

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 05
Nguyen Thai An – 20176677
Bui Tu Hoang – 20176761
Vu Minh Hoang – 20176765
Nguyen Manh Khang - 20176792

Hanoi, November, 2020

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 | Error! Bookmark not defined. |
| 2 Overall Description | 5 |
| 2.1 General Overview..... | 5 |
| 2.2 Assumptions/Constraints/Risks | 5 |
| 2.2.1 Assumptions..... | 5 |
| 2.2.2 Constraints | 6 |
| 2.2.3 Risks..... | 6 |
| 3 System Architecture and Architecture Design | 7 |
| 3.1 Architectural Patterns | 7 |
| 3.2 Interaction Diagrams | 7 |
| 3.3 Analysis Class Diagrams | 15 |
| 3.4 Unified Analysis Class Diagram | 17 |
| 3.5 Security Software Architecture | 17 |
| 4 Detailed Design | 18 |
| 4.1 User Interface Design | 18 |
| 4.1.1 Screen Configuration Standardization | 18 |
| 4.1.2 Screen Transition Diagrams..... | 20 |
| 4.1.3 Screen Specifications | 20 |
| 4.2 Data Modeling | 32 |
| 4.2.1 Conceptual Data Modeling | 32 |
| 4.2.2 Database Design..... | 32 |

| | | |
|-------|--|-----|
| 4.3 | Non-Database Management System Files | 45 |
| 4.4 | Class Design | 45 |
| 4.4.1 | General Class Diagram | 45 |
| 4.4.2 | Class Diagrams | 46 |
| 4.4.3 | Class Design..... | 48 |
| 5 | Design Considerations..... | 106 |
| 5.1 | Goals and Guidelines..... | 106 |
| 5.2 | Architectural Strategies | 106 |
| 5.3 | Coupling and Cohesion | 106 |
| 5.4 | Design Principles | 107 |
| 5.5 | Design Patterns | 108 |

1 Introduction

1.1 Objective

The purpose of this document is to present a detailed description of the designs of the Eco Bike Rental System, created for the Eco Park township. This document is intended for the programming group in Team 5, to use the designs as guidelines to implement the project. On the other hand, this document is for reviewing by Prof. Nguyen Thi Thu Trang and her assistants, as a part of the course. The document could also be used for designers who try to upgrade or modify the design of the system

1.2 Scope

Eco Bike Rental software's aim, as its name, is to provide a bike renting service to customers.

In this software, user will be provided with the ability to search for docks and see their information in details. At each dock, user can have knowledge of bikes parked at the dock. Bike renting and returning are key functions of this system.

Interbank will have the possibility to execute payment transactions. Interbank will take place in the validating, adding money and deducting money processes.

This document gives a detailed description of the software architecture of the inventory system. It specifies the structure and design of some of the modules discussed in the SRS. It also provides sequential and activity diagrams for some use case. The class diagrams show how the programming team would implement the specific module.

1.3 Glossary

| Word | Meaning |
|---------|--|
| deposit | to pay someone an amount of money when you make an agreement with that person to pay for or buy something, that either will be returned to you later, if the agreed arrangement is kept, or that forms part of the total payment |
| deduct | to take away an amount or part from a total |

| | |
|-------------|---|
| transaction | an occasion when someone buys or sells something, or when money is exchanged or the activity of buying or selling something |
|-------------|---|

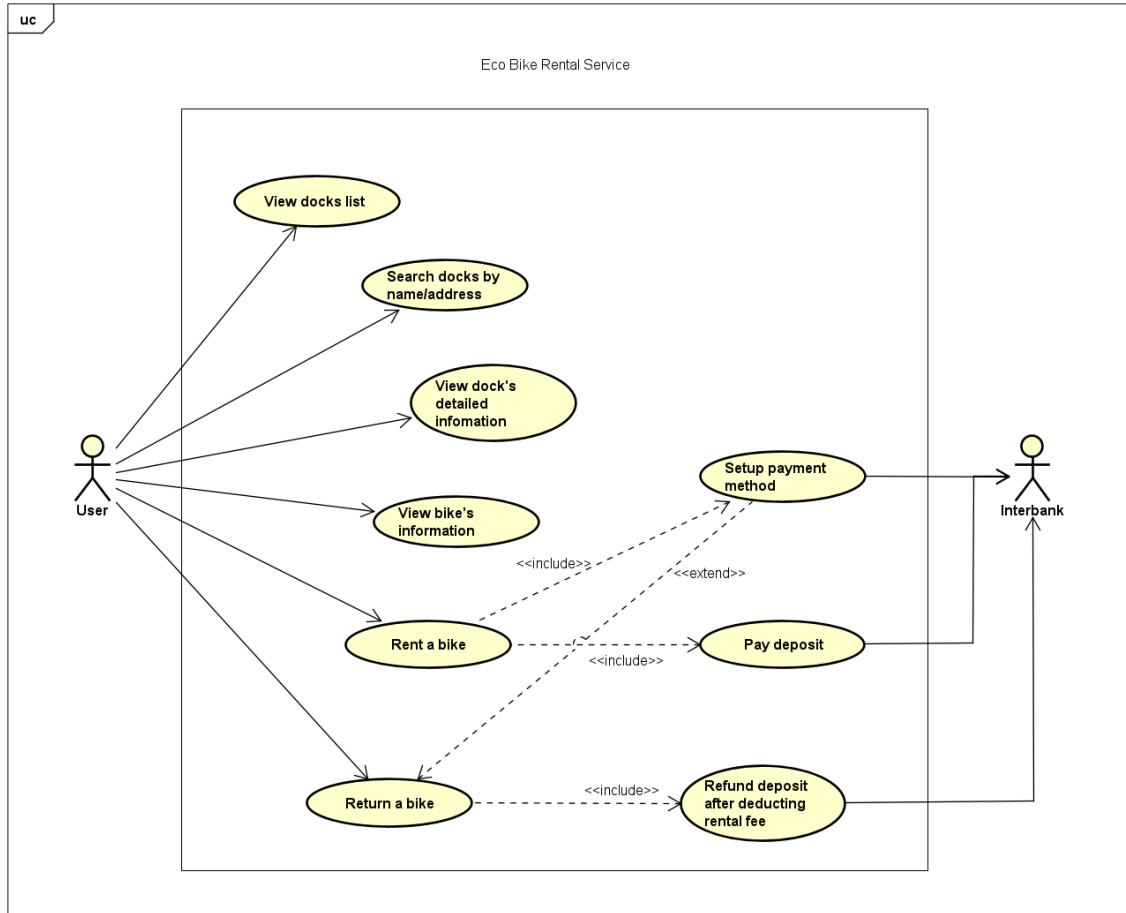
1.4 References

2 Overall Description

2.1 General Overview

The EBR system is a simulator for a bike rental service. This is a desktop software run on Java Runtime Environment. This software is designed using the MVC architectural design model.

The general use case diagram is as following:



2.2 Assumptions/Constraints/Risks

2.2.1 Assumptions

The design described in this document is based on simulation requirements instead of reality requirements. Therefore, functions related to authentication and personal customer information are neglected.

This software is designed to run on any operating system with Java Runtime Environment.

2.2.2 Constraints

The system is built accessible only through personal computer. The system is implemented using Java and uses JavaFX for making user interfaces.

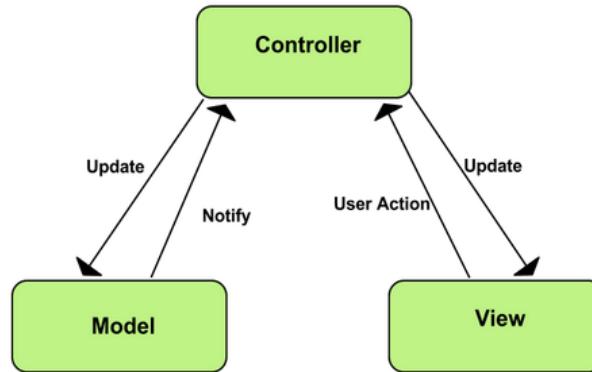
For storing data, PostgreSQL will be used with the help of JDBC

2.2.3 Risks

3 System Architecture and Architecture Design

3.1 Architectural Patterns

Model-View-Controller (MVC) is the chosen design pattern. The concept of MVC which used in this project can be described by the image below



MVC has its own advantages:

Simultaneous development - Multiple developers can work simultaneously on the model, controller and views.

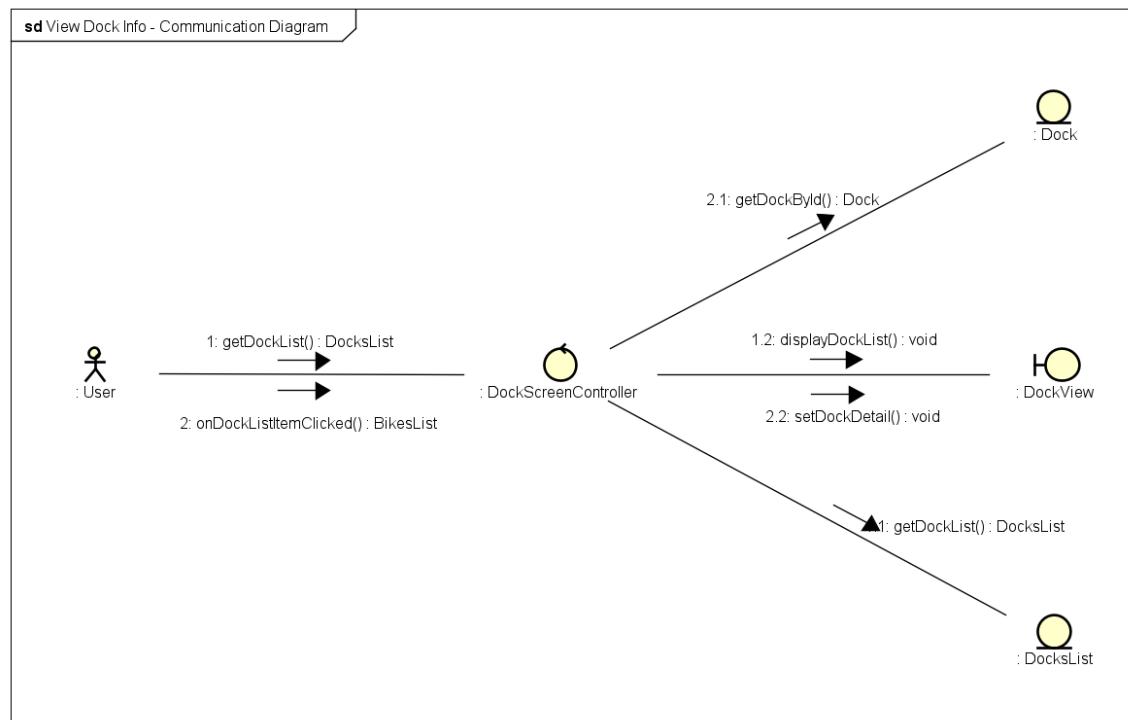
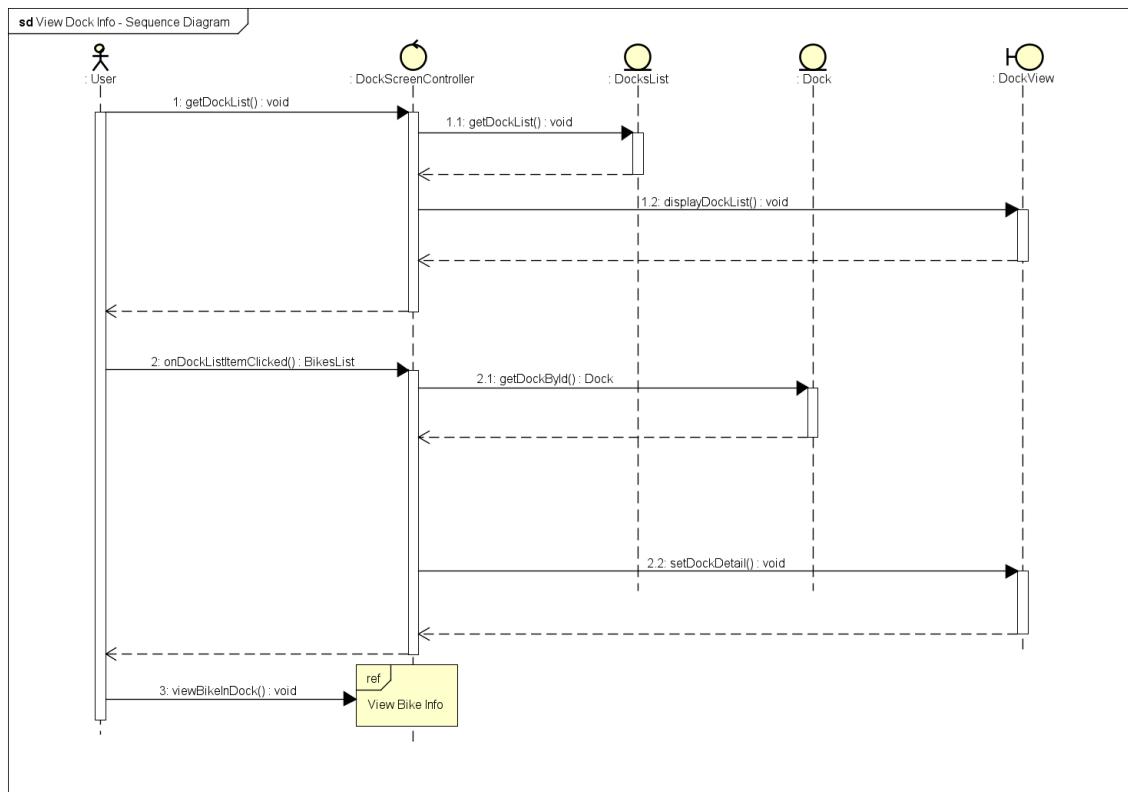
High cohesion - MVC enables logical grouping of related actions on a controller together.

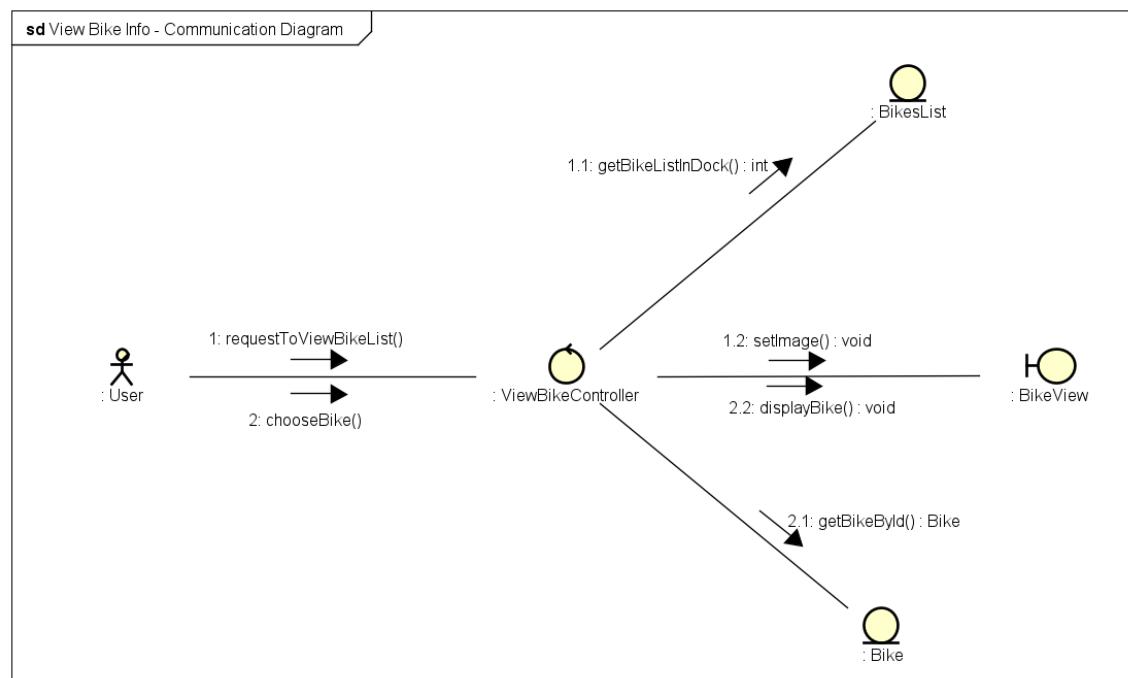
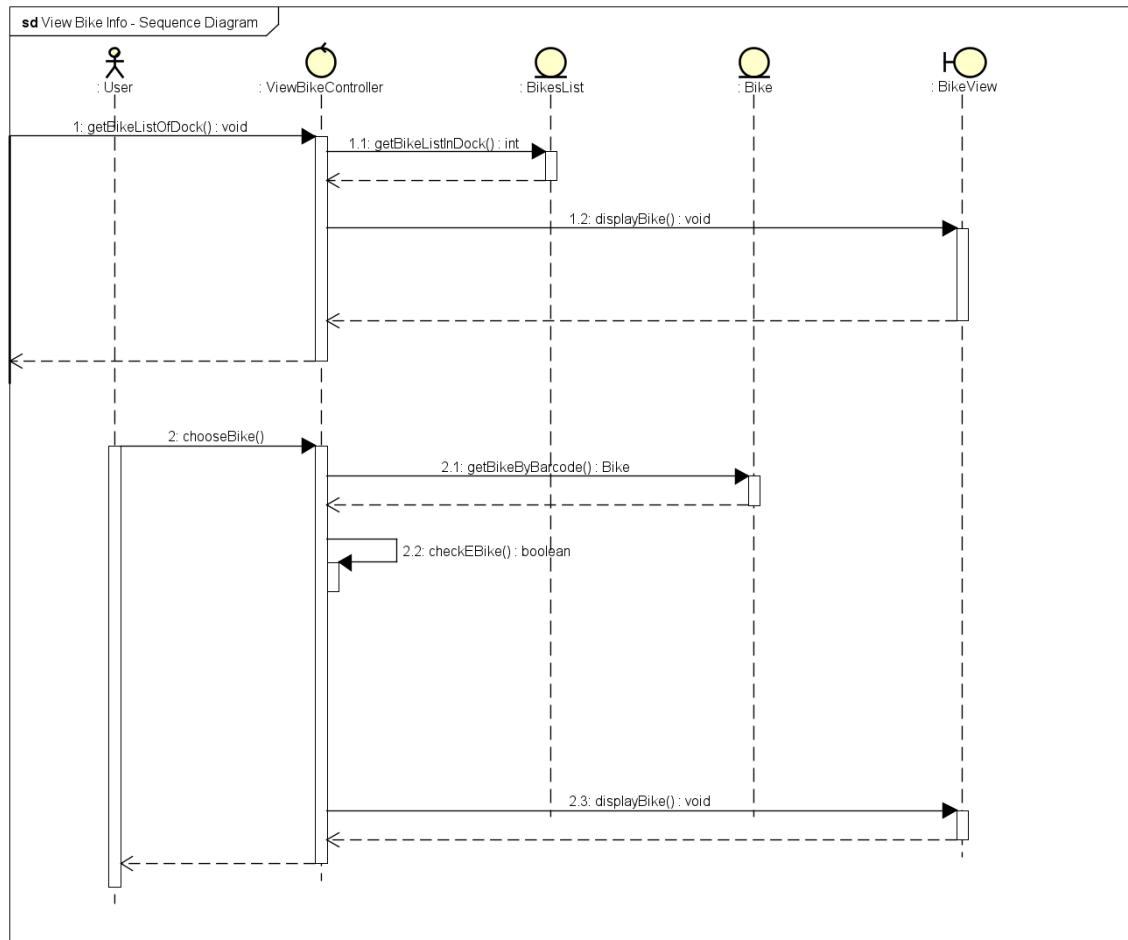
Low coupling - The very nature of the MVC framework is such that there is low coupling among models, views or controllers.

Ease of modification - Because of the separation, future development or modification is easier, and also the scalability.

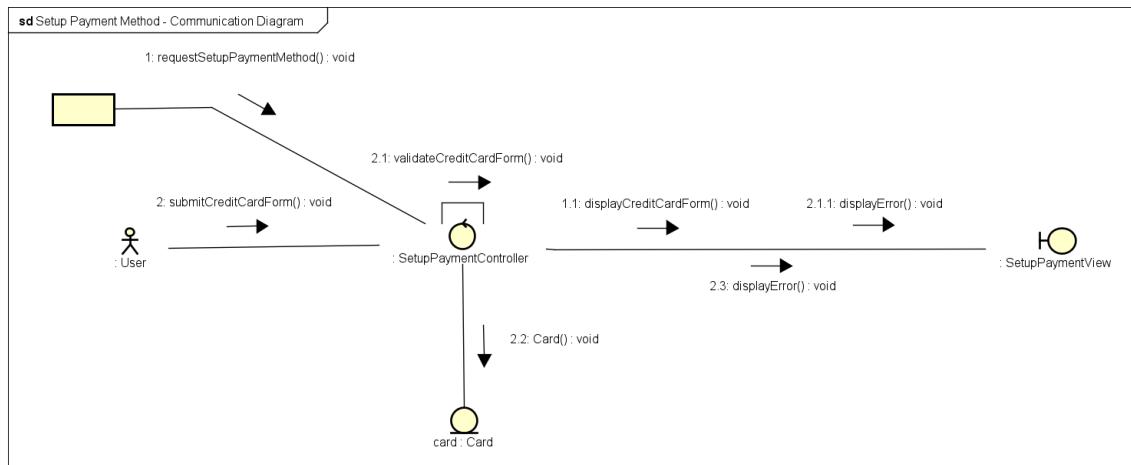
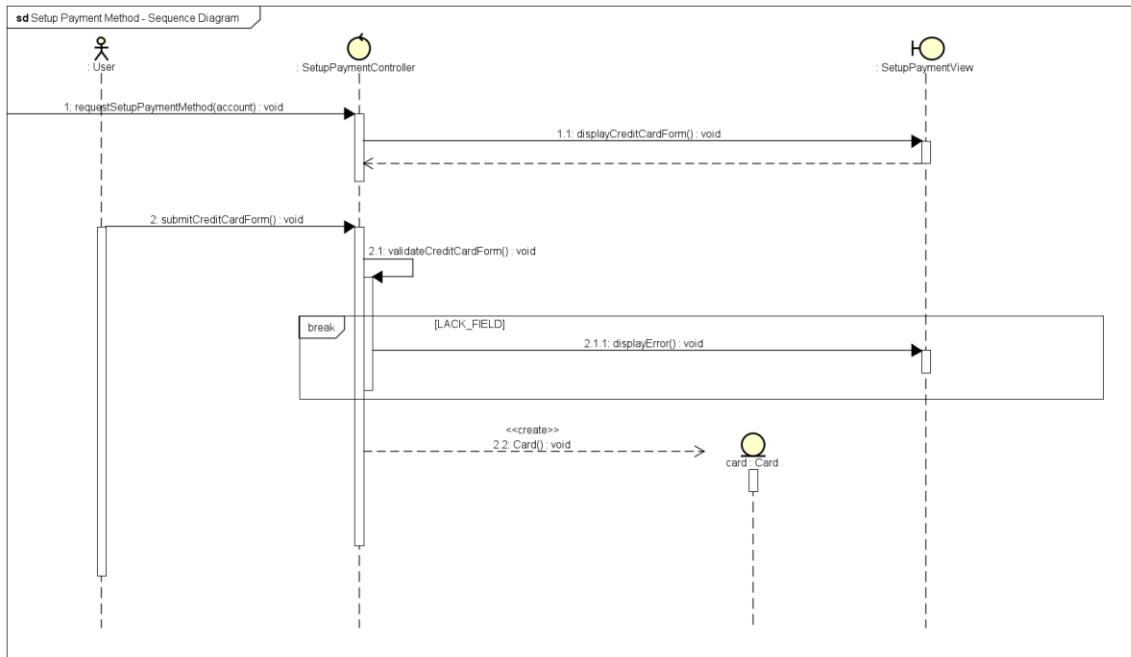
3.2 Interaction Diagrams

3.2.1 Browse Docks and Bikes

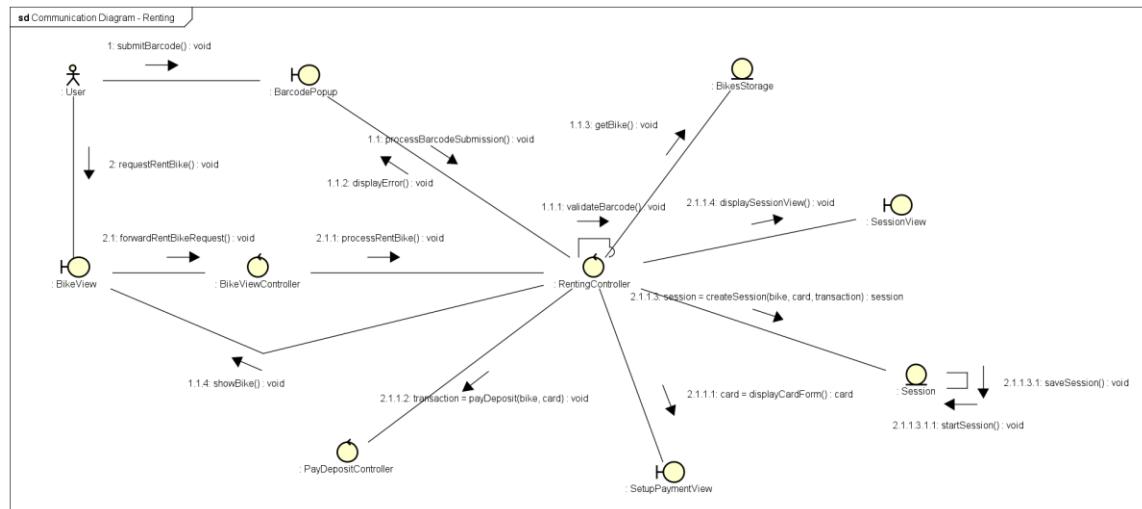
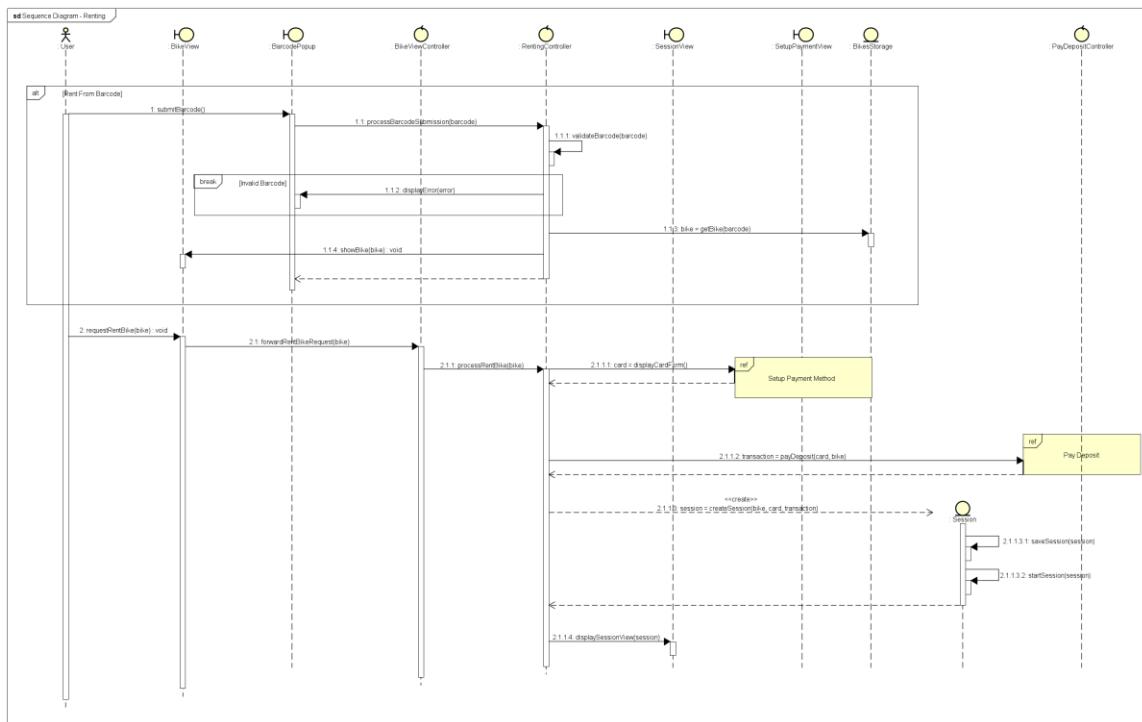


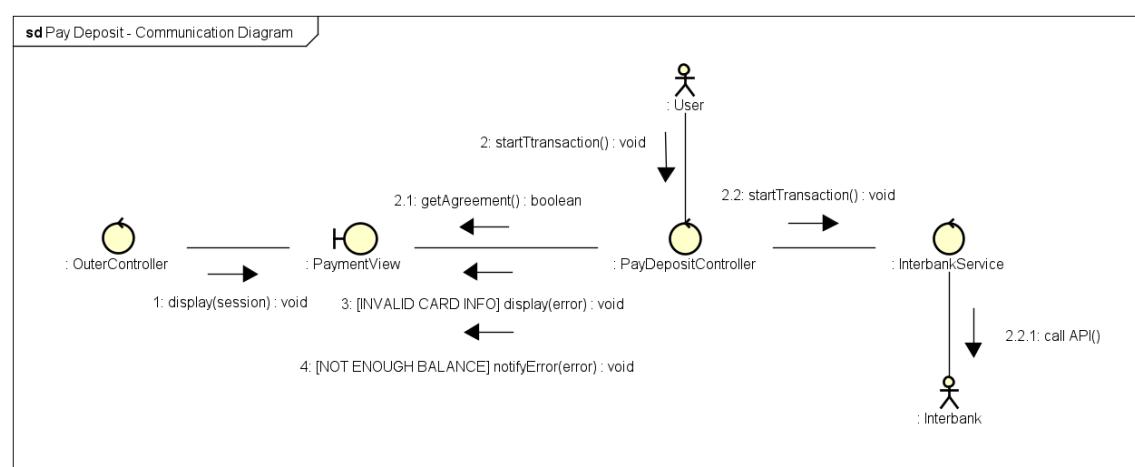
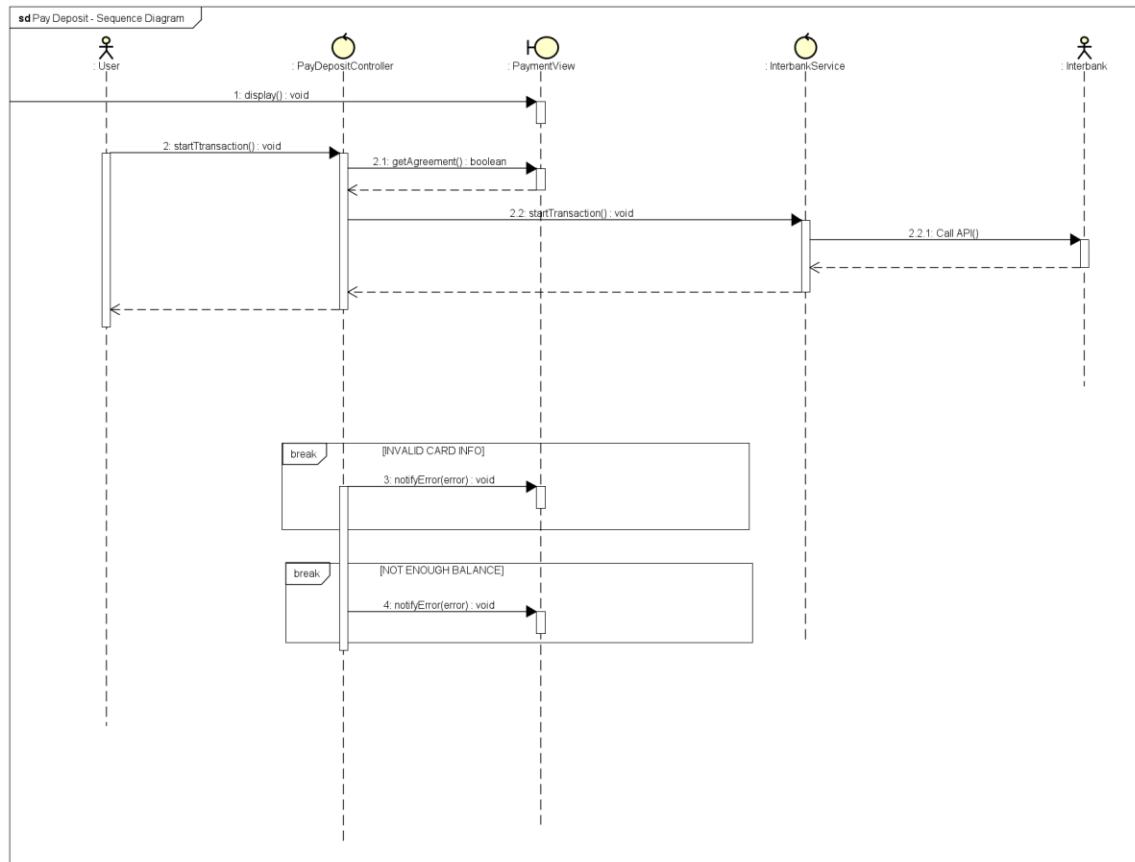


3.2.2 Setup Payment Method

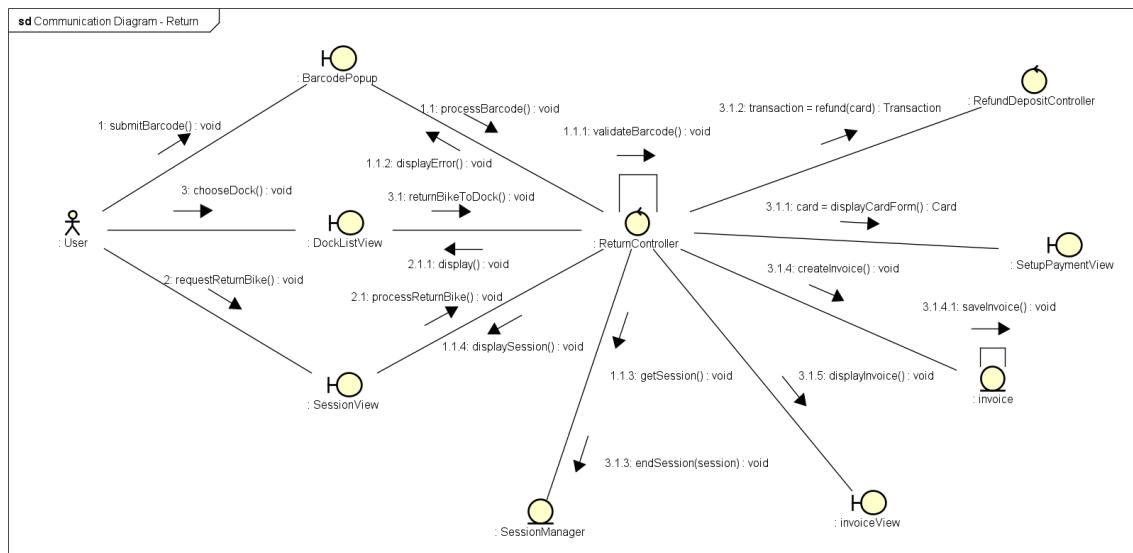
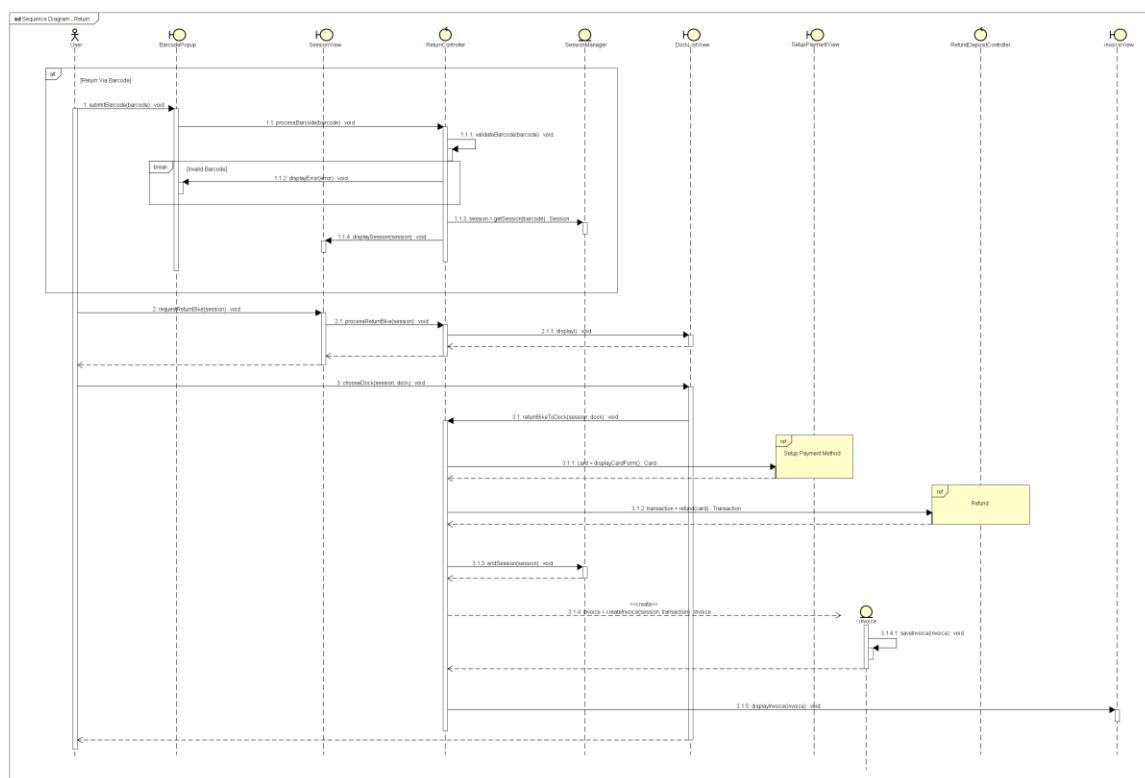


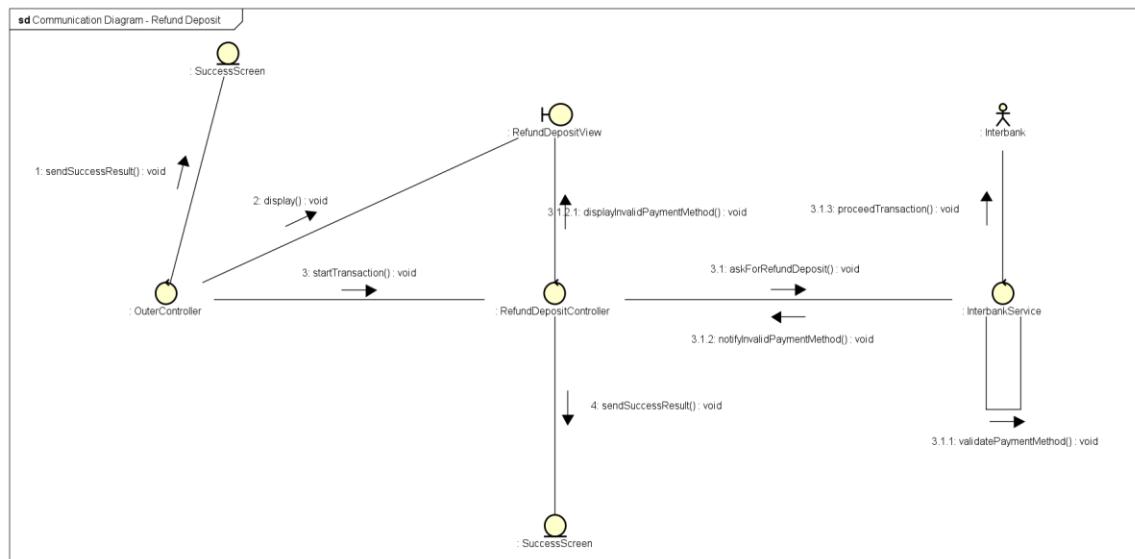
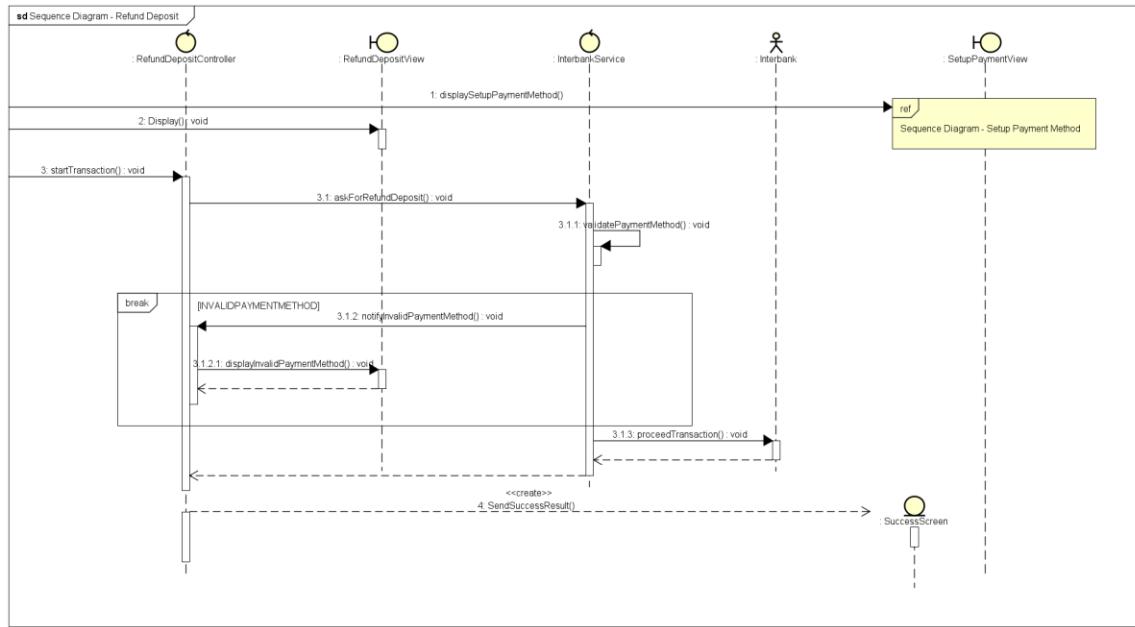
3.2.3 Rent Bike





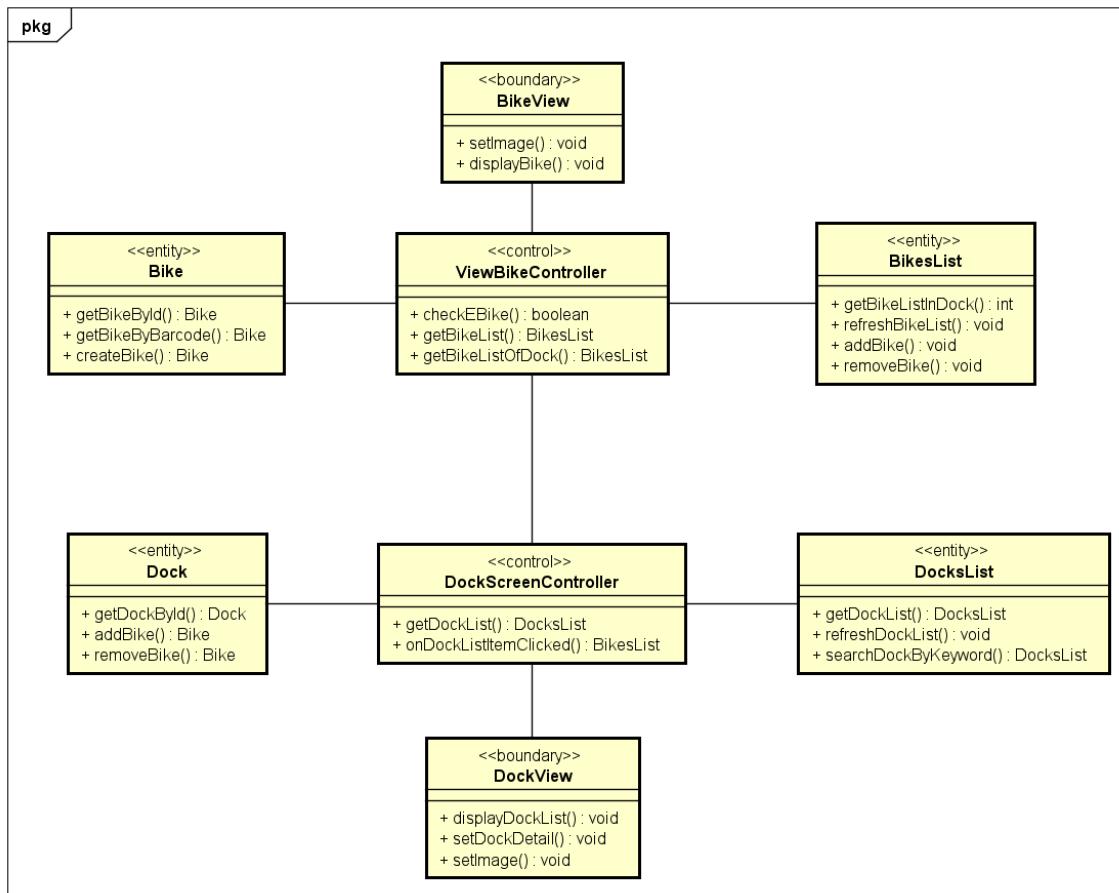
3.2.4 Return Bike



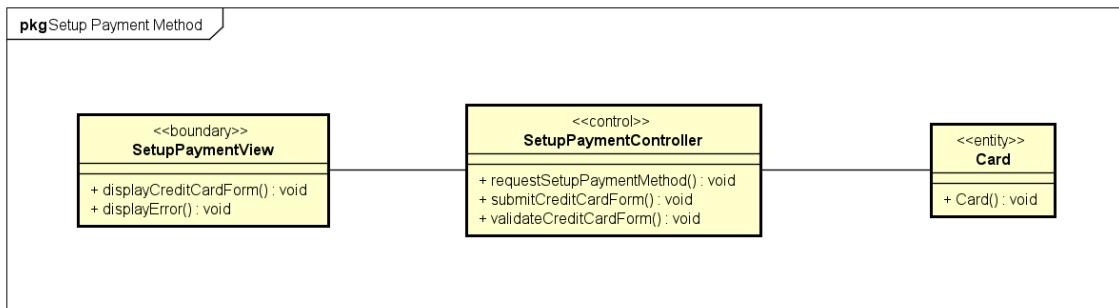


3.3 Analysis Class Diagrams

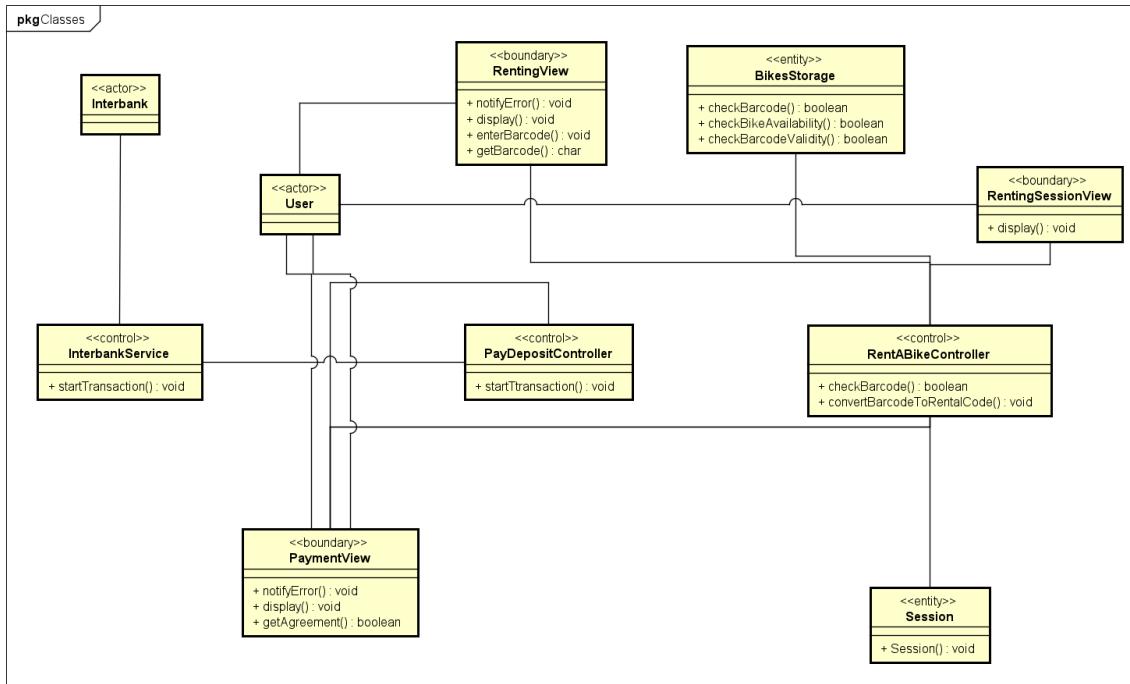
3.3.1 Docks and Bikes



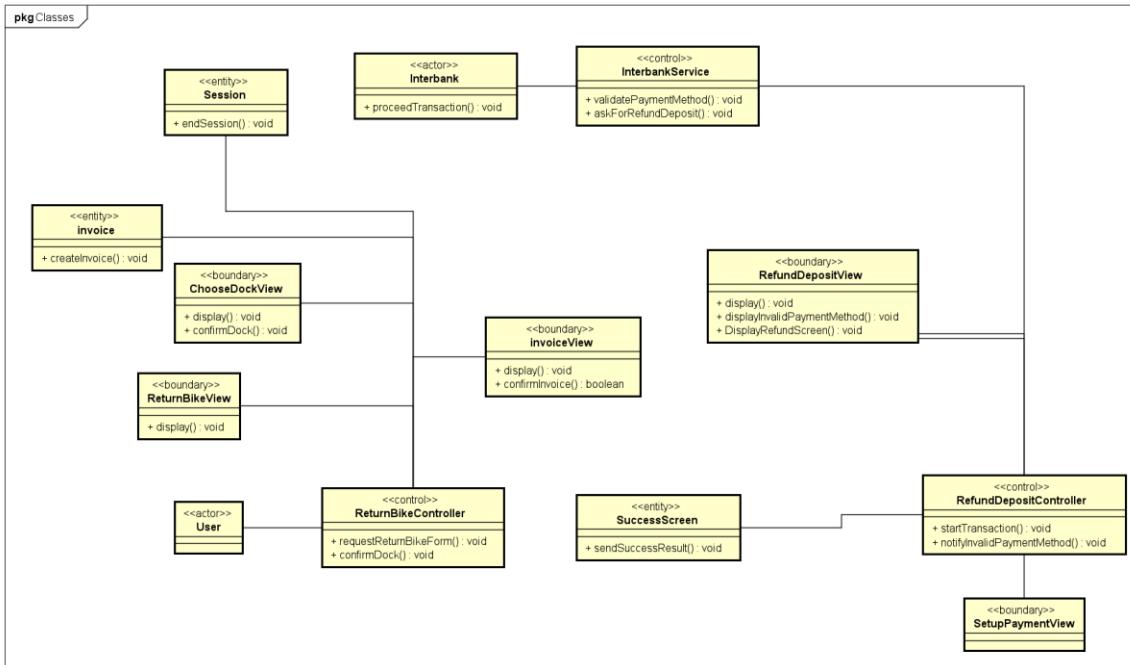
3.3.2 Setup Payment



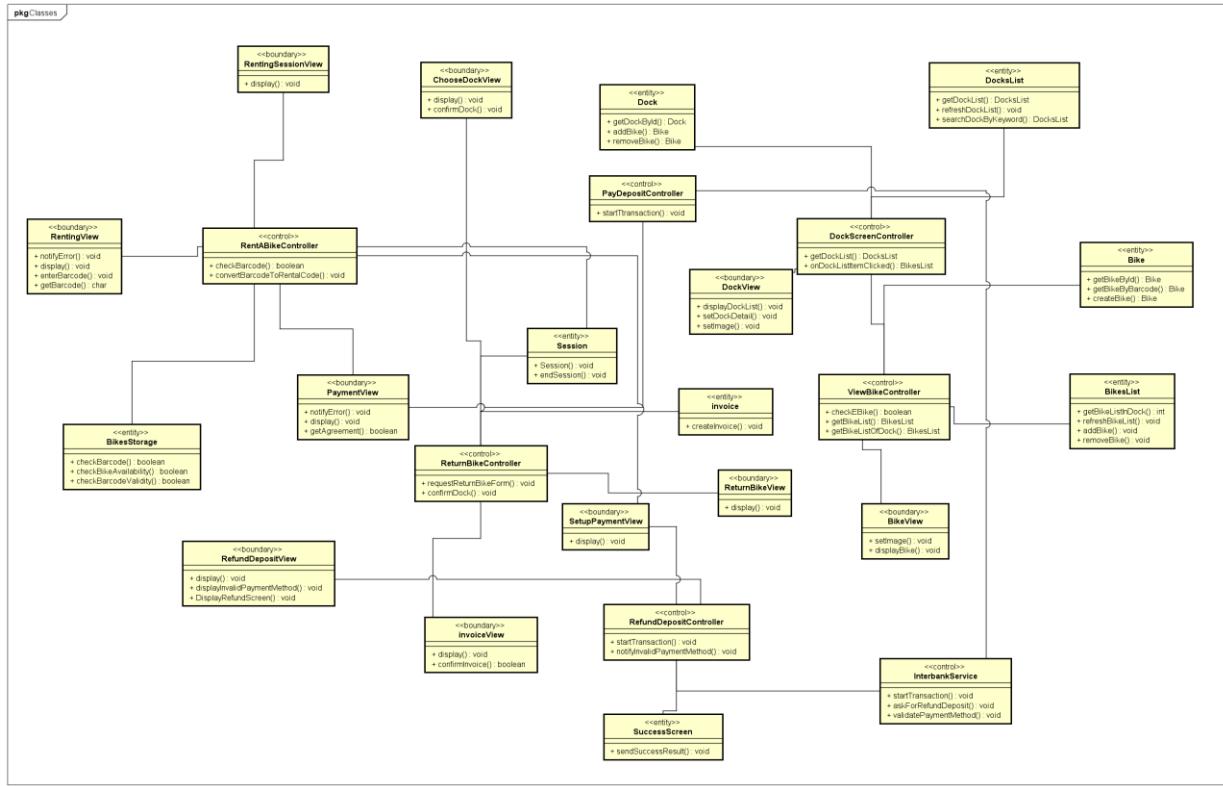
3.3.3 Rent Bike



3.3.4 Return Bike



3.4 Unified Analysis Class Diagram



3.5 Security Software Architecture

4 Detailed Design

4.1 User Interface Design

4.1.1 Screen Configuration Standardization

Display

Number of colors supported: 16,777,216 colors

Resolution: 1366 × 768 pixels

Screen

Location of standard buttons: At the bottom (vertically) and in the middle (horizontally) of the frame.

- Affirmative button to the right, background-color: #8C36C0, text-color: #FFFFFF
- Negative button to the left, background-color: #FFFFFF, text-color: #8C36C0

Location of the messages: Starting from the top vertically and in the middle horizontally of the frame down to the bottom.

Display of the screen title: The title is located at the center-top of the screen and also on the top-left of the window bar.

Consistency in expression of alphanumeric numbers: no separator for thousand while strings only consist of characters, digits, commas, dots, spaces, underscores, and hyphen symbol.

Control

Size of the text:

- Title and header text: font: Roboto, style: bold, size: 48px, color: #8C36C0
(medium-dark purple)
- Text in button: font: Roboto, style: black, size: 24px, color: #FFFFFF
- Other text: font: Roboto, style: regular, size: 14px, color: #000000

Input check process: Should check if it is empty or not. Next, check if the input is in the correct format or not

Sequence of moving the focus: There will be some popup screen. The popup screen will overlay the current parent screen. Cannot interact with parent screen while the popup screen overlaying. Click outside of the popup screen will close it, return active screen to the parent screen.

After the opening screen, the app will start with splash screen, and then the first screen (homescreen) will appear.

Sequences of the system screens:

1. Splash screen (first screen)
2. Home screen – show list of docks
3. Barcode Popup – for inputting the barcode (rent/return bike)
4. Dock View Screen – view all available bikes and dock's info
5. Bike View Screen – detailed bike's info
6. Payment Screen – fill in payment method (credit card)
7. Transaction Result Popup – show the result of transaction
8. Renting Session – all session's info (start time, current rental fee, bike's info...)
9. Invoice screen – view order details
10. Dock Selection Screen – for choosing dock when returning

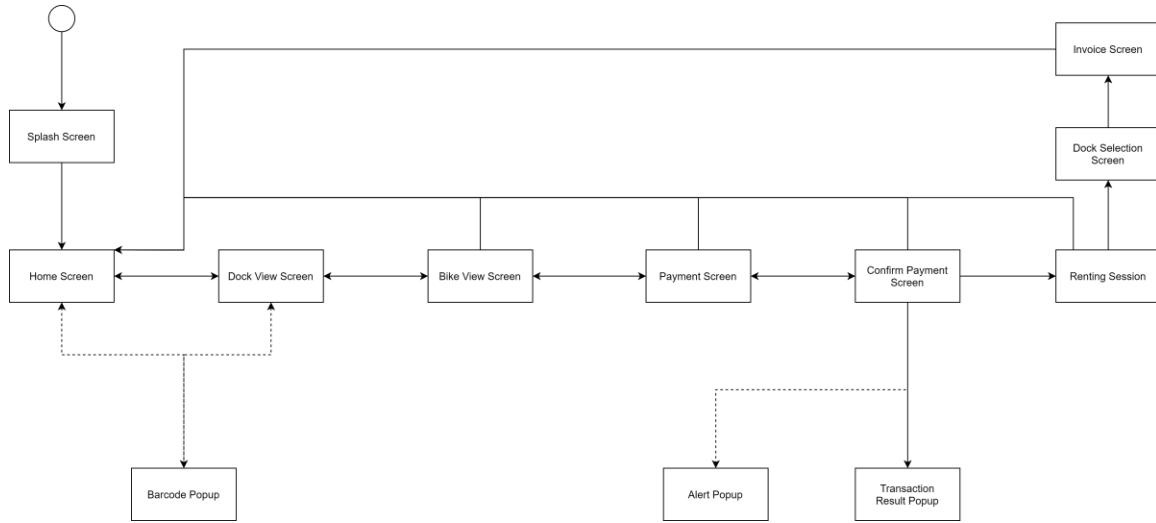
Direct input from the keyboard

There will be no shortcuts. There are back buttons to move back to the previous screen located at the top-right corner of each screen. All input fields take input directly from keyboard.

Error

A message will be given to notify the users what is the problem.

4.1.2 Screen Transition Diagrams



4.1.3 Screen Specifications

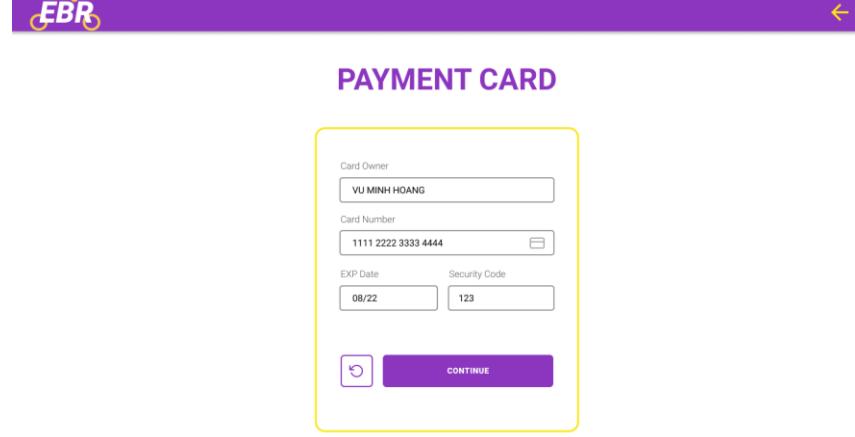
4.1.3.1 Home Screen

| EBR Software | | Date of creation | Approved by | Reviewed by | Person in charge |
|----------------------|--|--------------------------|-------------|------------------------------|------------------|
| Screen specification | Home Screen | 6/11/2020 | | | Nguyễn Thái An |
| | <p>The screenshot shows the EBR software interface. At the top left is the EBR logo. To its right is a search bar with a magnifying glass icon and the placeholder text "Search". Next to the search bar is a yellow button labeled "ENTER BARCODE" with a barcode icon. Below the search bar is a section titled "DOCKS" in purple. Underneath "DOCKS" are four cards, each representing a dock (Dock 1, Dock 2, Dock 3, Dock 4). Each card displays a small image of a bike dock, the dock's name, its address, and its bike number (166/200).</p> | Control | Operation | Function | |
| | | Enter Barcode Button | Click | Display Barcode Popup Screen | |
| | | Back button | Click | Back to Splash Screen | |
| | | Area for entering search | Initial | Display “search” faded word | |

Defining the field attributes

| Screen name | Home Screen | | | |
|----------------|-----------------------------|--------|-----------------|----------------|
| Item name | Number of digits (bytes) | Type | Field attribute | Remarks |
| Search field | 50 | String | Black | Left justified |
| Dock's name | 30 | String | White | Left justified |
| Dock's address | 50 | String | White | Left justified |
| Bike Number | 4 | Int | White | Left justified |

4.1.3.2 Payment Screen

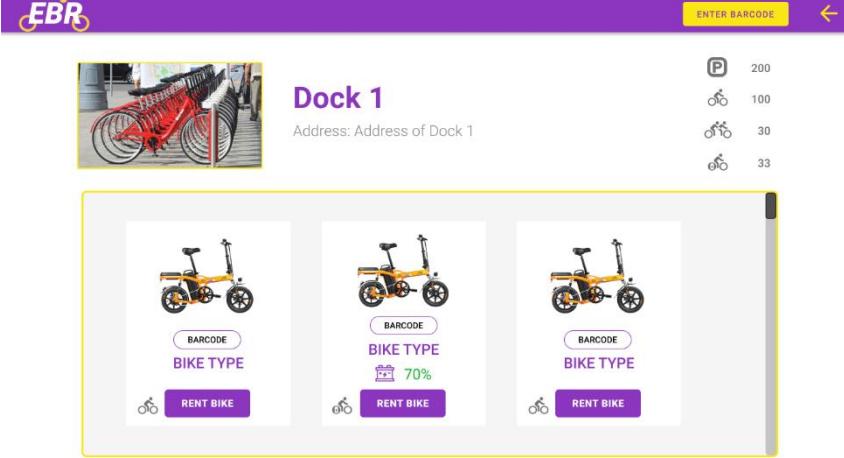
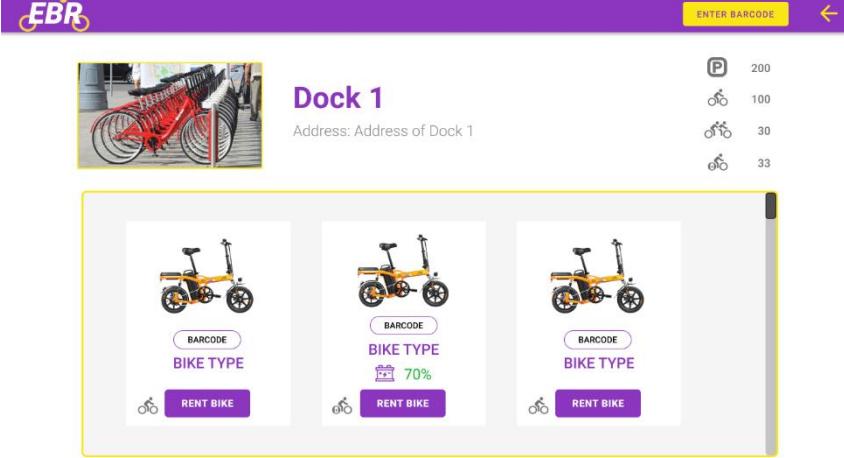
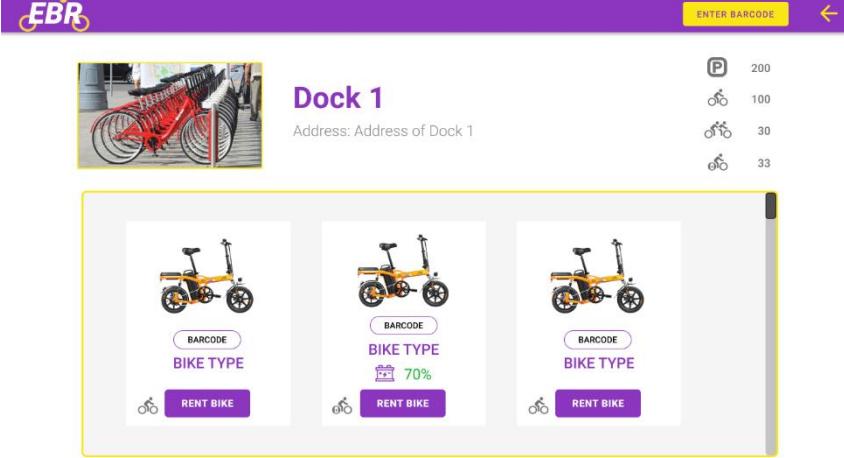
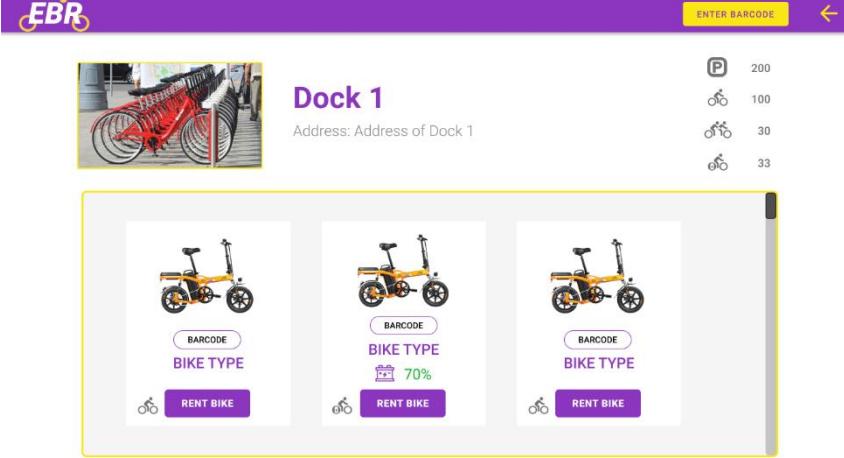
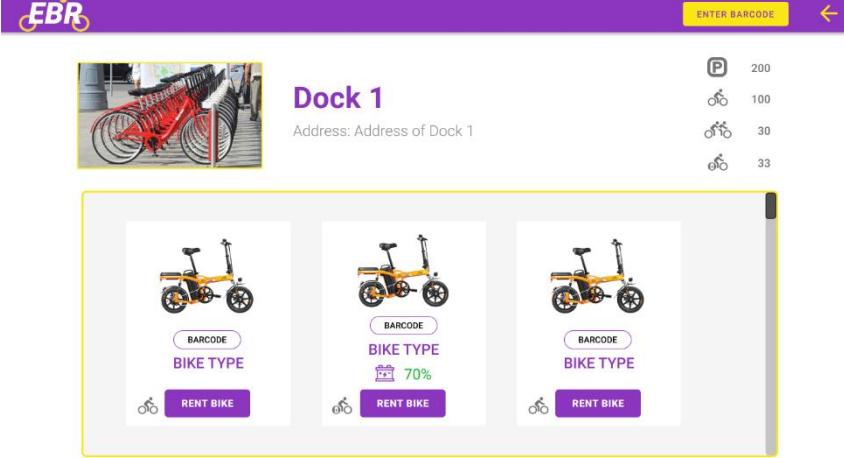
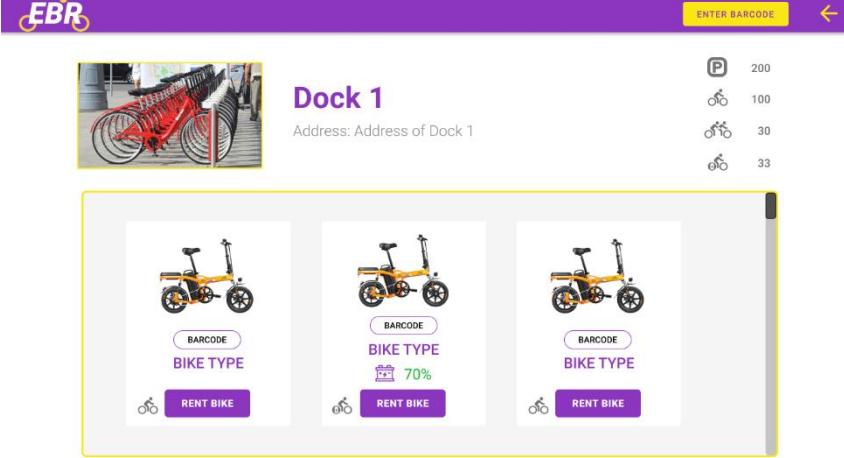
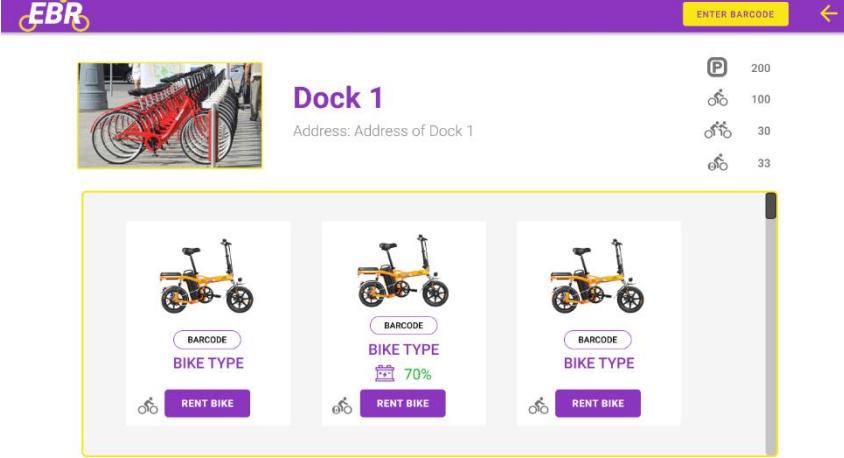
| EBR Software | Date of creation | Approved by | Reviewed by | Person in charge |
|--|-----------------------------------|-------------|---|------------------|
| Screen specification | Payment Screen | 6/11/2020 | | Nguyễn Thái An |
|  | Control | Operation | Function | |
| | Place holder text in input fields | Initial | Display what kind of information need for the field | |
| | Back button | Click | Navigate to the previous screen. | |
| | Confirm button | Click | Process payment, show Renting Session Screen | |
| | Cancel button | Click | Navigate back to Home Screen | |

Defining the field attributes

| Screen name | Payment Screen |
|-------------|----------------|
| | |

| Item name | Number of digits (bytes) | Type | Field attribute | Remarks |
|---------------|--------------------------|---------|-----------------|--------------------------|
| Card Owner | 30 | String | Black | Left justified |
| Card Number | 16 | Numeral | Black | Left justified |
| EXP date | 5 | String | Black | Left justified, MM/YY |
| Security Code | 3 | Numeral | Black | Left justified |

4.1.3.3 Dock View Screen

| EBR Software | | Date of creation | Approved by | Reviewed by | Person in charge |
|--|------------------|---|-------------|--------------------------------------|------------------|
| Screen specification | Dock View Screen | 7/11/2020 | | | Bùi Tú Hoàng |
|  | | 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 | |

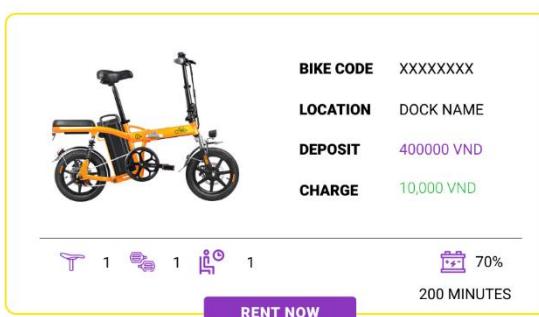
| | | | |
|--|------------------|--|--|
| | Bike in the Dock | | |
|--|------------------|--|--|

Defining the field attributes

| 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 |

4.1.3.4 Bike View Screen

| EBR Software | | Date of creation | Approved by | Reviewed by | Person in charge |
|----------------------|------------------|------------------|-------------|-------------------------------|------------------|
| Screen specification | Bike View Screen | 7/11/2020 | | | Bùi Tú Hoàng |
| | | Control | Operation | Function | |
| | | Back Button | Click | Back to the Dock View screen. | |
| | | Rent Now Button | Click | Display Payment Screen | |

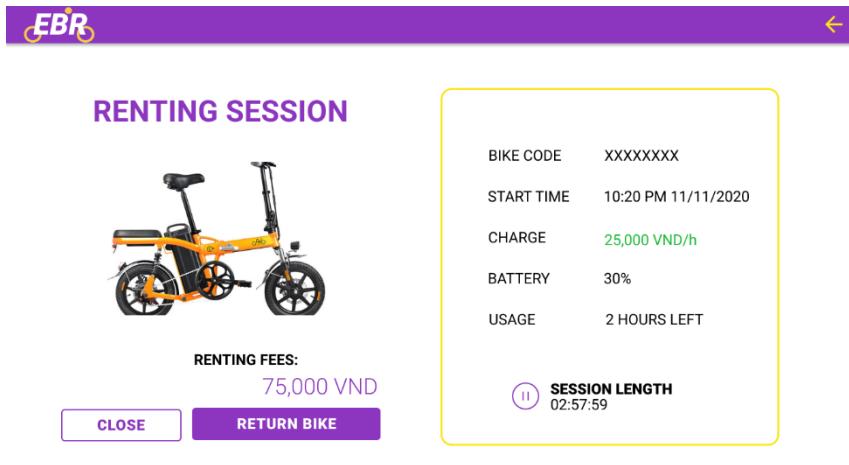
| | | | | |
|---|--|-------------------------------|---|---------------------|
|  | ← | BIKE INFORMATION | | |
|  | | EBR Home Screen Button | Click | Display Home Screen |
|  | Area for displaying the Bike Information | Initial | Displaying the Bike Information and Deposit | |

Defining the field attributes

| Screen name | Bike View Screen | | | |
|----------------------------|--------------------------|---------|-----------------|-----------------|
| Item name | Number of digits (bytes) | Type | Field attribute | Remarks |
| Bike Code | 8 | String | Black | Left justified |
| Location | 30 | String | Black | Left justified |
| Battery | 4 | String | Black | Left justified |
| Usage | 25 | String | Black | Right justified |
| Deposit | 15 | String | Purple | Right justified |
| Charge | 15 | String | Green | Right justified |
| Number of Saddle | 1 | Numeral | Black | Left justified |
| Number of Rear Seat | 1 | Numeral | Black | Left justified |
| Number of Pedal | 1 | Numeral | Black | Left justified |

4.1.3.5 Renting Session Screen

| | | | | |
|---------------------|-------------------------|--------------------|--------------------|-------------------------|
| EBR Software | Date of creation | Approved by | Reviewed by | Person in charge |
|---------------------|-------------------------|--------------------|--------------------|-------------------------|

| Screen specification | Renting Session Screen | 7/11/2020 | | | Nguyễn Mạnh Khang |
|----------------------|--|---|-----------|--|-------------------|
| |  | Control | Operation | Function | |
| | | Enter Barcode Button | Click | Display Barcode Popup Screen | |
| | | Back Button | Click | Navigate to the previous screen. | |
| | | Return Bike button | Click | Display Invoice Screen | |
| | | EBR Home Screen Button | Click | Display Home Screen | |
| | | Play/Pause Button | Click | Pause session | |
| | | Area for displaying the Session Information | Initial | Displaying the Session Information and Deposit | |
| | | Area for displaying Renting Fees | Initial | Displaying Renting Fees | |
| | | Close button | Click | Display Home Screen | |

Defining the field attributes

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

| | | | | |
|-----------------------|----|--------|--------|-----------------|
| Start time | 30 | String | Black | Left justified |
| Battery | 4 | String | Black | Left justified |
| Usage | 25 | String | Black | Left justified |
| Session length | 8 | String | Black | Left justified |
| Charge | 15 | String | Green | Right justified |
| Renting fees | 15 | String | Purple | Right justified |

4.1.3.6 Invoice Screen

| EBR Software | | Date of creation | Approved by | Reviewed by | Person in charge |
|----------------------|----------------|---|-------------|---|-------------------|
| Screen specification | Invoice Screen | 7/11/2020 | | | Nguyễn Mạnh Khang |
| | | Control | Operation | Function | |
| | | Return Home Button | Click | Display Payment Screen | |
| | | Area for displaying the Session Information | Initial | Displaying the Session Information and Fees | |
| | | Area for displaying Returned | Initial | Displaying Returned money | |

Defining the field attributes

| | | | | |
|-------------|--------------------------|--------|-----------------|----------------|
| Screen name | Invoice Screen | | | |
| Item name | Number of digits (bytes) | Type | Field attribute | Remarks |
| Card number | 20 | String | Black | Left justified |

| | | | | |
|----------------------|----|---------|-------|-----------------|
| Card Owner | 20 | String | Black | Left justified |
| Exp Date | 5 | Numeric | Black | Left justified |
| Security Code | 4 | Numeric | Black | Left justified |
| Start time | 30 | String | Black | Left justified |
| End time | 30 | String | Black | Left justified |
| Total fees | 30 | String | Green | Left justified |
| Deposit | 30 | String | Green | Left justified |
| Returned | 30 | String | Green | Right justified |

4.1.3.7 Splash Screen

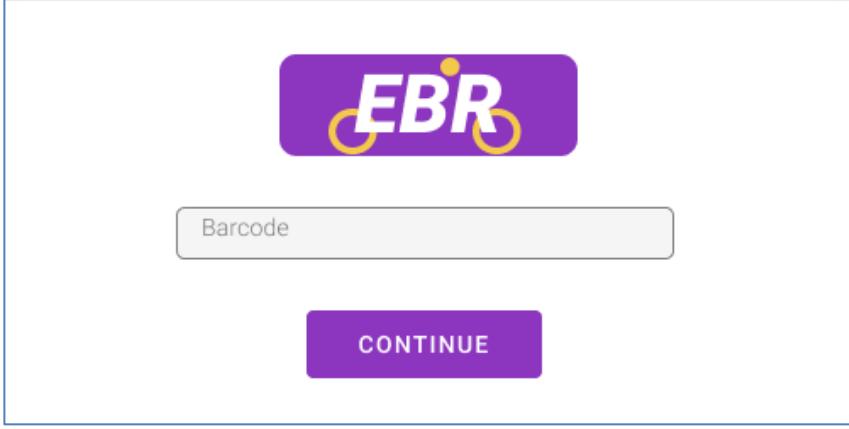
| EBR Software | | Date of creation | Approved by | Reviewed by | Person in charge |
|--|---------------|-------------------------|--------------------|---------------------|-------------------------|
| Screen specification | Splash Screen | 7/11/2020 | | | Vũ Minh Hoàng |
| | | Control | Operation | Function | |
|  LET'S RIDE | | Home Screen Button | Click | Display Home Screen | |
| | | | | | |

Defining the field attributes

| | |
|--------------------|-------------------------|
| Screen name | Dock View Screen |
|--------------------|-------------------------|

| Item name | Number of digits (bytes) | Type | Field attribute | Remarks |
|-----------|--------------------------|------|-----------------|---------|
|-----------|--------------------------|------|-----------------|---------|

4.1.3.8 Barcode Popup Screen

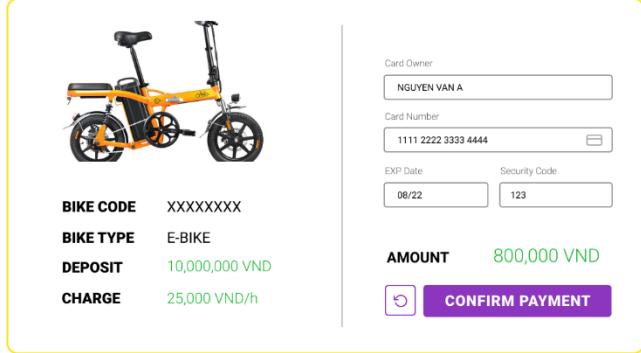
| EBR Software | | Date of creation | Approved by | Reviewed by | Person in charge |
|---|--|------------------|---|-------------|------------------|
| Screen specification | | 7/11/2020 | | | Vũ Minh Hoàng |
|  | | Control | Operation | Function | |
| Area for Entering Barcode | | Initial | Display a text holder for entering barcode for renting bike | | |
| Confirm Button | | Click | Navigate to the previous screen. | | |

Defining the field attributes

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

4.1.3.9 Payment Confirmation Screen

| EBR Software | Date of creation | Approved by | Reviewed by | Person in charge |
|--------------|------------------|-------------|-------------|------------------|
|--------------|------------------|-------------|-------------|------------------|

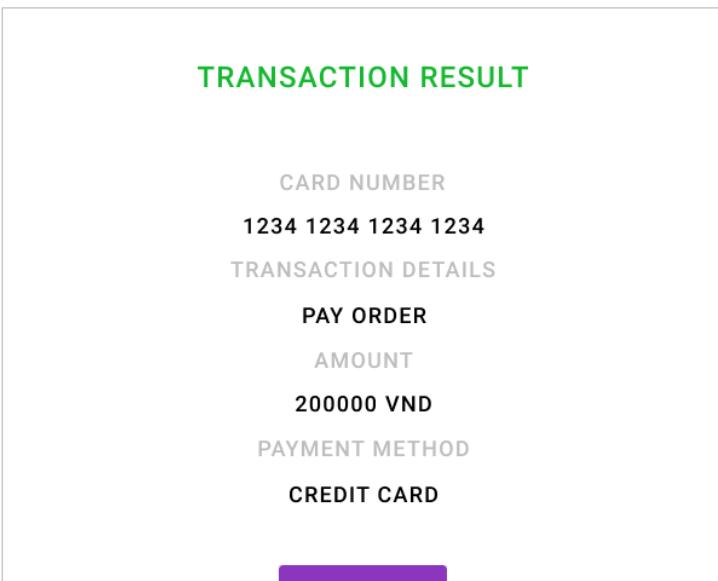
| | | | | |
|----------------------|---|--------------------------------------|-----------|--|
| Screen specification | Payment Confirmation | 7/11/2020 | | Vũ Minh Hoàng |
| | | Control | Operation | Function |
| |  | Enter Barcode Button | Click | Display Barcode Popup Screen |
| | | Back Button | Click | Back to the Payment Screen |
| | | Return Dock Button | Click | Return to the Dock Screen |
| | | EBR Home Screen Button | Click | Display Home Screen |
| | | Area for display Payment Information | Initial | Display the Bike Code, Card Number and Total money |
| | | Confirm Payment Button | Click | Display Renting Session |

Defining the field attributes

| Screen name | Bike View Screen | | | |
|-------------|--------------------------|--------|-----------------|----------------|
| Item name | Number of digits (bytes) | Type | Field attribute | Remarks |
| Bike Code | 8 | String | Black | Left justified |
| Bike Type | 30 | String | Black | Left justified |
| Card number | 20 | String | Black | Left justified |
| Card Owner | 20 | String | Black | Left justified |

| | | | | |
|----------------------|----|---------|-------|-----------------|
| Exp Date | 5 | Numeric | Black | Left justified |
| Security Code | 4 | Numeric | Black | Left justified |
| Deposit | 15 | String | Green | Left justified |
| Charge | 15 | String | Green | Left justified |
| Amount | 15 | String | Green | Right justified |

4.1.3.10 Transaction Result Popup Screen

| EBR Software | | Date of creation | Approved by | Reviewed by | Person in charge |
|----------------------|---|--|-------------|---|------------------|
| Screen specification | Transaction Result | 26/12/2020 | | | Vũ Minh Hoàng |
| | | Control | Operation | Function | |
| | | Close Button | Click | Close Popup | |
| | | Area for display transaction Information | Initial | Display transaction result and transaction info | |
| |  | | | | |

Defining the field attributes

| | | | | |
|----------------------------|--------------------------|--------|-----------------|------------------|
| Screen name | Bike View Screen | | | |
| Item name | Number of digits (bytes) | Type | Field attribute | Remarks |
| Transaction result | 30 | String | Green/Red | Center justified |
| Transaction Details | 30 | String | Black | Center justified |
| Amount | 20 | String | Black | Center justified |
| Payment method | 20 | String | Black | Center justified |
| Card Number | 20 | String | Black | Center justified |

4.1.3.11 Alert Popup Screen

| EBR Software | | Date of creation | Approved by | Reviewed by | Person in charge |
|--|-------------|---------------------------------|-------------|-------------------------------|------------------|
| Screen specification | Alert Popup | 26/12/2020 | | | Vũ Minh Hoàng |
|  ERROR MESSAGE | | Control | Operation | Function | |
| | | Area for Displaying Err Message | Initial | Display a text of err message | |

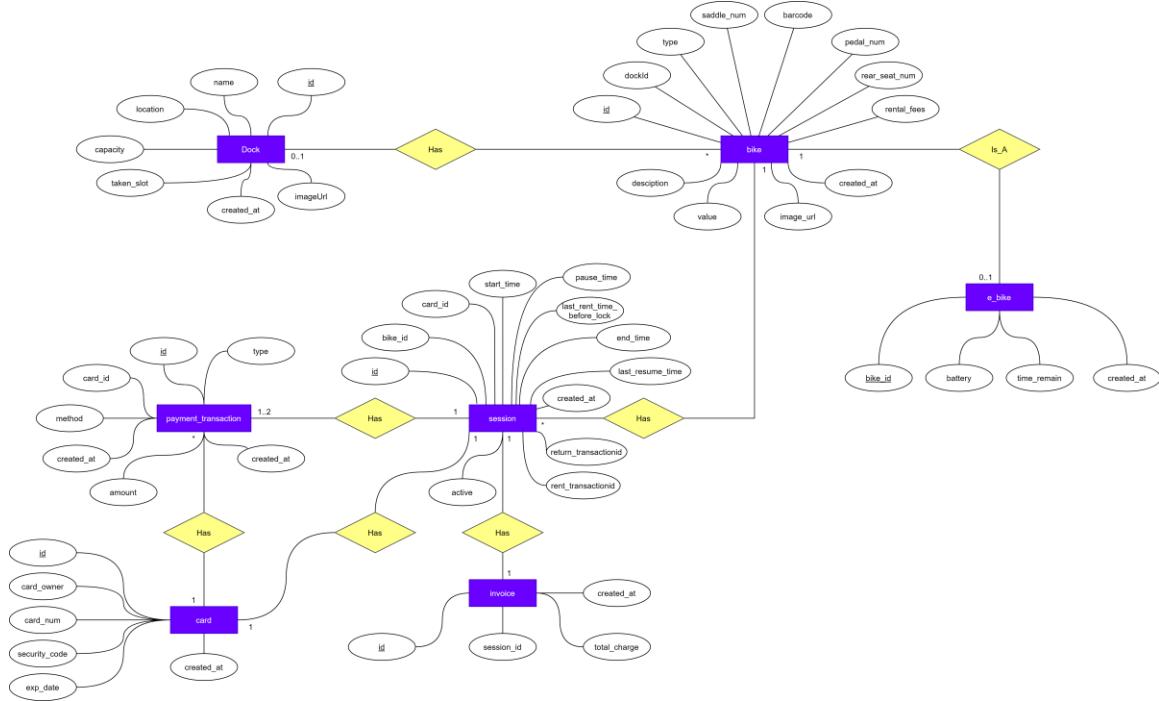
Defining the field attributes

| | | | | |
|--------------------|--------------------------|------|-----------------|---------|
| Screen name | Bike View Screen | | | |
| Item name | Number of digits (bytes) | Type | Field attribute | Remarks |

| | | | | |
|----------|-----|--------|-------|----------------|
| Err Text | 100 | String | Black | Left justified |
|----------|-----|--------|-------|----------------|

4.2 Data Modeling

4.2.1 Conceptual Data Modeling



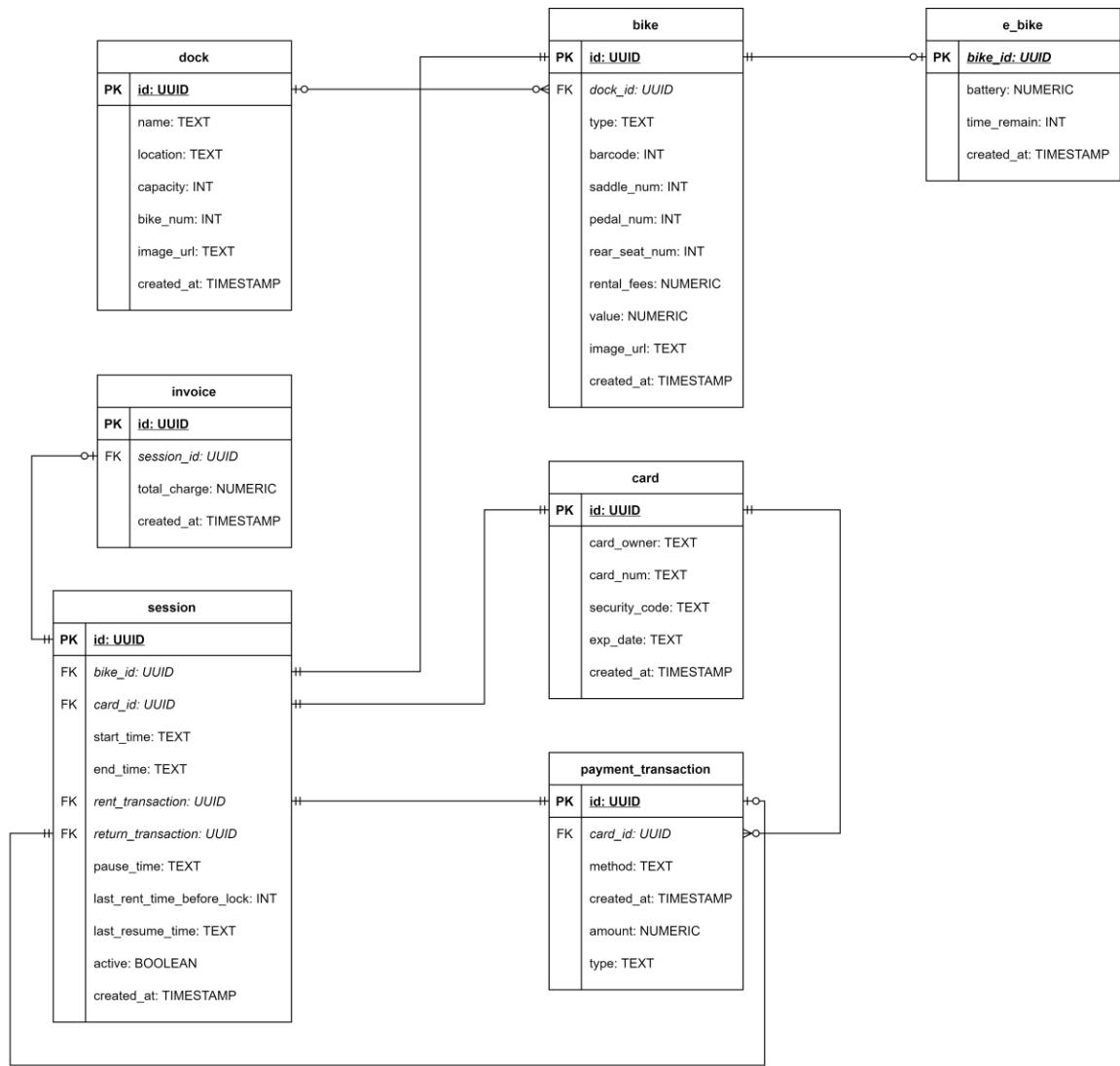
4.2.2 Database Design

4.2.2.1 Database Management Systems

Database Management System: PostgreSQL

PostgreSQL is a powerful, open-source object-relational database system with over 30 years of active development that has earned it a strong reputation for reliability, feature robustness, and performance.

4.2.2.2 Logical Data Model



4.2.2.3 Physical Data Model

bike

| # | PK | FK | Column Name | Data Type | Mandatory | Description |
|---|----|----|-------------|-----------|-----------|---|
| 1 | X | | id | UUID | Yes | UUID |
| 2 | | | type | INT | Yes | Bike type, e.g., Standard Bike, E-Bike, ... |
| 3 | | | barcode | INT | Yes | Barcode for renting |
| 4 | | | saddle_num | INT | Yes | Number of saddles |
| 5 | | | pedal_num | INT | Yes | Number of pedals |

| | | | | | | |
|-----------|--|---|---------------|--------------------------------|-----|----------------------|
| 6 | | | rear_seat_num | INT | Yes | Number of rear seats |
| 7 | | | value | INT | Yes | Value of the bike |
| 8 | | | rental_fees | INT | Yes | Rental fees |
| 9 | | X | dock_id | UUID | No | Dock ID |
| 10 | | | image_url | TEXT | Yes | URL of image of bike |
| 11 | | | description | TEXT | No | |
| 12 | | | created_at | TIMESTAMP WITH TIME ZONE | No | Created time |

e_bike

| # | PK | FK | Column Name | Data Type | Mandatory | Description |
|----------|----|----|-------------|--------------------------------|-----------|--|
| 1 | | X | bike_id | UUID | Yes | ID, same as ID of Bike of which type is E-Bike |
| 2 | | | battery | NUMERIC | Yes | Battery left percentage |
| 3 | | | time_remain | INT | Yes | Time remaining |
| 4 | | | created_at | TIMESTAMP WITH TIME ZONE | No | Created time |

dock

| # | PK | FK | Column Name | Data Type | Mandatory | Description |
|----------|----|----|-------------|-----------|-----------|----------------------------|
| 1 | X | | id | UUID | Yes | UUID |
| 2 | | | name | TEXT | Yes | Name of the dock |
| 3 | | | location | TEXT | Yes | Address of the dock |
| 4 | | | capacity | INT | Yes | Total parking slot |
| 5 | | | bike_num | INT | Yes | Number of bike in the dock |
| 6 | | | image_url | TEXT | Yes | URL of image of dock |

| | | | | | | |
|---|--|--|------------|--------------------------|-----|----------------------|
| 7 | | | created_at | TIMESTAMP WITH TIME ZONE | No | Created time |
| 8 | | | taken_slot | integer | Yes | Taken slot in a dock |

session

| # | PK | FK | Column Name | Data Type | Mandatory | Description |
|----|----|----|----------------------------|--------------------------|-----------|----------------------------|
| 1 | X | | id | UUID | Yes | UUID |
| 2 | | X | card_id | UUID | Yes | Card Id |
| 3 | | X | bike_id | UUID | Yes | Bike Id |
| 4 | | | start_time | TEXT | No | Start time of session |
| 5 | | | end_time | TEXT | No | End time of session |
| 6 | | X | rent_transaction_id | UUID | Yes | Renting transaction ID |
| 7 | | X | return_transaction_id | UUID | No | Returning transaction ID |
| 8 | | | last_resume_time | TEXT | No | Last resume time |
| 9 | | | active | boolean | No | Session state |
| 10 | | | created_at | TIMESTAMP WITH TIME ZONE | No | Created time |
| 11 | | | last_rent_time_before_lock | INT | No | Last rent time before lock |

payment_transaction

| # | PK | FK | Column Name | Data Type | Mandatory | Description |
|---|----|----|-------------|-----------|-----------|-------------|
| 1 | X | | id | UUID | Yes | UUID |

| | | | | | | |
|---|--|---|------------|--------------------------------|-----|--------------------------------------|
| 2 | | X | card_id | UUID | Yes | Card ID |
| 3 | | | type | TEXT | Yes | Type of the transaction |
| 4 | | | create_at | TIMESTAMP | Yes | Time when the transaction is created |
| 5 | | | amount | NUMERIC | Yes | Transaction amount |
| 6 | | | method | TEXT | Yes | Method of payment |
| 7 | | | created_at | TIMESTAMP WITH TIME ZONE | No | Created time |

card

| # | PK | FK | Column Name | Data Type | Mandatory | Description |
|---|----|----|---------------|--------------------------------|-----------|-----------------------------|
| 1 | X | | id | UUID | Yes | UUID |
| 2 | | | card_owner | TEXT | Yes | Full name of the card owner |
| 3 | | | card_num | TEXT | Yes | Card number, hashed |
| 4 | | | security_code | TEXT | Yes | Security code, hashed |
| 5 | | | exp_date | TEXT | Yes | Expiration date |
| 6 | | | created_at | TIMESTAMP WITH TIME ZONE | No | Created time |

invoice

| # | PK | FK | Column Name | Data Type | Mandatory | Description |
|---|----|----|--------------|--------------------------------|-----------|--------------------|
| 1 | X | | id | UUID | Yes | UUID |
| 2 | | X | session_id | UUID | Yes | Session ID |
| 3 | | | total_charge | NUMERIC | No | Total charge money |
| 4 | | | created_at | TIMESTAMP WITH TIME ZONE | No | Created time |

```

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_v
4() 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
        d_generate_v4() NOT NULL,
    card_id                          uuid          NOT NULL,
    bike_id                          uuid          NOT NULL,
    rent_transactionid              uuid          NOT NULL,
    return_transactionid            uuid          NOT NULL,
    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_se-
at_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\Tw-
inBike.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_se-
at_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\Tw-
inBike.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_se-
at_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_se-
at_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 (i
d) 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);

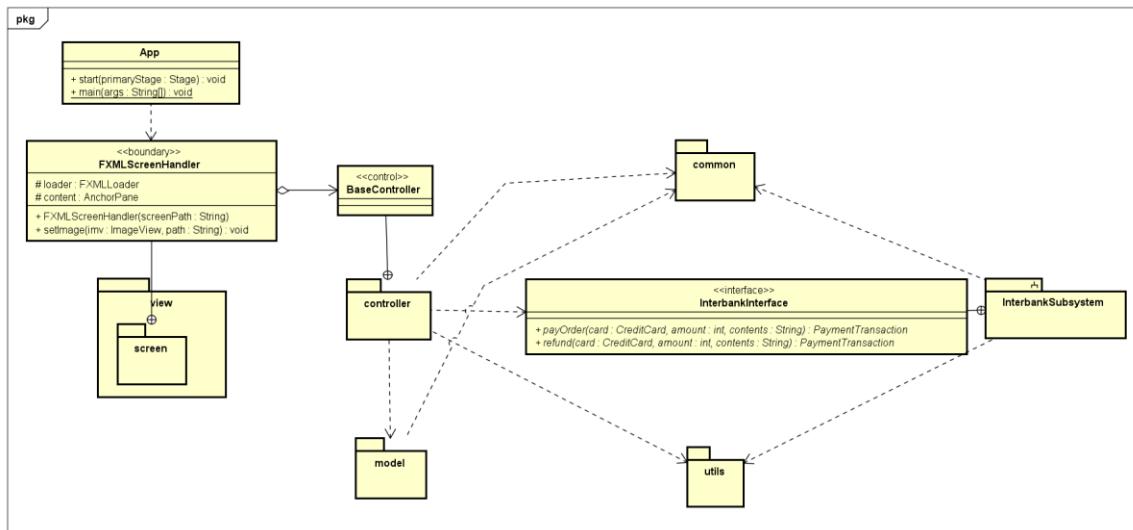
ALTER TABLE ONLY public.session
    ADD CONSTRAINT session_fk4 FOREIGN KEY (return_transactionid) REFERENCES public.payment_transaction (id);

```

4.3 Non-Database Management System Files

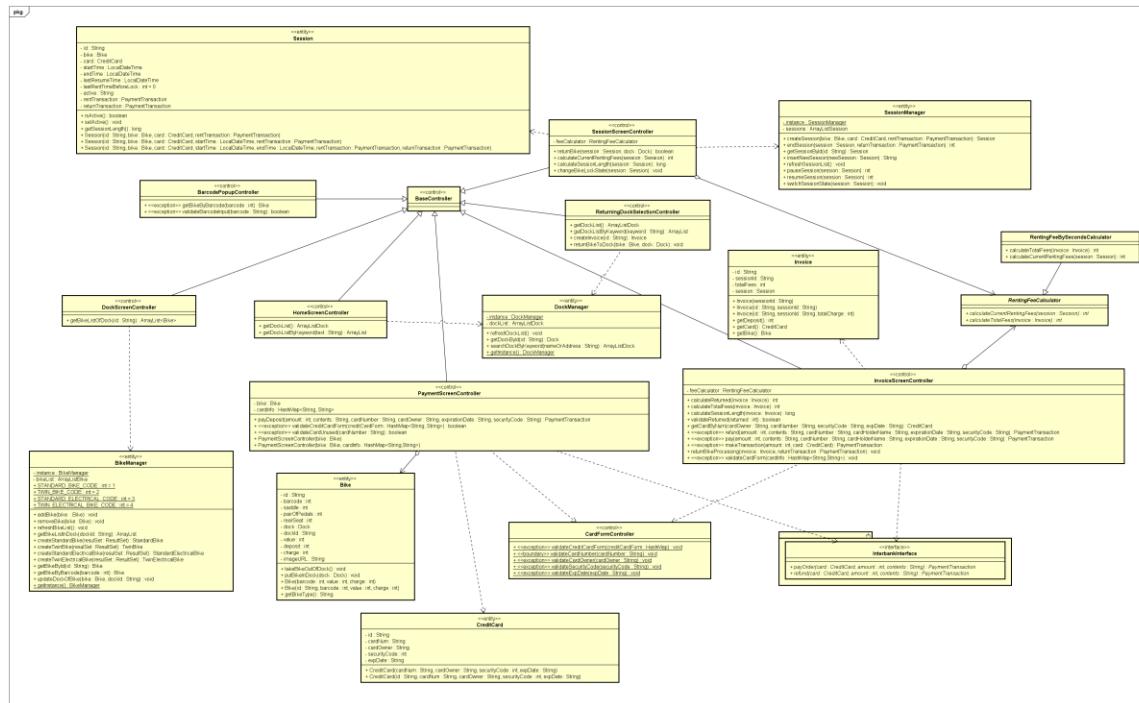
4.4 Class Design

4.4.1 General Class Diagram

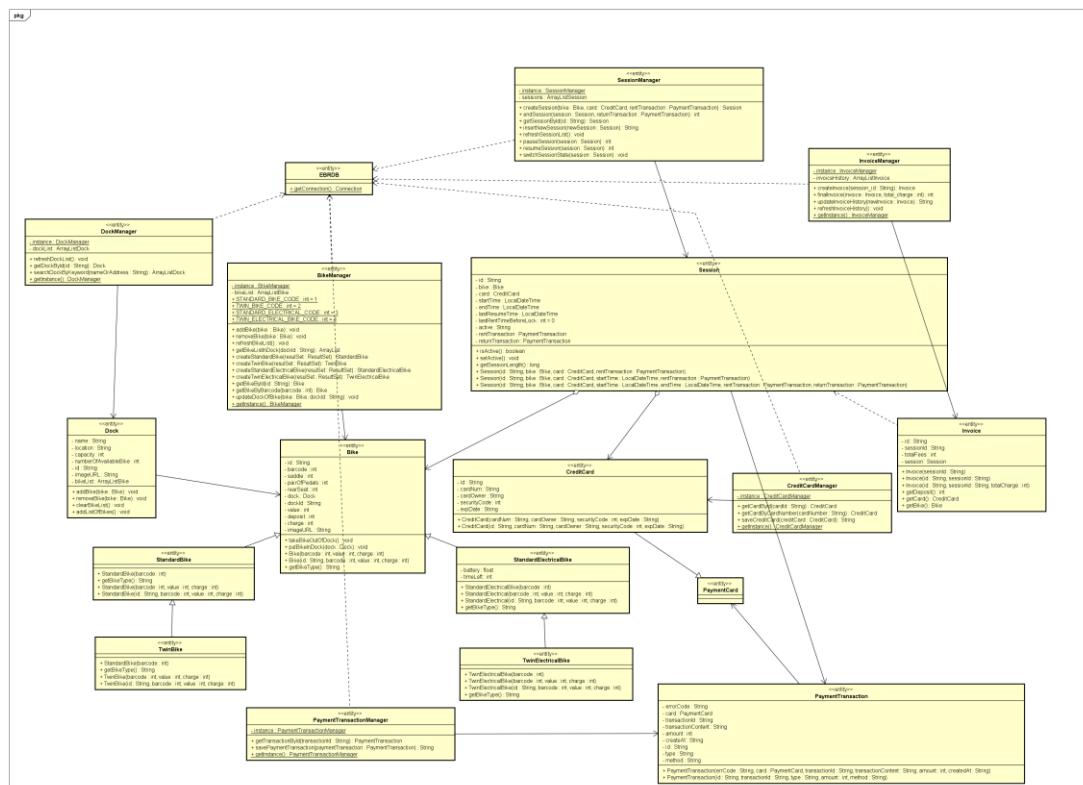


4.4.2 Class Diagrams

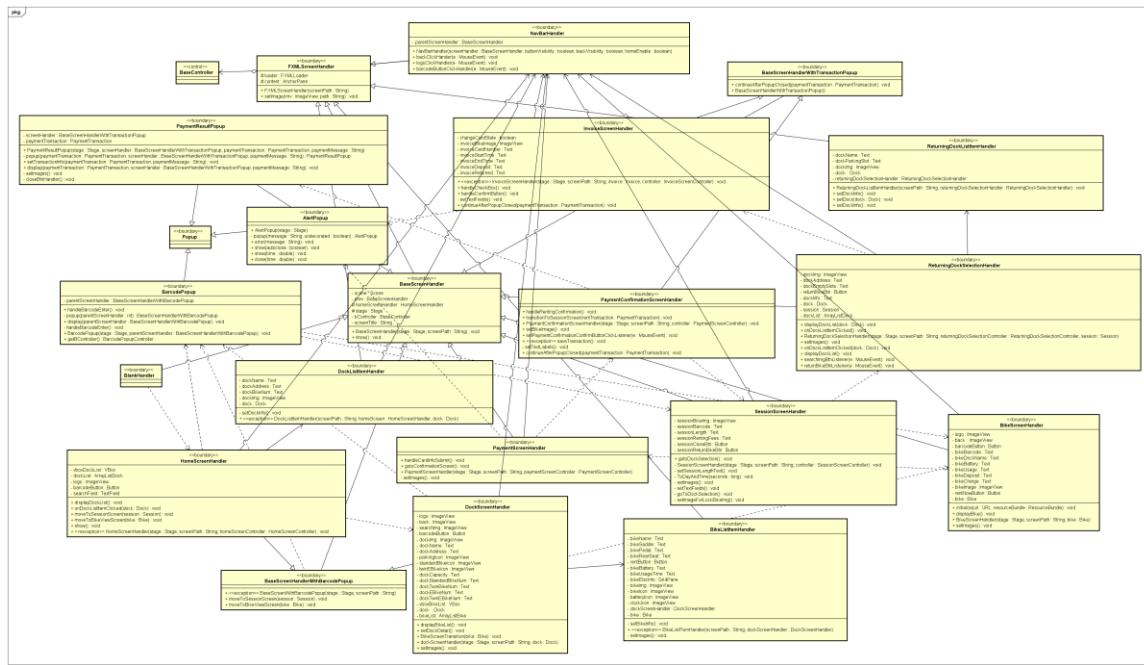
4.4.2.1 Class Diagram for Package “controller”



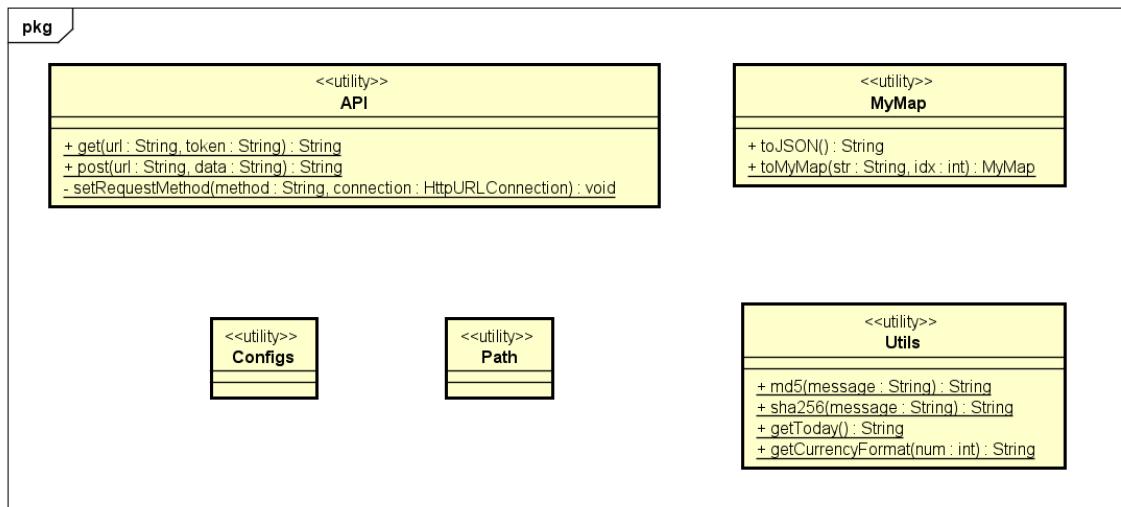
4.4.2.2 Class Diagram for Package “model”



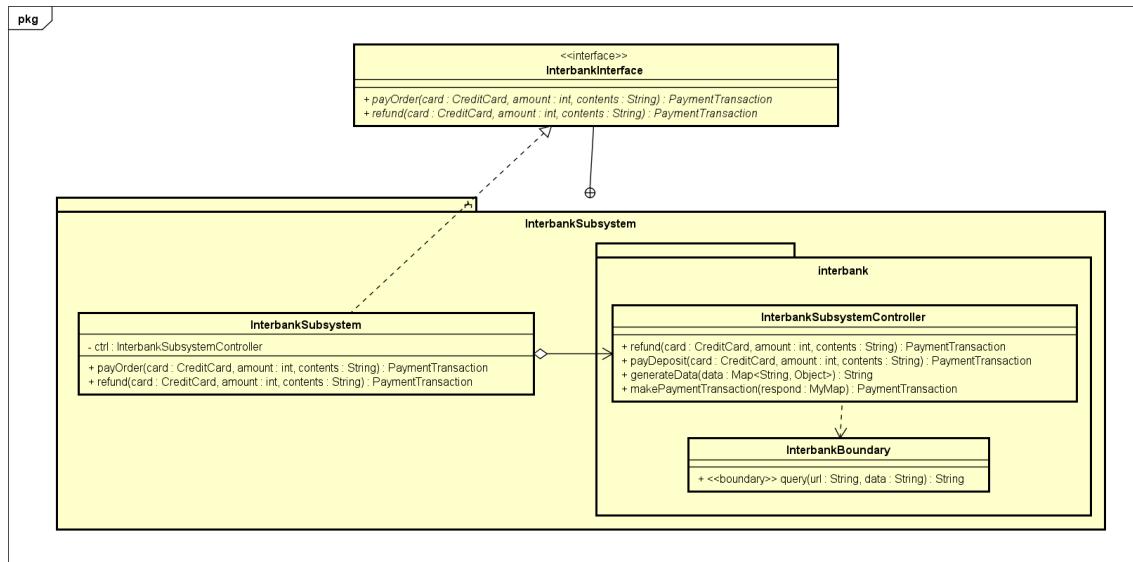
4.4.2.3 Class Diagram for Package “view”



4.4.2.4 Class Diagram for Package “utils”

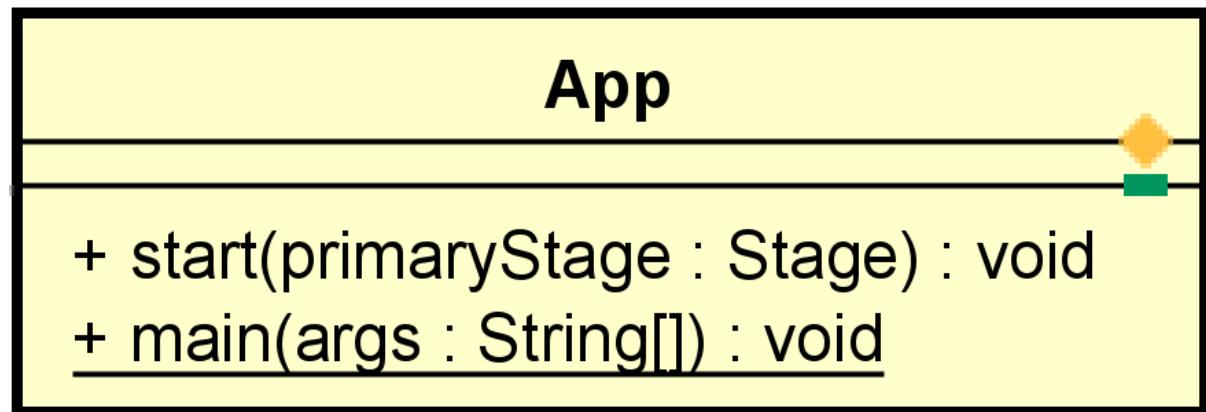


4.4.2.5 Class Diagram for Subsystem “interbank”



4.4.3 Class Design

4.4.3.1 Class “App”



Attribute

None

Operation

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

Parameter:

stage: javafx main stage

args: environment variable

Exception:

None

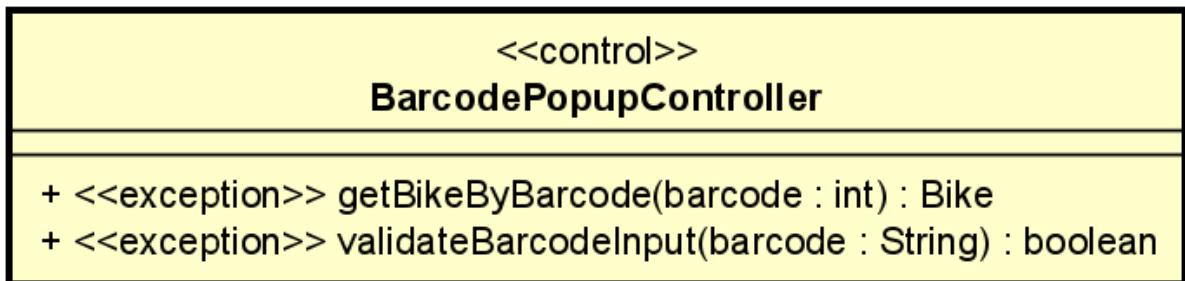
Method

None

State

None

4.4.3.2 Class “BarcodePopupController”



Attribute

None

Operation

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

Parameter:

barcode— int

Barcode - string

Exception:

- BarcodeNotFoundException
- NullBarcodeException
- InvalidBarcodeException

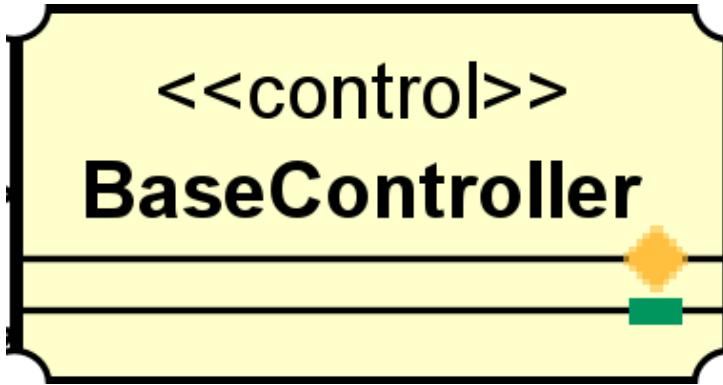
Method

None

State

None

4.4.3.3 Class “BaseController”



Attribute

None

Operation

None

Parameter:

None

Exception:

None

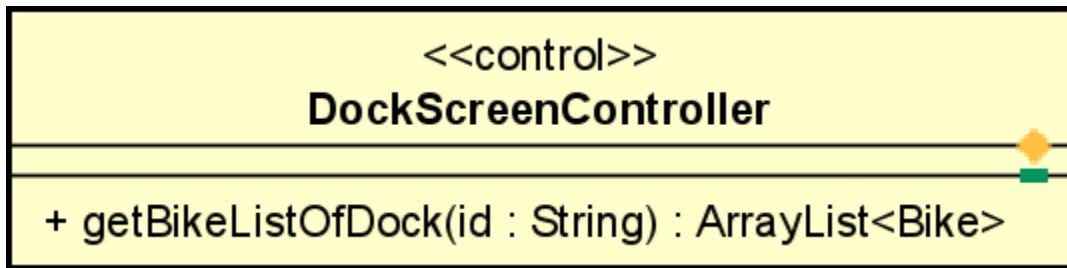
Method

None

State

None

4.4.3.4 Class “DockScreenController”



Attribute

None

Operation

| # | Name | Return type | Description |
|---|-------------------|-------------|--------------------------------|
| 1 | getBikeListOfDock | ArrayList | Get bike list by using dock id |

Parameter:

Id – dock id

Exception:

None

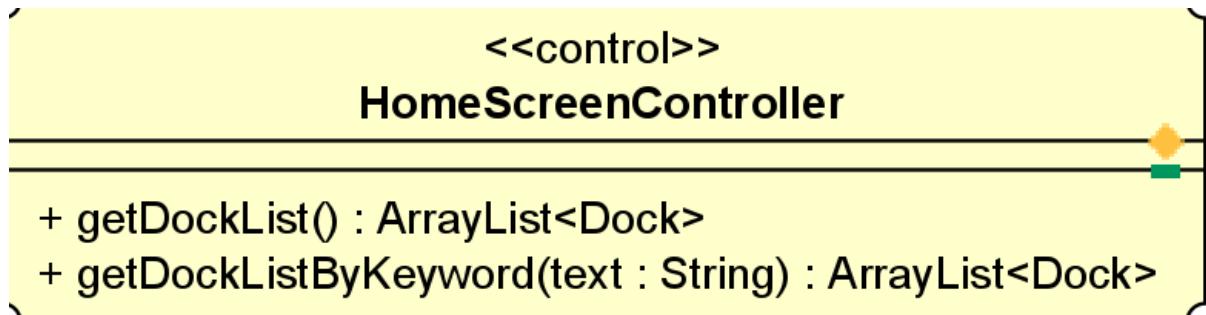
Method

None

State

None

4.4.3.5 Class “HomeScreenController”



Attribute

None

Operation

| # | Name | Return type | Description |
|---|----------------------|-------------|--------------------------|
| 1 | getDockList | ArrayList | Get dock list |
| 2 | getDockListByKeyword | ArrayList | Get dock list by keyword |

Parameter:

Text - keyword

Exception:

None

Method

None

State

None

4.4.3.6 Class “InvoiceScreenController”

| <<control>> InvoiceScreenController | |
|--|--|
| - feeCalculator : RentingFeeCalculator | |
| + calculateReturned(invoice : Invoice) : int | |
| + calculateTotalFees(invoice : Invoice) : int | |
| + calculateSessionLength(invoice : Invoice) : long | |
| + validateReturned(returned : int) : boolean | |
| + getCardByNum(cardOwner : String, cardNumber : String, securityCode : String, expDate : String) : CreditCard | |
| + <<exception>> refund(amount : int, contents : String, cardNumber : String, cardHolderName : String, expirationDate : String, securityCode : String) : PaymentTransaction | |
| + <<exception>> pay(amount : int, contents : String, cardNumber : String, cardHolderName : String, expirationDate : String, securityCode : String) : PaymentTransaction | |
| + <<exception>> makeTransaction(amount : int, card : CreditCard) : PaymentTransaction | |
| + returnBikeProcessing(invoice : Invoice, returnTransaction : PaymentTransaction) : void | |
| + <<exception>> validateCardForm(cardInfo : HashMap<String, String>) : void | |

Attribute

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

Operation

| # | Name | Return type | Description |
|----|------------------------|--------------------|---|
| 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 | validateCardForm | void | Check card form validity |

Parameter:

Invoice
Amount
Contents
CardNumber
CardOwner
ExpirationDate
SecurityCode
Returned

Exception:

PaymentException
NullCardNumberException
NullCardOwnerException
NullExpDateException
NullSecurityCodeException
CardExpiredException

Method

None

State

None

4.4.3.7 Class “*PaymentScreenController*”

| <<control>> PaymentScreenController | |
|---|--|
| - bike : Bike | |
| - cardInfo : HashMap<String, String> | |
| + payDeposit(amount : int, contents : String, cardNumber : String, cardOwner : String, expirationDate : String, securityCode : String) : PaymentTransaction | |
| + <<exception>> validateCreditCardForm(creditCardForm : HashMap<String, String>) : boolean | |
| + <<exception>> validateCardUnused(cardNumber : String) : boolean | |
| + PaymentScreenController(bike : Bike) | |
| + PaymentScreenController(bike : Bike, cardInfo : HashMap<String, String>) | |

Attribute

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

| | | | | |
|----------|----------|-------------------------|------|-----------|
| 1 | bike | Bike | NULL | bike |
| 2 | cardInfo | HashMap<String, String> | NULL | Card info |

Operation

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

Parameter:

- Amount
- Contents
- CardNumber
- CreditCardForm
- CardInfo
- Bike

Exception:

- None

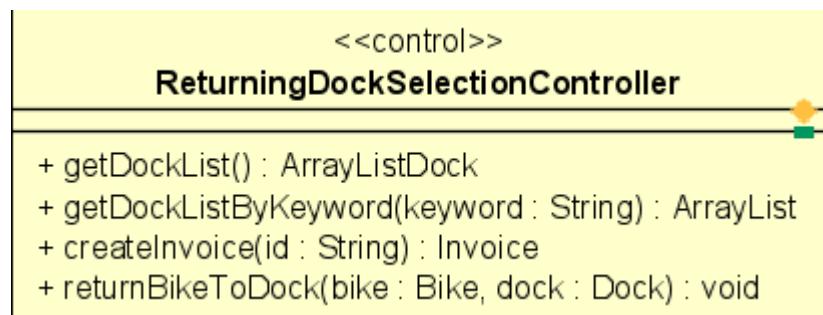
Method

- None

State

- None

4.4.3.8 Class “Class “ReturningDockSelectionController”



Attribute

None

Operation

| # | Name | Return type | Description |
|---|----------------------|-----------------|--------------------------------|
| 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:

Keyword

Id

Bike

Dock

Exception:

None

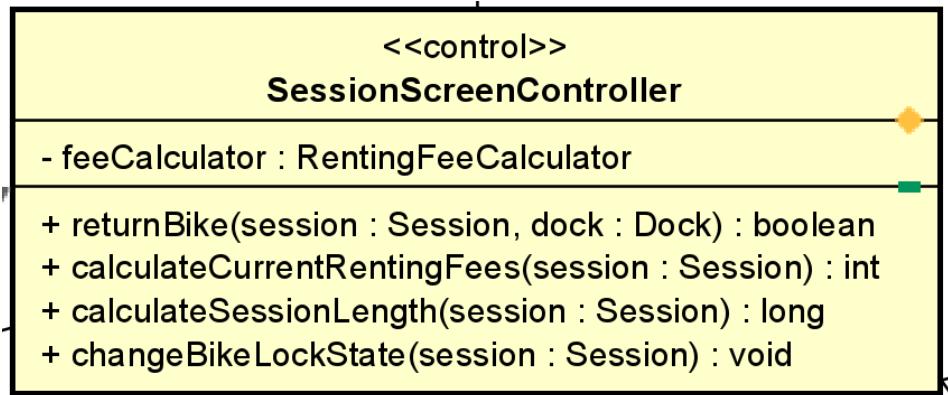
Method

None

State

None

4.4.3.9 Class “SessionScreenController”



Attribute

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

Operation

| # | Name | Return type | Description |
|---|-----------------------------|-------------|--------------------------------|
| 1 | returnBike | boolean | Check if bike is available |
| 2 | calculateCurrentRentingFees | int | Calculate current renting fees |
| 3 | calculateSessionLength | long | Calculate session length |
| 4 | changeBikeLockState | void | Change bike lock state |

Parameter:

Session

Dock

Exception:

None

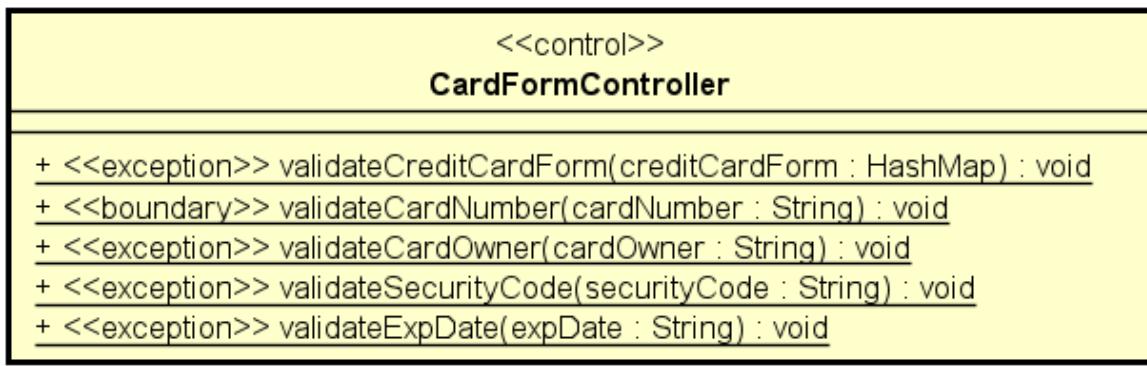
Method

None

State

None

4.4.3.10 Class “CardFormController”



Attribute

None

Operation

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

Parameter:

CreditCardForm

CardNumber

CardOwner
SecurityCode
expDate

Exception:

None

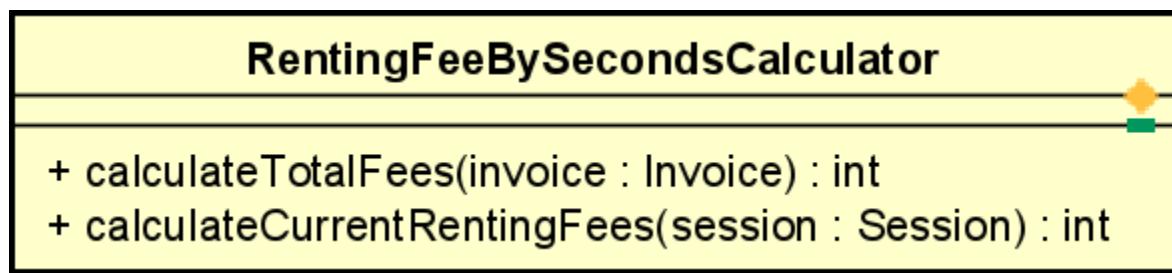
Method

None

State

None

4.4.3.11 Class “*RentingFeeBySecondsCalculator*”



Attribute

None

Operation

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

Parameter:

Invoice
session

Exception:

None

Method

None

State

None

4.4.3.12 Class “*RentingFeeCalculator*”

RentingFeeCalculator

+ *calculateCurrentRentingFees(session : Session) : int*
+ *calculateTotalFees(invoice : Invoice) : int*

Attribute

None

Operation

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

Parameter:

Invoice

session

Exception:

None

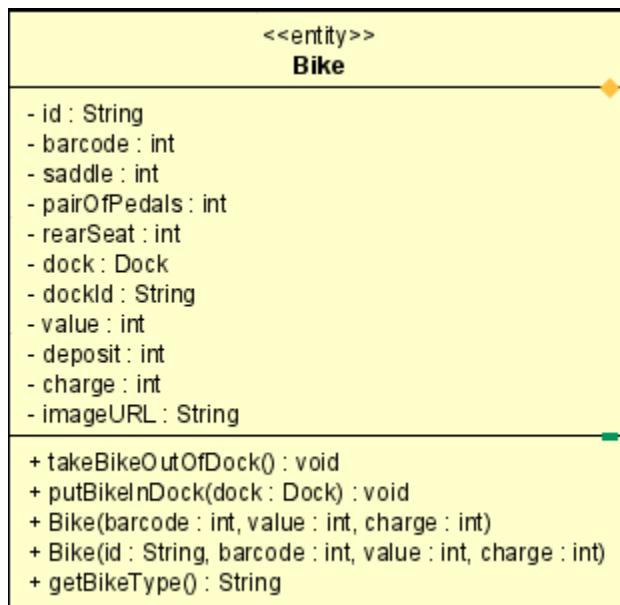
Method

None

State

None

4.4.3.13 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 | deposit | int | NULL | deposit |
| 10 | charge | int | NULL | charge |
| 11 | imageURL | String | NULL | Image URL |

Operation

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

Parameter:

dock

Barcode

Value

Charge

id

Exception:

None

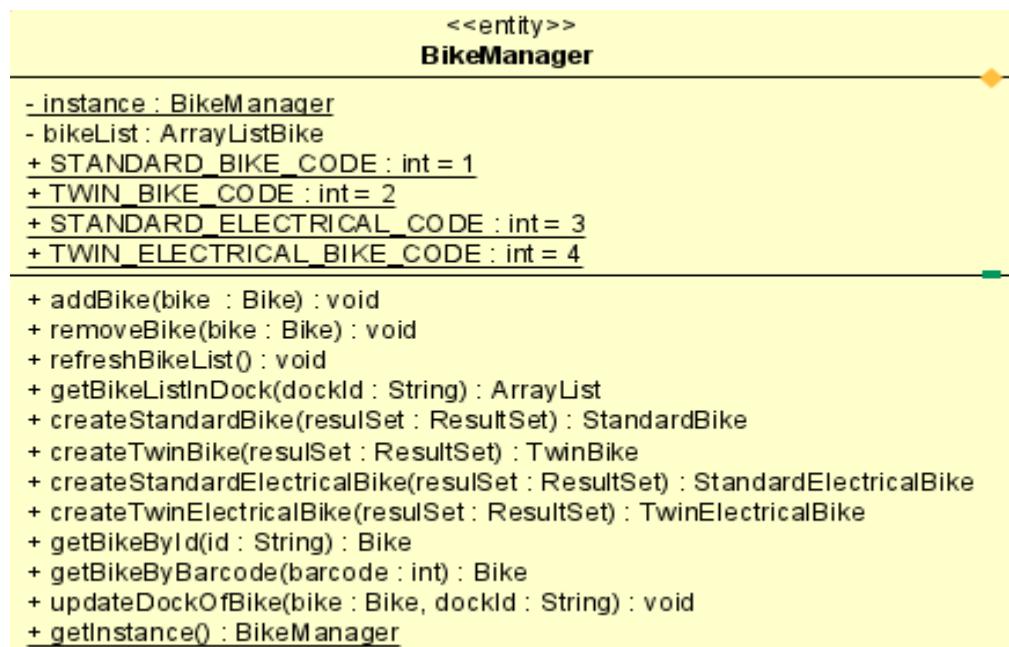
Method

None

State

None

4.4.3.14 Class “BikeManager”



Attribute

| # | Name | Data type | Default value | Description |
|---|-------------------------------|-------------|---------------|-------------------------|
| 1 | instance | BikeManager | NULL | Bike instance |
| 2 | bikeList | ArrayList | NULL | Bike list |
| 3 | STANDARD BIKE CODE | int | 1 | Code of standard bike |
| 4 | TWIN BIKE CODE | int | 2 | Code of twin bike |
| 5 | STANDARD ELECTRICAL BIKE CODE | int | 3 | Code of standard e bike |
| 6 | TWIN ELECTRICAL BIKE CODE | int | 4 | Code of twin e bike |

Operation

| # | Name | Return type | Description |
|-----------|---------------------|---------------|------------------------|
| 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:

bike

DockId

ResultSet

Id – bikeId

barcode

Exception:

None

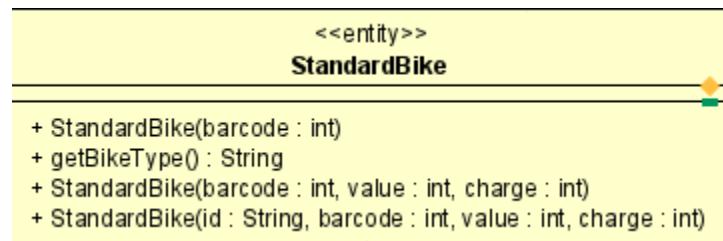
Method

None

State

None

4.4.3.15 Class “*StandardBike*”



Attribute

None

Operation

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

Parameter:

None

Exception:

None

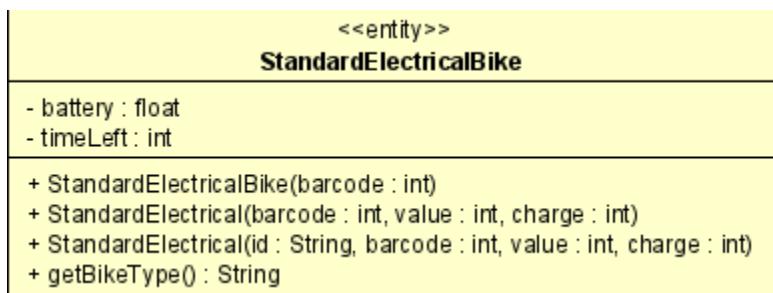
Method

None

State

None

4.4.3.16 Class “*StandardElectricalBike*”



Attribute

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

Operation

None

Parameter:

None

Exception:

None

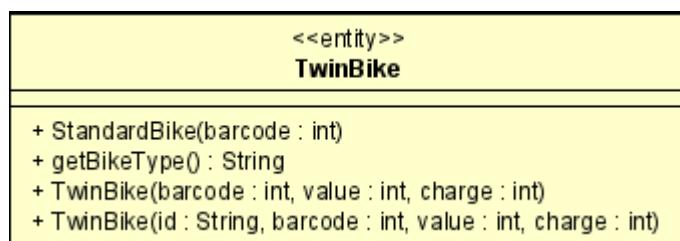
Method

None

State

None

4.4.3.17 Class "TwinBike"



Attribute

None

Operation

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

Parameter:

None

Exception:

None

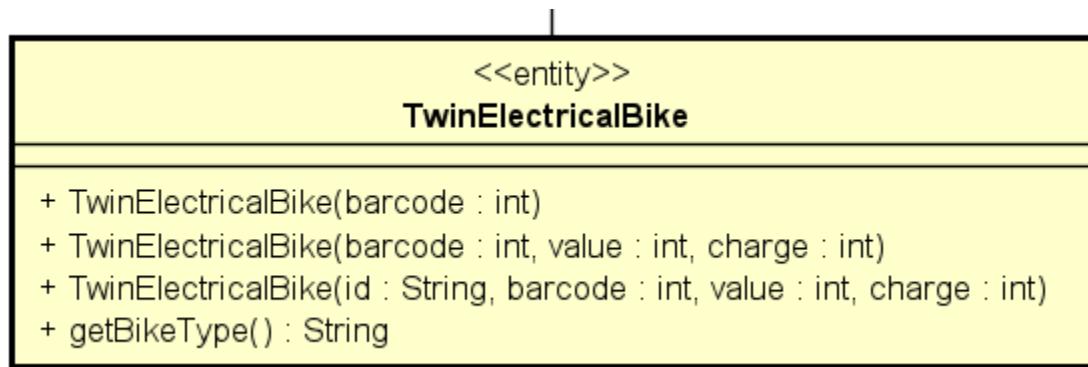
Method

None

State

None

4.4.3.18 Class “TwinElectricalBike”



Attribute

None

Operation

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

Parameter:

None

Exception:

None

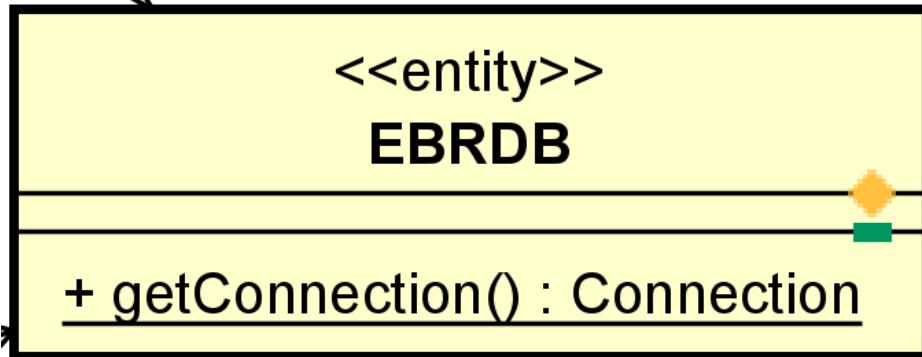
Method

None

State

None

4.4.3.19 Class “EBRDB”



Attribute

None

Operation

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

Parameter:

None

Exception:

None

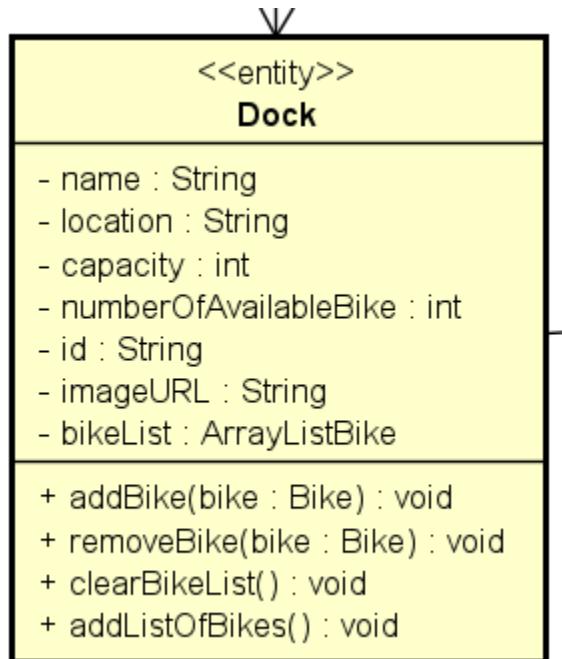
Method

None

State

None

4.4.3.20 Class "Dock"



3

Attribute

| # | Name | Data type | Default value | Description |
|---|-----------------------|-----------------|---------------|---------------------------|
| 1 | name | String | NULL | Name of dock |
| 2 | location | String | NULL | Dock location |
| 3 | capacity | int | NULL | Dock size |
| 4 | numberOfAvailableBike | int | NULL | Number of available bikes |
| 5 | id | String | NULL | Dock id |
| 6 | imageURL | String | NULL | Dock image |
| 7 | bikeList | ArrayList<Bike> | NULL | Bike list in dock |

Operation

| # | Name | Return type | Description |
|---|---------|-------------|------------------|
| 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:

bike

Exception:

None

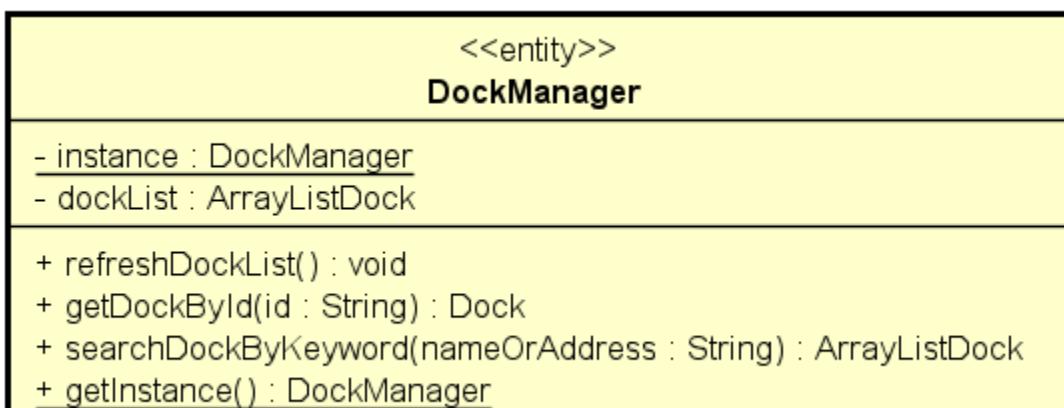
Method

None

State

None

4.4.3.21 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 | Description |
|----------|---------------------|-----------------|----------------------|
| 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:

Id - dock id

NameOfAddress – dock address

Exception:

None

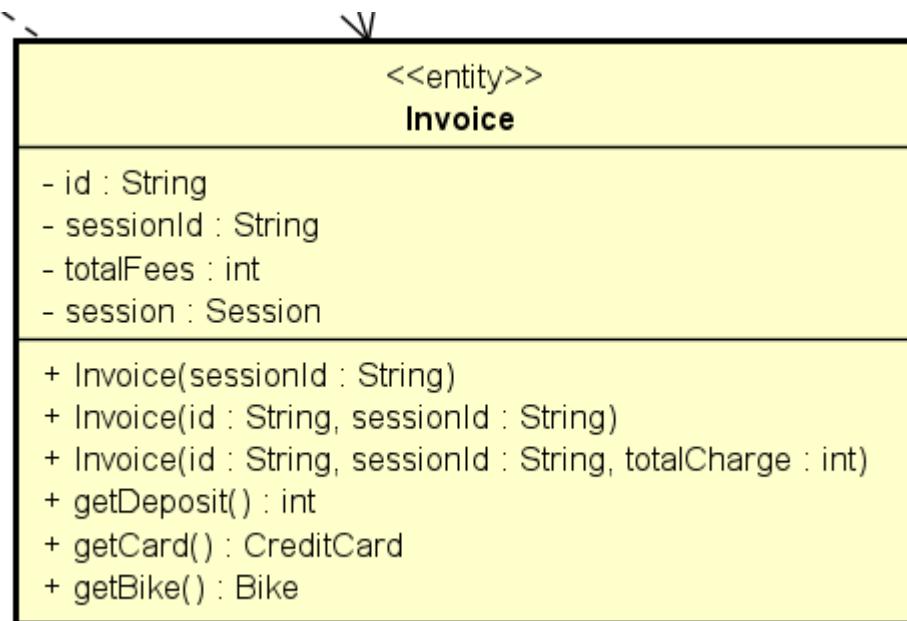
Method

None

State

None

4.4.3.22 Class “Invoice”



Attribute

| # | Name | Data type | Default value | Description |
|---|-----------|-----------|---------------|-------------------------------|
| 1 | id | String | NULL | Invoice id |
| 2 | sessionId | String | NULL | Session id |
| 3 | totalFees | int | NULL | Total charge |
| 4 | session | Session | NULL | Session correspond to invoice |

Operation

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

Parameter:

None

Exception:

None

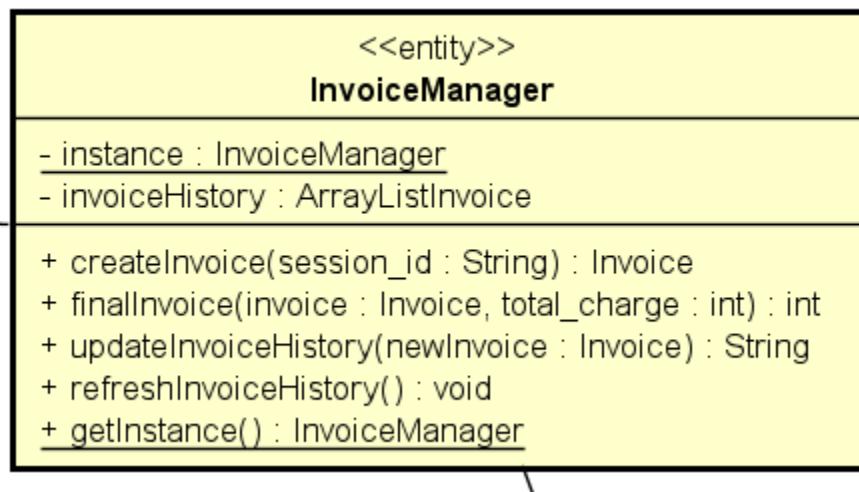
Method

None

State

None

4.4.3.23 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 | Description |
|----------|----------------------|-------------|------------------------|
| 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:

- Session_id
- Invoice
- Total_charge
- newInvoice

Exception:

- None

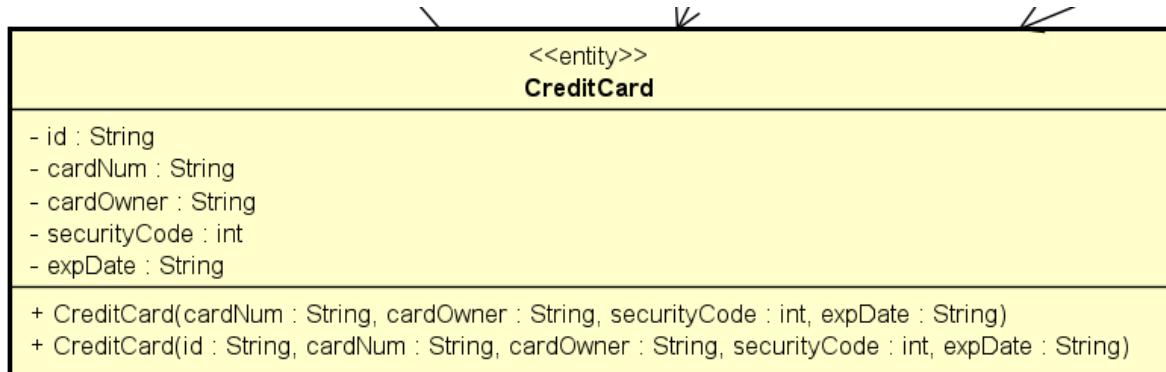
Method

- None

State

- None

4.4.3.24 Class “CreditCard”



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 | int | NULL | Card security code |
| 5 | expDate | String | NULL | Card exp date |

Operation

None

Parameter:

None

Exception:

None

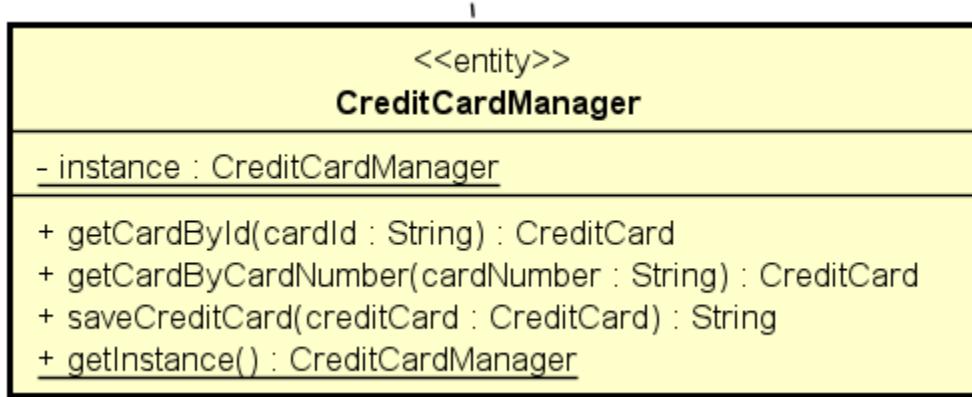
Method

None

State

None

4.4.3.25 Class “CreditCardManager”



Attribute

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

Operation

| # | Name | Return type | Description |
|---|---------------------|-------------------|-------------------------|
| 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:

CardId – card id

CardNumber – card number

creditCard

Exception:

None

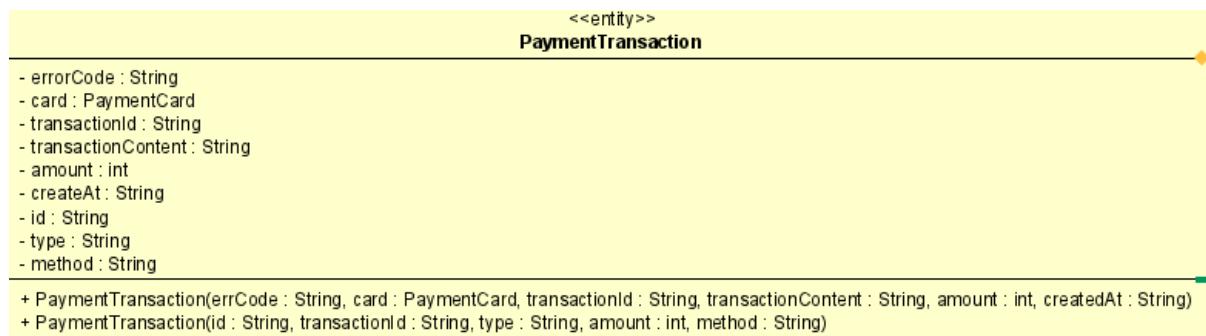
Method

None

State

None

4.4.3.26 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

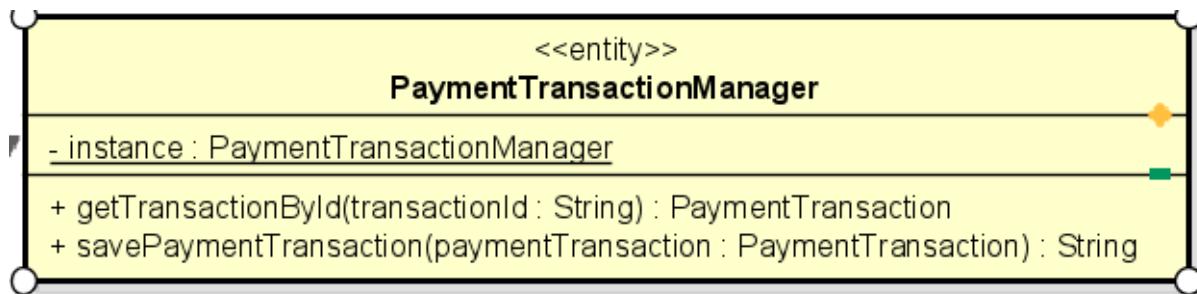
Method

None

State

None

4.4.3.27 Class “*PaymentTransactionManager*”



Attribute

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

Operation

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

Parameter:

 TransactionId – transaction id

 paymentTransaction

Exception:

 None

Method

 None

State

 None

4.4.3.28 Class “Session”

| <pre><<entity>> Session</pre> | |
|--|---|
| <pre>- id : String - bike : Bike - card : CreditCard - startTime : LocalDateTime - endTime : LocalDateTime - lastResumeTime : LocalDateTime - lastRentTimeBeforeLock : int = 0 - active : String - rentTransaction : PaymentTransaction - returnTransaction : PaymentTransaction</pre> | <pre>+ isActive(): boolean + setActive(): void + getSessionLength(): Long + Session(id: String, bike: Bike, card: CreditCard, rentTransaction: PaymentTransaction) + Session(id: String, bike: Bike, card: CreditCard, startTime: LocalDateTime, rentTransaction: PaymentTransaction) + Session(id: String, bike: Bike, card: CreditCard, startTime: LocalDateTime, endTime: LocalDateTime, rentTransaction: PaymentTransaction, returnTransaction: PaymentTransaction)</pre> |

Attribute

| # | Name | Data type | Default value | Description |
|----|------------------------|--------------------|---------------|----------------------------|
| 1 | id | String | NULL | Session id |
| 2 | bike | Bike | NULL | Bike in session |
| 3 | card | CreditCard | NULL | Card in session |
| 4 | startTime | LocalDateTime | NULL | Start time |
| 5 | endTime | LocalDateTime | NULL | End time |
| 6 | lastResumeTime | LocalDateTime | NULL | Last resume time |
| 7 | lastRentTimeBeforeLock | int | 0 | Last rent time before lock |
| 8 | active | String | NULL | Session active or not |
| 9 | rentTransaction | PaymentTransaction | NULL | Rent transaction |
| 10 | returnTransaction | PaymentTransaction | NULL | Return transaction |

Operation

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

Parameter:

None

Exception:

None

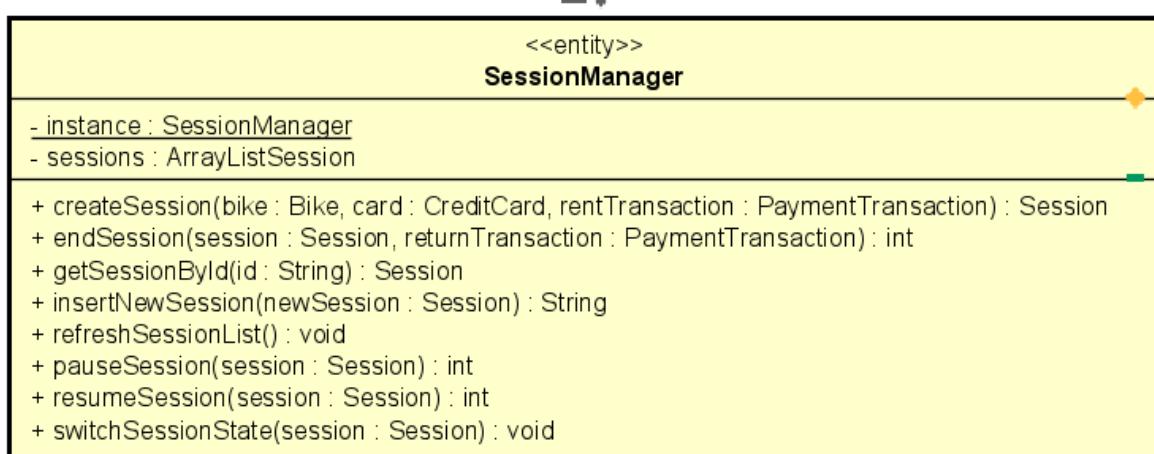
Method

None

State

None

4.4.3.29 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 | Description |
|----------|--------------------|-------------|--------------------------|
| 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 |

Parameter:

Bike

Card

PaymentTransaction

Session

newSession

Exception:

None

Method

None

State

None

4.4.3.30 Class “*BikeScreenHandler*”

| <code><<boundary>></code> BikeScreenHandler | |
|--|--|
| - logo : ImageView | |
| - back : ImageView | |
| - barcodeButton : Button | |
| - bikeBarcode : Text | |
| - bikeDockName : Text | |
| - bikeBattery : Text | |
| - bikeUsage : Text | |
| - bikeDeposit : Text | |
| - bikeCharge : Text | |
| - bikeImage : ImageView | |
| - rentNowButton : Button | |
| - bike : Bike | |
| + initialize(url : URL, resourceBundle : ResourceBundle) : void | |
| + displayBike() : void | |
| + BikeScreenHandler(stage : Stage, screenPath : String, bike : Bike) | |
| + setImages() : void | |

Attribute

| # | Name | Data type | Default value | Description |
|---|---------------|-----------|---------------|--------------------|
| 1 | logo | ImageView | NULL | Logo image fxml |
| 2 | back | imageView | NULL | Back image fxml |
| 3 | barcodeButton | Button | NULL | BarcodeButton fxml |
| 4 | bikeBarcode | Text | NULL | Bike barcode fxml |
| 5 | bikeDockName | Text | NULL | Dock name fxml |
| 6 | bikeBattery | Text | NULL | Bike battery fxml |
| 8 | bikeUsage | Text | NULL | Bike usage fxml |
| 9 | bikeDeposit | Text | NULL | Bike deposit fxml |

| | | | | |
|-----------|---------------|-----------|------|------------------|
| 10 | bikeCharge | Text | NULL | Bike charge fxml |
| 11 | bikeImage | ImageView | NULL | Bike image |
| 12 | rentNowButton | Button | NULL | Rent button fxml |
| 13 | 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:

URL

ResourceBundle

Exception:

None

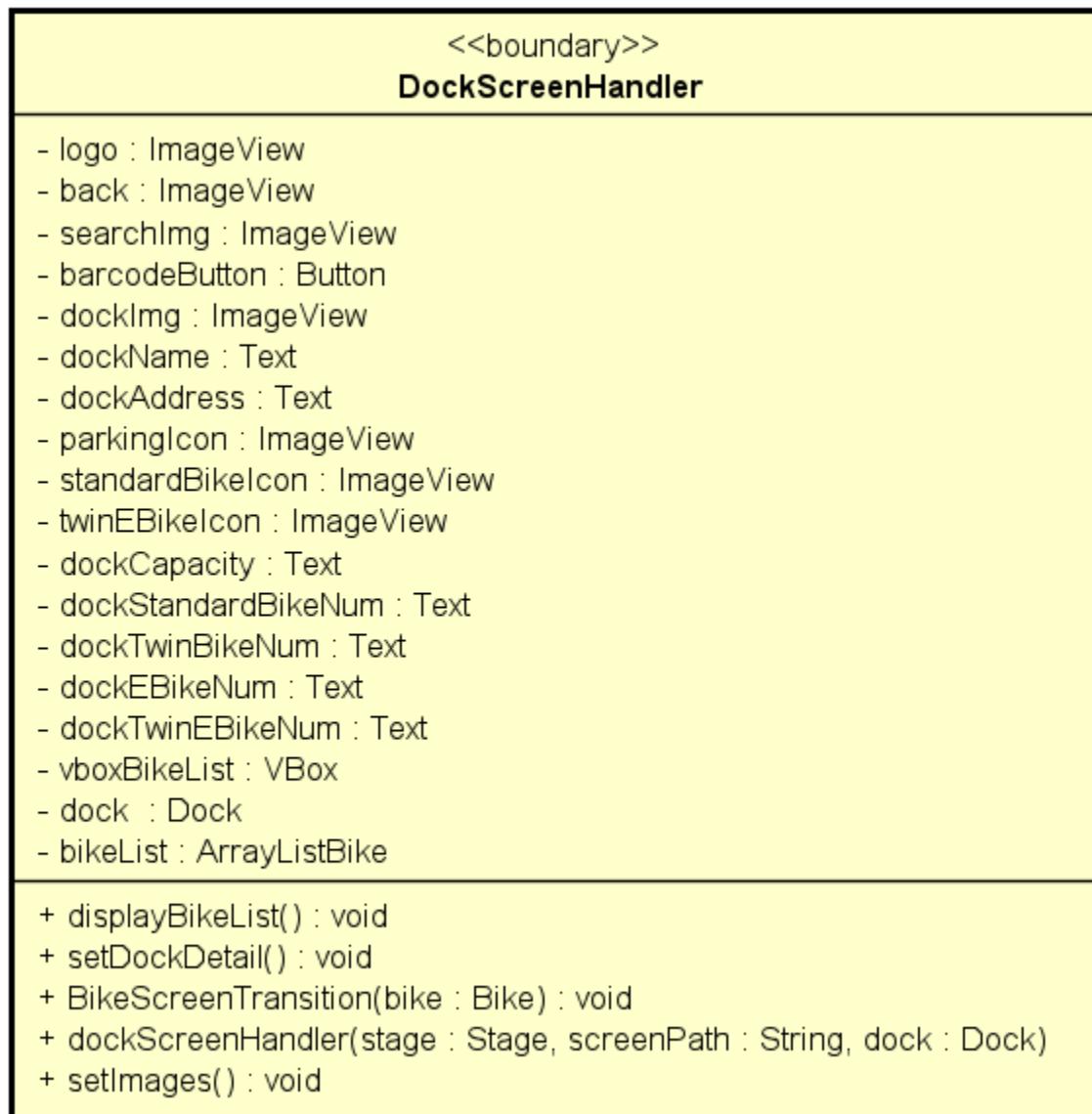
Method

None

State

None

4.4.3.31 Class “DockScreenHandler”



Attribute

| # | Name | Data type | Default value | Description |
|---|---------------|-----------|---------------|--------------------|
| 1 | logo | ImageView | NULL | Logo image fxml |
| 2 | back | imageView | NULL | Back image fxml |
| 3 | searchImg | imageView | NULL | Search image fxml |
| 4 | barcodeButton | Button | NULL | BarcodeButton fxml |

| | | | | |
|-----------|---------------------|-----------------|------|---------------------------|
| 5 | dockName | Text | NULL | Dock name fxml |
| 6 | dockAddress | Text | NULL | Dock address fxml |
| 7 | dockImg | ImageView | NULL | Dock image fxml |
| | parkingIcon | ImageView | NULL | Parking icon fxml |
| | standardBikeIcon | ImageView | NULL | Standard bike icon fxml |
| | twinEBikeIcon | ImageView | NULL | Twin bike icon fxml |
| 8 | dockStandardBikeNum | Text | NULL | Standard bike number fxml |
| 9 | dockTwinBikeNum | Text | NULL | Twin bike number fxml |
| 10 | dockEBikeNum | Text | NULL | E-bike number fxml |
| 11 | dockTwinEBikeNum | Text | NULL | Twin e-bikenumber fxml |
| 12 | vboxBikeList | VBox | NULL | Bike list fxml |
| 13 | dock | Dock | NULL | dock |
| 14 | 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:

bike

Exception:

None

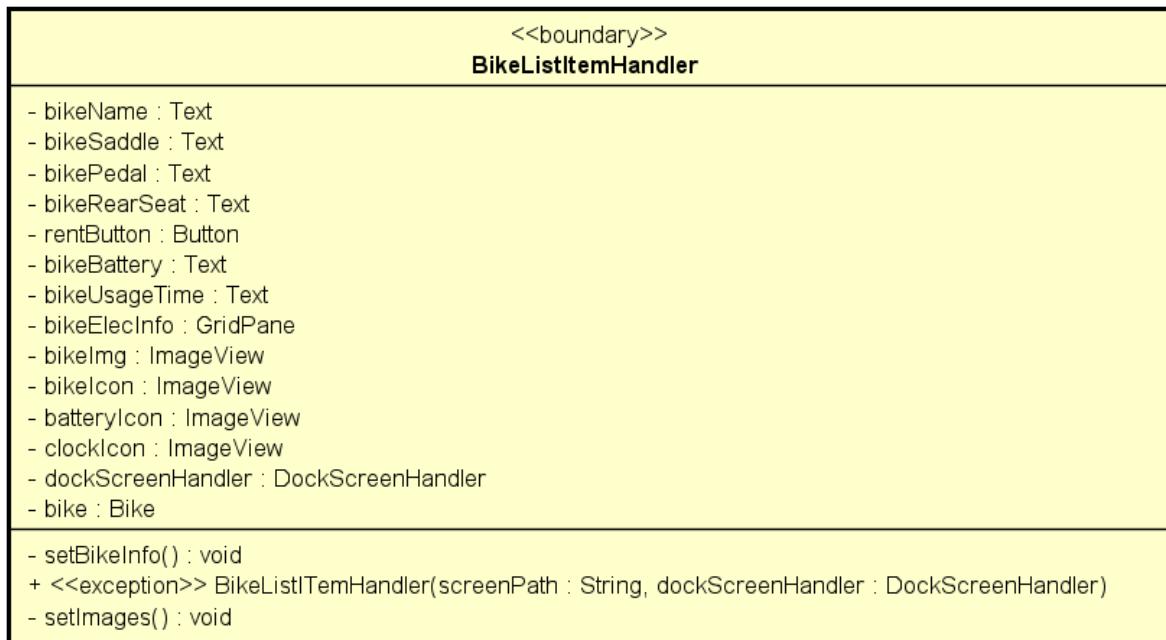
Method

None

State

None

4.4.3.32 Class “*BikeListItemHandler*”



Attribute

| # | Name | Data type | Default value | Description |
|---|------------|-----------|---------------|------------------|
| 1 | bikeImg | ImageView | NULL | Bike image fxml |
| 2 | bikeIcon | imageView | NULL | Bike image fxml |
| 3 | rentButton | Button | NULL | Rent button fxml |
| 4 | bikeName | Text | NULL | Bike name fxml |
| 5 | bikeSaddle | Text | NULL | Bike saddle fxml |

| | | | | |
|----------|-------------------|-------------------|------|---------------------|
| 6 | bikePedal | Text | NULL | Bike pedal fxml |
| 7 | bikeRearSeat | Text | NULL | Bike rear seat fxml |
| 8 | dockScreenHandler | DockScreenHandler | NULL | object |
| 9 | bike | Bike | NULL | object |

Operation

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

Parameter:

none

Exception:

IOException

Method

None

State

None

4.4.3.33 Class “BarcodePopup”

| |
|---|
| <<boundary>> BarcodePopup |
| - parentScreenHandler : BaseScreenHandlerWithBarcodePopup |
| + handleBarcodeEnter() : void - popup(parentScreenHandler : int) : BaseScreenHandlerWithBarcodePopup + display(parentScreenHandler : BaseScreenHandlerWithBarcodePopup) : void - handleBarcodeEnter() : void + BarcodePopup(stage : Stage, parentScreenHandler : BaseScreenHandlerWithBarcodePopup) : void + getBController() : BarcodePopupController |

Attribute

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

Operation

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

Parameter:

parentScreenHandler

stage

Exception:

None

Method

None

State

None

4.4.3.34 Class “*DockListItemHandler*”

| <<boundary>> DockListItemHandler |
|---|
| - dockName : Text - dockAddress : Text - dockBikeNum : Text - dockImg : ImageView - dock : Dock |
| - setDockInfo() : void + <<exception>> DockListItemHandler(screenPath : String, homeScreen : HomeScreenHandler, dock : Dock) |

Attribute

| # | Name | Data type | Default value | Description |
|----------|-------------|-----------|---------------|-------------------------|
| 1 | dockName | Text | NULL | Dock name |
| 2 | dockAddress | Text | NULL | Dock address |
| 3 | dockBikeNum | Text | 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:

None

Method

None

State

None

4.4.3.35 Class “HomeScreenHandler”

| <<boundary>> HomeScreenHandler | |
|---|--|
| - vboxDockList : VBox - dockList : ArrayListDock - logo : ImageView - barcodeButton : Button - searchField : TextField | |
| + displayDockList() : void + onDockListItemSelected(dock : Dock) : void + moveToSessionScreen(session : Session) : void + moveToBikeViewScreen(bike : Bike) : void + show() : void + <<exception>> HomeScreenHandler(stage : Stage, screenPath : String, homeScreenController : HomeScreenController) : void | |

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 | onDockListItemSelected | 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:

dock
session
bike

Exception:

None

Method

None

State

None

4.4.3.36 Class “InvoiceScreenHandler”

| <<boundary>> InvoiceScreenHandler | |
|--|---|
| - changeCardState : boolean - invoiceBikeImage : ImageView - invoiceCardNumber : Text - invoiceStartTime : Text - invoiceEndTime : Text - invoiceDeposit : Text - invoiceReturned : Text | + <<exception>> InvoiceScreenHandler(stage : Stage, screenPath : String, invoice : Invoice, controller : InvoiceScreenController) : void + handleCheckBox() : void + handleConfirmButton() : void - setTextFields() : void + continueAfterPopupClosed(paymentTransaction : PaymentTransaction) : void |

Attribute

| # | Name | Data type | Default value | Description |
|---|-------------------|-----------|---------------|---------------------|
| 1 | changeCardState | boolean | NULL | Card state |
| 2 | invoiceBikeImage | ImageView | NULL | Bike image |
| 3 | invoiceCardNumber | Text | NULL | Invoice card number |
| 4 | invoiceStartTime | Text | NULL | Invoice start time |
| 5 | invoiceEndTime | Text | NULL | Invoice end time |

| | | | | |
|----------|-----------------|------|------|------------------|
| 6 | invoiceDeposit | Text | NULL | Invoice deposit |
| 7 | invoiceReturned | Text | NULL | Invoice returned |

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:

paymentTransaction

Exception:

None

Method

None

State

None

4.4.3.37 Class “*PaymentConfirmationScreenHandler*”

| <<boundary>> PaymentConfirmationScreenHandler | |
|---|--|
| + handleRentingConfirmation() : void + transitionToSessionScreen(rentTransaction : PaymentTransaction) : void + PaymentConfirmationScreenHandler(stage : Stage, screenPath : String, controller : PaymentScreenController) : void + setBikeImage() : void + setPaymentConfirmationConfirmButtonClickListener(e : MouseEvent) : void + <<exception>> saveTransaction() : void - setTextLabels() : void + continueAfterPopupClosed(paymentTransaction : PaymentTransaction) : void | |

Attribute

None

Operation

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

Parameter:

rentTransaction

paymentTransaction

mouseEvent

Exception:

IOException

Method

None

State

None

4.4.3.38 Class “*PaymentScreenHandler*”

| <<boundary>> PaymentScreenHandler | |
|---|--|
| + handleCardInfoSubmit() : void + gotoConfirmationScreen() : void + PaymentScreenHandler(stage : Stage, screenPath : String, paymentScreenController : PaymentScreenController) - setImages() : void | |

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

4.4.3.39 Class “*ReturningDockSelectionHandler*”

| <code><<boundary>></code> ReturningDockSelectionHandler | |
|---|--|
| - dockImg : ImageView | |
| - dockAddress : Text | |
| - dockEmptySlots : Text | |
| - returnBikeBtn : Button | |
| - dockInfo : Text | |
| - dock : Dock | |
| - session : Session | |
| - dockList : ArrayListDock | |
| + displayDockList(dock : Dock) : void | |
| + onDockListClicked() : void | |
| + ReturningDockSelectionHandler(stage : Stage, screenPath : String, returningDockSelectionController : ReturningDockSelectionController, session : Session) | |
| + setImages() : void | |
| + onDockListitemClicked(dock : Dock) : void | |
| + displayDockList() : void | |
| + searchImgBtnListener(e : MouseEvent) : void | |
| + returnBikeBtnListener(e : MouseEvent) : void | |

Attribute

| # | Name | Data type | Default value | Description |
|---|-----------------|---------------|---------------|------------------------|
| 1 | dockName | Text | NULL | Dock name fxml |
| 2 | dockParkingSlot | Text | NULL | Dock parking slot fxml |
| 3 | dockImg | ImageView | NULL | Dock image fxml |
| 4 | dockInfo | Text | NULL | Dock information fxml |
| 5 | dockAdress | Text | NULL | Dock address fxml |
| 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

4.4.3.40 Class “ReturningDockListItemHandler”

| <<boundary>> ReturningDockListItemHandler | |
|---|--|
| - dockName : Text | |
| - dockParkingSlot : Text | |
| - dockImg : ImageView | |
| - dock : Dock | |
| - returningDockSelectionHandler : ReturningDockSelectionHandler | |
| + ReturningDockListItemHandler(screenPath : String, returningDockSelectionHandler : ReturningDockSelectionHandler) : void | |
| + setDockInfo() : void | |
| + setDock(dock : Dock) : void | |
| + setDockInfo() : void | |

Attribute

| # | Name | Data type | Default value | Description |
|----------|-----------------|-----------|---------------|-------------------|
| 1 | dockName | Text | NULL | Dock name fxml |
| 2 | dockParkingSlot | Text | 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:

dock

Exception:

None

Method

None

State

None

4.4.3.41 Class “SessionScreenHandler”

| <<boundary>> SessionScreenHandler | |
|--|--|
| - sessionBikeImg : ImageView - sessionBarcode : Text - sessionLength : Text - sessionRentingFees : Text - sessionCloseBtn : Button - sessionReturnBikeBtn : Button | |
| + gotoDockSelection() : void - SessionScreenHandler(stage : Stage, screenPath : String, controller : SessionScreenController) : void - setSessionLengthText() : void - ToDayAndTime(seconds : long) : void - setImages() : void - setTextFields() : void - goToDockSelection() : void - setImageForLockBikeImg() : void | |

Attribute

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

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:

seconds

Exception:

None

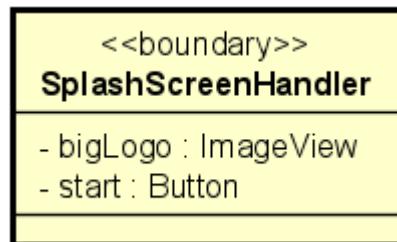
Method

None

State

None

4.4.3.42 Class “SplashScreenHandler”



Attribute

| # | Name | Data type | Default value | Description |
|----------|---------|-----------|---------------|-----------------|
| 1 | bigLogo | ImageView | NULL | Logo image fxml |
| 2 | start | Button | NULL | Button fxml |

Operation

None

Parameter:

None

Exception:

None

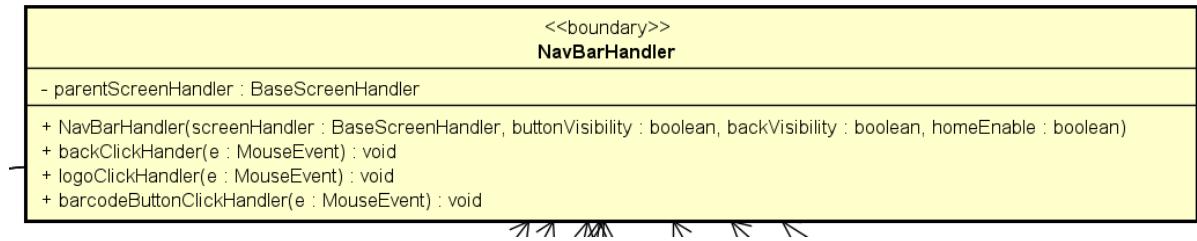
Method

None

State

None

4.4.3.43 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:

e – mouse event

Exception:

None

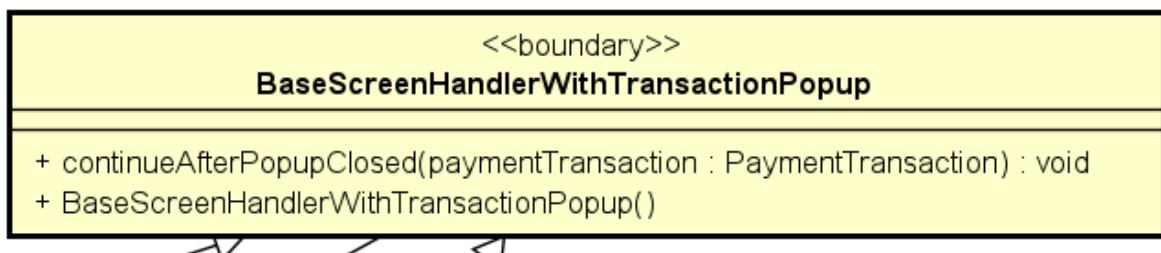
Method

None

State

None

4.4.3.44 *Class “BaseScreenHandlerWithTransactionPopup”*



Attribute

None

Operation

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

Parameter:

paymentTransaction – transaction to be shown result

Exception:

None

Method

None

State

None

4.4.3.45 Class “*PaymentResultPopup*”

| <<boundary>> PaymentResultPopup | |
|--|--|
| - screenHandler : BaseScreenHandlerWithTransactionPopup | |
| - paymentTransaction : PaymentTransaction | |
| + PaymentResultPopup(stage : Stage, screenHandler : BaseScreenHandlerWithTransactionPopup, paymentTransaction : PaymentTransaction, paymentMessage : String) | |
| - popup(paymentTransaction : PaymentTransaction, screenHandler : BaseScreenHandlerWithTransactionPopup, paymentMessage : String) : PaymentResultPopup | |
| + setTransactionInfo(paymentTransaction : PaymentTransaction, paymentMessage : String) : void | |
| + display(paymentTransaction : PaymentTransaction, screenHandler : BaseScreenHandlerWithTransactionPopup, paymentMessage : String) : void | |
| - setImage() : void | |
| - closeBtnHandler() : void | |

Attribute

| # | Name | Data type | Default value | Description |
|---|------------------------|--|---------------|----------------------------------|
| 1 | screenHandler | BaseScreenHandlerWithTransactionPop pup | new | handler |
| 2 | paymentTransacti on | 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:

screenHandler

paymentTransaction

paymentMessage

Exception:

None

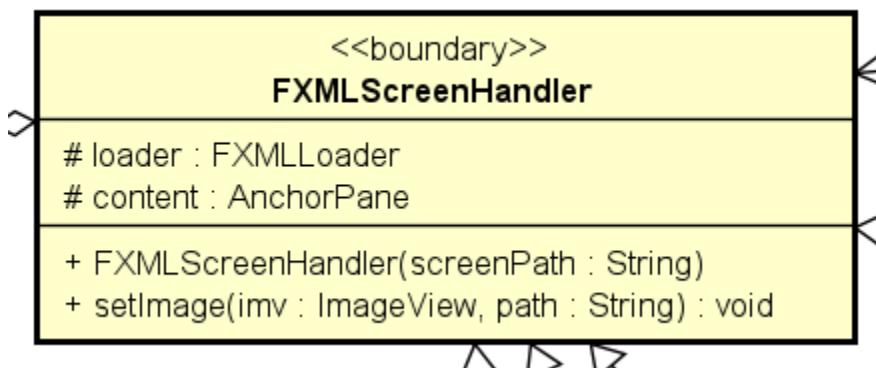
Method

None

State

None

4.4.3.46 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:

imv – image holder to be set

path – path to image

Exception:

None

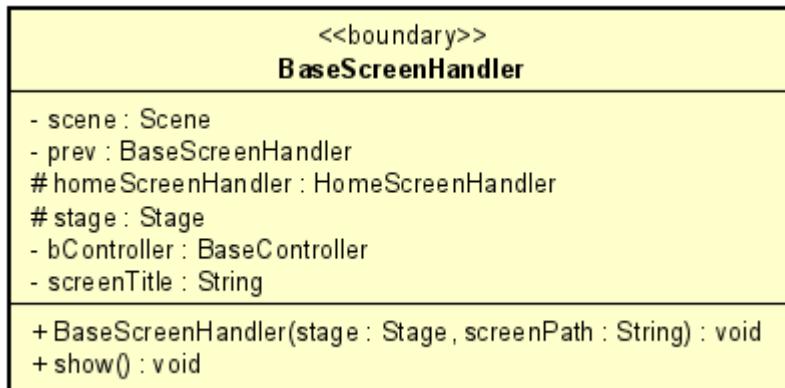
Method

None

State

None

4.4.3.47 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

5 Design Considerations

5.1 Goals and Guidelines

Goals:

- The database design is focused on speed over optimal storage. As the non-functional requirements of the software demands for fast responsiveness and the EBR software is only used in the Eco park area; hence, the amount of data is not too large.

Guidelines:

- The naming convention will follow the Java naming convention.

5.2 Architectural Strategies

- Design architecture: MVC – the reason has been stated above.
- Design principles: the design will follow the SOLID principles. These principles set has long history and, via practical usages, proved as effective.
- Database design: the software will use PostgreSQL as the DBMS.
- Framework to be used to develop the software will be JavaFX.

5.3 Coupling and Cohesion

- Coupling:
 - There are no coupling issues at Content, Common and Control level.
 - Most of our classes and packages are at Data Coupling level, which is already a good design.
 - There some parts of design are at Stamp Coupling level such as in calculate renting fee as we pass the whole instance of Session into the method, but as we use the bike instance in the session and multiple other fields, this coupling is acceptable, a nice trade-off for the breeze of use. This holds true for other Stamp Coupling issues as well.

- Cohesion:
 - There is no component with Cohesion issues at Coincidental, Temporal, Procedural and Communicational level.
 - There's an exception in utility class Utils: it contains method for both security methods (hash) and formatting methods (time, date, ...). However, all those methods are static and this class acts only as helper/utility.

5.4 Design Principles

- Single Responsibility Principle

Each of our designed classes follows this principle. Only exception is the Utils class as it contains method for both security and formatting. However, as explained in cohesion section, all these methods are global, so split it into 2 utility classes doesn't affect much.

- Open / Closed Principle

For additional requirement of extending multiple renting fees formulas, we already using Strategy Pattern to solve the problem. And for others classes, the design also has many parent classes to use for generalization (e.g., StandardBike and StandardElectricalBike for adding new kind of bike, PaymentCard for adding new payment method, ...)

- Liskov Substitution Principle

Yes, our design does follow the Liskov Substitution Principle. All the parent class can be substituted by its children classes. Our design has a clear and logical inheritance hierarchy.

- Interface Segregation Principle

The interbank system interface has already designed very simply and contains only 2 methods that needed for pay and refund.

- Dependency Inversion Principle

The higher-level modules do not depend on lower-level modules. All subsystem has their own interface for other to use. For the additional requirement, to add more Card type, we only need to make it extends the PaymentCard class.

5.5 Design Patterns

- Singleton:
 - All manager model classes are designed using Singleton Pattern as these classes are used to provide and manipulate lots of share data throughout the software.
 - For example, the DockManager classes will store all queried docks for displaying, reading data. These manager classes should have only 1 instance as some are expensive and has no functional use to create extra instance.
- Strategy:
 - This pattern is used to solve the violation of Open/Closed Principle. Since, this pattern helps the design easy to scaleup when adding new variations of a method.
 - For example of implementation, the design uses this pattern in renting fee calculation by making a parent abstract class RentingFeeCalculator then let others type of Calculator extends it.
- Façade:
 - This pattern is used to reduce complexity of API of a service or interface of a subsystem.
 - The current design applies this pattern on the interbank subsystem, to make it interface simple and compact.