**HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY**

School of Information and communications technology

*******

Software Design Document

*Version 1.3*

# EcoBike Rental Software

Subject: Phát triển phần mềm theo chuẩn kỹ năng ITSS

**ISD.VN.20211-Group2**

Vũ Văn Long - 20184146

Trần Xuân Trường - 20184212

Mai Hoàng Minh - 20184151

*Hanoi, November 25th, 2021*

# Table of Contents


Table of Contents ..... 1

1 Introduction ..... 5

1.1 Objective ..... 5

1.2 Scope ..... 5

1.3 Glossary ..... 5

1.4 References ..... 5

2 Overall Description ..... 6

2.1 General Overview ..... 6

2.2 Assumptions/Constraints/Risks ..... 6

2.2.1 Assumptions ..... 6

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 ..... 12

3.4 Unified Analysis Class Diagram ..... 14

3.5 Security Software Architecture ..... 14

4 Detailed Design ..... 15

4.1 User Interface Design ..... 15

4.1.1 Screen Configuration Standardization ..... 15

4.1.2 Screen Transition Diagrams ..... 16

4.1.3 Screen Specifications ..... 16

4.2 Data Modeling ..... 26

4.2.1 Conceptual Data Modeling ..... 26

4.2.2 Database Design ..... 27

4.3 Non-Database Management System Files ..... 30

4.4 Class Design ..... 30
4.4.1 General Class Diagram ..... 30
4.4.2 Class Diagrams ..... 31
4.4.3 Class Design ..... 34
5 Design Considerations ..... 37
5.1 Goals and Guidelines ..... 37
5.2 Architectural Strategies ..... 37
5.3 Coupling and Cohesion ..... 37
5.4 Design Principles ..... 38
5.5 Design Patterns ..... 39

# List of Figures

Figure 1: View Dock Sequence Diagram ..... 7
Figure 2: View Dock Communication Diagram ..... 7
Figure 3: Rental Bike Sequence Diagram ..... 8
Figure 4: Rental Bike Communication Diagram ..... 9
Figure 5: Pay Bike Deposit Sequence Diagram ..... 10
Figure 6: Pay Bike Deposit Communication Diagram ..... 10
Figure 7: Return Bike Sequence Diagram ..... 11
Figure 8: Return Bike Communication Diagram ..... 11
Figure 9: View Dock UC ..... 12
Figure 10: Pay Bike Deposit UC ..... 12
Figure 11: Rent Bike UC ..... 13
Figure 12: Return Bike UC ..... 13
Figure 13: Unified Analysis Class Diagram ..... 14
Figure 14: Screen Transition Diagram ..... 16
Figure 15: Conceptual Data Modeling ..... 26
Figure 16: Logical Data Model ..... 27
Figure 17: General Class Diagram ..... 30
Figure 18: Identify InterbankSubsystem ..... 31
Figure 19: Identify InterbankSubsystem Interface ..... 31
Figure 20: Distribute subsystem behavior to subsystem elements ..... 32
Figure 21: Document subsystem elements ..... 33
Figure 22: Describe subsystem dependencies ..... 33
Figure 23: Checkpoints ..... 34

# List of Tables

Table 1: Screen Specification – Home Screen ..... 16
Table 2: Define the field attributes – Home screen ..... 17
Table 3: Screen Specification – The detailed information of a dock screen ..... 18
Table 4: Define the field attributes – The detailed information of a dock ..... 18
Table 5: Screen Specification – The detailed information of a bike ..... 19
Table 6: Define the field attributes – The detailed information of a bike ..... 19
Table 7: Screen Specification – Credit card information screen ..... 20
Table 8: Define the field attributes – Credit card information screen ..... 20
Table 9: Screen Specification – Information confirmation screen ..... 21
Table 10: Define the field attributes – Information confirmation screen ..... 21
Table 11: Screen Specification - Notification screen ..... 22
Table 12: Define the field attributes – Notification screen ..... 22
Table 13: Screen Specification– Bike state information screen ..... 22
Table 14: Define the field attributes – Bike state information screen ..... 23
Table 15: Screen Specification – Return bike screen ..... 23
Table 16: Define the field attributes – Return bike screen ..... 24
Table 17: Screen Specification – Invoice screen ..... 24
Table 18: Define the field attributes – Invoice screen ..... 25
Table 19: Dock DB ..... 28
Table 20: Bike DB ..... 28
Table 21: EBike DB ..... 28
Table 22: Type DB ..... 28
Table 23: Invoice DB ..... 29
Table 24: Transaction DB ..... 29

# 1 Introduction

The project is to create an EcoBike Rental software for Ecopark township. The Software Design Document provides documentation which will be used to aid in software development by providing the details for how the software should be built. The designs described, follow the requirements specified in the Software Requirements Specifications document prepared for the project. Within the Software Design Document are narrative and graphical documentation of the software design for the project including interaction diagrams, system and subsystem architecture, user interface design, database design, class diagrams and other supporting requirement information.

## *1.1 Objective*

The purpose of this document is to present a detailed description of the designs of the EcoBike Rental Software. This document is intended for the programming group in Group 2, to use the designs as guidelines to implement the project.

This document provides a comprehensive architectural overview of the system, using a number of different architectural views to depict different aspects of the system. It is intended to capture and convey the significant architectural decisions which have been made on the software.

## *1.2 Scope*

Our application will provide managers with automatic bike rental and return services in dock lots. But in this problem, we will skip the login part and focus on the function related to renting and returning the bike. The goal is to create a fully automated management application that provides customers with the most basic functionality.

A simple way the user after accessing, will see a list of dock lots. After that, users can view information about dock lots and perform bike rental functions through bike codes. After the initial payment process, customers can use the registered bike. Before or during use, customers can also view information about their rented bike such as bike status, battery status. At the end of the usage period, the user will be able to perform the function of returning the bike and making the final payment.

## *1.3 Glossary*

We assume that the reader of this document has relatively good base knowledge about computer/software in general. Still, the document will be written in general audience-friendly way that most reader can understand. Scholarly terms, if any, in this document will be briefly explained after it has been used.

## *1.4 References*

Centers for Medicare & Medicaid Services. (n.d.). *System Design Document Template.* Retrieved from Centers for Medicare & Medicaid Services: https://www.cms.gov/Research-Statistics-Data-and-Systems/CMS-Information-Technology/XLC/Downloads/SystemDesignDocument.docx

# 2 Overall Description

## *2.1 General Overview*

The Software Design Document is divided into 5 sections with various subsections.

Section 1 and 2 are introduction and overall description about the Software Design Document of the EcoBike Rental Software. Section 3 contains discussions of the system architecture and architecture design for the project with interaction diagrams and analysis class diagrams…. Section 4 shows samples of user interface design for the software, database design and contains the class diagrams. Lastly, section 5 discusses about design considerations such as coupling and cohesion, design principles, design patterns.

## *2.2 Assumptions/Constraints/Risks*

### 2.2.1 Assumptions

Users that use the software should have a good connection to the Internet. Also, our software is a desktop application, so the user also must have a laptop/desktop with an OS (we recommended 64 Bit Microsoft Windows 8 or later; macOS 10.13 or later; or any Linux distribution that supports running application) to run the apps. About the system requirement, we would say 2 GB RAM minimum, 8 GB RAM recommended; for storage 2.5 GB and another 1 GB for caches minimum, solid-state drive with at least 5 GB of free space recommended; require latest version of JRE; 1280×800 is a recommended screen resolution.

### 2.2.2 Constraints

### 2.2.3 Risks

# 3 System Architecture and Architecture Design

## *3.1 Architectural Patterns*

## *3.2 Interaction Diagrams*

**Figure 1: View Dock Sequence Diagram**

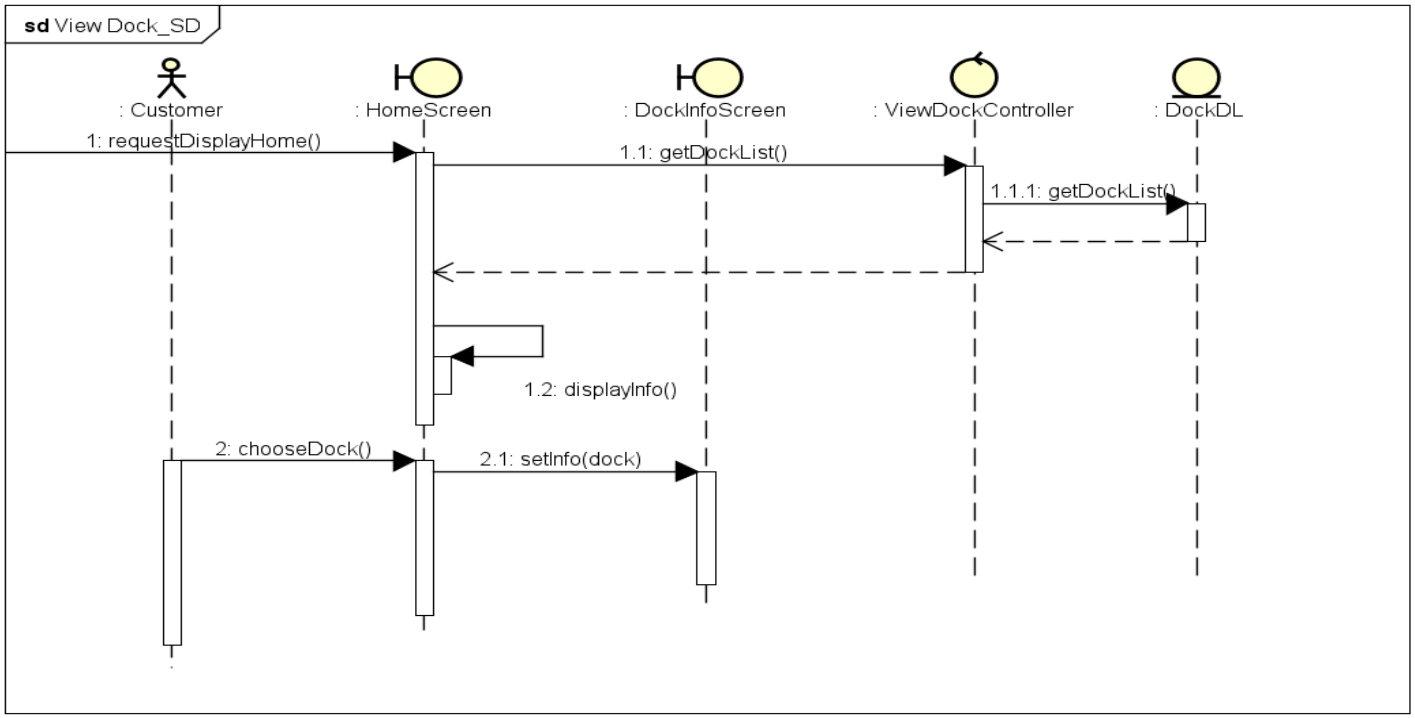



**Figure 2: View Dock Communication Diagram**

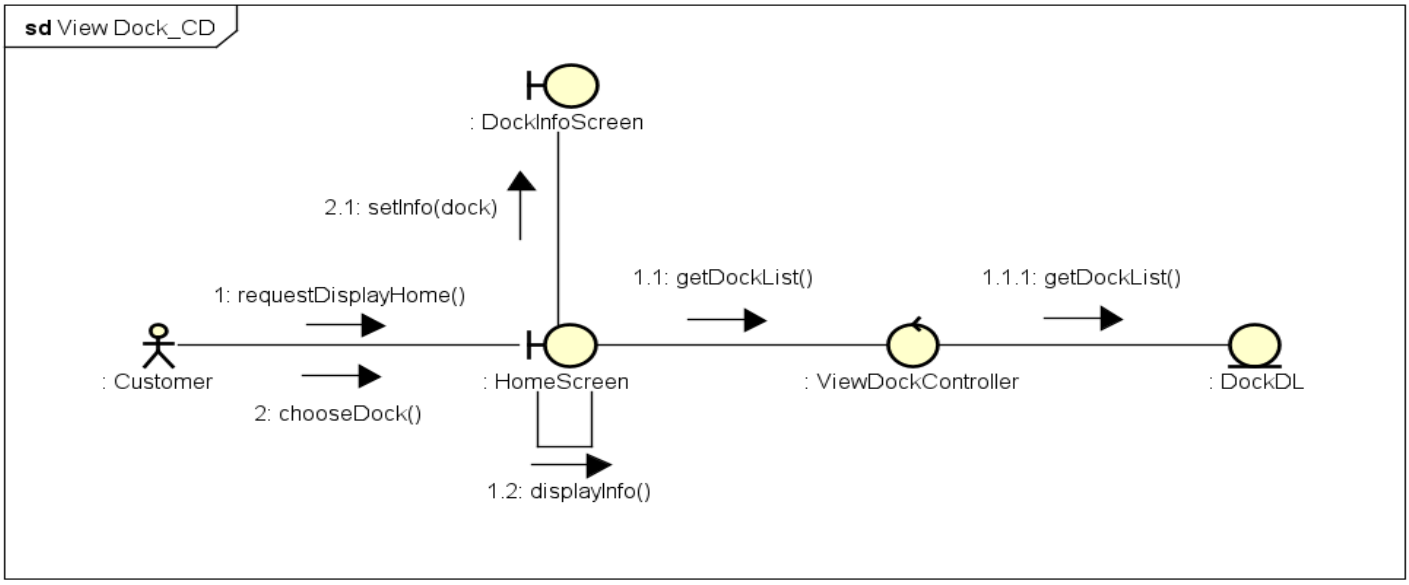


**Figure 3: Rental Bike Sequence Diagram**

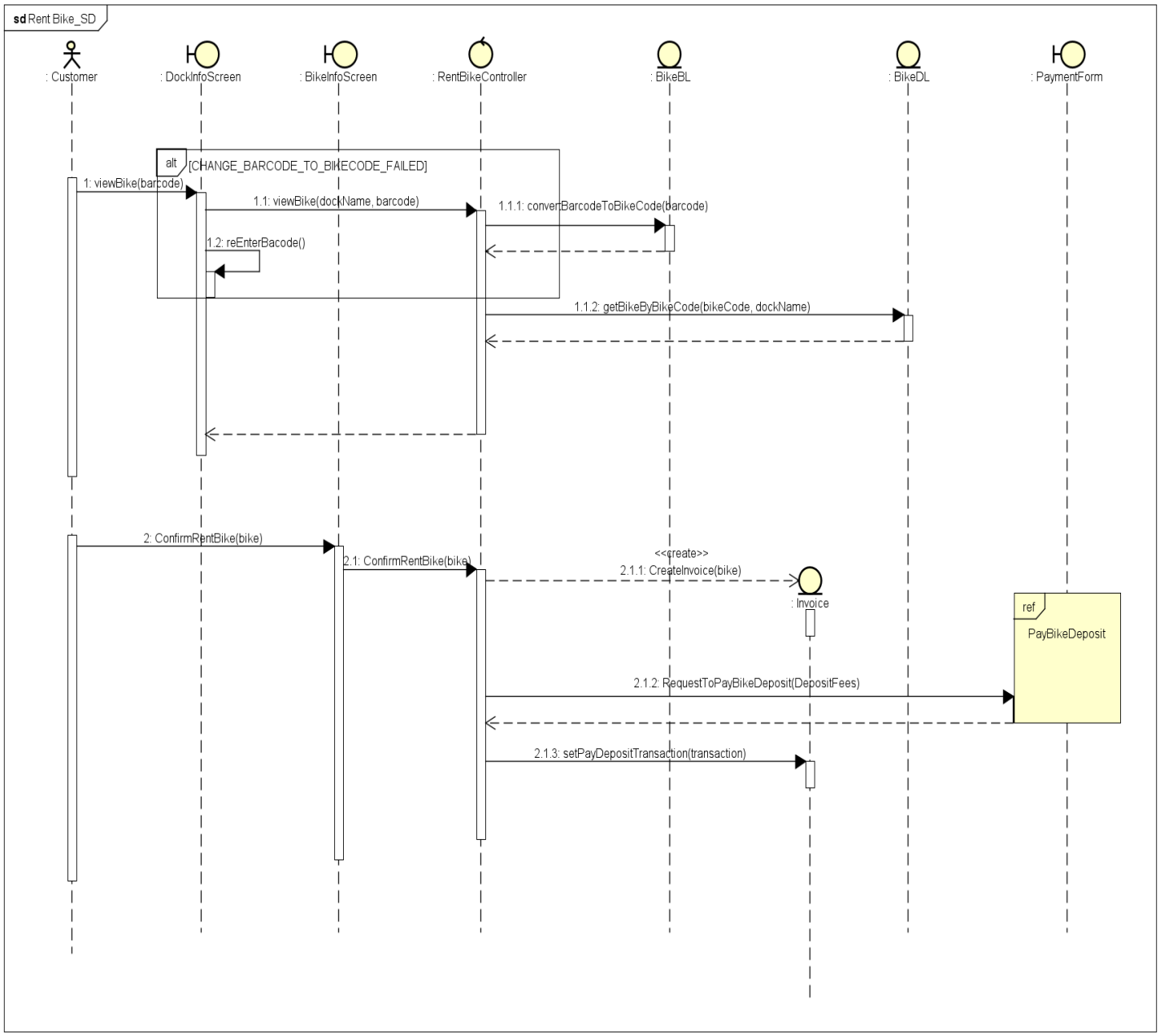


**Figure 4: Rental Bike Communication Diagram**

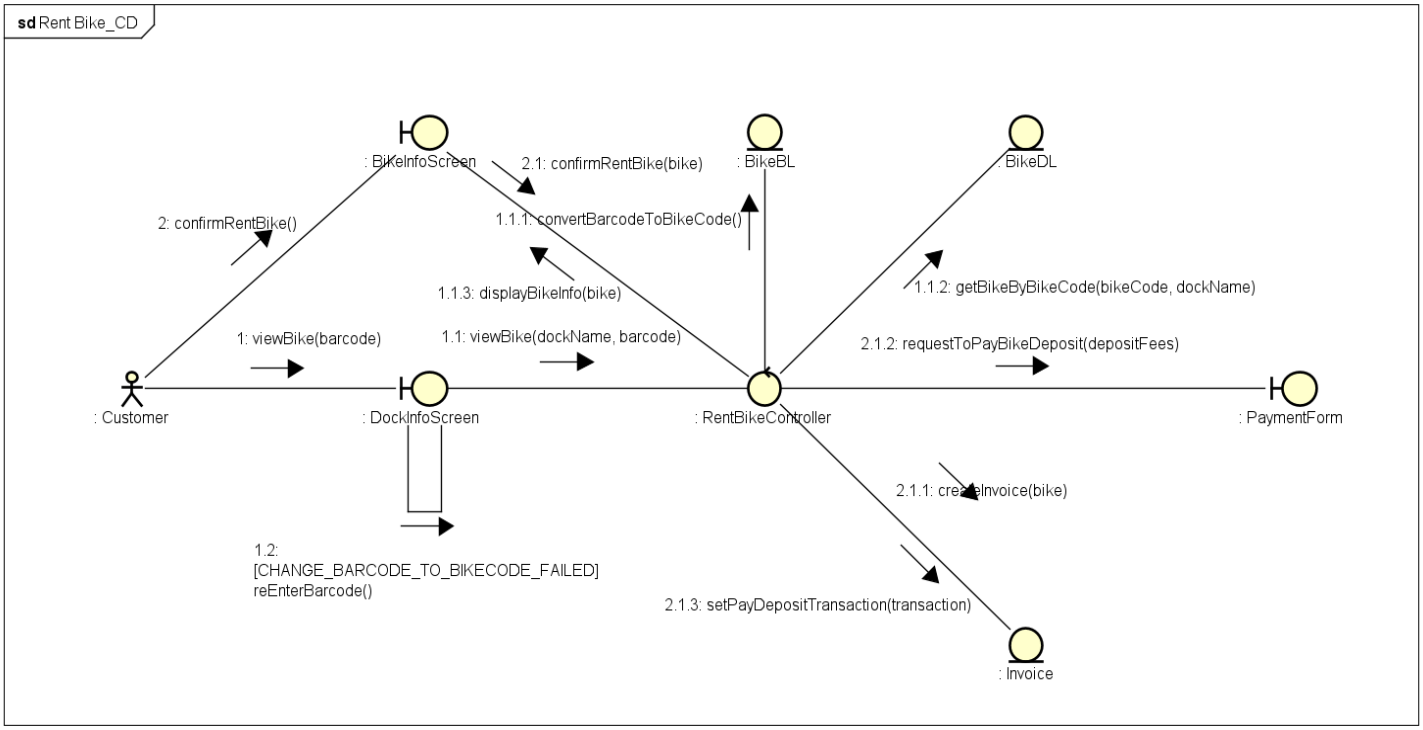


**Figure 5: Pay Bike Deposit Sequence Diagram**

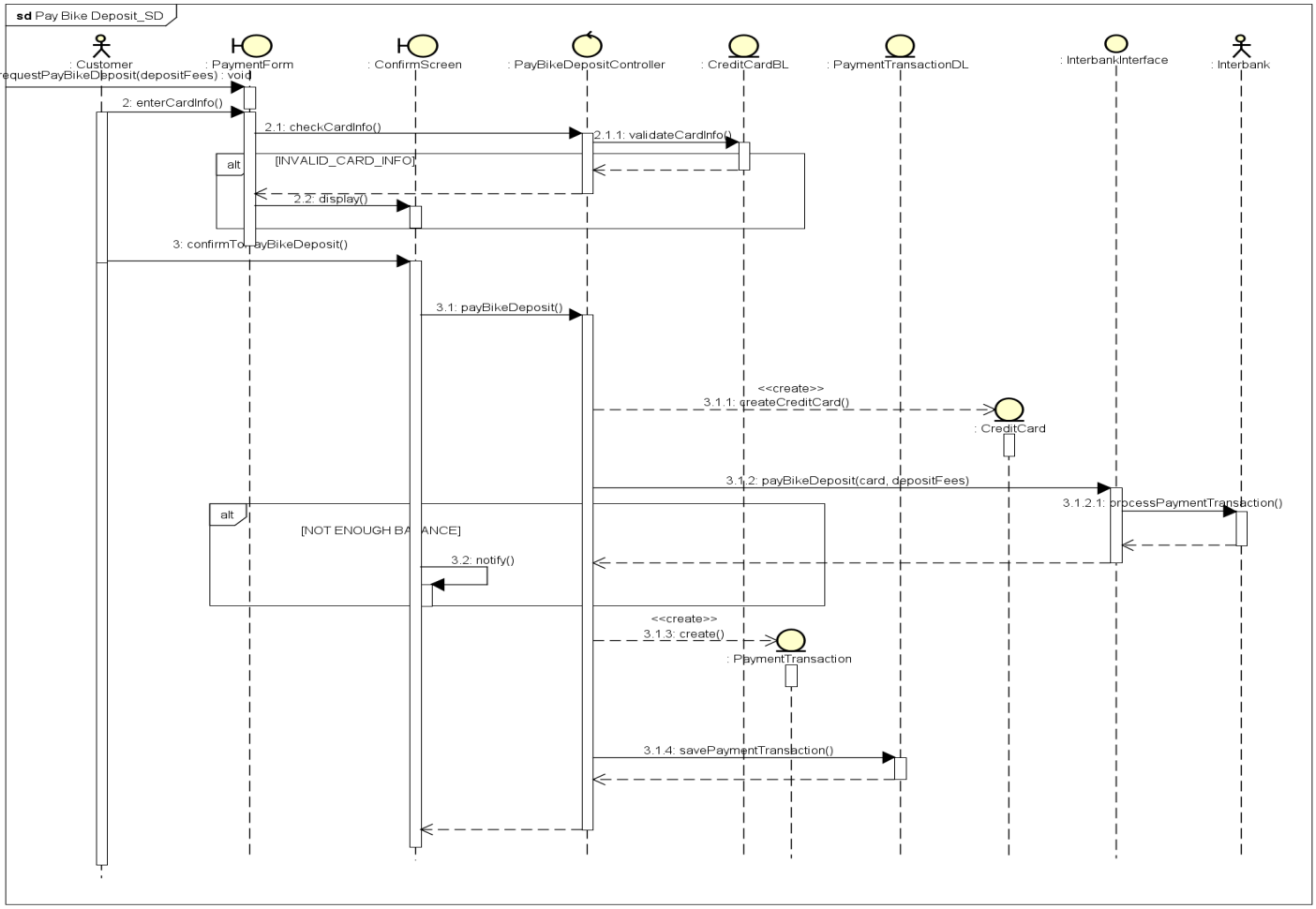



**Figure 6: Pay Bike Deposit Communication Diagram**

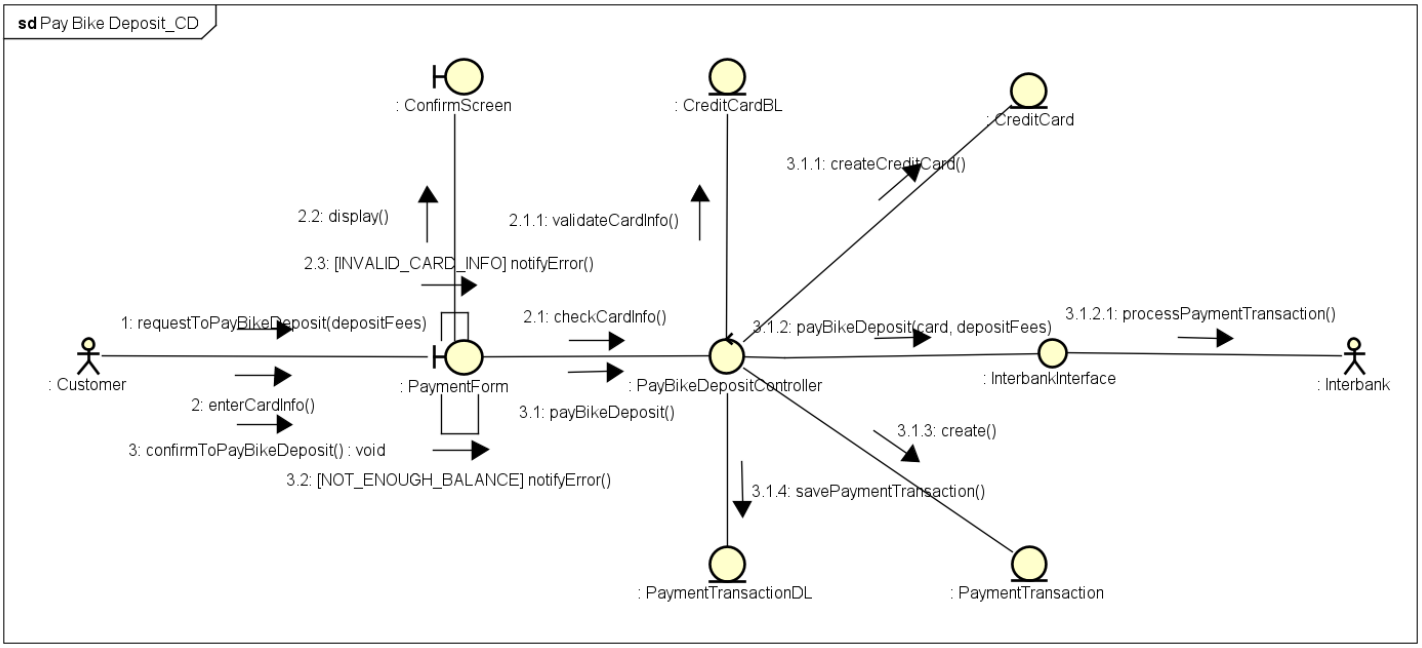


**Figure 7: Return Bike Sequence Diagram**

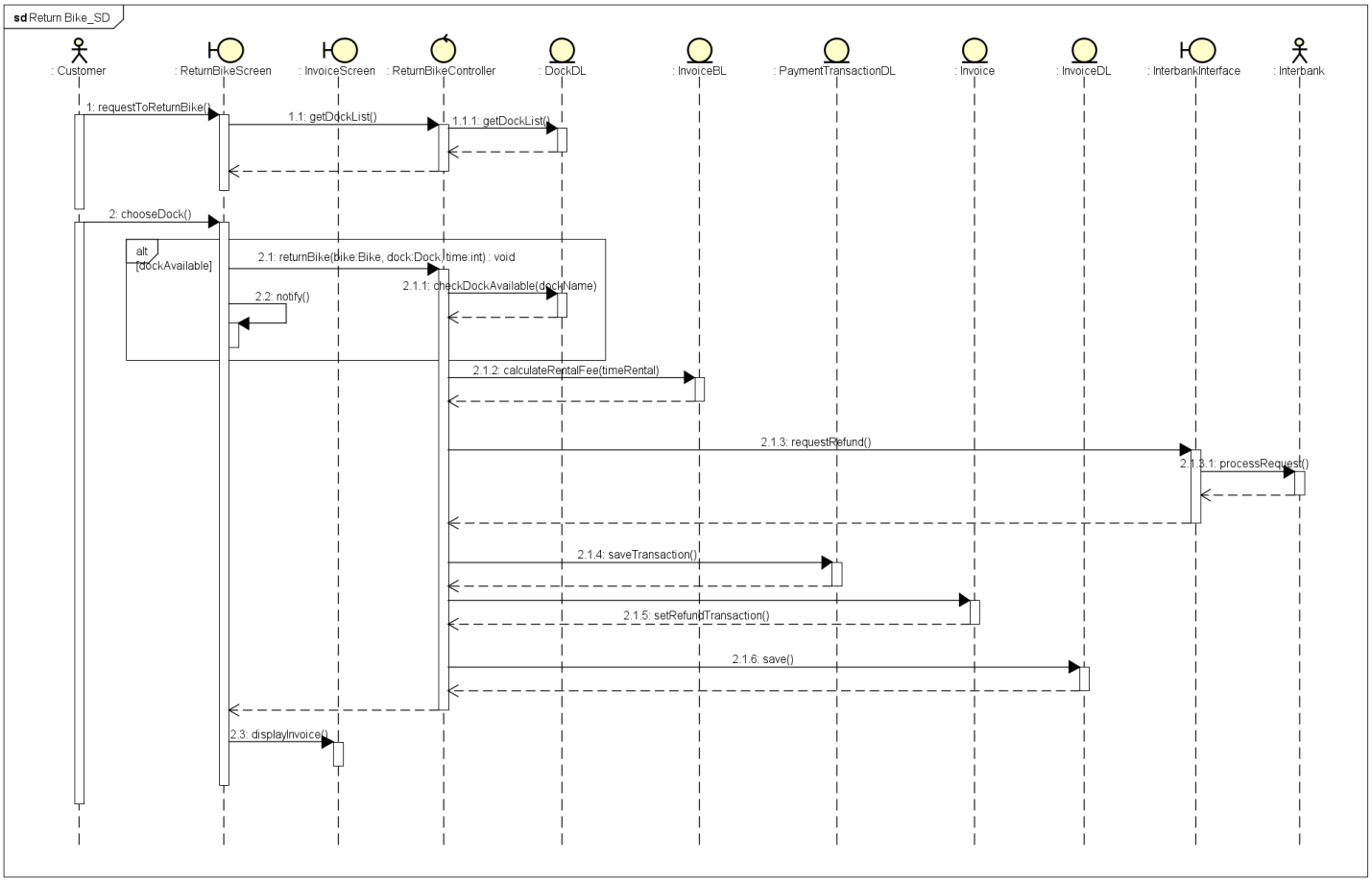
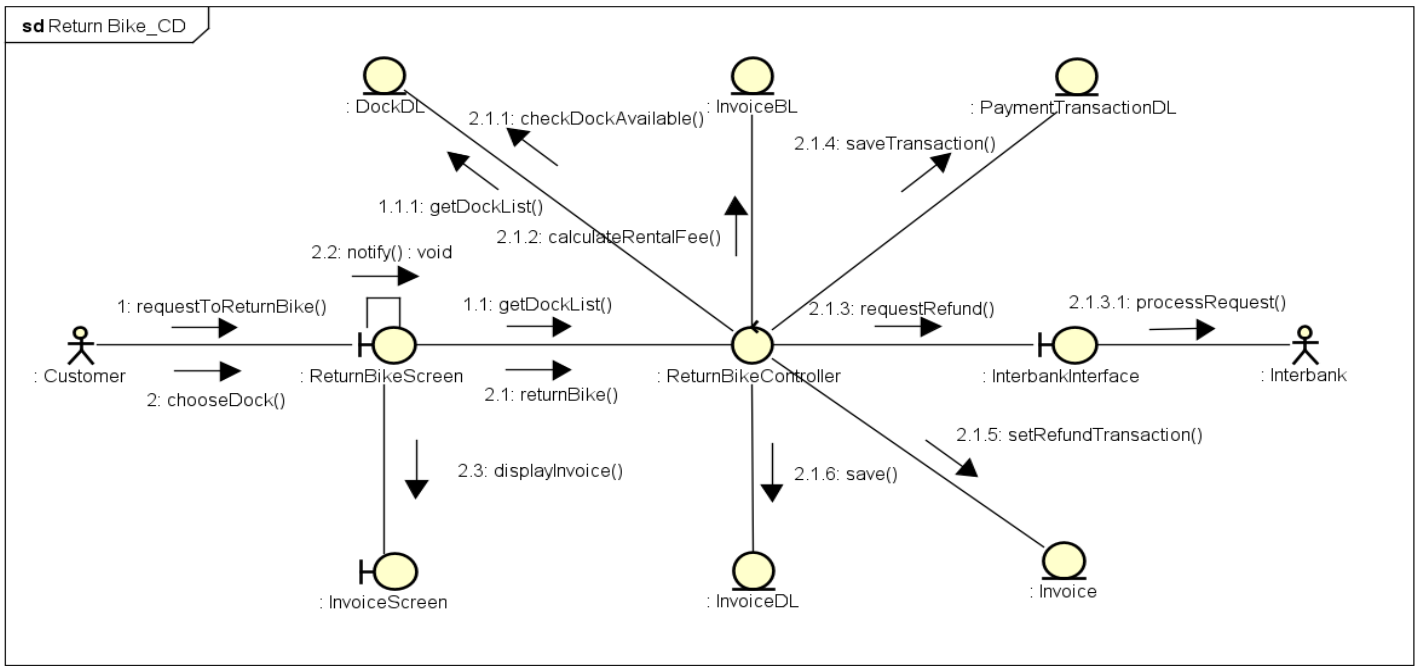



**Figure 8: Return Bike Communication Diagram**



## *3.3 Analysis Class Diagrams*

**Figure 9: View Dock UC**

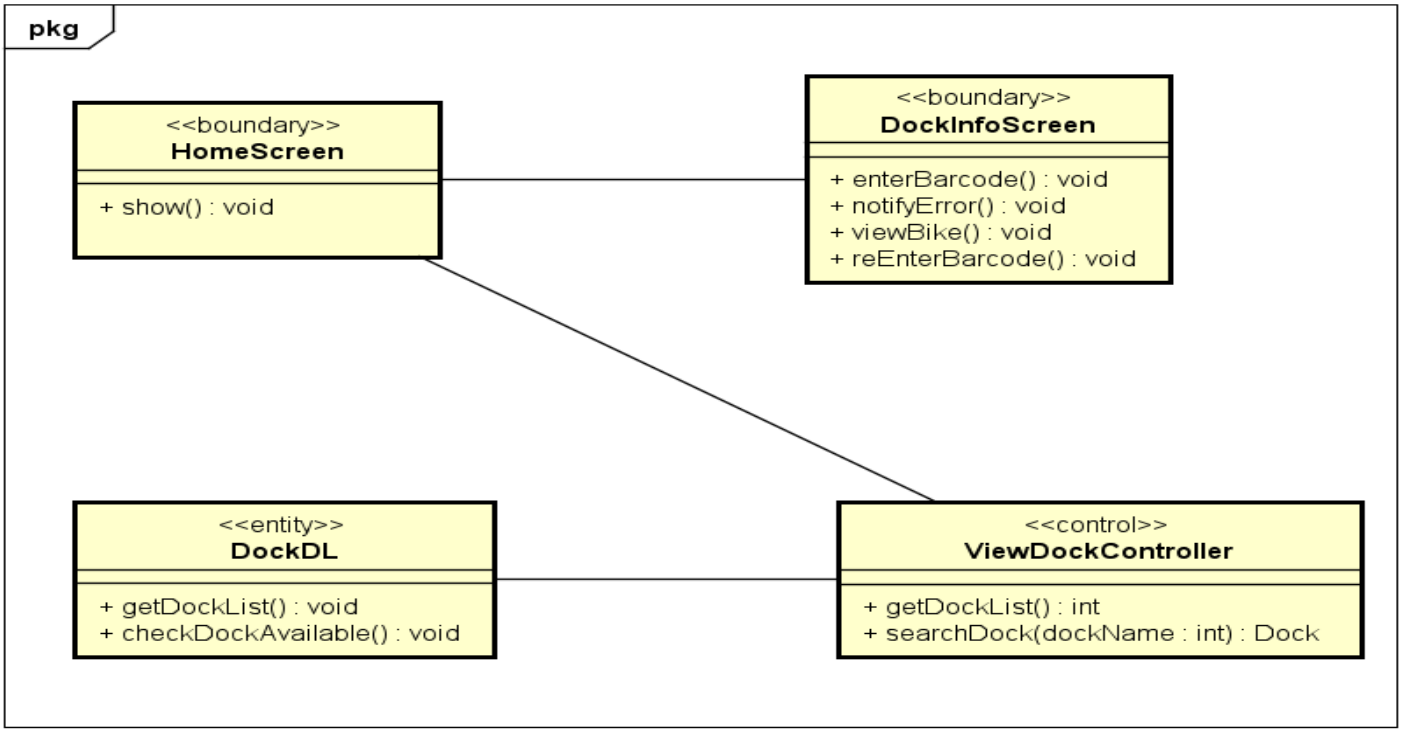



**Figure 10: Pay Bike Deposit UC**

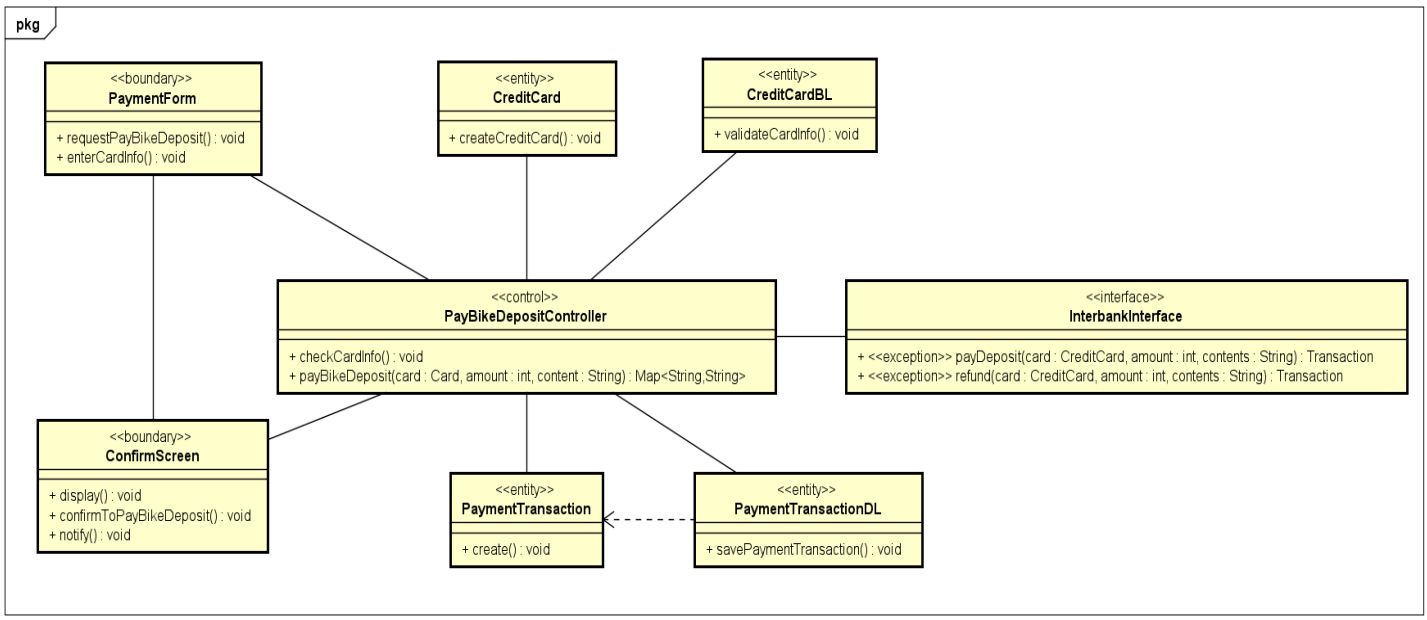


**Figure 11: Rent Bike UC**

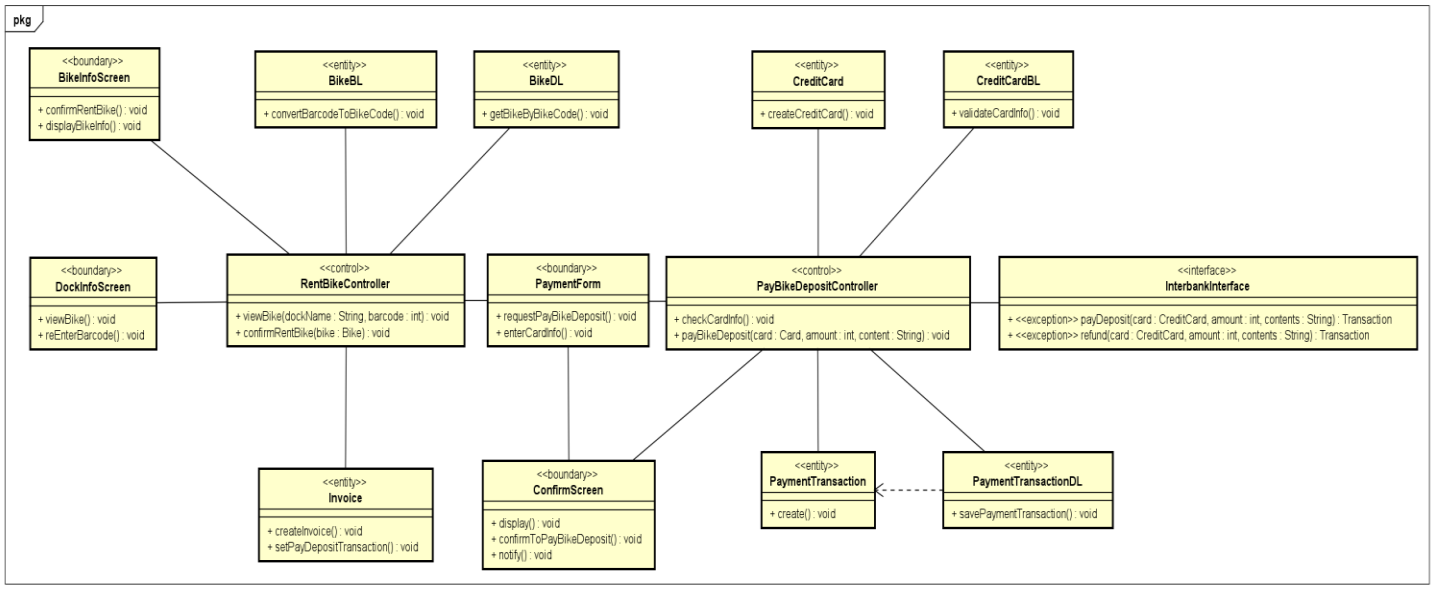
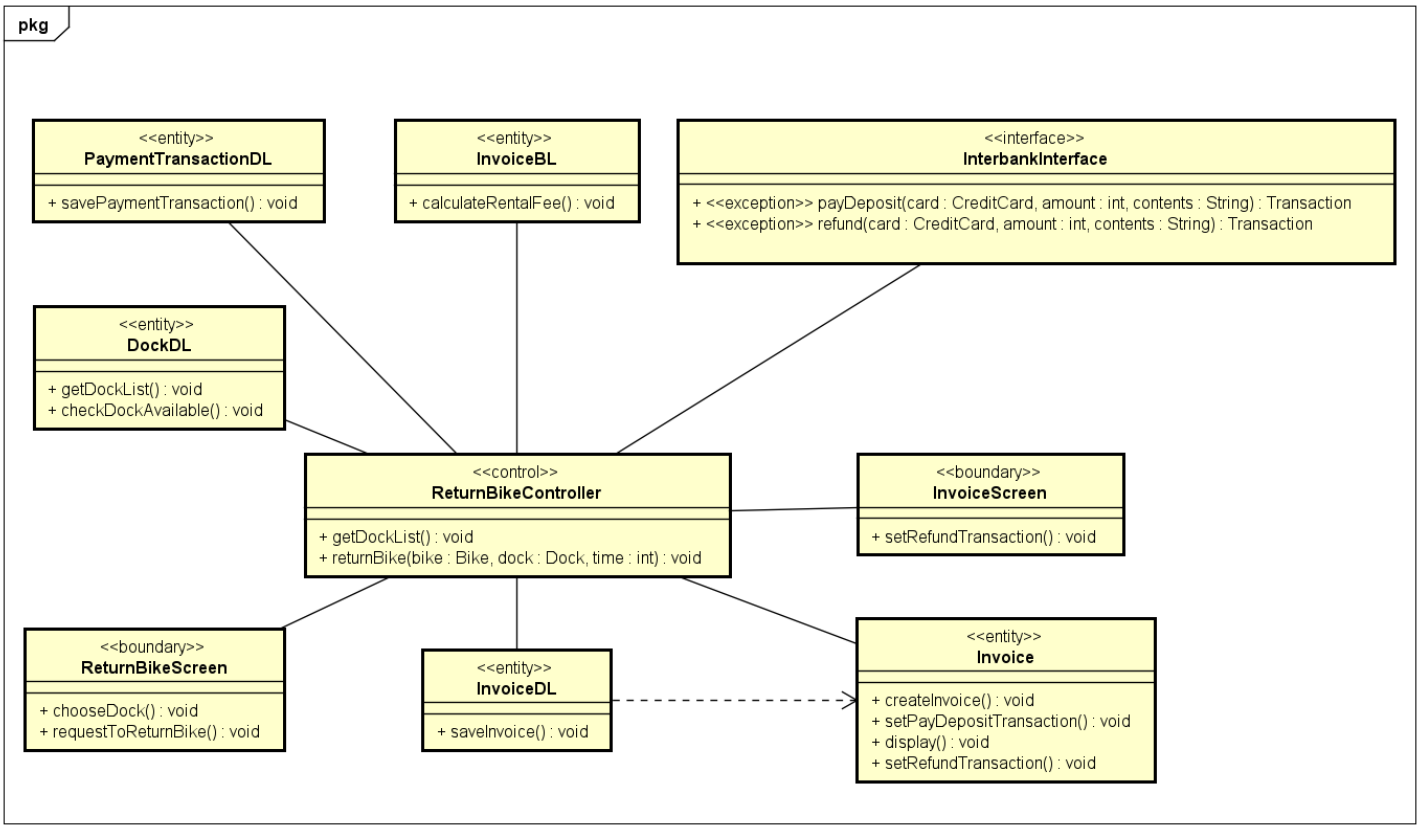
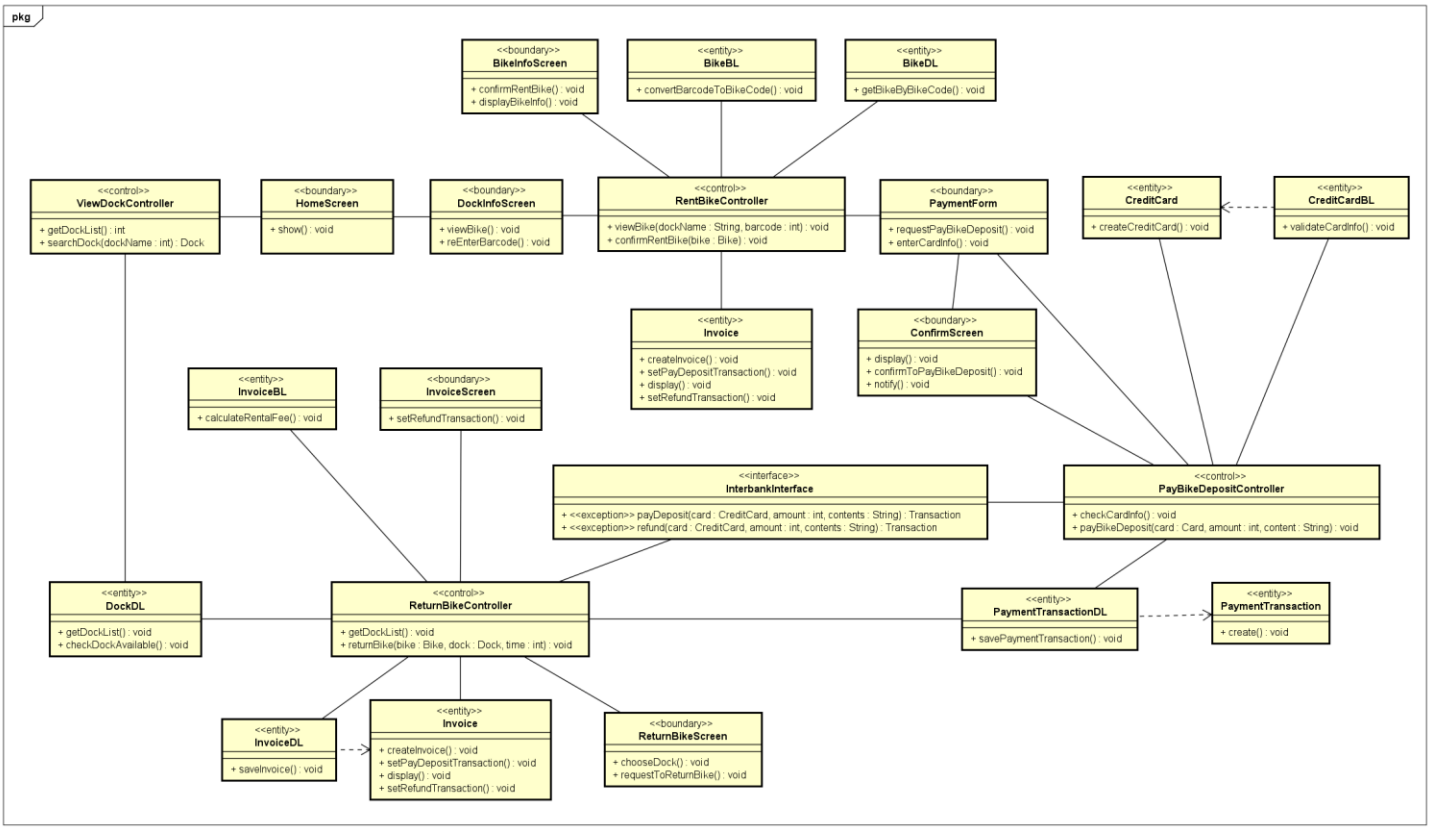



**Figure 12: Return Bike UC**



## *3.4 Unified Analysis Class Diagram*

**Figure 13: Unified Analysis Class Diagram**




## *3.5 Security Software Architecture*

In this project, we will not consider features such as user authentication (sign up, sign in, sign out), but we focus on features related to bike renting and returning.

# 4 Detailed Design

## *4.1 User Interface Design*

### 4.1.1 Screen Configuration Standardization

**Display**

Number of colors supported: 16,777,216 colors

Resolution: 1280 × 800 $pixels$

**Screen**

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

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 top of the frame in the middle. Consistency in expression of alphanumeric numbers: comma for separator of thousand while strings only consist of characters, digits, commas, dots, spaces, underscores, and hyphen symbol.

**Control**

Size of the text: medium size (mostly 24px). Font: Segoe UI. Color: #000000 or #160C67

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 no stack frames. Each screen will be separated.

However, the manual is considered a popup message, as the main screen cannot be operated while the manual screen is shown. After the opening screen, the app will start with splash screen, and then the first screen (home screen) will appear.

*Sequences of the system screens:*

1. Splash screen (first screen)
2. Home screen
3. Dock details screen – view dock information
4. Bike details screen – view bike information
5. Credit card form screen – enter credit card information
6. Notification screen – notify status payment
7. Renting bike information screen – view Renting bike information
8. Return bike screen – return renting bike
9. Invoice screen – view rented bill

**Direct input from the keyboard**

There will be no shortcuts. There are back buttons to move back to the previous screen.

**Error**

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

### 4.1.2 Screen Transition Diagrams

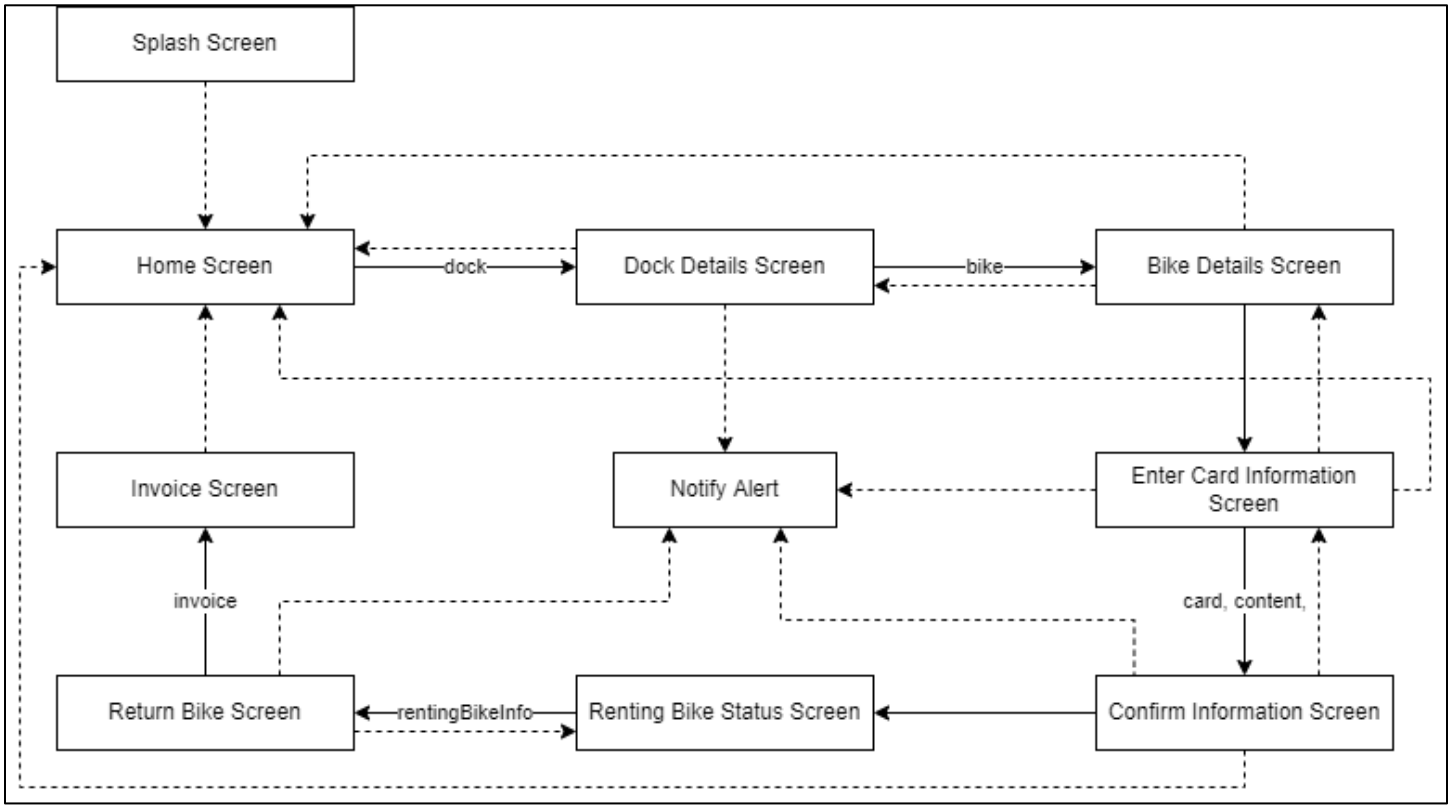



**Figure 14: Screen Transition Diagram**

### 4.1.3 Screen Specifications

**Table 1: Screen Specification – Home Screen**

| **EcoBikeRental Software** | | **Date of creation** | **Approved by** | **Reviewed by** | **Personal in charge** |
|---|---|---|---|---|---|
| **Screen Specification** | **Home screen** | **08/11/2021** | | **Vũ Văn Long** | **Trần Xuân Trường** |
| | | **Control** | **Operation** | **Function** | |
| | | Area of searching dock | Initial | Searching a dock | |
| | | Area of displaying the list of docks | Initial | Displaying the list of docks | |

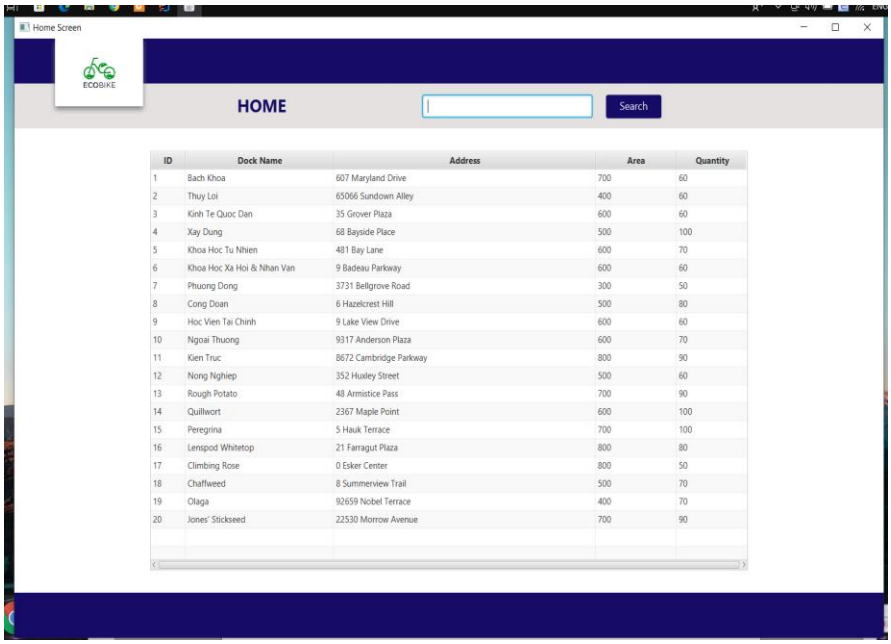
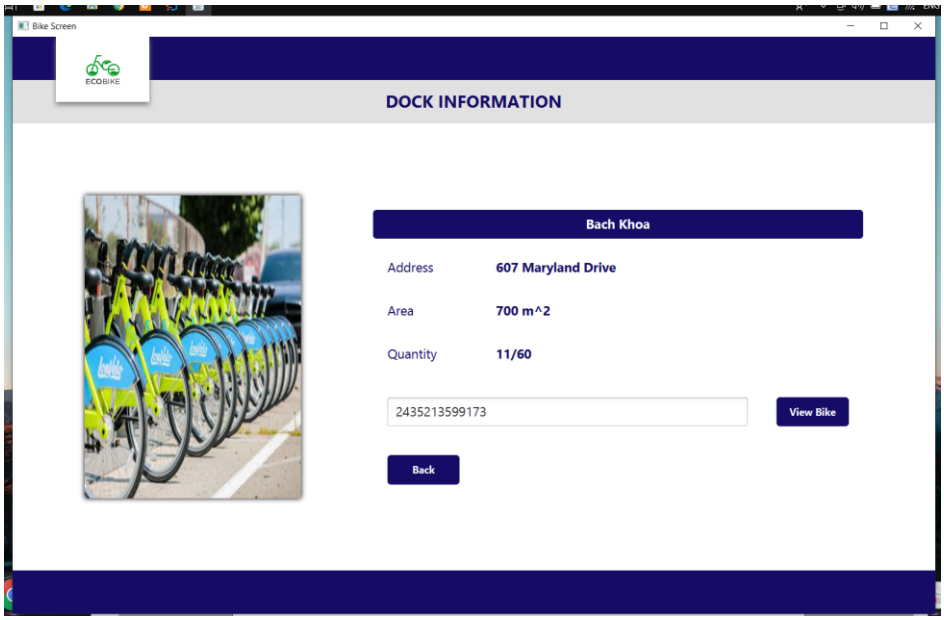
| | | | |
|---|---|---|---|
|  | Search button | Click | Search dock with entered data |

**Table 2: Define the field attributes – Home screen**

| **Screen name** | **Home screen** | | | |
|---|---|---|---|---|
| **Item name** | **Number of digits (bytes)** | **Type** | **Field attribute** | **Remarks** |
| Dock ID | 5 | Numeral | Black | Left-justified |
| Dock Name | 50 | String | Black | Left-justified |
| Address | 100 | String | Black | Center |
| Area | 5 | Numeral | Black | Right-justified |
| Quantity | 5 | Numeral | Black | Right-justified |
| Search Dock | 100 | String (address) | Box | Center |

**Table 3: Screen Specification – The detailed information of a dock screen**

| EcoBikeRental Software | | Date of creation | Approved by | Reviewed by | Personal in charge |
|---|---|---|---|---|---|
| **Screen Specification** | **The detailed information of a dock** | **08/11/2021** | | **Vũ Văn Long** | **Trần Xuân Trường** |



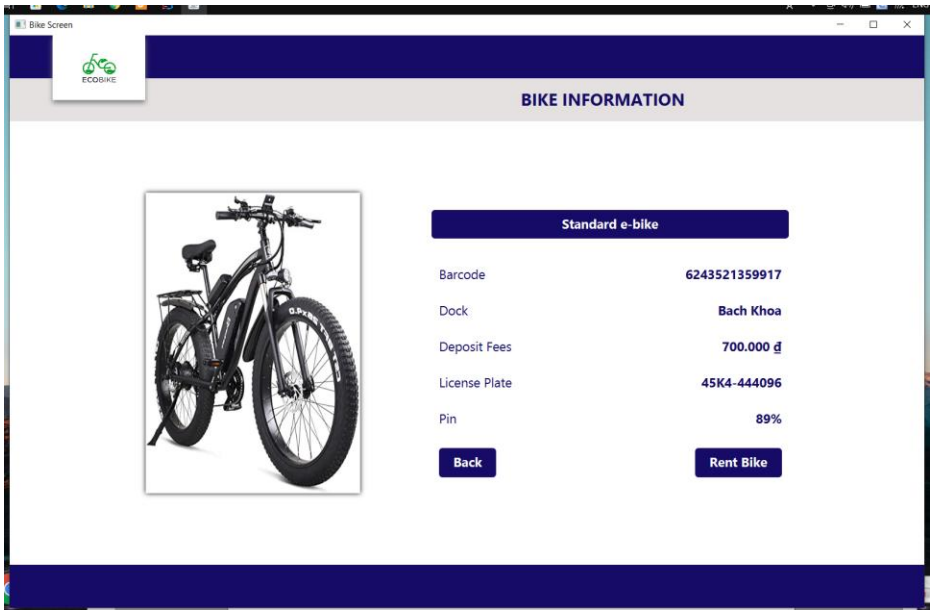

| Control | Operation | Function |
|---|---|---|
| Area of displaying the title | Initial | Display the title: DOCK INFORMATION |
| Area of displaying the dock details | Initial | Display a dock details screen |
| Enter barcode area | Enter barcode | Enter barcode of a bike |
| View bike button | Click | Display detailed information of a bike |
| Back button | Click | Back to list of docks/HOME screen |

**Table 4: Define the field attributes – The detailed information of a dock**

| **Screen name** | **The detailed information of a dock** | | | |
|---|---|---|---|---|
| **Item name** | **Number of digits (bytes)** | **Type** | **Field attribute** | **Remarks** |
| Dock Name | 50 | String | White | Center |
| Address | 100 | String | Blue | Left-justified |
| Area | 5 | Numeral | Blue | Left-justified |
| Quantity | 5 | Numeral | Blue | Left-justified |
| Enter barcode area | 15 | Numeral | Box | Left-justified |

**Table 5: Screen Specification – The detailed information of a bike**

| EcoBikeRental Software | | Date of creation | Approved by | Reviewed by | Personal in charge |
|---|---|---|---|---|---|
| **Screen Specification** | **The detailed information of a bike** | **08/11/2021** | | **Vũ Văn Long** | **Trần Xuân Trường** |



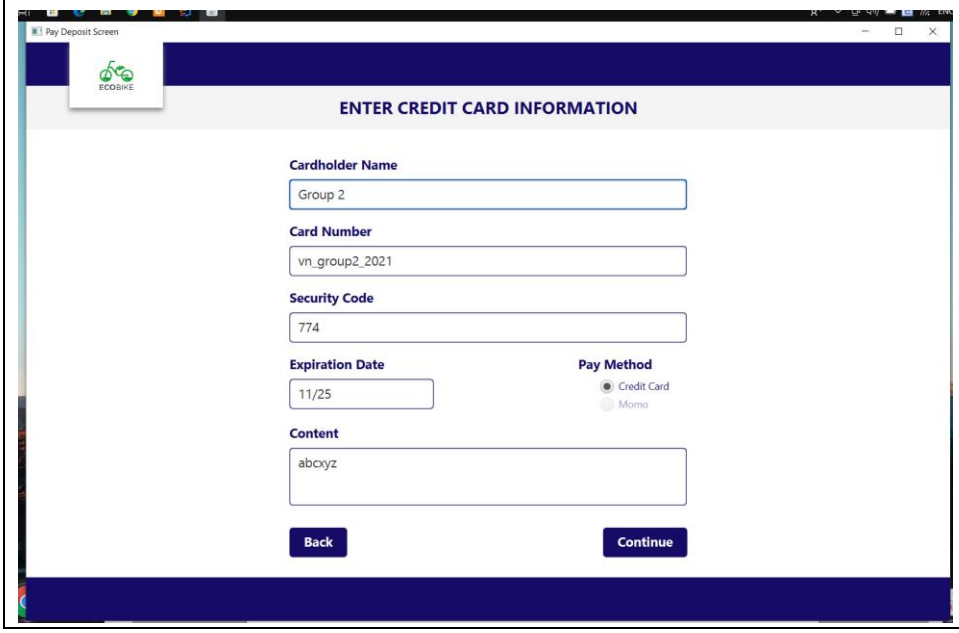

| Control | Operation | Function |
|---|---|---|
| Area of displaying the title | Initial | Display the title BIKE DETAILS |
| Area of displaying the bike details | Initial | Display the bike details |
| 'Rent Bike' button | Click | Display ENTER CREDIT CARD INFORMATION screen |
| 'Back' button | Click | Back to DOCK INFORMATION screen |

**Table 6: Define the field attributes – The detailed information of a bike**

| **Screen name** | **The detailed information of a bike** | | | |
|---|---|---|---|---|
| **Item name** | **Number of digits (bytes)** | **Type** | **Field attribute** | **Remarks** |
| Bike type name | 50 | String | White | Center |
| Barcode | 10 | Numeral | Blue | Right-justified |
| License plate | 20 | String | Blue | Right-justified |
| Bike type | 50 | String | Blue | Right-justified |
| Dock name | 50 | String | Blue | Right-justified |
| Deposit | 10 | Numeral | Blue | Right-justified |

**Table 7: Screen Specification – Credit card information screen**

| EcoBikeRental Software | | Date of creation | Approved by | Reviewed by | Personal in charge |
|---|---|---|---|---|---|
| Screen Specification | Credit card information screen | 09/11/2021 | | Vũ Văn Long | Trần Xuân Trường |



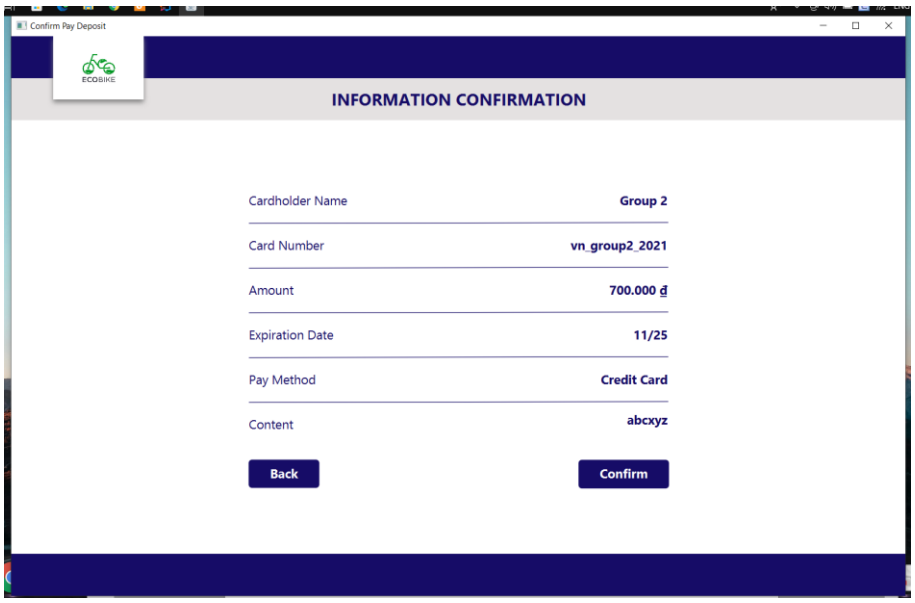

| Control | Operation | Function |
|---|---|---|
| Area of displaying the title | Initial | Display the title ENTER CREDIT CARD INFORMATION |
| Area of displaying the card information | Initial | Display the card information |
| 'Back' button | Click | Back to BIKE INFORMATION screen |
| 'Continue' button | Click | Display the INFORMATION CONFIRMATION screen |

**Table 8: Define the field attributes – Credit card information screen**

| **Screen name** | **Credit card information screen** | | | |
|---|---|---|---|---|
| **Item name** | **Number of digits (bytes)** | **Type** | **Field attribute** | **Remarks** |
| Cardholder Name | 50 | String | Box | Center |
| Card Number | 50 | Numeral | Box | Center |
| Security Code | 10 | Numeral | Box | Center |
| Expiration Date | 4 | Date: month/year<br>mm/yy | Box | Center |
| Content | 200 | String | Box | Center |

**Table 9: Screen Specification – Information confirmation screen**

| EcoBikeRental Software | | Date of creation | Approved by | Reviewed by | Personal in charge |
|---|---|---|---|---|---|
| Screen Specification | Information Confirmarion screen | 09/11/2021 | | Vũ Văn Long | Trần Xuân Trường |

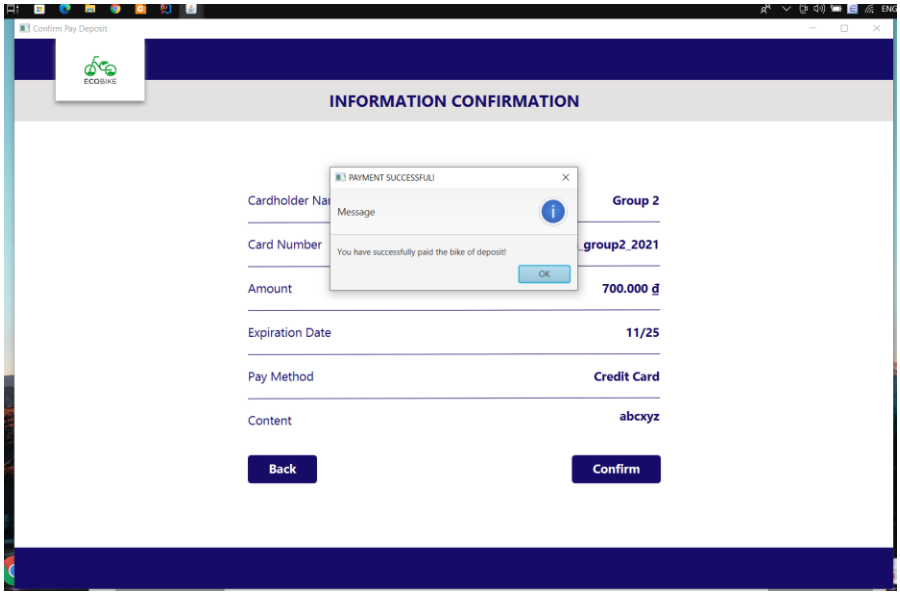



| Control | Operation | Function |
|---|---|---|
| Area of displaying the title | Initial | Display the title INFORMATION CONFIRMATION |
| Area of displaying the information need to confirm | Initial | Display the pay deposit bike transaction need to confirm |
| 'Back' button | Click | Back to ENTER CREDIT CARD INFORMATION screen |
| 'Confirm' button | Click | Confirm the credit card information. Display the notification. |

**Table 10: Define the field attributes – Information confirmation screen**

| **Screen name** | **Information Confirmation screen** | | | |
|---|---|---|---|---|
| **Item name** | **Number of digits (bytes)** | **Type** | **Field attribute** | **Remarks** |
| Cardholder Name | 50 | String | Blue | Center |
| Card Number | 20 | Numeral | Blue | Center |
| Expiration Date | 20 | Date: day/month/year | Blue | Center |
| Amount | 10 | Numeral | Blue | Center |
| Pay Method | 20 | String | Blue | Center |
| Content | 200 | String | Blue | Center |

## Table 11: Screen Specification - Notification screen

| EcoBikeRental Software | | Date of creation | Approved by | Reviewed by | Personal in charge |
|---|---|---|---|---|---|
| Screen Specification | Notification screen | 09/11/2021 | | Vũ Văn Long | Trần Xuân Trường |
|  | | **Control** | **Operation** | **Function** | |
| | | Area of displaying the notification of payment transaction | | Displaying the notification of payment transaction, display successful transaction or error transaction, if transaction has error, display the detailed error. | |
| | | 'OK' button | Click | If the payment transaction is successful, display next screen.<br>If not, display the current screen. | |

## Table 12: Define the field attributes – Notification screen

| **Screen name** | **Notification screen** | | | |
|---|---|---|---|---|
| **Item name** | **Number of digits (bytes)** | **Type** | **Field attribute** | **Remarks** |
| Status | 30 | String | White | |

## Table 13: Screen Specification– Bike state information screen

| EcoBikeRental Software | | Date of creation | Approved by | Reviewed by | Personal in charge |
|---|---|---|---|---|---|
| Screen Specification | Bike State Information | 09/11/2021 | | Vũ Văn Long | Trần Xuân Trường |
| | | Control | Operation | Function | |

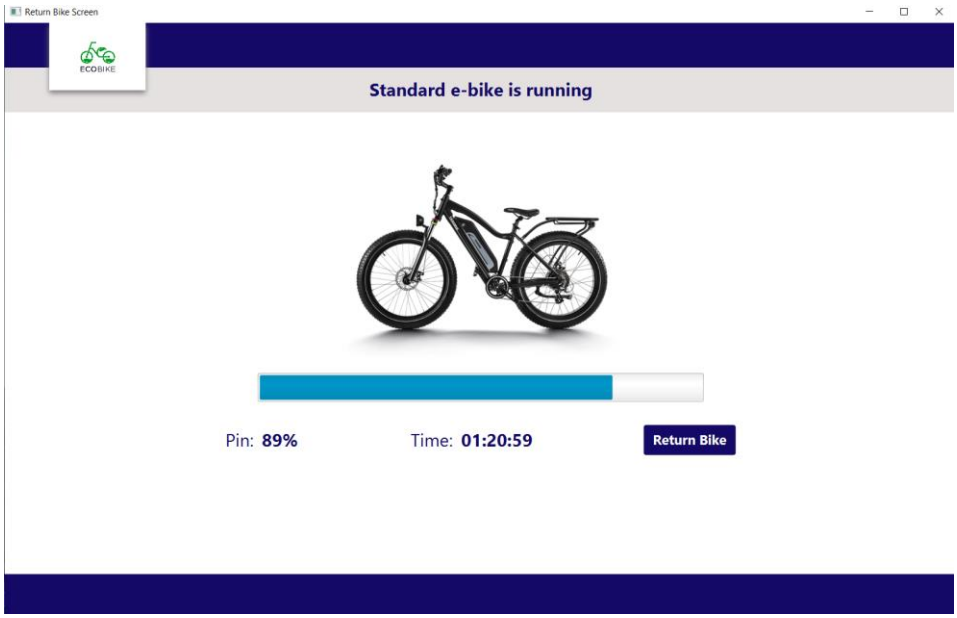
| | | | |
|---|---|---|---|
|  | Area of displaying the title | Initial | Display the title |
| | Area of displaying the bike state details | Initial | Display the bike current state details that the customer is renting |
| | 'Return Bike' button | Click | The customer stops to return that bike. Display RETURN BIKE screen |

**Table 14: Define the field attributes – Bike state information screen**

| **Screen name** | **Bike State Information** | | | |
|---|---|---|---|---|
| **Item name** | **Number of digits (bytes)** | **Type** | **Field attribute** | **Remarks** |
| Pin | 10 | String | Blue | Left-justified |
| Time | 10 | String | Blue | Center |

**Table 15: Screen Specification – Return bike screen**

| **EcoBikeRental Software** | | **Date of creation** | **Approved by** | **Reviewed by** | **Personal in charge** |
|---|---|---|---|---|---|
| **Screen Specification** | **Return bike screen** | **09/11/2021** | | **Vũ Văn Long** | **Trần Xuân Trường** |
| | | **Control** | **Operation** | **Function** | |
| | | Area of displaying the title | Initial | Display the title RETURN BIKE | |
| | | Area of displaying the return | Initial | Display the return bike information | |

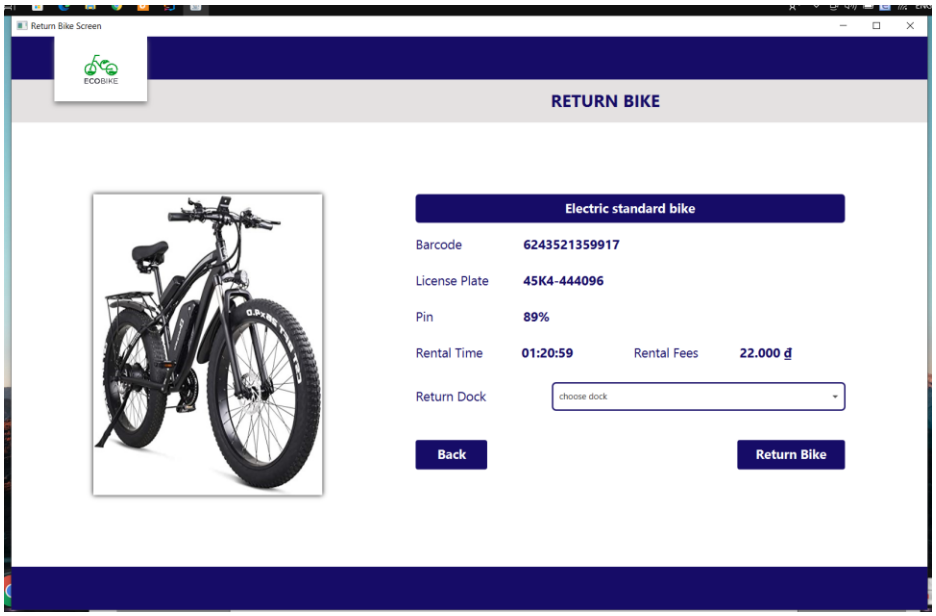
| | | | |
|---|---|---|---|
|  | bike information | | |
| | ‘Back’ button | Initial | Back to current state of the bike screen |
| | ‘Return Bike’ button | Initial | Confirm to return bike. Display the INVOICE screen |

**Table 16: Define the field attributes – Return bike screen**

| **Screen name** | **Return bike screen** | | | |
|---|---|---|---|---|
| **Item name** | **Number of digits (bytes)** | **Type** | **Field attribute** | **Remarks** |
| Barcode | 10 | Numeral | Blue | Center |
| License Plate | 20 | String | Blue | Center |
| Rental Time | 10 | String | Blue | Left-justified |
| Pin | 10 | String | Blue | Center |
| Rental Fees | 10 | Numeral | Blue | Right-justified |
| Return Dock | 50 | String | List of dock | Right justified |

**Table 17: Screen Specification – Invoice screen**

| **EcoBikeRental Software** | | **Date of creation** | **Approved by** | **Reviewed by** | **Personal in charge** |
|---|---|---|---|---|---|
| **Screen Specification** | **Invoice screen** | **09/11/2021** | | **Vũ Văn Long** | **Trần Xuân Trường** |
| | | **Control** | **Operation** | **Function** | |
| | | Area of displaying the title | Initial | Display the title INVOICE | |

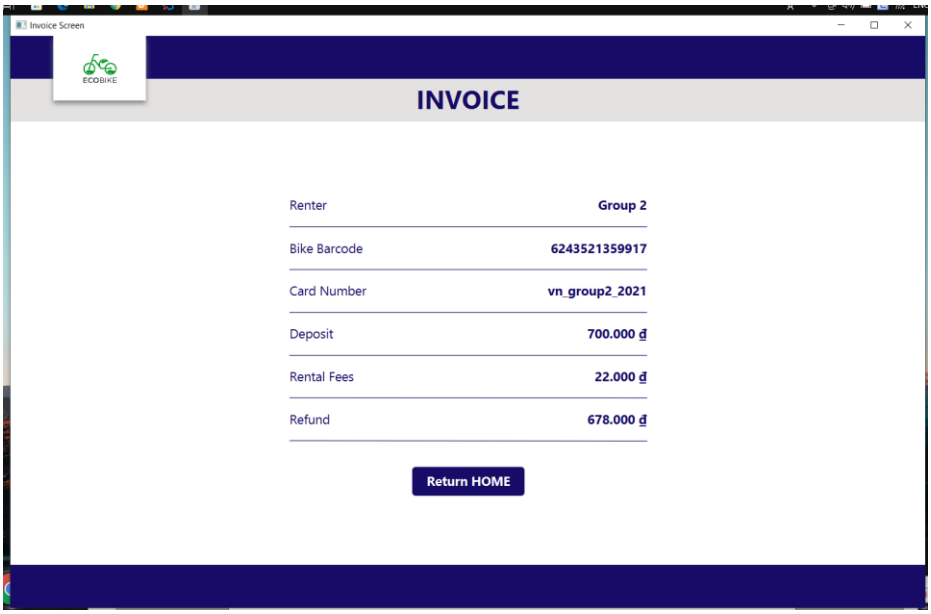
| | | | |
|---|---|---|---|
|  | Area of displaying the invoice information | Initial | Display the invoice information |
| | Return HOME button | Click | Notify success transaction and return to HOME screen |

**Table 18: Define the field attributes – Invoice screen**

| **Screen name** | **Invoice screen** | | | |
|---|---|---|---|---|
| **Item name** | **Number of digits (bytes)** | **Type** | **Field attribute** | **Remarks** |
| Barcode | 10 | Numeral | Blue | Center |
| Renter | 50 | String | Blue | Center |
| Card Number | 20 | Numeral | Blue | Center |
| Deposit | 10 | Numeral | Blue | Center |
| Rental fees | 10 | Numeral | Blue | Center |
| Refund | 10 | Numeral | Blue | Center |

## *4.2 Data Modeling*

### 4.2.1 Conceptual Data Modeling

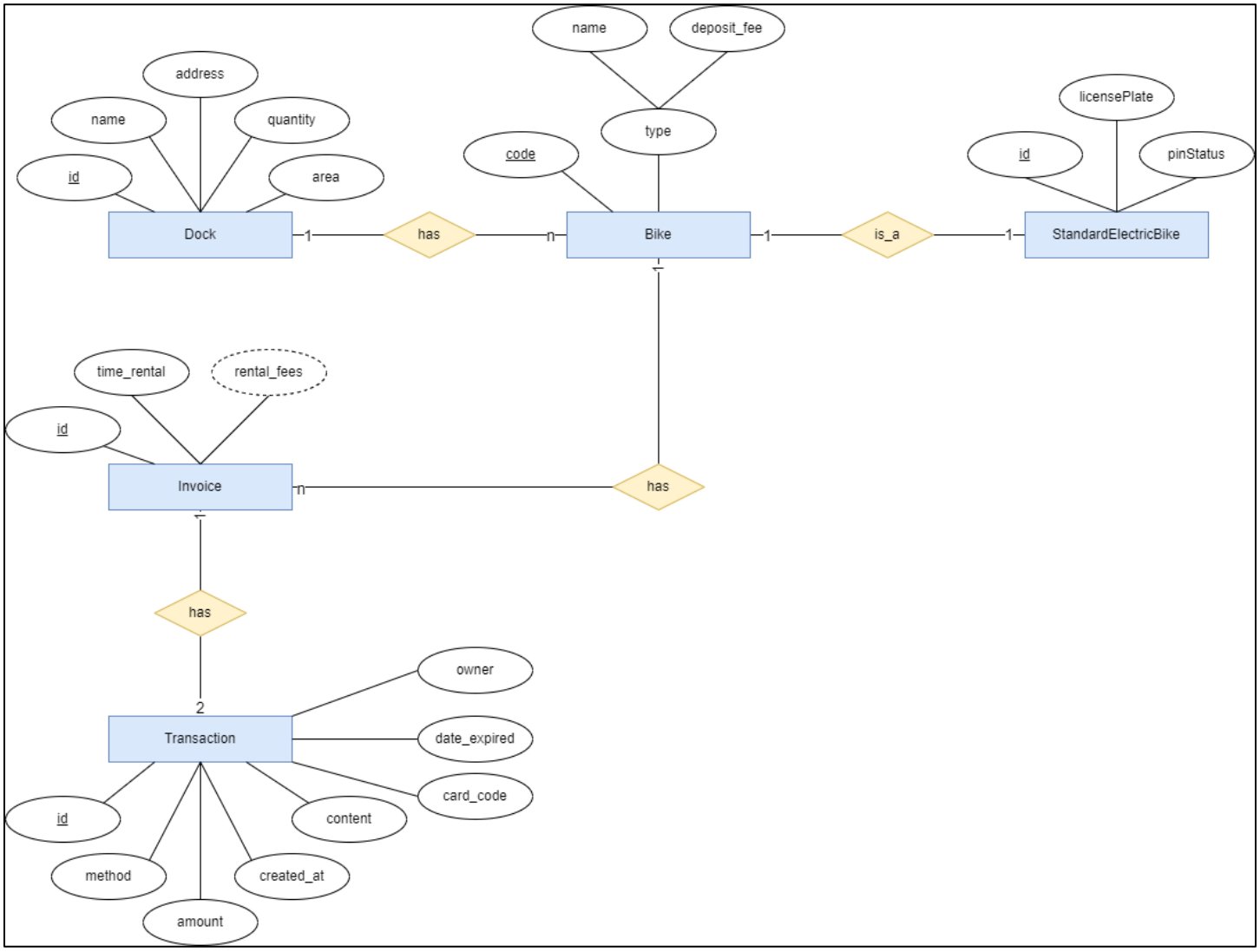



**Figure 15: Conceptual Data Modeling**

### 4.2.2 Database Design

#### *4.2.2.1 Database Management Systems*

#### *4.2.2.2 Logical Data Model*

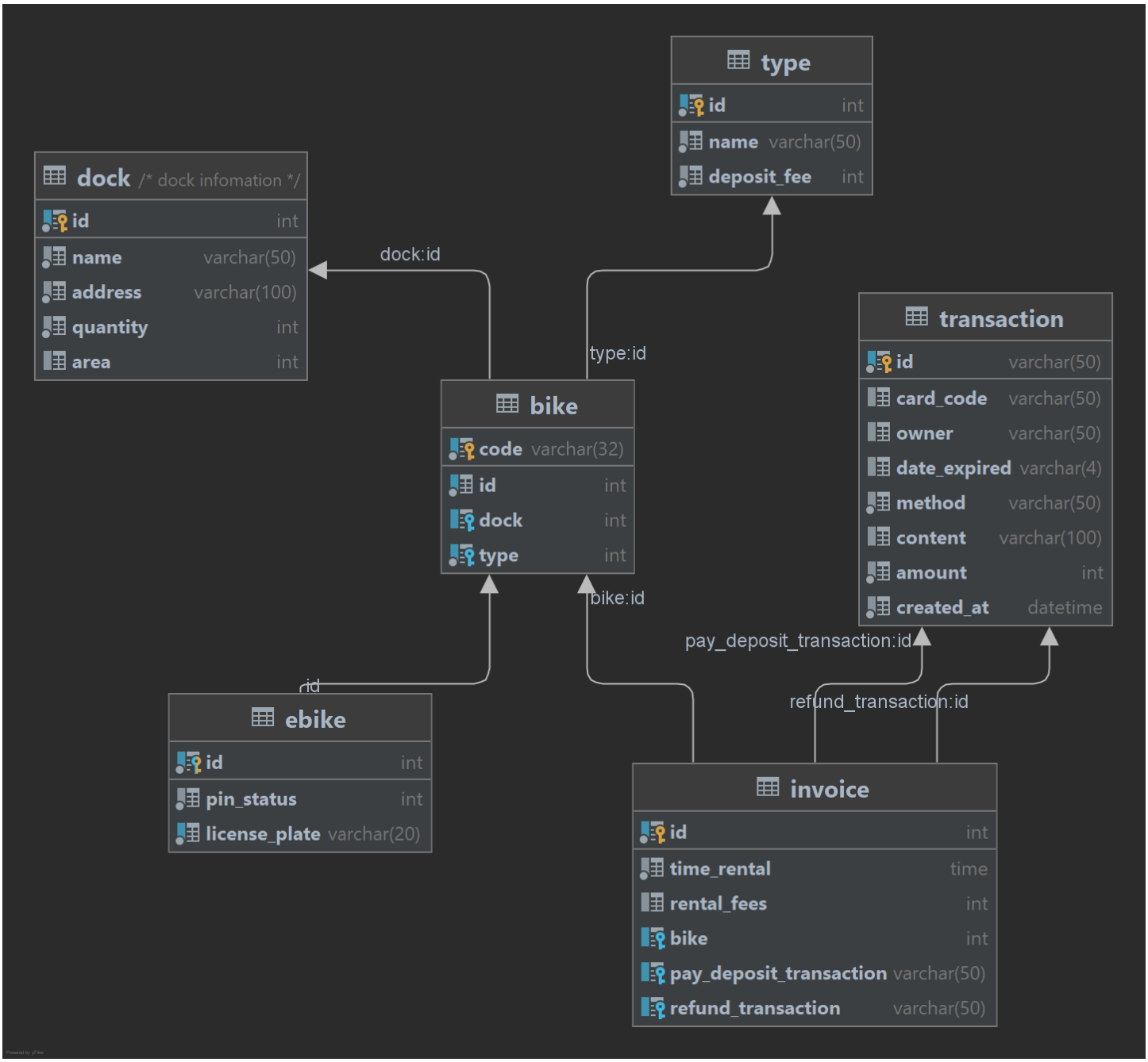



**Figure 16: Logical Data Model**

#### *4.2.2.3 Physical Data Model*

**Table 19: Dock DB**

| # | PK | FK | Column Name | Data Type | Mandatory | Description |
|---|---|---|---|---|---|---|
| **1** | x | | id | Integer | Yes | ID of dock |
| **2** | | | name | VARCHAR(50) | Yes | Name of dock |
| **3** | | | address | VARCHAR(100) | Yes | Address of dock |
| **4** | | | quantity | Integer | Yes | Quantity of bikes at the dock |
| **5** | | | area | Integer | Yes | Area of dock |

**Table 20: Bike DB**

| # | PK | FK | Column Name | Data Type | Mandatory | Description |
|---|---|---|---|---|---|---|
| **1** | x | | code | VARCHAR(32) | Yes | Code of bike |
| | | | id | Integer | Yes | Id of bike |
| **4** | | x | type | Integer | Yes | ID of bikeType |
| **5** | | x | dock | Integer | No | ID of dock where bike is located |

**Table 21: EBike DB**

| # | PK | FK | Column Name | Data Type | Mandatory | Description |
|---|---|---|---|---|---|---|
| **1** | x | x | id | Integer | Yes | ID, same of the ID of bikeType is Standard e-bike |
| **2** | | | license_plate | VARCHAR(20) | Yes | License plate of bike |
| **3** | | | pin_status | Integer | Yes | Current battery percentage of e-bike |

**Table 22: Type DB**

| # | PK | FK | Column Name | Data Type | Mandatory | Description |
|---|---|---|---|---|---|---|
| **1** | x | | id | Integer | Yes | ID of bikeType |
| **2** | | | typeName | VARCHAR(50) | Yes | Name of bikeType |
| **3** | | | deposit_fee | Integer | Yes | Fee to deposit when renting |

**Table 23: Invoice DB**

| # | PK | FK | Column Name | Data Type | Mandatory | Description |
|---|---|---|---|---|---|---|
| **1** | x | | id | Integer | Yes | ID |
| **2** | | | time_rental | VARCHAR(10) | Yes | Rental time |
| **3** | | | rental_fees | Integer | Yes | Rental fees |
| **4** | | x | bike | Integer | Yes | ID of the bike |
| **5** | | x | pay_deposit_transaction | Integer | Yes | ID of the pay deposit transaction |
| **6** | | x | refund_transaction | Integer | Yes | ID of the refund transaction |

**Table 24: Transaction DB**

| # | PK | FK | Column Name | Data Type | Mandatory | Description |
|---|---|---|---|---|---|---|
| **1** | x | | id | Integer | Yes | ID |
| **2** | | | method | VARCHAR(50) | Yes | Method: pay or refund |
| **3** | | | amount | Integer | Yes | Amount |
| **4** | | | created_at | DATETIME | Yes | Date of creation |
| **5** | | | content | VARCHAR(100) | No | Transaction content |
| **6** | | | owner | VARCHAR(50) | Yes | Name of renter |
| **7** | | | card_code | VARCHAR(50) | Yes | Code of the credit card |
| **8** | | | date_expired | VARCHAR(4) | Yes | Expired date of the credit card |

## *4.3 Non-Database Management System Files*

## *4.4 Class Design*

### 4.4.1 General Class Diagram

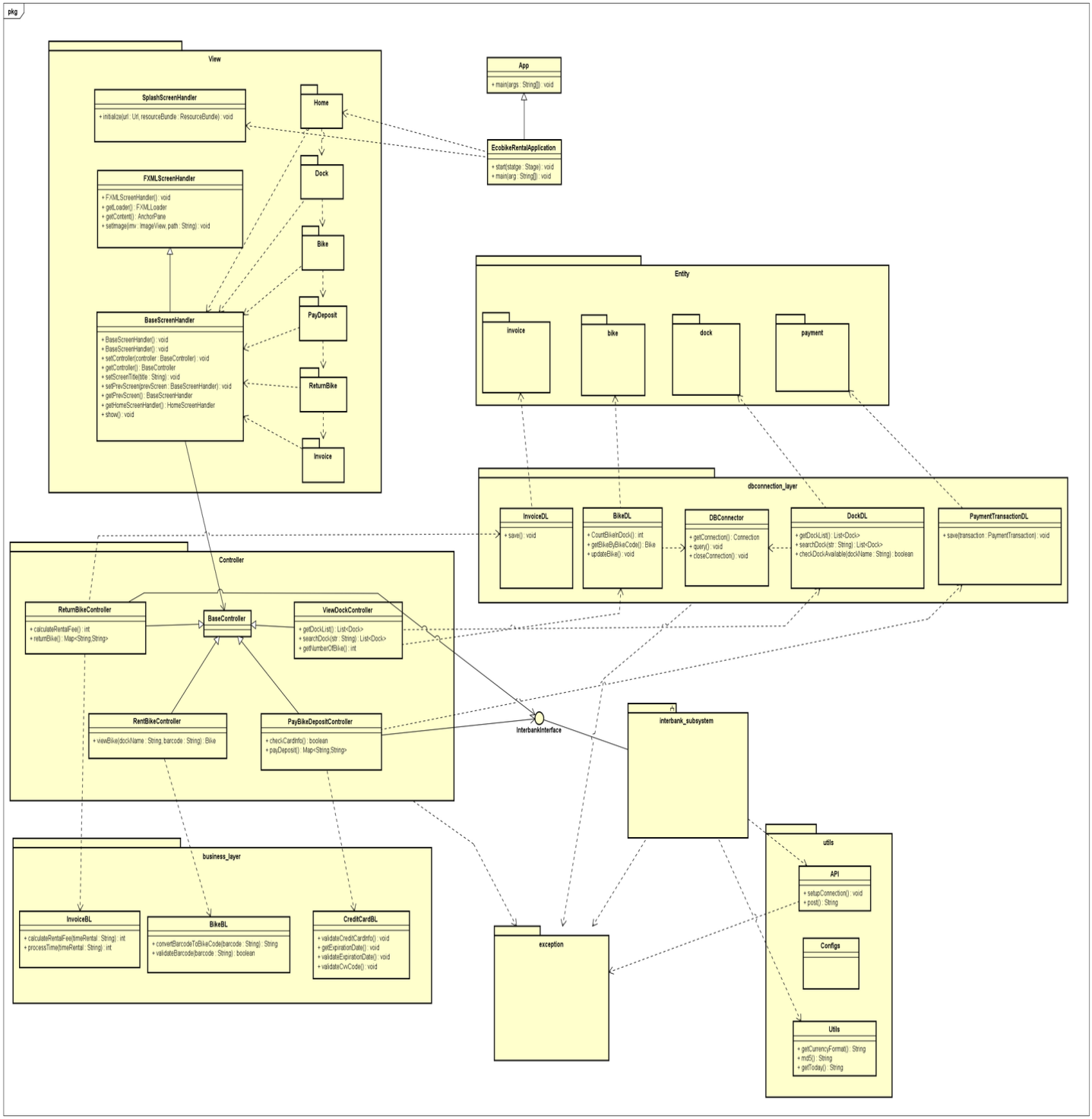



**Figure 17: General Class Diagram**

### 4.4.2 Class Diagrams

#### *4.4.2.1 Class Diagram for Interbank Subsystem*

**Figure 18: Identify InterbankSubsystem**

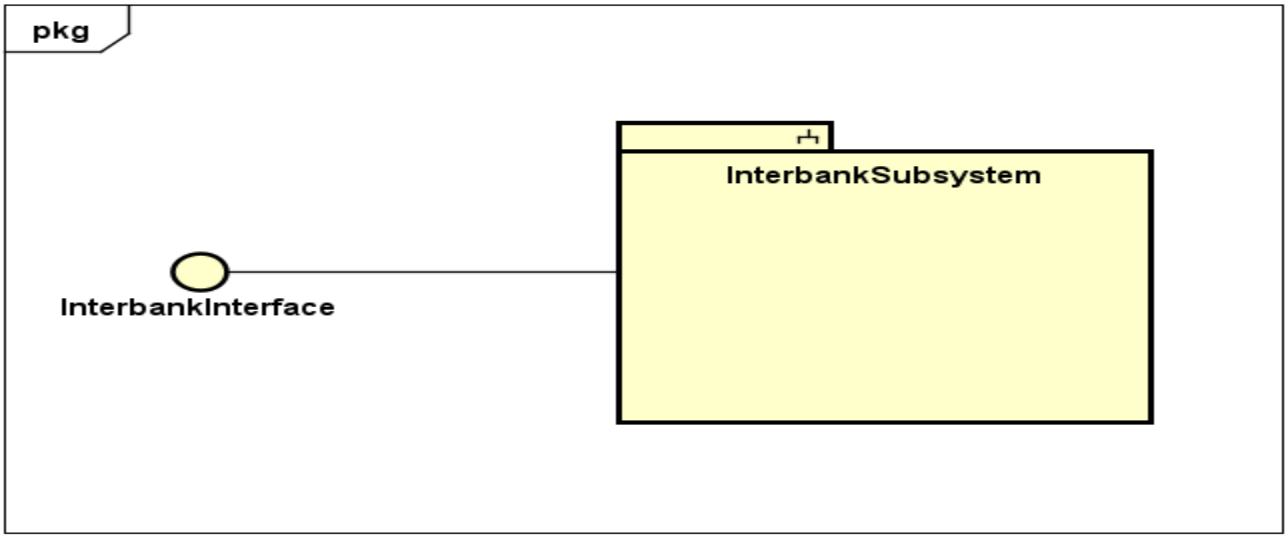



**Figure 19: Identify InterbankSubsystem Interface**

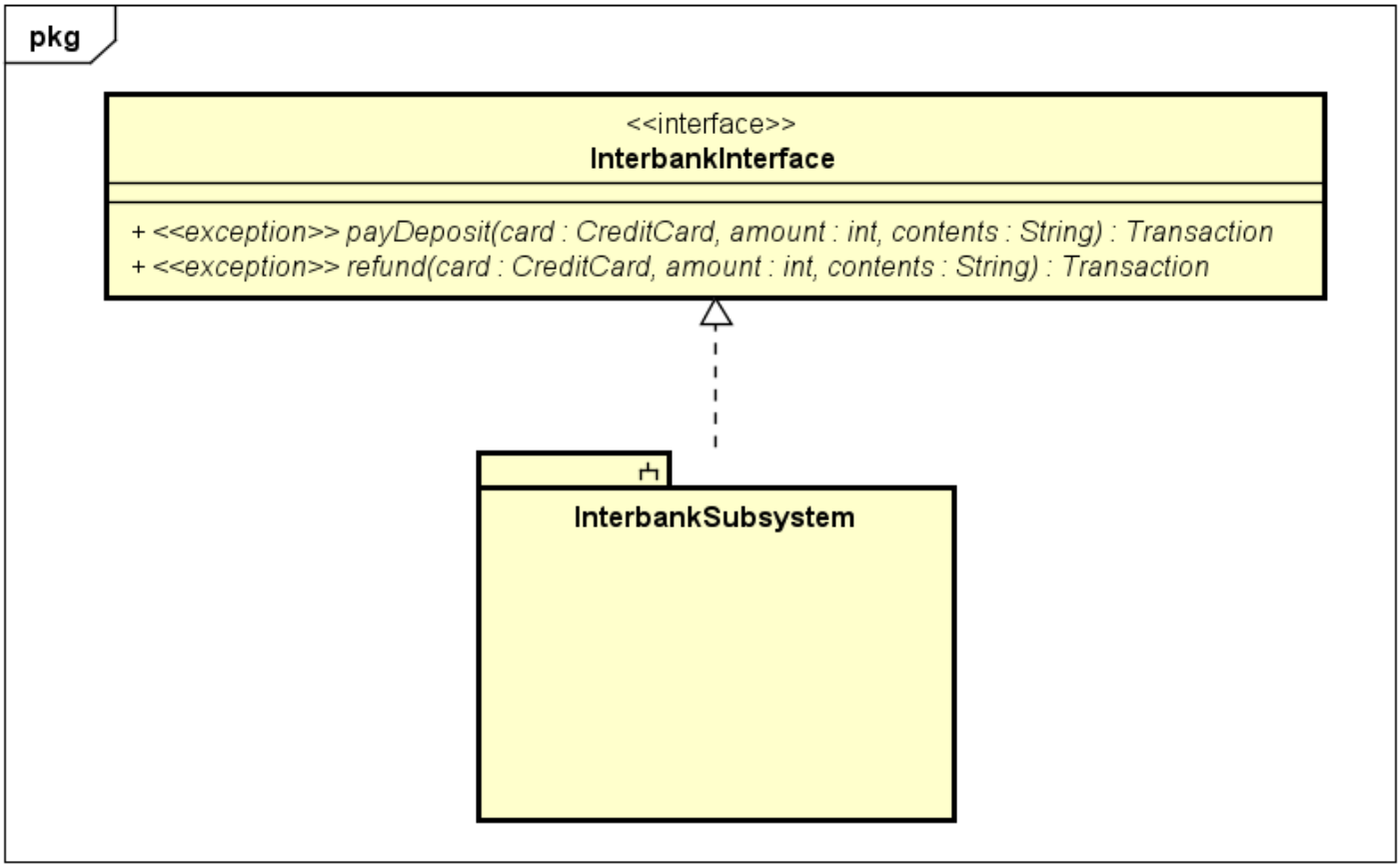
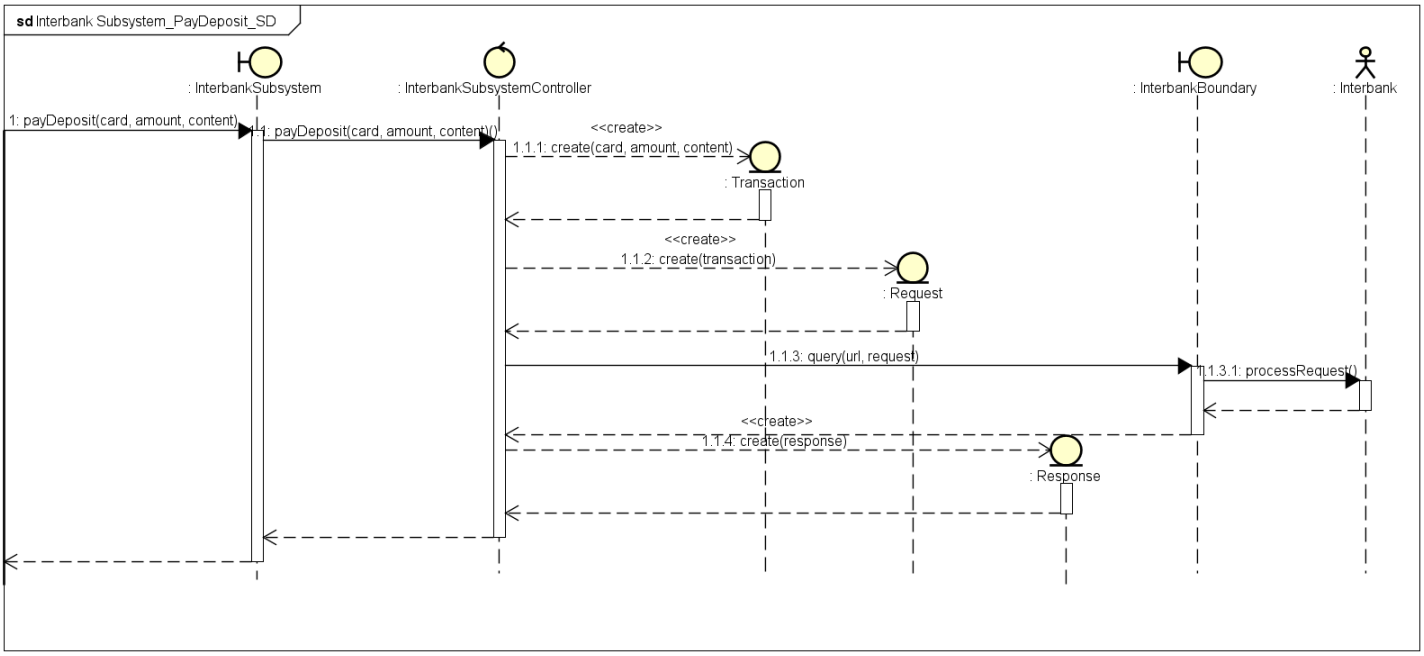


**Figure 20: Distribute subsystem behavior to subsystem elements**



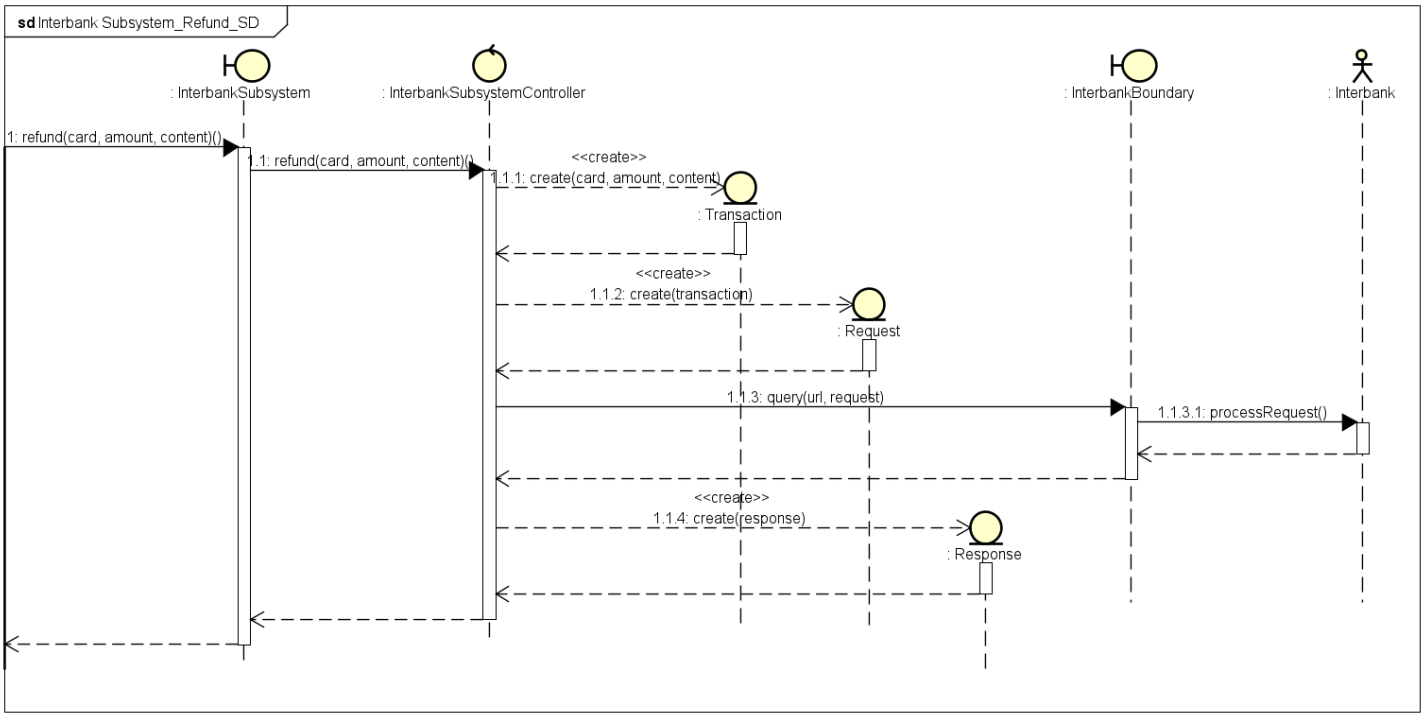



**Figure 21: Document subsystem elements**

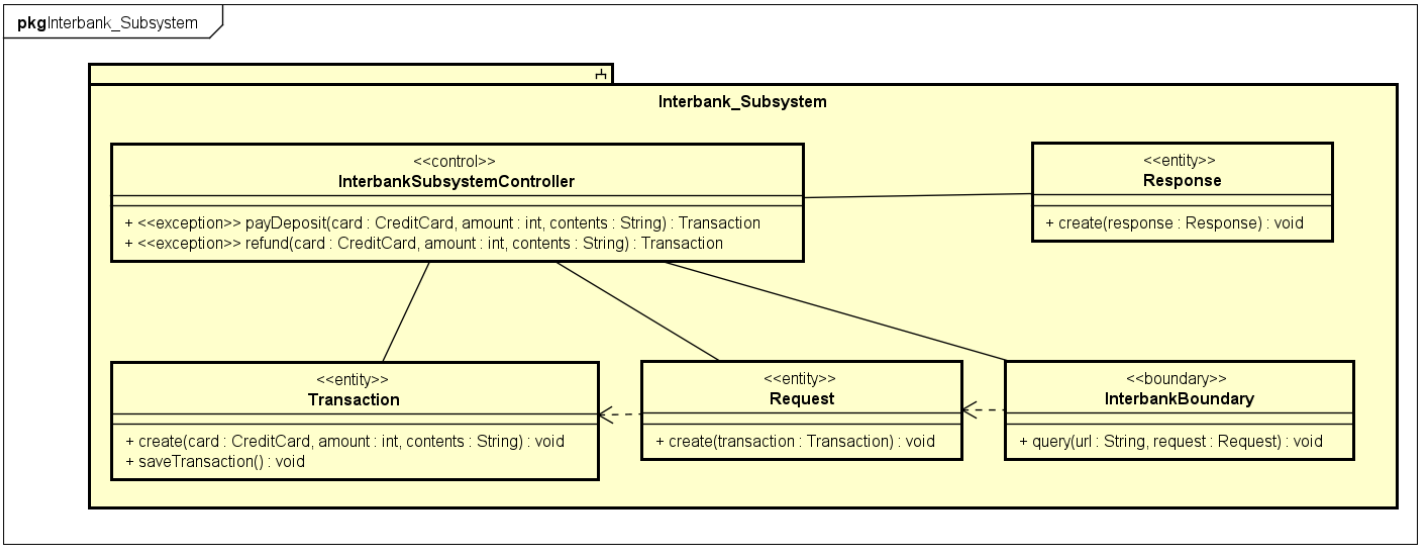



**Figure 22: Describe subsystem dependencies**

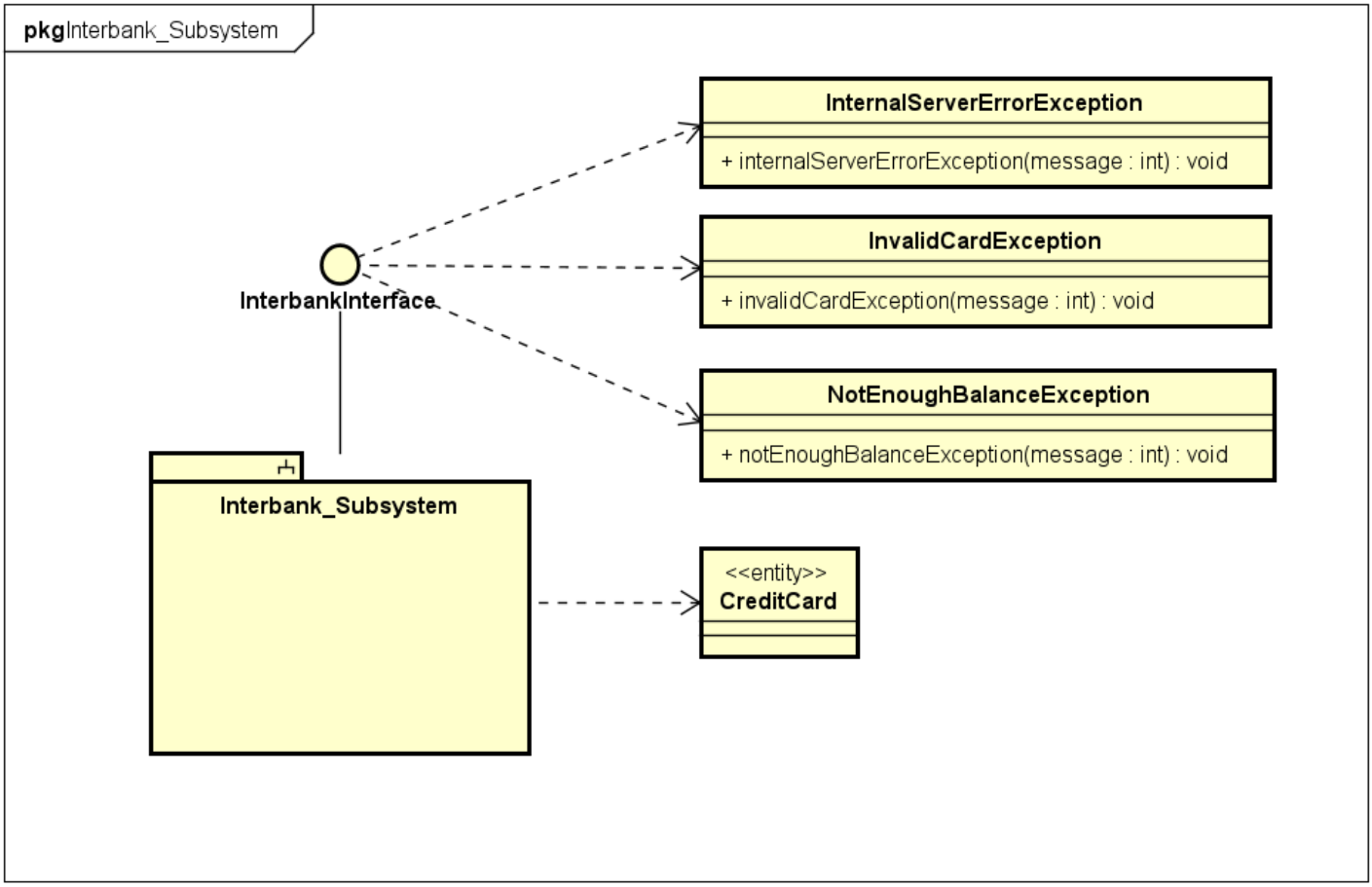
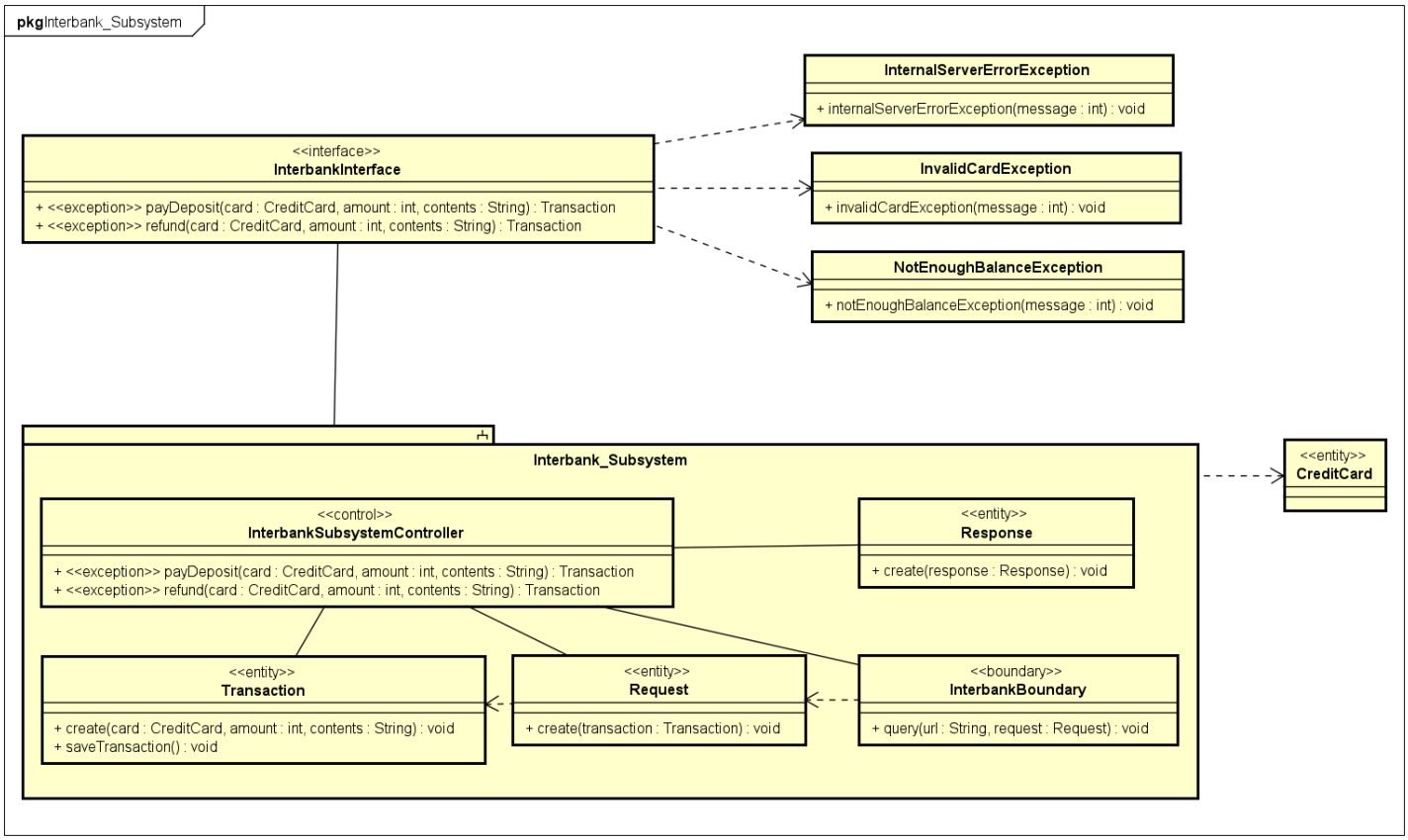
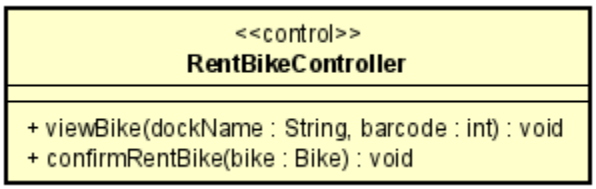


**Figure 23: Checkpoints**




### 4.4.3 Class Design

#### *4.4.3.1 Class "RentBikeController"*




**Attribute**

None

**Operation**

| *#* | *Name* | *Return type* | *Description* |
|---|---|---|---|
| *1* | viewBike | void | Display the bike information in dockName with this barcode |
| *2* | confirmRentBike | void | User confirm, system process |

*Parameter:*

1. viewBike()
    a. Parameter
        - dockName: name of dock
        - barcode: the barcode that user entered
    b. Exception
        - None
2. confirmRentBike
    a. Parameter
        - bike: the bike with information
    b. Exception

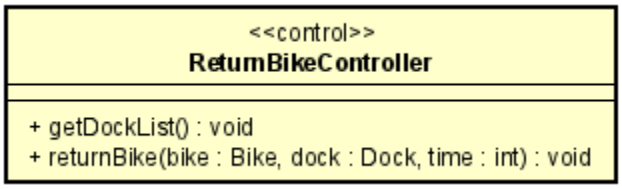
**Method**

1. viewBike(): System process barcode, convert to bike code then display bike information.
2. confirmRentBike(): Get bike information to pay order.

**State**

None

#### *4.4.3.2 Class "ReturnBikeController"*




**Attribute**

None

**Operation**

| # | *Name* | *Return type* | *Description (purpose)* |
|---|---|---|---|
| 1 | getDockList | void | Get list of docks |
| 2 | returnBike | void | Return the bike to dock |

*Parameter*:

- bike – the renting bike
- time – bike rental time

- dock – selected dock to return the bike

*Exception*:

None

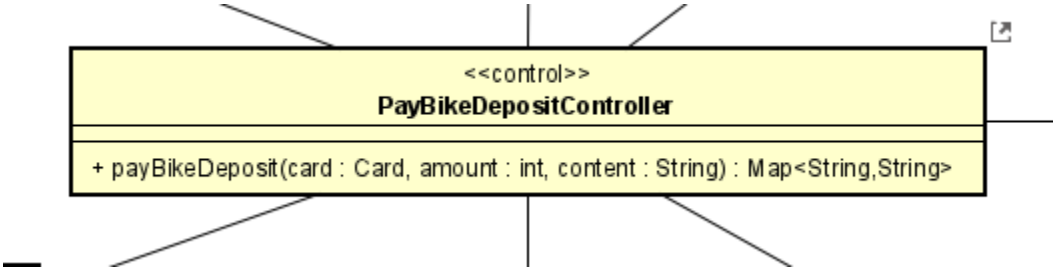
**Method**

None

**State**

None

#### *4.4.3.3 Class "PayBikeDepositController"*




**Attribute**

None

**Operation**

| # | *Name* | *Return type* | *Description (purpose)* |
|---|---|---|---|
| 1 | payBikeDeposit() | Map<String, String> | Return message success or fail when pay deposit |

*Parameter*:

- Card – card of user
- amount – money to deposit
- content – content of deposit

*Exception*:

None

**Method**

None

**State**

None

# 5 Design Considerations

## *5.1 Goals and Guidelines*

- Goals:
  - Bring a good looking and good experience for users
  - The response time for the system is 1 second at normal and 2 seconds during a peak load
- Guidelines
  - Observe java convention in coding, OOP principles
  - Avoid hash code
  - Explain code, write java doc for maintenance

## *5.2 Architectural Strategies*

Our design decisions focus on reusing components, unified system following

- Programing Language: Java
- Database: MySQL
- Unified on error detection and recovery

We always toward save memory and spaces, also speed up response time and nice looking. In the future, we plan to extend software: have site for admin to add, delete bike, statistics, business strategies. These targets make us concentrate totally on architectural design.

## *5.3 Coupling and Cohesion*

- Coupling
  - Content:
    - HelloApplication - FadeTransition, HomeScreenHandle (Set value in other module)

```
fadeOut.setOnFinished(event -> {
    try {
        HomeScreenHandler homeHandler = new HomeScreenHandler(stage, Configs.HOME_SCREEN_PATH);
        homeHandler.setScreenTitle("Home Screen");
        homeHandler.show();
    } catch (IOException e) {
        e.printStackTrace();
    }
```

- Cohesion:
  - Procedural:
    - RentBikeController – Barcode process + Invoice process (Steps, not relate)

### *5.4 Design Principles*

- RentBikeController: Vi phạm nguyên lí S.
  Method không liên quan đến nhau -> có thể chia thành các module BarcodeController, InvoiceController.
  Các method validate không nên đặt trong lớp này -> Đặt sang lớp Utils chuyên về validate

=> Đưa vào lớp Business để chứa các xử lí liên quan đến Barcode và Validate

```
public class BikeBL {
    /**
     * this method validate format barcode.
     *
     * @param barcode - barcode of bike
     *
     * @return - true if format barcode is valid else false
     */
    public static boolean validateBarcode(final String barcode) {
        if (barcode == null || barcode.length() != 13) {
            return false;
        }
        return barcode.matches("^[0-9]+");
    }

    /**
     * @param barcode -
     *
     * @return String: bike code
     */
    public static String convertBarcodeToBikeCode(final String barcode) {
        if (!validateBarcode(barcode)) {
            return null;
        }
        return barcode + "123456" + barcode;
    }
```

- EntityClass: Vi phạm nguyên lí S
  Không nên có các phương thức lấy dữ liệu từ database trong lớp entity -> Thêm các lớp DatabaseConnectionLayer để lấy dữ liệu từ database

```java
public class BikeDL {
    private List<Bike> bikeList;

    public BikeDL() throws SQLException {
        this.bikeList = new ArrayList<>();
        String sql =
            "select bike.id as id, code, type.name as type, dock.name as dock_name,
        ResultSet res = DBConnector.query(sql);
        Bike bike;
        while (res.next()) {
            bike = new BikeFactory().getBike(res.getString("type"));
            bike.setBikeId(res.getInt("id"));
            bike.setBikeCode(res.getString("code"));
            bike.setType(res.getString("type"));
            bike.setDockName(res.getString("dock_name"));
            bike.setDeposit(res.getInt("deposit_fee"));
            bikeList.add(bike);
//              System.out.println(bike.toString());
        }
    }
```

- InterBankSubsystem: Vi phạm nguyên lí O
  Khi thanh toán, phương thức payDeposit và refund đều được truyền CreditCart, muốn thêm phương thức thanh toán khác sẽ phải sửa mã nguồn -> Thay bằng một lớp abstract.

### *5.5 Design Patterns*

- *Singleton:*
  - *Invoice*
  - *Dock*
- *Factory:*
  - *Bike (Standard Bike – Twin Bike)*

```java
public class Invoice {
    /**
     * Represent for bike
     */
    private Bike bike;
    private String rentalTime;
    private int rentalFee;
    /**
     * Represent for payment transaction.
     */
    private PaymentTransaction payDepositTransaction;
    private PaymentTransaction refundTransaction;

    private static Invoice instance = new Invoice();

    private Invoice() {
    }

    /**
     * This method saves the invoice.
     */
    public void save() {
```

```
public class BikeFactory {
    public Bike getBike(String type) {
        if (type.equals("Standard bike") || type.equals("Twin bike")) {
            return new Bike();
        } else if (type.equals("Standard e-bike")) {
            return new ElectricBike();
        }
        return null;
    }
}
```