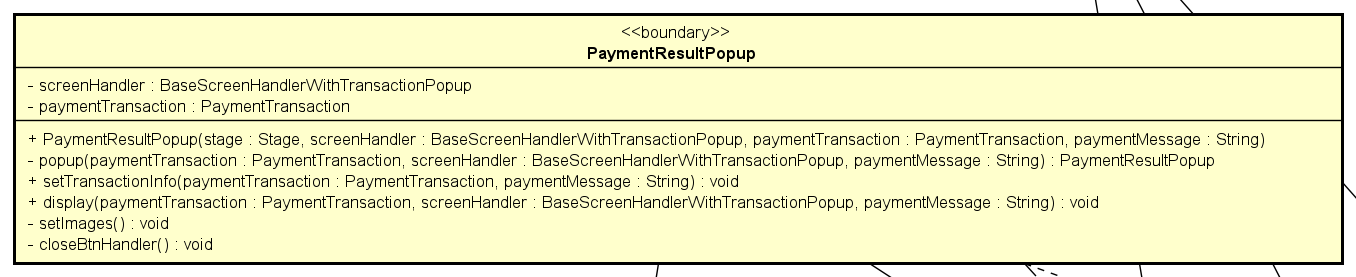
**Method**

None

**State**

None

* **Class “PaymentResultPopup”**



**Attribute**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Name | Data type | Default value | Description |
| 1 | screenHandler | BaseScreenHandlerWithTransactionPopup | new | handler |
| 2 | paymentTransaction | PaymentTransaction | NULL | transaction to be shown |

**Operation**

|  |  |  |  |
| --- | --- | --- | --- |
| # | Name | Return type | Description |
| 1 | setTransactionInfo | void | Display result |
| 2 | display | void | Render result |
| 3 | setImage | void | Set image for screen |
| 4 | closeBtnHandler | void | Close the popup and move to next screen |
| 5 | popup | PaymentResultPopup | Create then return popup |

**Parameter:**

* popup
  + screenHandler
  + paymentTransaction
  + paymentMessage

**Exception:**

None

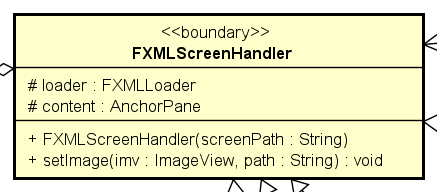
**Method**

None

**State**

None

* **Class “FXMLScreenHandler”**



**Attribute**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Name | Data type | Default value | Description |
| 1 | loader | FXMLLoader | new |  |
| 2 | content | AnchorPane | NULL | main app anchor |

**Operation**

|  |  |  |  |
| --- | --- | --- | --- |
| # | Name | Return type | Description |
| 1 | setImage | void | set image for screen |

**Parameter:**

* setImage
  + imv – image holder to be set
  + path – path to image

**Exception:**

None

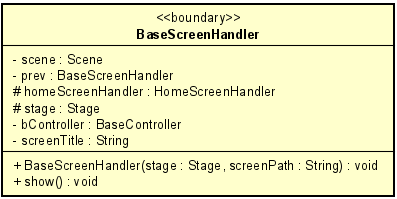
**Method**

None

**State**

None

* **Class “BaseScreenHandler”**



**Attribute**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Name | Data type | Default value | Description |
| 1 | scene | Scene | NULL | Scene to render |
| 2 | prev | BaseScreenHandler | NULL | Previous screen |
| 3 | homeScreenHandler | HomeScreenHandler | NULL | Home screen |
| 4 | stage | Stage | NULL | JavaFx stage |
| 5 | bController | BaseController | NULL | Controller for this screen |
| 6 | screenTitle | String | NULL | Title of this screen |

**Operation**

|  |  |  |  |
| --- | --- | --- | --- |
| # | Name | Return type | Description |
| 1 | show | void | Render the screen |

**Parameter:**

**Exception:**

None

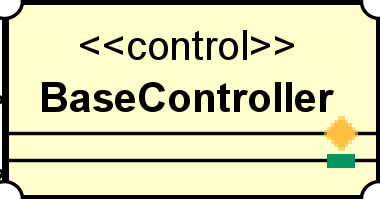
**Method**

None

**State**

None

* **Class “BaseController”**



**Attribute**

None

**Operation**

None

**Parameter:**

None

**Exception:**

None

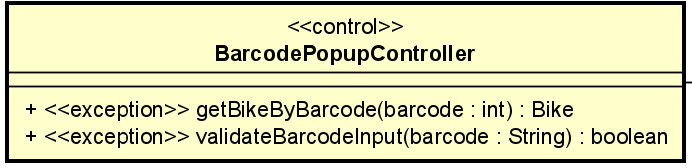
**Method**

None

**State**

None

* **Class “BarcodePopupController”**



**Attribute**

None

**Operation**

|  |  |  |  |
| --- | --- | --- | --- |
| # | Name | Return type | Desciption |
| 1 | getBikeByBarcode | Bike | Get bike by using barcode |
| 2 | ValidateBarcodeInput | boolean | Checks if barcode is entered or not |

**Parameter:**

* Parameter of getBikeByBarcode()
* barcode– int
* Parameter of ValidateBarcodeInput()
* barcode - string

**Exception:**

* BarcodeNotFoundException
* NullBarcodeException
* InvalidBarcodeException

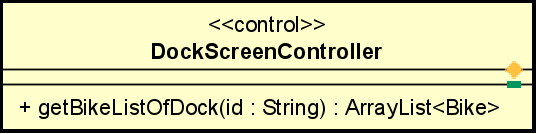
**Method**

None

**State**

None

* **Class “DockScreenController”**



**Attribute**

None

**Operation**

|  |  |  |  |
| --- | --- | --- | --- |
| # | Name | Return type | Desciption |
| 1 | getBikeListOfDock | ArrayList<Bike> | Get bike list by using dock id |

**Parameter:**

* Parameter of getBikeListOfDock()
  + Id – dock id

**Exception:**

None

**Method**

None

**State**

None

* **Class “HomeScreenController”**

Diagram, table

Description automatically generated with medium confidence

**Attribute**

None

**Operation**

|  |  |  |  |
| --- | --- | --- | --- |
| # | Name | Return type | Desciption |
| 1 | getDockList | ArrayList<Dock> | Get dock list |
| 2 | getDockListByKeyword | ArrayList<Dock> | Get dock list by keyword |

**Parameter:**

* Parameter of getDockListByKeyword()
* Text - keyword

**Exception:**

None

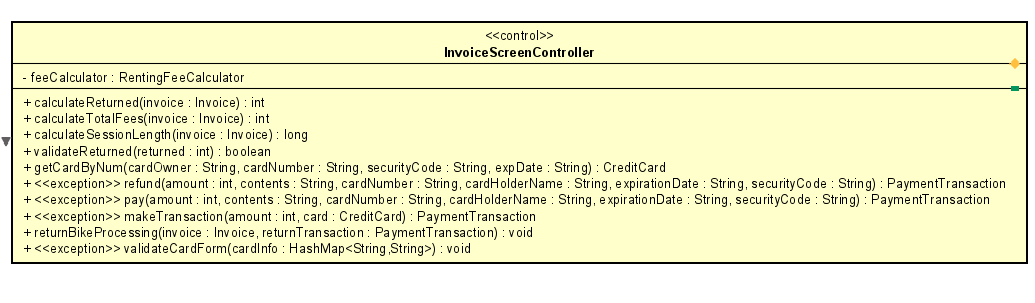
**Method**

None

**State**

None

* **Class “InvoiceScreenController”**



**Attribute**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Name | Data type | Default value | Description |
| 1 | feeCalculator | RentingFeeCalculator | new RentingFeeBySecondsCalculator | calculate fee |

**Operation**

|  |  |  |  |
| --- | --- | --- | --- |
| # | Name | Return type | Desciption |
| 1 | calculateReturned | Int | Calculate returned after deducting renting fees |
| 2 | calculateTotalFees | Int | Calculate total fees |
| 3 | calculateSessionLength | long | Session length |
| 4 | refund | PaymentTransaction | Create return transaction |
| 5 | validateReturned | boolean | Check returned validity |
| 6 | getCardByNum | CreditCard | Get credit card info |
| 7 | refund | PaymentTransaction | Setup Return transaction |
| 8 | pay | PaymentTransaction | Setup transaction |
| 9 | makeTransaction | PaymentTransaction | Proceed transaction |
| 10 | returnBikeProcessing | void | Process return bike |
| 11 | validateCardForm | void | Check card form validity |

**Parameter:**

* Parameter of calculateReturned()
* Invoice
* Parameter of calculateTotalFees()
* Invoice
* Parameter of calculateSessionLength()
* Invoice
* Amount Parameter of getCardByNum()
  + cardOwner
  + cardNumber
  + securityCode
  + expDate
* Parameter of refund()
* amount
* contents
* cardNumber
* cardHolderName
* expirationDate
* securityCode
* Parameter of pay()
* amount
* contents
* cardNumber
* cardHolderName
* expirationDate
* securityCode
* ExpirationDate Parameter of makeTransaction()
* amount
* card
* Parameter of validateReturned()
* returned
* Parameter of returnBikeProcessing()
* Invoice
* returnTransaction
* Parameter of validateCardForm()
* cardInfo

**Exception:**

PaymentException

NullCardNumberException

NullCardOwnerException

NullExpDateException

NullSecurityCodeException

CardExpiredException

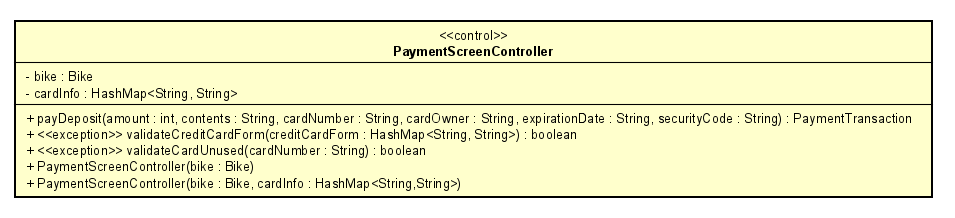
**Method**

None

**State**

None

* **Class “PaymentScreenController”**



**Attribute**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Name | Data type | Default value | Description |
| 1 | bike | Bike | NULL | bike |
| 2 | cardInfo | HashMap<String, String> | NULL | Card info |

**Operation**

|  |  |  |  |
| --- | --- | --- | --- |
| # | Name | Return type | Desciption |
| 1 | payDeposit | PaymentTransaction | Set rent transaction |
| 2 | validateCreditCardForm | boolean | Check creditcard form |
| 3 | validateCardUnused | boolean | Check creditcard used or not |

**Parameter:**

* Parameter of payDeposit()
* amount
* contents
* cardNumber
* cardOwner
* expirationDate
* securityCode
* Parameter of validateCreditCardForm()
  + creditCardForm
* Parameter of validateCardUnused()
  + cardNumber

**Exception:**

None

**Method**

None

**State**

None

* **Class “ReturningDockSelectionController”**

Text

Description automatically generated

**Attribute**

None

**Operation**

|  |  |  |  |
| --- | --- | --- | --- |
| # | Name | Return type | Desciption |
| 1 | getDockList | ArrayList<Dock> | Get dock list |
| 2 | getDockListByKeyWord | ArrayList<Dock> | Get dock list by keyword |
| 3 | createInvoice | Invoice | Create new invoice |
| 4 | returnBikeToDock | void | Return bike to the chosen dock |

Parameter:

* Parameter of getDockListByKeyWord()
  + keyword
* Parameter of createInvoice()
  + id
* Parameter of returnBikeToDock()
  + bike
  + dock

**Exception:**

None

**Method**

None

**State**

None

* **Class “SessionScreenController”**

Text

Description automatically generated

**Attribute**

None

**Operation**

|  |  |  |  |
| --- | --- | --- | --- |
| # | Name | Return type | Desciption |
| 1 | returnBike | boolean | Check if bike is available |
| 2 | calculateCurrenRentingFees | int | Calculate current renting fees |
| 3 | calculateSessionLength | long | Calculate session length |

**Parameter:**

* Parameter of returnBikeToDock()
  + session
  + dock
* Parameter of returnBikeToDock()
  + session
* Parameter of returnBikeToDock()
  + session

**Exception:**

None

**Method**

None

**State**

None

* **Class “CardFormController”**

Text

Description automatically generated with medium confidence

**Attribute**

None

**Operation**

|  |  |  |  |
| --- | --- | --- | --- |
| # | Name | Return type | Desciption |
| 1 | validateCreditCardForm | boolean | Check if card is valid |
| 2 | validateCardNumber | boolean | Validate card number |
| 3 | validateCardOwner | boolean | Validate card owner |
| 4 | validateSecurityCode | boolean | Validate security code |
| 5 | validateExpDate | boolean | Validate exp date |

**Parameter:**

* Parameter of validateCreditCardForm()
  + creditCardForm
* Parameter of validateCardNumber()
  + CardNumber
* Parameter of validateCardOwner()
  + cardOwner
* Parameter of validateSecurityCode()
  + securityCode
* Parameter of validateExpDate()
  + expDate

**Exception:**

None

**Method**

None

**State**

None

* **Class “RentingFeeCaculator”**

Text

Description automatically generated

**Attribute**

None

**Operation**

|  |  |  |  |
| --- | --- | --- | --- |
| # | Name | Return type | Desciption |
| 1 | calculateCurrentRentingFees | int | Calculate current renting fees |
| 2 | calculateTotalFees | int | Calculate total fees |

**Parameter:**

* Parameter of calculateCurrentRentingFees()
  + session
* Parameter of calculateTotalFees()
  + invoice

**Exception:**

None

**Method**

None

**State**

None

* **Class “RentingFeeForEBikeCalculator”**

Text

Description automatically generated

**Attribute**

None

**Operation**

|  |  |  |  |
| --- | --- | --- | --- |
| # | Name | Return type | Desciption |
| 1 | calculateCurrentRentingFees | int | Calculate current renting fees |
| 2 | calculateTotalFees | int | Calculate total fees |

**Parameter:**

* Parameter of calculateCurrentRentingFees()
  + session
* Parameter of calculateTotalFees()
  + invoice

**Exception:**

None

**Method**

None

**State**

None

* **Class “RentingFeeForTwinEBikeCalculator”**

Text

Description automatically generated with medium confidence

**Attribute**

None

**Operation**

|  |  |  |  |
| --- | --- | --- | --- |
| # | Name | Return type | Desciption |
| 1 | calculateCurrentRentingFees | int | Calculate current renting fees |
| 2 | calculateTotalFees | int | Calculate total fees |

**Parameter:**

* Parameter of calculateCurrentRentingFees()
  + session
* Parameter of calculateTotalFees()
  + invoice

**Exception:**

None

**Method**

None

**State**

None

* **Class “RentingFeeForTwinBikeCalculator”**

A picture containing text

Description automatically generated

**Attribute**

None

**Operation**

|  |  |  |  |
| --- | --- | --- | --- |
| # | Name | Return type | Desciption |
| 1 | calculateCurrentRentingFees | int | Calculate current renting fees |
| 2 | calculateTotalFees | int | Calculate total fees |

**Parameter:**

* Parameter of calculateCurrentRentingFees()
  + session
* Parameter of calculateTotalFees()
  + invoice

**Exception:**

None

**Method**

None

**State**

None

* **Class “RentingFeeForStandardBikeCalculator”**

A picture containing text

Description automatically generated

**Attribute**

None

**Operation**

|  |  |  |  |
| --- | --- | --- | --- |
| # | Name | Return type | Desciption |
| 1 | calculateCurrentRentingFees | int | Calculate current renting fees |
| 2 | calculateTotalFees | int | Calculate total fees |

**Parameter:**

* Parameter of calculateCurrentRentingFees()
  + session
* Parameter of calculateTotalFees()
  + invoice

**Exception:**

None

**Method**

None

**State**

None

* **Class “RentingFeeBySecondsCalculator”**

A picture containing text

Description automatically generated

**Attribute**

None

**Operation**

|  |  |  |  |
| --- | --- | --- | --- |
| # | Name | Return type | Desciption |
| 1 | calculateCurrentRentingFees | int | Calculate current renting fees |
| 2 | calculateTotalFees | int | Calculate total fees |

**Parameter:**

* Parameter of calculateCurrentRentingFees()
  + session
* Parameter of calculateTotalFees()
  + invoice

**Exception:**

None

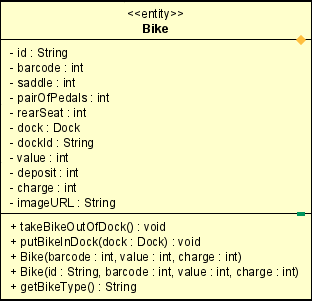
**Method**

None

**State**

None

* **Class “Bike”**



**Attribute**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Name | Data type | Default value | Description |
| 1 | id | String | NULL | Bike id |
| 2 | barcode | int | NULL | barcode |
| 3 | saddle | int | NULL | saddle |
| 4 | pairOfPedals | int | NULL | Pair of pedals |
| 5 | rearSeat | int | NULL | Number of rear seats |
| 6 | dock | Dock | NULL | dock |
| 7 | dockId | String | NULL | Dock id |
| 8 | value | int | NULL | value |
| 9 | doposit | int | NULL | deposit |
| 10 | charge | int | NULL | charge |
| 11 | imageURL | String | NULL | Image URL |

**Operation**

|  |  |  |  |
| --- | --- | --- | --- |
| # | Name | Return type | Desciption |
| 1 | takeBikeOutOfDock | void | Take bike out of dock |
| 2 | putBikeInDock | void | Put bike to a dock |
| 3 | getBikeType | String | Get bike type |

**Parameter:**

* putBikeInDock
  + dock - Dock

**Exception:**

None

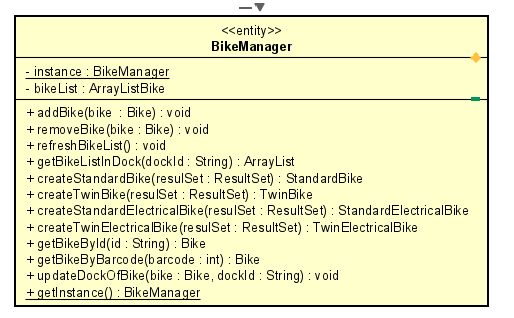
**Method**

None

**State**

None

* **Class “BikeManager”**



**Attribute**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Name | Data type | Default value | Description |
| 1 | instance | BikeManager | NULL | Bike instance |
| 2 | bikeList | ArrayList | NULL | Bike list |

**Operation**

|  |  |  |  |
| --- | --- | --- | --- |
| # | Name | Return type | Desciption |
| 1 | addBike | void | Add bike |
| 2 | removeBike | void | Remove bike |
| 3 | refreshBikeList | void | Refresh bike list |
| 4 | getBikeListInDock | ArrayList | Get bike list in dock |
| 5 | createStandardBike | StandardBike | Create standard bike |
| 6 | createTwinBike | TwinBike | Create twin bike |
| 7 | createStandardEBike | StandardEBike | Create standard e bike |
| 8 | createTwinEBike | TwinEBike | Create twin e bike |
| 9 | getBikeById | Bike | Get bike by id |
| 10 | getBikeByBarcode | Bike | Get bike by barcode |
| 11 | updateDockOfBike | void | Update dock |
| 12 | getInstance | BikeManager | Get bike instance |

**Parameter:**

* addBike
  + bike - Bike
* getBikeListInDock
  + dockId - String
* createStandardBike
  + resultSet - ResultSet
* createTwinBike
  + resultSet - ResultSet
* createTwinEBike
  + resultSet - ResultSet
* getBikeById
  + id – String
* getBikeByBarcode
  + barcode - int
* updateDockOfBike
  + dockId – String
  + bike - Bike

**Exception:**

None

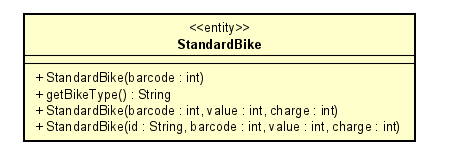
**Method**

None

**State**

None

* **Class “StandardBike”**



**Attribute**

None

**Operation**

|  |  |  |  |
| --- | --- | --- | --- |
| # | Name | Return type | Desciption |
| 1 | getBikeType | String | Get bike type |

**Parameter:**

None

**Exception:**

None

**Method**

None

**State**

None

* **Class “StandardElectricalBike”**

Table

Description automatically generated

**Attribute**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Name | Data type | Default value | Description |
| 1 | battery | float | 100.0 | battery remained |
| 2 | timeRemain | Time | NULL | usage time base on battery and bike |

**Operation**

|  |  |  |  |
| --- | --- | --- | --- |
| # | Name | Return type | Desciption |
| 1 | getBikeType | String | Get bike type |

**Parameter:**

None

**Exception:**

None

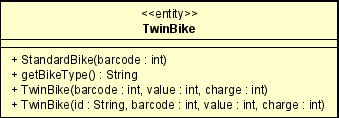
**Method**

None

**State**

None

* **Class “TwinBike”**



**Attribute**

None

**Operation**

|  |  |  |  |
| --- | --- | --- | --- |
| # | Name | Return type | Desciption |
| 1 | getBikeType | String | Get bike type |

**Parameter:**

None

**Exception:**

None

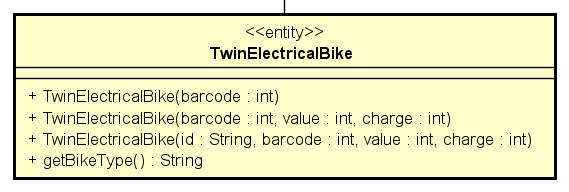
**Method**

None

**State**

None

* **Class “TwinElectricalBike”**



**Attribute**

None

**Operation**

|  |  |  |  |
| --- | --- | --- | --- |
| # | Name | Return type | Desciption |
| 1 | getBikeType | String | Get bike type |

**Parameter:**

None

**Exception:**

None

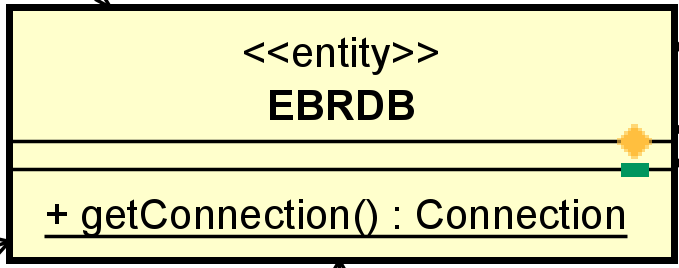
**Method**

None

**State**

None

* **Class “EBRDB”**



**Attribute**

None

**Operation**

|  |  |  |  |
| --- | --- | --- | --- |
| # | Name | Return type | Desciption |
| 1 | getConnection | Connect | get a connection to PostgreSQL database |

**Parameter:**

None

**Exception:**

None

**Method**

None

**State**

None

* **Class “Dock”**

Text

Description automatically generated

**Attribute**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Name | Data type | Default value | Description |
| 1 | id | String | NULL | Dock id |
| 2 | name | String | NULL | Name of dock |
| 3 | location | String | NULL | Dock location |
| 4 | capacity | int | NULL | Dock size |
| 5 | imageURL | String | NULL | Dock image |
| 6 | bikeList | ArrayList<Bike> | NULL | Bike list in dock |
| 7 | createAt | Time | NULL | Time Create Dock |
| 8 | Availalbity | Int | NULL | Number of availability bike |

**Operation**

|  |  |  |  |
| --- | --- | --- | --- |
| # | Name | Return type | Desciption |
| 1 | addBike | void | Add bike to dock |
| 2 | removeBike | void | Remove bike out of dock |
| 3 | clearBikeList | void | Clear bike list in dock |
| 4 | addListOfBike | void | Add bike list to dock |

**Parameter:**

* addBike
  + bike – Bike
* removeBike
  + bike – Bike

**Exception:**

None

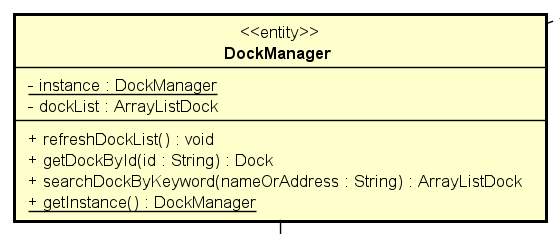
**Method**

None

**State**

None

* **Class “DockManager”**



**Attribute**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Name | Data type | Default value | Description |
| 1 | instance | DockManager | NULL | Instance of a dock |
| 2 | dockList | ArrayList<Dock> | NULL | Dock list |

**Operation**

|  |  |  |  |
| --- | --- | --- | --- |
| # | Name | Return type | Desciption |
| 1 | refreshDockList | void | Refresh dock list |
| 2 | getDockById | Dock | Get dock by id |
| 3 | searchDockByKeyword | ArrayList<Dock> | Get docks by keyword |
| 4 | getInstance | DockManager | Get dock instance |

**Parameter:**

* getDockById
  + id - String
* searchDockByKeyWord
  + nameOrAddress - String

**Exception:**

None

**Method**

None

**State**

None

* **Class “Invoice”**

Table

Description automatically generated

**Attribute**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Name | Data type | Default value | Description |
| 1 | id | String | NULL | Invoice id |
| 2 | sessionId | String | NULL | Session id |
| 3 | createAt | Time | NULL | Create time |
| 4 | type | String | NULL | Session corresponds to invoice |
| 5 | totalCharge | int | NULL | Total Charge |

**Operation**

|  |  |  |  |
| --- | --- | --- | --- |
| # | Name | Return type | Desciption |
| 1 | getDeposit | int | Get deposit |
| 2 | getCard | CreditCard | Get credit card |
| 3 | getBike | Bike | Get bike |

**Parameter:**

None

**Exception:**

None

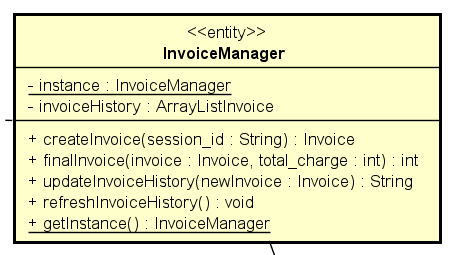
**Method**

None

**State**

None

* **Class “InvoiceManager”**



**Attribute**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Name | Data type | Default value | Description |
| 1 | instance | DockManager | NULL | Instance of a dock |
| 2 | invoiceHistory | ArrayList<Invoice> | NULL | Invoice History |

**Operation**

|  |  |  |  |
| --- | --- | --- | --- |
| # | Name | Return type | Desciption |
| 1 | createInvoice | Invoice | Create new invoice |
| 2 | finalInvoice | int | Complete invoice |
| 3 | updateInvoiceHistory | String | Update invoice history |
| 4 | refreshInvoiceHistory | void | Refresh invoice history |
| 5 | getInstance | InvoiceManager | Get invoice instance |

**Parameter:**

* createInvoice
  + session\_id - String
* finallInvoice
  + invoice – Invoice
  + total\_charge: int
* updateInvoiceHistory
  + newInvoice - Invoice

**Exception:**

None

**Method**

None

**State**

None

* **Class “CreditCard”**

A picture containing table

Description automatically generated

**Attribute**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Name | Data type | Default value | Description |
| 1 | id | String | NULL | Card id |
| 2 | cardNum | String | NULL | Card number |
| 3 | cardOwner | String | NULL | Card owner’s name |
| 4 | securityCode | String | NULL | Card security code |
| 5 | expDate | String | NULL | Card exp date |
| 6 | issuingBank | String | NULL | IssuingBank |

**Operation**

None

**Parameter:**

None

**Exception:**

None

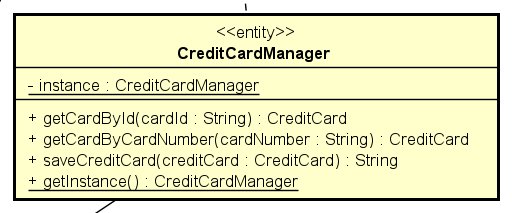
**Method**

None

**State**

None

* **Class “CreditCardManager”**



**Attribute**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Name | Data type | Default value | Description |
| 1 | instance | CreditCardManager | NULL | Instance of a card |

**Operation**

|  |  |  |  |
| --- | --- | --- | --- |
| # | Name | Return type | Desciption |
| 1 | getCardById | CreditCard | Get card by id |
| 2 | getCardByCardNumber | CreditCard | Get card by card number |
| 3 | saveCreditCard | String | Save credit card |
| 4 | getInstance | CreditCardManager | Get card instance |

**Parameter:**

* getCardById
  + id - String
* getCardByCardNumber
  + carNumber – String
* saveCreditCard
  + credidCard - CreditCard

**Exception:**

None

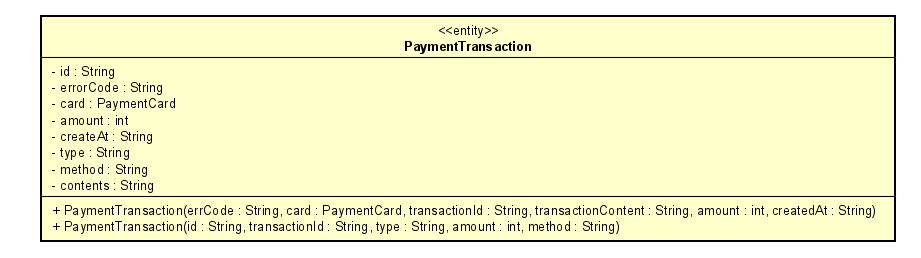
**Method**

None

**State**

None

* **Class “PaymentTransaction”**



**Attribute**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Name | Data type | Default value | Description |
| 1 | errorCode | String | NULL | Error code |
| 2 | card | PaymentCard | NULL | Credit card |
| 3 | transactionId | String | NULL | Transaction id |
| 4 | transactionContent | String | NULL | Contents of transaction |
| 5 | amount | int | NULL | Transaction amount |
| 6 | createAt | String | NULL | Transaction create time |
| 7 | id | String | NULL | PaymentTransaction id |
| 8 | type | String | NULL | Transaction type |
| 9 | method | String | NULL | Transaction method |

**Operation**

None

**Parameter:**

None

**Exception:**

None

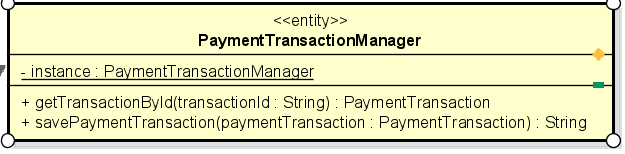
**Method**

None

**State**

None

* **Class “PaymentTransactionManager”**



**Attribute**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Name | Data type | Default value | Description |
| 1 | instance | PaymentTransaction | NULL | Instance of transaction |

**Operation**

|  |  |  |  |
| --- | --- | --- | --- |
| # | Name | Return type | Desciption |
| 1 | getTransactionById | PaymentTransaction | Get transaction by Id |
| 2 | savePaymentTransaction | String | Save payment transaction |

**Parameter:**

* getTransactionById
  + transactionId - String
* savePaymentTransaction
  + paymentTransaction - Paymentransaction

**Exception:**

None

**Method**

None

**State**

None

* **Class “Session”**

Text, letter

Description automatically generated

**Attribute**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Name | Data type | Default value | Description |
| 1 | id | String | NULL | Session id |
| 2 | bikeId | String | NULL | Bike Id |
| 3 | rentDockId | String | NULL | Dock rented Id |
| 4 | returnDockId | String | NULL | Dock Return Id |
| 5 | endTime | LocalDateTime | NULL | End time |
| 6 | startTime | LocalDateTime | NULL | Start time |

**Operation**

|  |  |  |  |
| --- | --- | --- | --- |
| # | Name | Return type | Desciption |
| 1 | isActive | boolean | Session is active or not |
| 2 | setActive | void | Set session active |
| 3 | getSessionLength | long | Get session length |

**Parameter:**

None

**Exception:**

None

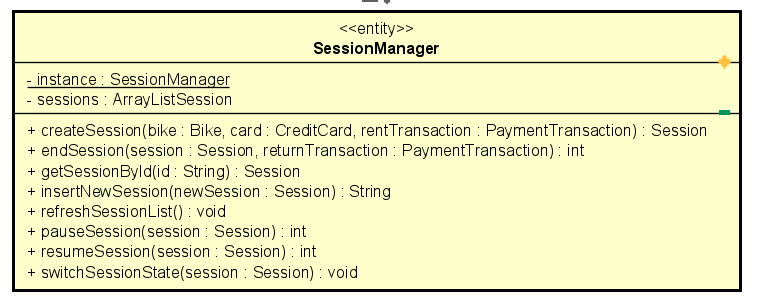
**Method**

None

**State**

None

* **Class “SessionManager”**



**Attribute**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Name | Data type | Default value | Description |
| 1 | instance | SessionManager | NULL | Session instance |
| 2 | sessions | ArrayList<Session> | NULL | Session list |

**Operation**

|  |  |  |  |
| --- | --- | --- | --- |
| # | Name | Return type | Desciption |
| 1 | createSession | Session | Create a session |
| 2 | endSession | int | Complete session |
| 3 | getSessionById | Session | Get session by id |
| 4 | insertNewSession | String | Insert new session to db |
| 5 | refreshSessionList | void | Refresh session list |
| 6 | pauseSession | int | Pause session |
| 7 | resumeSession | int | Resume session |
| 8 | switchSessionState | void | Switch session state |

**Parameter:**

* createSession
  + bike - Bike
  + card – CreditCard
  + rentTransaction - PaymentTransaction
* endSession
  + session – Session
  + returnTransaction - PaymentTransaction
* getSessionById
  + id - String
* insertNewSession
  + newSession – Session
* pauseSession
  + session – Session
* resumeSession
  + session – Session
* switchSessionState
  + session – Session

**Exception:**

None

**Method**

None

**State**

None

* **Class “InterbankInterface”**

Diagram

Description automatically generated with medium confidence

**Attribute**

None

**Operation**

|  |  |  |  |
| --- | --- | --- | --- |
| # | Name | Return type | Description |
| 1 | payOrder | PaymentTransaction | Pay order |
| 2 | refund | PaymentTransaction | Refund money |

Parameter:

* payOrder
  + creditCard
  + amount
  + contents
* refund
  + card
  + amount
  + contents

Exception:

None

**Method**

None

**State**

None

* **Class “InterbankSubsystem”**

Table

Description automatically generated

**Attribute**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Name | Data type | Default value | Description |
| 1 | ctrl | InterbankSubsystemController | NULL | Controller payment transaction |

**Operation**

|  |  |  |  |
| --- | --- | --- | --- |
| # | Name | Return type | Description |
| 1 | payOrder | PaymentTransaction | Pay order |
| 2 | refund | PaymentTransaction | Refund money |

Parameter:

* payOrder
  + card
  + amount
  + contents
* refund
  + card
  + amount
  + contents

Exception:

None

**Method**

None

**State**

None

* **Class “InterbankSubsystemController”**

Text

Description automatically generated

**Attribute**

None

**Operation**

|  |  |  |  |
| --- | --- | --- | --- |
| # | Name | Return type | Description |
| 1 | refund | PaymentTransaction | Refund money |
| 2 | payDeposit | PaymentTransaction | Pay deposit money |
| 3 | generateData | String | Generate data |
| 4 | makePaymentTransaction | PaymentTransaction | Make payment transaction |

Parameter:

* payDeposit
  + creditCard
  + amount
  + contents
* refund
  + card
  + amount
  + contents
* generateData
  + data
* makePaymentTransaction
  + respond

Exception:

None

**Method**

None

**State**

None

* **Class “InterbankBoundary”**

Diagram

Description automatically generated with medium confidence

**Attribute**

None

**Operation**

|  |  |  |  |
| --- | --- | --- | --- |
| # | Name | Return type | Description |
| 1 | query | String | Query url and data with boundary bank |

Parameter:

* url
* data

Exception:

None

**Method**

None

**State**

None

# Design Considerations

## Goals and Guidelines

Goals:

* Thiết kế cơ sở dữ liệu tập trung vào tốc độ lưu trữ tối ưu. Do các yêu cầu phi chức năng của phần mềm đòi hỏi khả năng đáp ứng nhanh, vì vậy, lượng dữ liệu không quá lớn.

Guidelines:

* The naming convention will follow the Java naming convention.

## Architectural Strategies

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

## Coupling and Cohesion

* Coupling:
  + Không có coupling mức độ Content, Common, Control
  + Hầu hết các package đạt Data Coupling.
* Cohesion:
  + Không có Cohesion mức độ Temporal, Procedural và Communicational.
  + Class Utils bị vi phạm Concidental Cohesion nhưng đây là class hỗ trợ cho nên không phải là vấn đề.

## Design Principles

* Single Responsibility

Mỗi lớp được thiết kế tuân theo nguyên tắc này. Chỉ có ngoại lệ là lớp Utils vì nó chứa phương thức cho cả bảo mật và định dạng. Tuy nhiên, như đã giải thích trong phần kết hợp, tất cả các phương thức này là toàn cục, vì vậy việc chia nó thành 2 lớp tiện ích không ảnh hưởng nhiều.

* Open / Closed

Sử dụng các class cha để tổng quát hóa, ví dụ khi thêm cách tính phí thuê xe, thêm loại xe, thêm phương thức thanh toán

* Liskov Substitution Principle

Tất cả các lớp cha có thể được thay thế bằng các lớp con của nó. Thiết kế có một hệ thống phân cấp kế thừa rõ ràng và hợp lý.

* Interface Segregation Principle

InterbankSubsytem Interface đã được thiết kế rất đơn giản và chỉ chứa 2 phương thức cần thiết để thanh toán và hoàn tiền.

* Dependency Inversion Principle

Các mô-đun cấp cao hơn không phụ thuộc vào các mô-đun cấp thấp hơn. Tất cả các hệ thống con đều có giao diện riêng để người khác sử dụng.