HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

School of Information and communications technology

Software Design Document

Version 1.3

EcoBike Application

Subject: ITSS Software Management

Group 6

| No. | Student Name | Student ID |
| --- | --- | --- |
| 1 | Nguyen Thi Minh Chau | 20184238 |
| 2 | Tran Le Hai Duong | 20184248 |
| 3 | Nguyen Thanh Long | 20184287 |

*Hanoi, 01/2022*

# Table of Contents

[**Table of Contents**](#_gx1nopg5943e) **3**

[**List of Figures**](#_6h5t8z315scf) **6**

[**List of Tables**](#_oumk3ckg6521) **7**

[**1. Introduction**](#_8q4adbvgu9zl) **8**

[1.1. Objective](#_fnnt24ot1eqi) 8

[1.2. Scope](#_rcv793thro0p) 8

[1.3. Glossary](#_v809z21pa8w) 8

[1.4. References](#_xmhv9mx0on3j) 9

[**2. Overall Description**](#_q2qg99kednq) **10**

[2.1. General Overview](#_cqcgoa37vh1) 10

[2.2. Assumptions/Constraints/Risks](#_1e98v535ep7a) 11

[2.2.1. Assumptions](#_79ygrcg6yli4) 11

[2.2.2. Constraints](#_caayy0q93xqy) 11

[2.2.3. Risks](#_9pu1ub11yq5o) 11

[**3. System Architecture and Architecture Design**](#_qnp241ql79m1) **12**

[3.1. Architectural Patterns](#_ywsc1gx70fcq) 12

[3.2. Interaction Diagrams](#_y2lswh3zwro7) 12

[3.2.1. Communication Diagrams](#_m836lie65ar4) 12

[3.2.2. Sequence Diagrams](#_8xtz4cdr5q3g) 15

[3.3. Analysis Class Diagrams](#_2opymiu37umd) 18

[3.4. Unified Analysis Class Diagram](#_cqb17kba9hog) 21

[3.5. Security Software Architecture](#_72v5u1xdgvwg) 21

[**4. Detailed Design**](#_u1gr57nvdbpy) **22**

[4.1. User Interface Design](#_uitbrso1jumn) 22

[4.1.1. Screen Configuration Standardization](#_mx7lpg5f3m7w) 22

[4.1.2. Screen Transition Diagrams](#_62w4491mhudw) 22

[4.1.3. Screen Specifications](#_csvdn4igq4x7) 22

[4.1.3.1. Splash Screen](#_m49fcuafcxjy) 22

[4.1.3.2. Main Screen](#_jb9nqunwjef) 23

[4.1.3.3. Dock Screen](#_b0ooxvodt4fo) 23

[4.1.3.4. Bike Screen](#_1fvmw06pqv63) 24

[4.1.3.5. Payment Method Screen](#_2kzq3jq1ljh) 25

[4.1.3.6. Deposit screen](#_gnpg9df5fc0u) 25

[4.1.3.7. Payment screen](#_a2kjl3twkisn) 25

[4.2. Data Modeling](#_nep01yykpou9) 27

[4.2.1. Conceptual Data Modeling](#_5mzguh3t2f32) 27

[4.2.2. Database Design](#_kmxjh3er0s8h) 27

[4.2.2.1. Database Management System](#_xlmrgcsvemos) 27

[4.2.2.2. Database Diagram](#_9ozeafh11a0r) 28

[4.2.2.3. Database Detail Design](#_agi51g46nnsp) 28

[4.3. Non-Database Management System Files](#_h6a63it86r9x) 35

[4.4. Class Design](#_y1yk30evber3) 35

[4.4.1. General Class Diagram](#_527h4ccj8g45) 35

[4.4.2. Class Diagrams](#_fbis8074m97o) 36

[4.4.2.1. Class Diagram for Package BikeInformation](#_ixhuw5uhml38) 36

[4.4.2.2. Class Diagram for Subsystem RentBike](#_1uv8ryw9ljbu) 37

[4.4.2.3. Class Diagram for Subsystem InterBank](#_cve699nnty55) 37

[4.4.3. Class Design](#_h4c5sevm9hwg) 38

[4.4.3.1. Class RentBikeController](#_yhimh3uoior6) 38

[4.4.3.2. Class ReturnBikeController](#_vel20b335w0) 39

[4.4.3.3. Class PaymentController](#_pdimm7irpvf3) 41

[4.4.3.4. Class BikeInformationScreenHandler](#_rehtlt43cs2u) 43

[**5. Design Considerations**](#_sz97q180ce8z) **45**

[5.1. Goals and Guidelines](#_9z4lweuesmmu) 45

[5.2. Architectural Strategies](#_ezj2ww1ltcw8) 45

[5.3. Coupling and Cohesion](#_9j5qab63z9zp) 45

[5.3.1. Coupling](#_6byvdp3fz7xu) 45

[5.3.1.1. Content coupling](#_aylu86up1s81) 45

[5.3.1.2. Common coupling](#_3bgsw8lf2e48) 46

[5.3.1.3. Control coupling](#_58q9jhes7d5) 46

[5.3.1.4. Stamp coupling](#_1kcoatsyo1iy) 46

[5.3.1.5. Data coupling](#_ftaymwj1sqhw) 46

[5.3.2. Cohesion](#_r43hhlnggau2) 47

[5.3.2.1. Coincidental cohesion](#_yjezyq6xorcn) 47

[5.3.2.2. Logical cohesion](#_10arwcec9l99) 47

[5.3.2.3. Temporal cohesion](#_cwcrbievaplo) 47

[5.3.2.4. Procedure cohesion](#_97wu953v2p9l) 47

[5.3.2.5. Communicational cohesion](#_j5caldij94u0) 47

[5.3.2.6. Sequential cohesion](#_4llg7w3p7uky) 48

[5.3.2.7. Information cohesion](#_nfdyht8dadr) 48

[5.3.2.8. Functional cohesion](#_6xjld66ijm42) 48

[5.4. Design Principles](#_q8wbdxhdduxm) 48

[5.5. Design Patterns](#_y403t1elj8ie) 48

# List of Figures

*Figure 1.1: General use case diagram 11*

*Figure 3.1: Communication Diagram for Rent Bike Use Case 13*

*Figure 3.2: Communication Diagram for Deposit Use Case 13*

*Figure 3.3: Communication Diagram for Update Payment Method Use Case 14*

*Figure 3.4: Communication Diagram for Return Bike Use Case 14*

*Figure 3.5: Communication Diagram for Return Deposit Use Case 15*

*Figure 3.6: Communication Diagram for Pay For Rental Use Case 15*

*Figure 3.7: Sequence Diagram for Rent Bike Use Case 16*

*Figure 3.8: Sequence Diagram for Deposit Use Case 16*

*Figure 3.9: Sequence Diagram for Update Payment Method Use Case 17*

*Figure 3.10: Sequence Diagram for Return Bike Use Case 18*

*Figure 3.11: Sequence Diagram for Return Deposit Use Case 18*

*Figure 3.12: Sequence Diagram for Pay For Rental Use Case 19*

*Figure 3.13: Class Diagram for View Bike Use Case 20*

*Figure 3.14: Class Diagram for View Bike Use Case 21*

*Figure 3.15: Class Diagram for Deposit Use Case 21*

*Figure 3.16: Class Diagram for Return Bike Use Case 22*

*Figure 3.17: Class Diagram for Return Deposit Use Case 22*

*Figure 3.18: Class Diagram for Pay Rental Use Case 23*

*Figure 3.19: Unified Class Diagram for EcoBike Application 24*

*Figure 4.1: Screen Transition Diagram for EcoBike Application 25*

*Figure 4.2. ER Diagram for EcoBike Application 29*

*Figure 4.3. Database Diagram for EcoBike Application 30*

*Figure 4.4. General Class Diagram for EcoBike Application 37*

*Figure 4.5. Class Diagram for Package BikeInformation 38*

*Figure 4.6. Class Diagram for Subsystem RentBike 39*

*Figure 4.7. Class Diagram for Subsystem InterBank 39*

*Figure 4.8. RentBikeController Class Diagram 40*

*Figure 4.9. Payment ControllerClass Diagram 42*

*Figure 4.10. BikeInformationController Class Diagram 45*

# List of Tables

*Table 1.1: Terms used in the document 10*

*Table 4.1. Splash Screen Specification 24*

*Table 4.2. Main Screen Specification 24*

*Table 4.3. View Dock Screen Specification 25*

*Table 4.4. View Bike Screen Specification 25*

*Table 4.5. Payment Method Screen Specification 26*

*Table 4.6. Deposit Screen Specification 26*

*Table 4.7. Payment Screen Specification 27*

*Table 4.8. Customer table design 29*

*Table 4.9. Administrator table design 30*

*Table 4.10. Dock table design 30*

*Table 4.11. Bike table design 31*

*Table 4.12. Bike In Dock table design 31*

*Table 4.13. Bike Status table design 32*

*Table 4.14. Invoice table design 32*

*Table 4.15. Transaction table design 33*

*Table 4.16. Rent Bike table design 33*

*Table 4.17. Credit Card table design 34*

*Table 4.18. RentBikeController attributes 39*

*Table 4.19. RentBikeController operations 40*

*Table 4.20. BikeTracker attributes 41*

*Table 4.21. BikeTracker operations 41*

*Table 4.24. BikeInformationScreenHandler operations 45*

*Table 4.25. BikeInformationScreenHandler operations 45*

# 1. Introduction

## 1.1. Objective

This Software Design Document provides the design of EcoBike Application. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli.

The expected audience is the user of the EcoBike Application, including Mrs. Trang and the ITSS Software development Course’s students, the developer of the project, and the people who will maintain the EcoBike Application.

## 1.2. Scope

This document contains a complete description of the design of EcoBike Application.

The EcoBike Application is a mobile application that allows the resident of Hanoi to use the EcoBike service, including finding out information of nearby docks and renting bikes for personal usage and online payment for the renting process.

The objective of the EcoBike Application is to serve a maximum of 100.000 users concurrently, with a friendly and easy-to-use user interface with the aim of helping the user to find the most suitable place to rent or return the bike.

## 1.3. Glossary

| **Term** | **Definition** |
| --- | --- |
| Administrator | The person who uses EcoBike application system for the purposes of monitoring list of bicycles in the system |
| Admin | as “administrator” |
| Bicycle | The transportation mean to be rent in this application system |
| Bike | as “bicycle” |
| Card number | The ID number of the credit card, printed on the credit card |
| Cardholder name | The name of the owner of the credit card, printed on the credit card |
| Credit card | A card connected to the interbank, used for performing transaction |
| Customer | The person who uses EcoBike application system for the purposes of renting bike |
| Database | Collection of all information monitored by this system |
| Deposit | An amount of money customer has to pay at first in order to rent a bike |
| Dock | A place where bicycles are put |
| Interbank | The organization in charges of performing payment and return deposit transactions in the system |
| Payment | An amount of money customer has to pay to rent a bike, including deposit and rental fee |
| Rent a bike | The action of using a bike in a period of time, with paying deposit and rental fee |
| Rental fee | An amount of money customer has to pay, outside of the deposit, which depends on the rental time |
| Rental time | The time period when the bike is being rented |
| Return a bike | The action of stopping using a bike after having rented |
| Software Requirement Specification | A document that completely describes all of the functions of a proposed system and the constraints under which it must operate. For example, this document. |
| Station | as “dock” |
| Transaction | The action of paying for bike deposit, bike rental or returning deposit |
| User | Customer or Administrator |

*Table 1.1: Terms used in the document*

## 1.4. References

| [1] | Centers for Medicare & Medicaid Services, "System Design Document Template," [Online]. Available: https://www.cms.gov/Research-Statistics-Data-and-Systems/CMS-Information-Technology/XLC/Downloads/SystemDesignDocument.docx. |
| --- | --- |

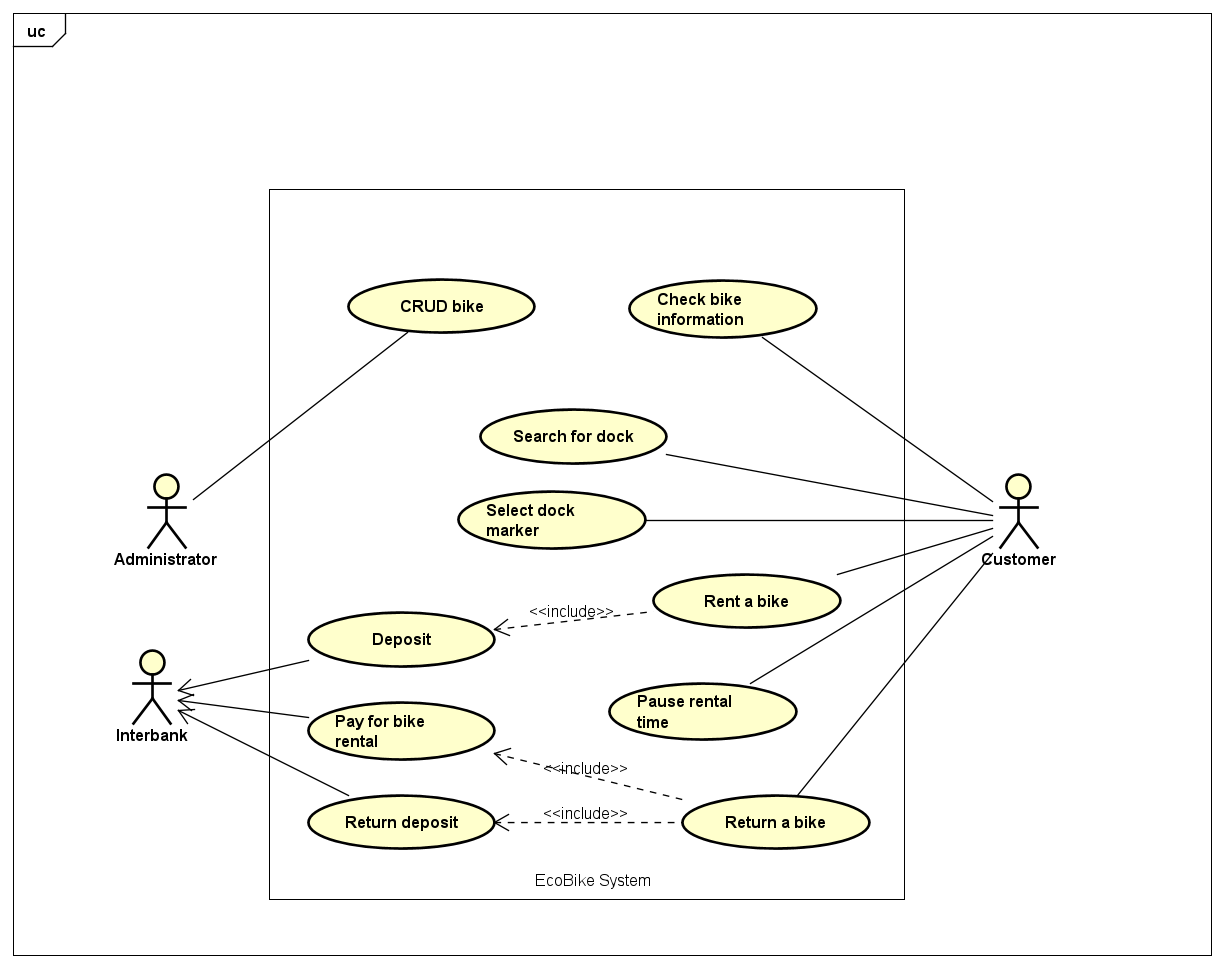
# 2. Overall Description

## 2.1. General Overview

EcoBike Application is a desktop application through which users can view docks and rent or return bikes. We design a clean and clear interface for users. Users can interact with the apps by clicking on the interface, and the request is processed by the controller.

Additionally, we have our own database to store information and data that is related to our system, as well as a subsystem to proceed payment transactions.

The below figure is the general use-case diagram for our design:



*Figure 1.1: General use case diagram*

## 2.2. Assumptions/Constraints/Risks

### 2.2.1. Assumptions

In order to use the application, users must have an internet connection as well as a personal computer to run the app. We would also require the latest version of JRE in order to ensure the application’ stability.

### 2.2.2. Constraints

· *Hardware or software environment*

· *End-user environment*

· *Availability or volatility of resources*

· *Standards compliance*

· *Interoperability requirements*

· *Interface/protocol requirements*

· *Licensing requirements*

· *Data repository and distribution requirements*

· *Security requirements (or other such regulations)*

· *Memory or other capacity limitations*

· *Performance requirements*

· *Network communications*

· *Verification and validation requirements (testing)*

· *Other means of addressing quality goals*

· *Other requirements described in the Requirements Document*

### 2.2.3. Risks

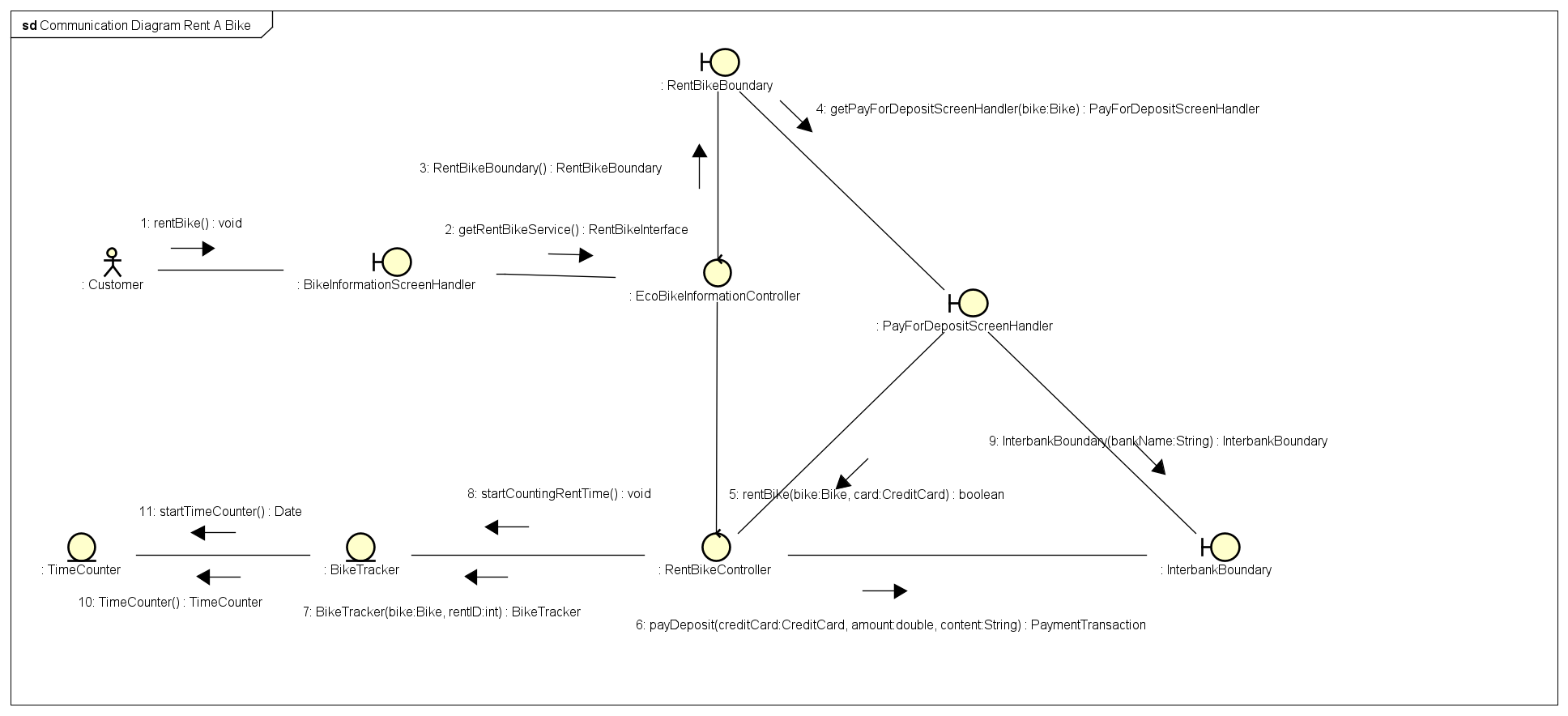
# 3. System Architecture and Architecture Design

## 3.1. Architectural Patterns

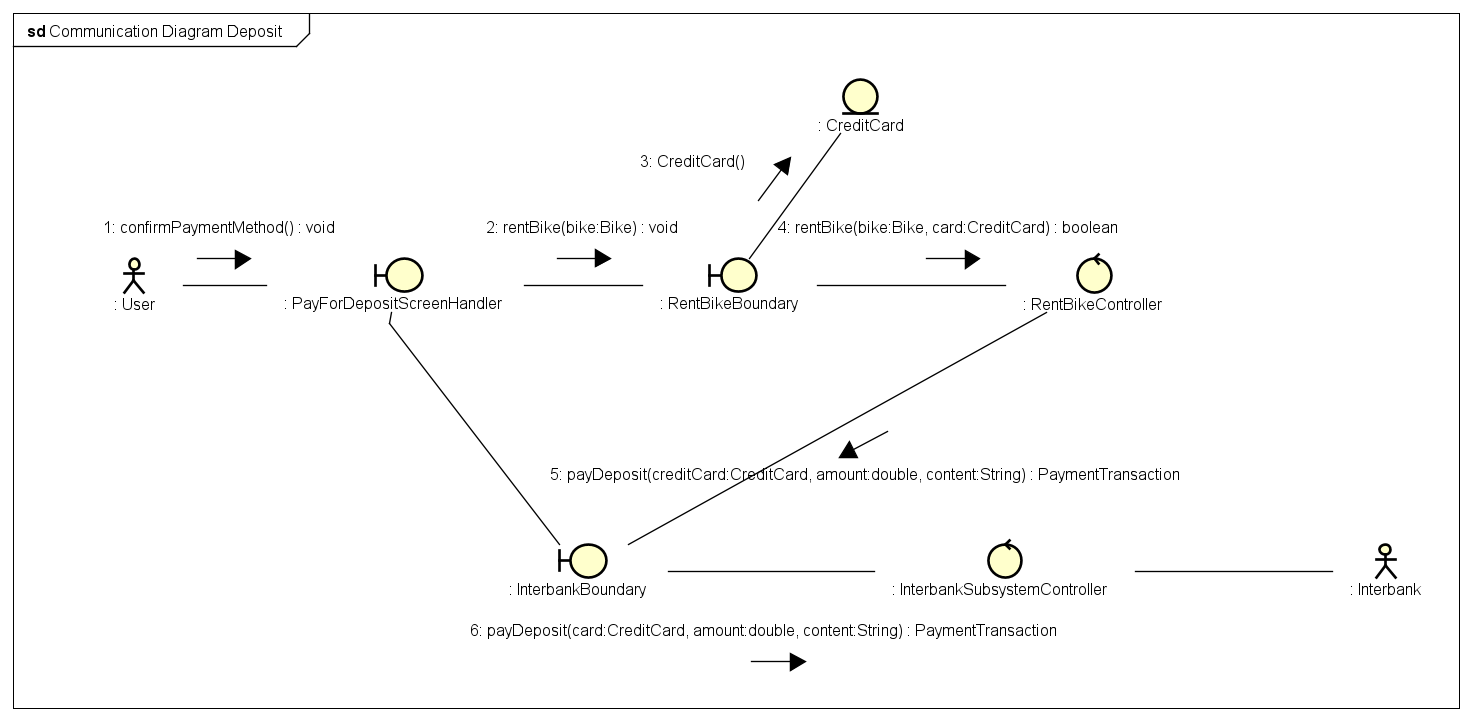
*<Specify and briefly describe the chosen architectural patterns and the reasons why they were chosen>*

## 3.2. Interaction Diagrams

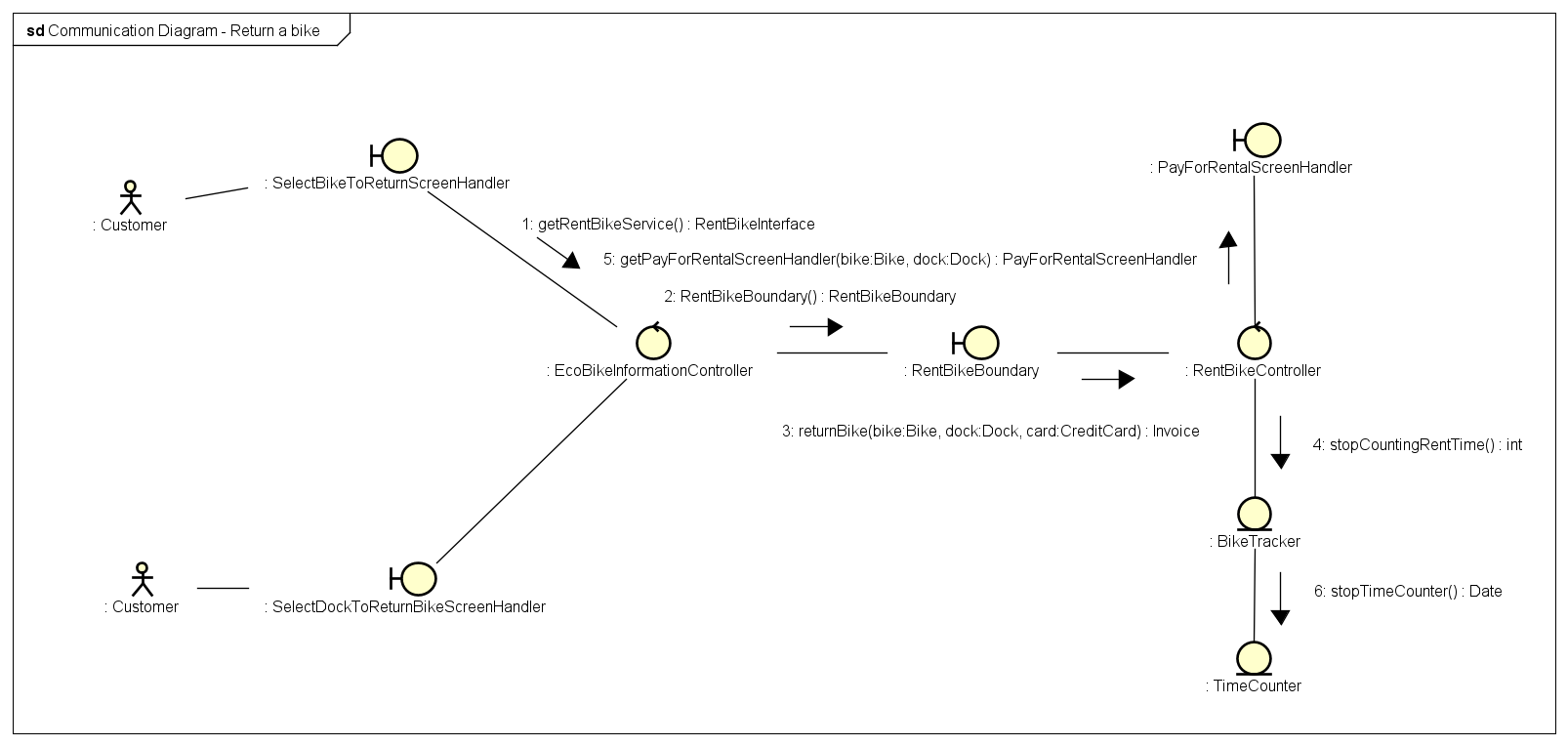
### 3.2.1. Communication Diagrams



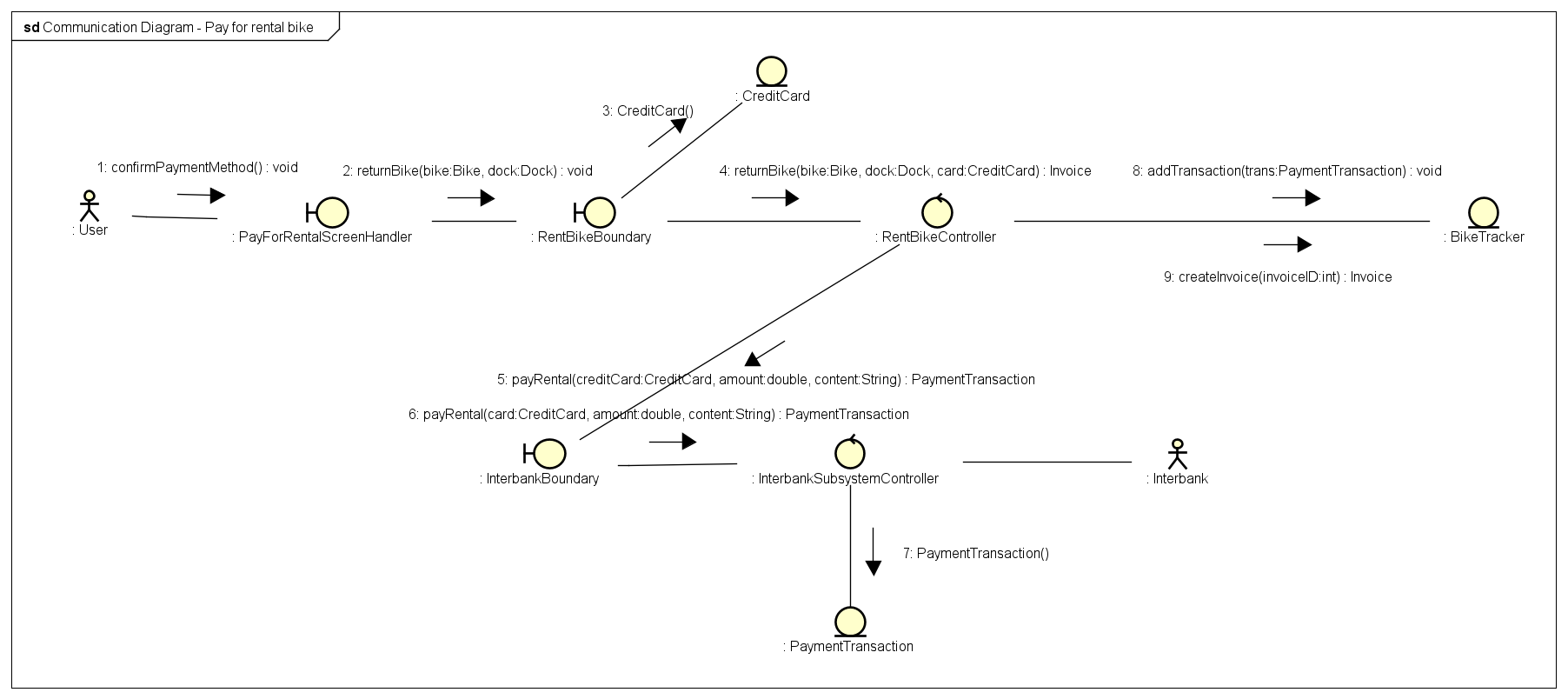
*Figure 3.1: Communication Diagram for Rent Bike Use Case*



*Figure 3.2: Communication Diagram for Deposit Use Case*

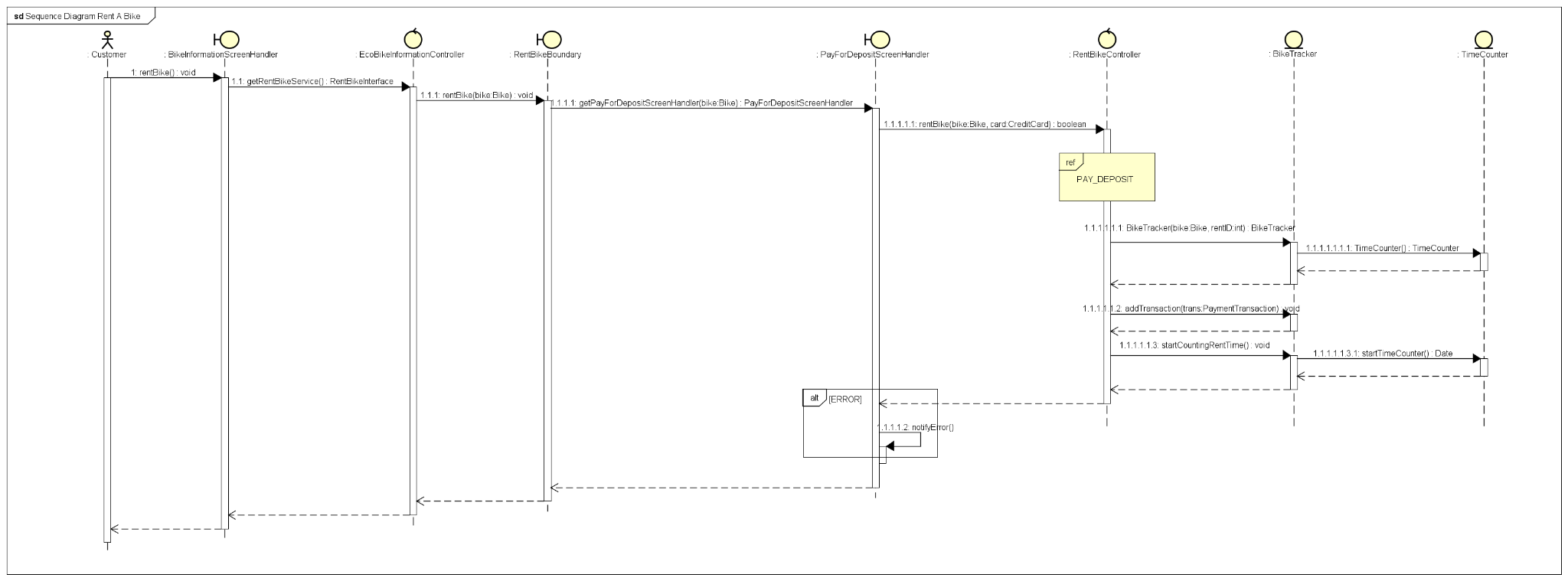


*Figure 3.4: Communication Diagram for Return Bike Use Case*

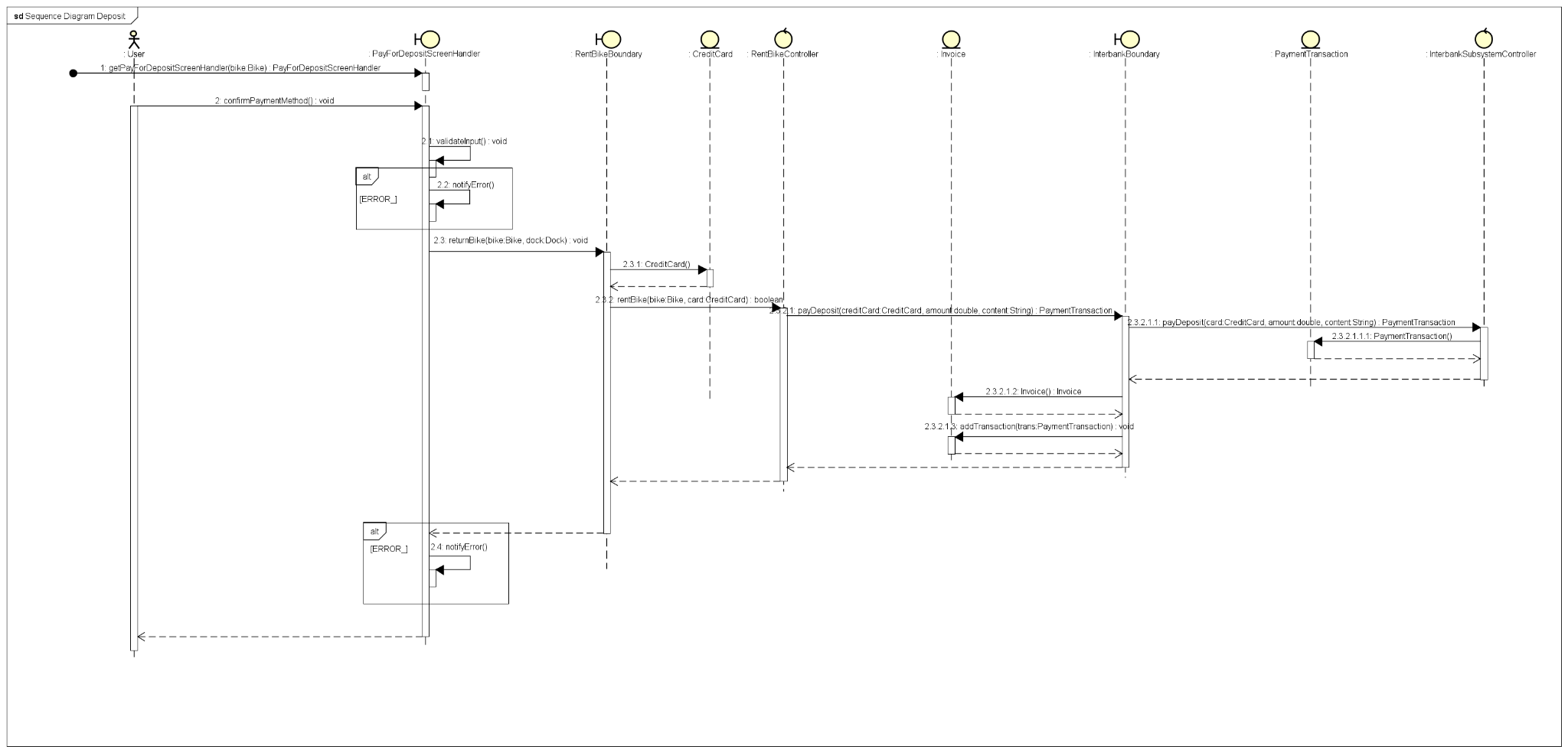


*Figure 3.6: Communication Diagram for Pay For Rental Use Case*

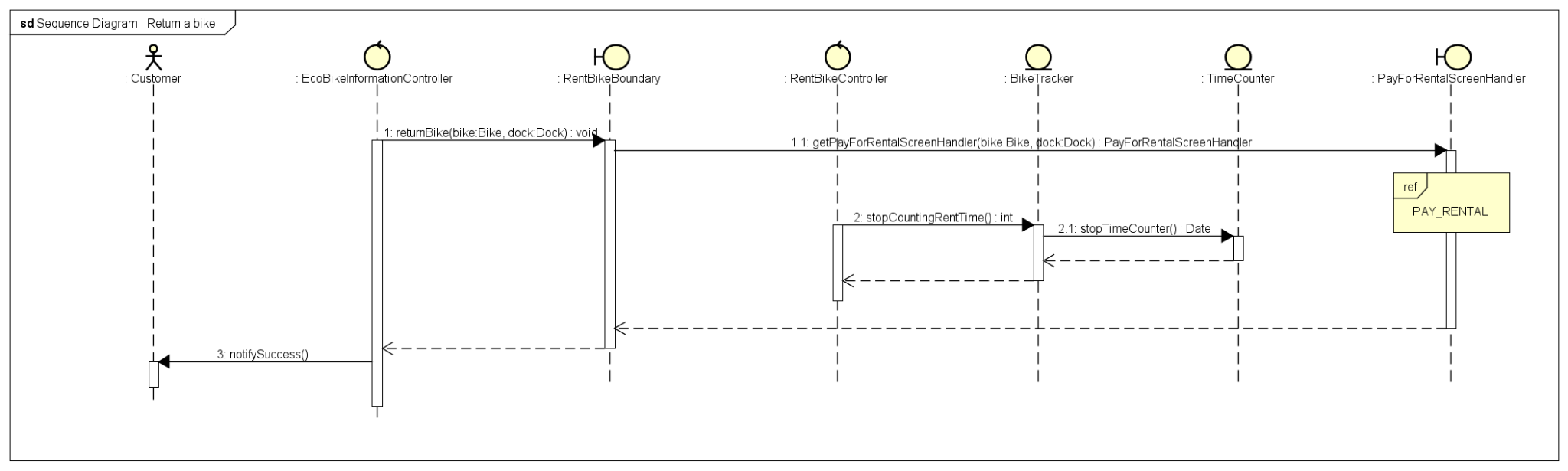
### 3.2.2. Sequence Diagrams



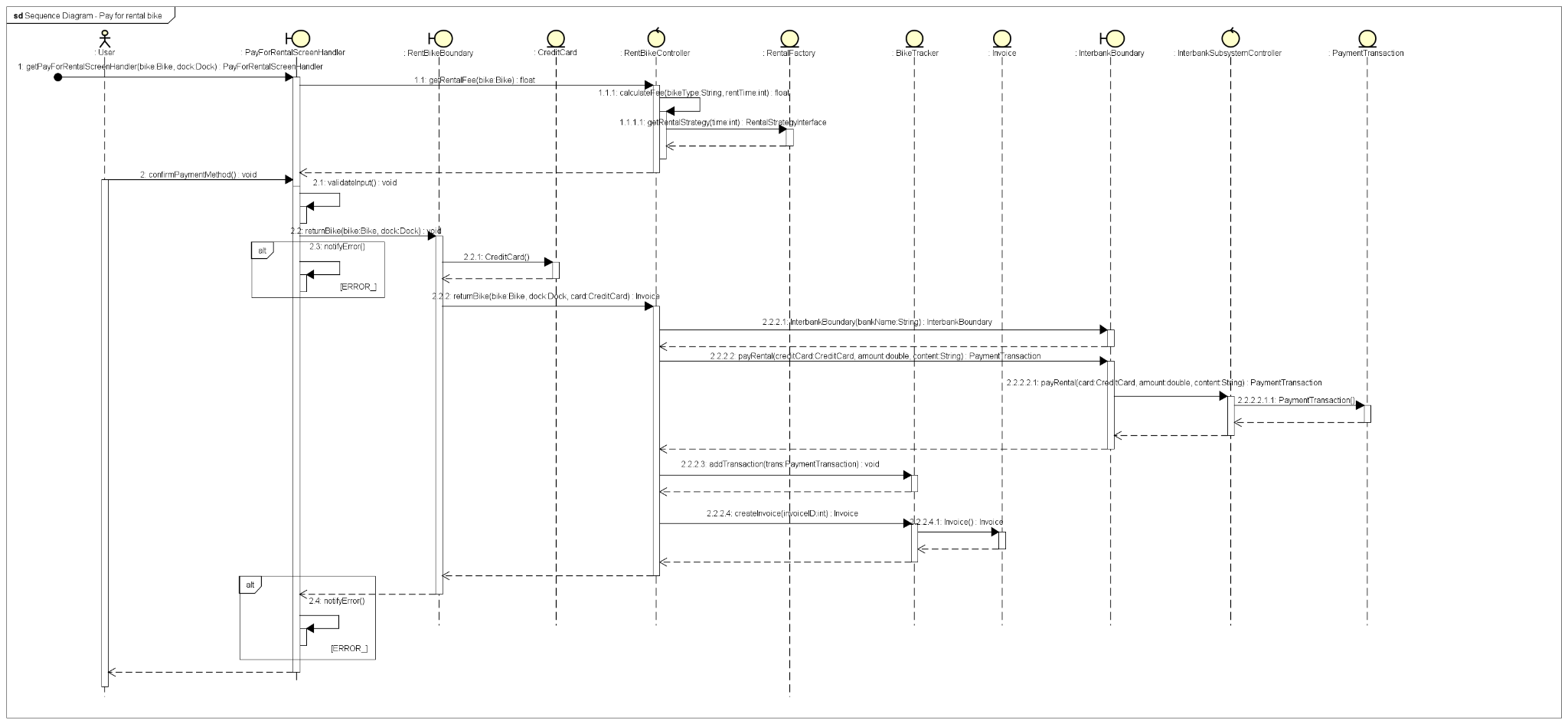
*Figure 3.7: Sequence Diagram for Rent Bike Use Case*

**

*Figure 3.8: Sequence Diagram for Deposit Use Case*

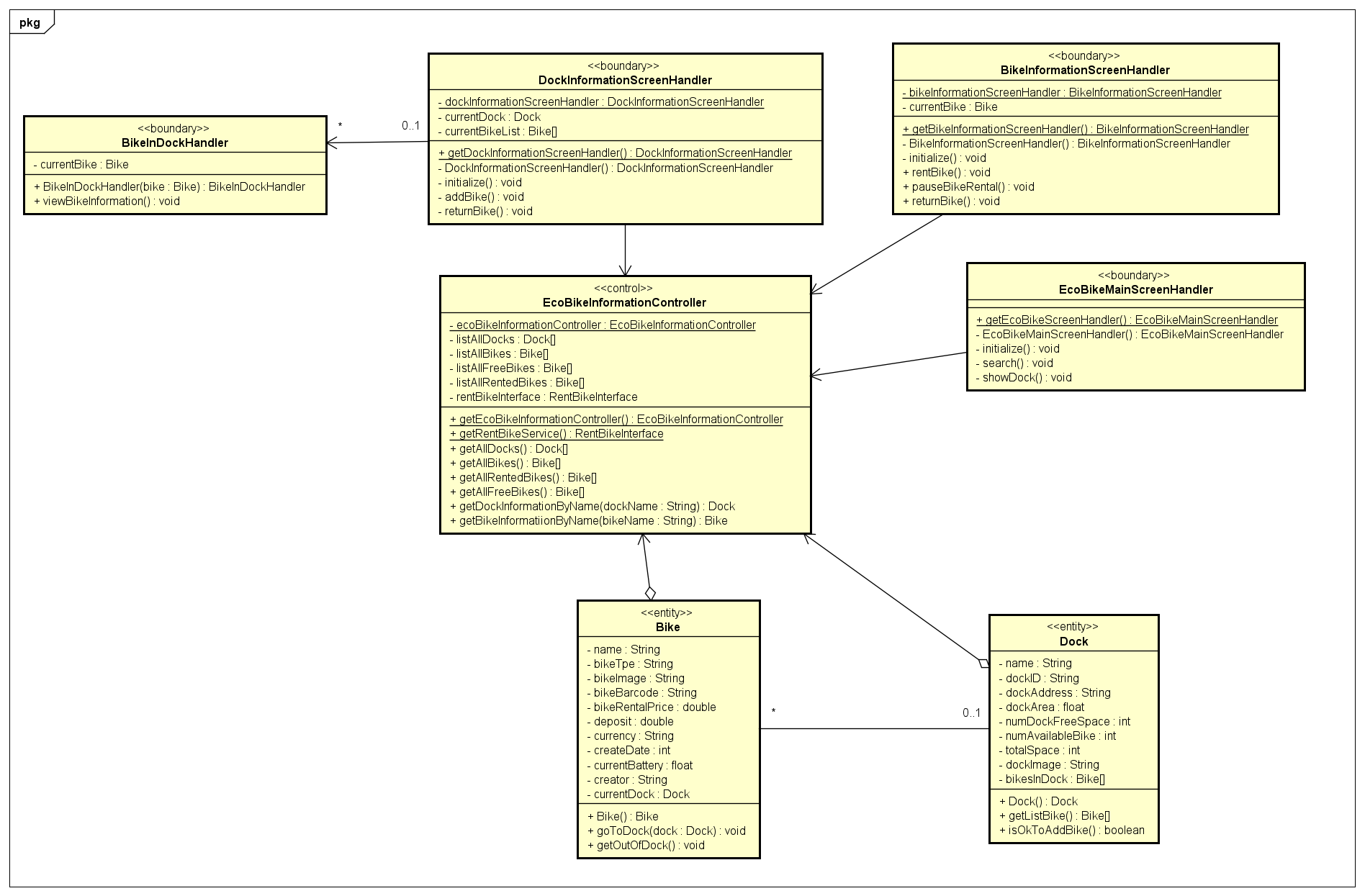
**

*Figure 3.10: Sequence Diagram for Return Bike Use Case*

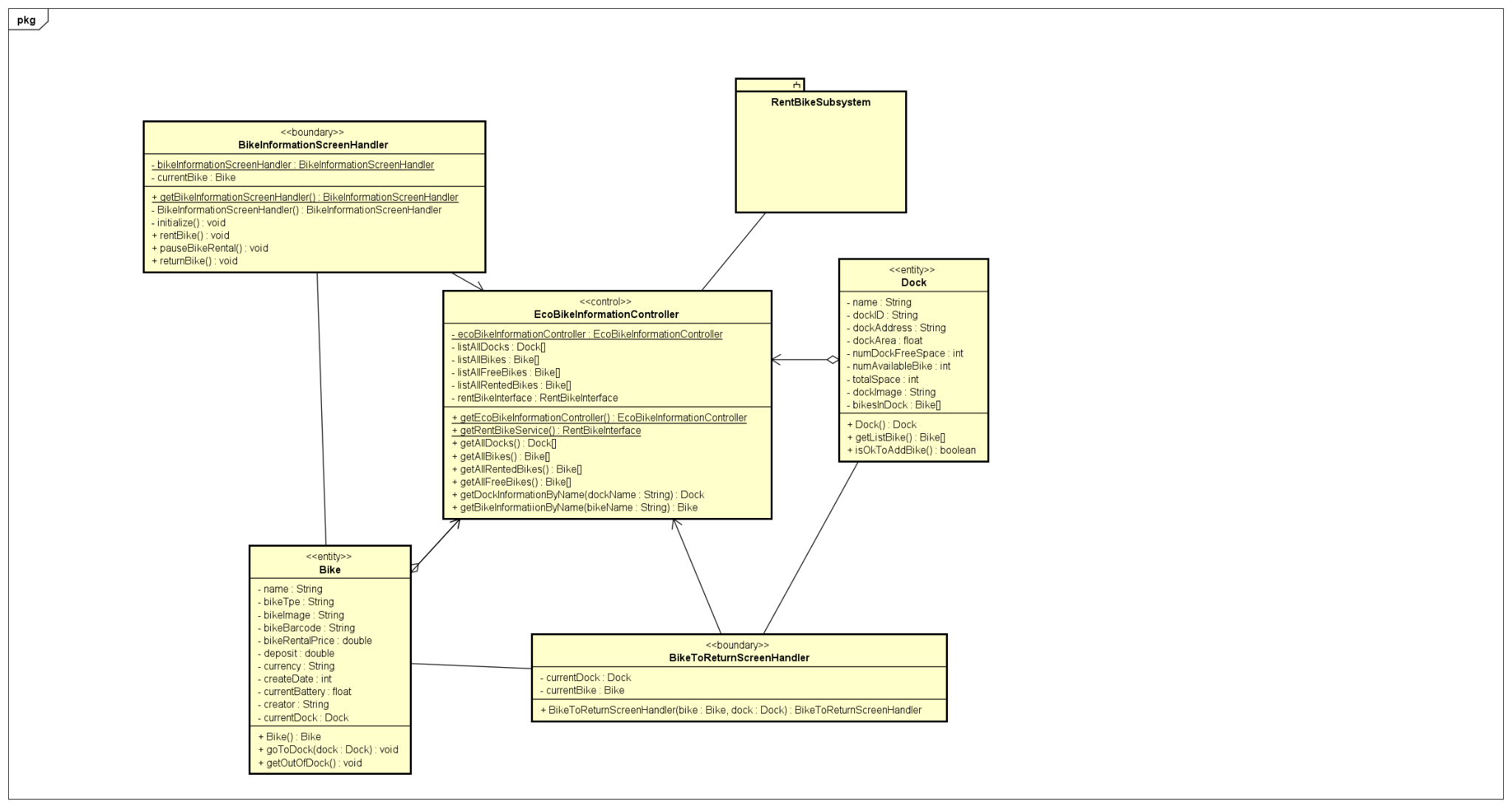
**

*Figure 3.12: Sequence Diagram for Pay For Rental Use Case*

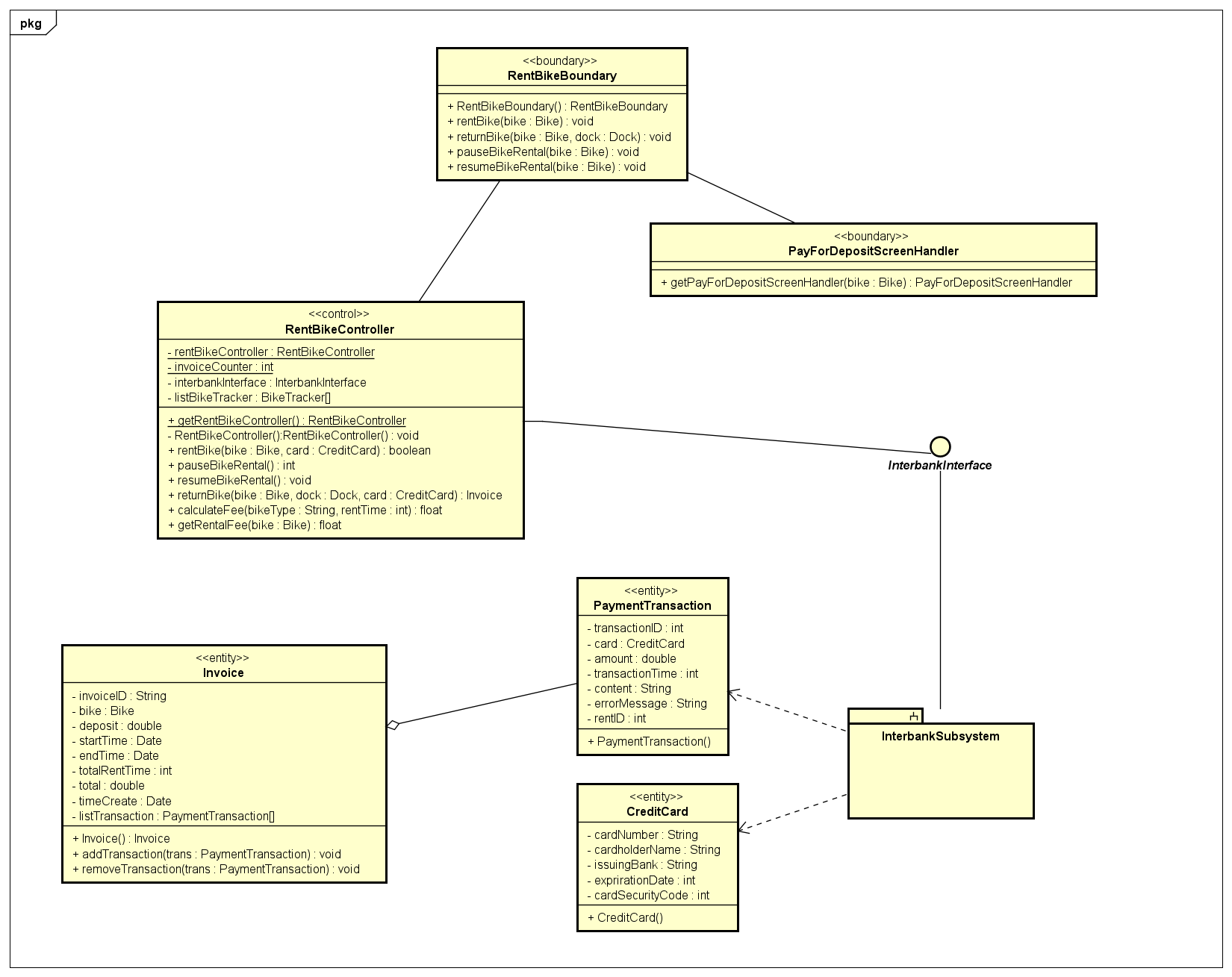
## 3.3. Analysis Class Diagrams



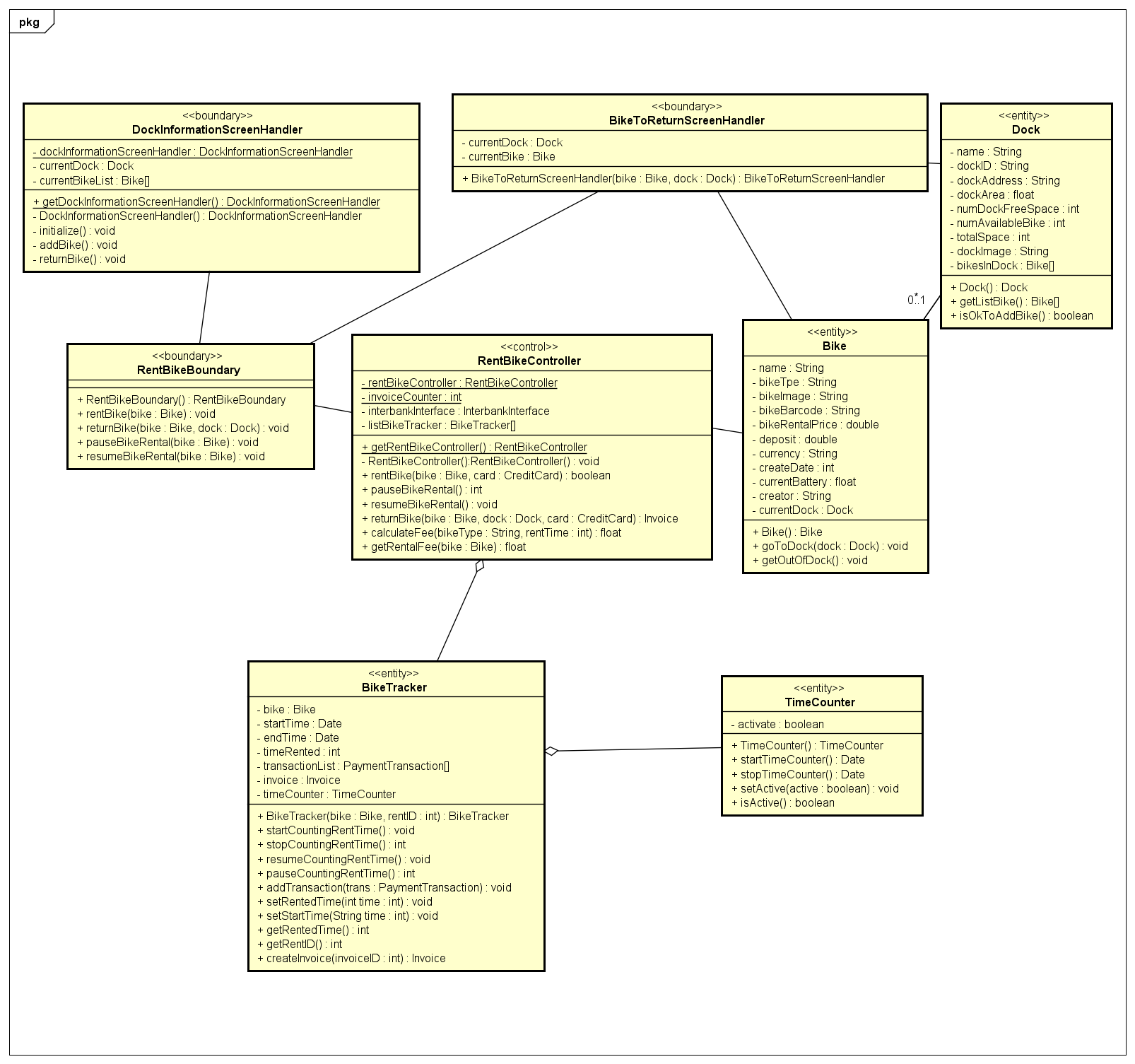
*Figure 3.13: Class Diagram for View Bike Use Case*

**

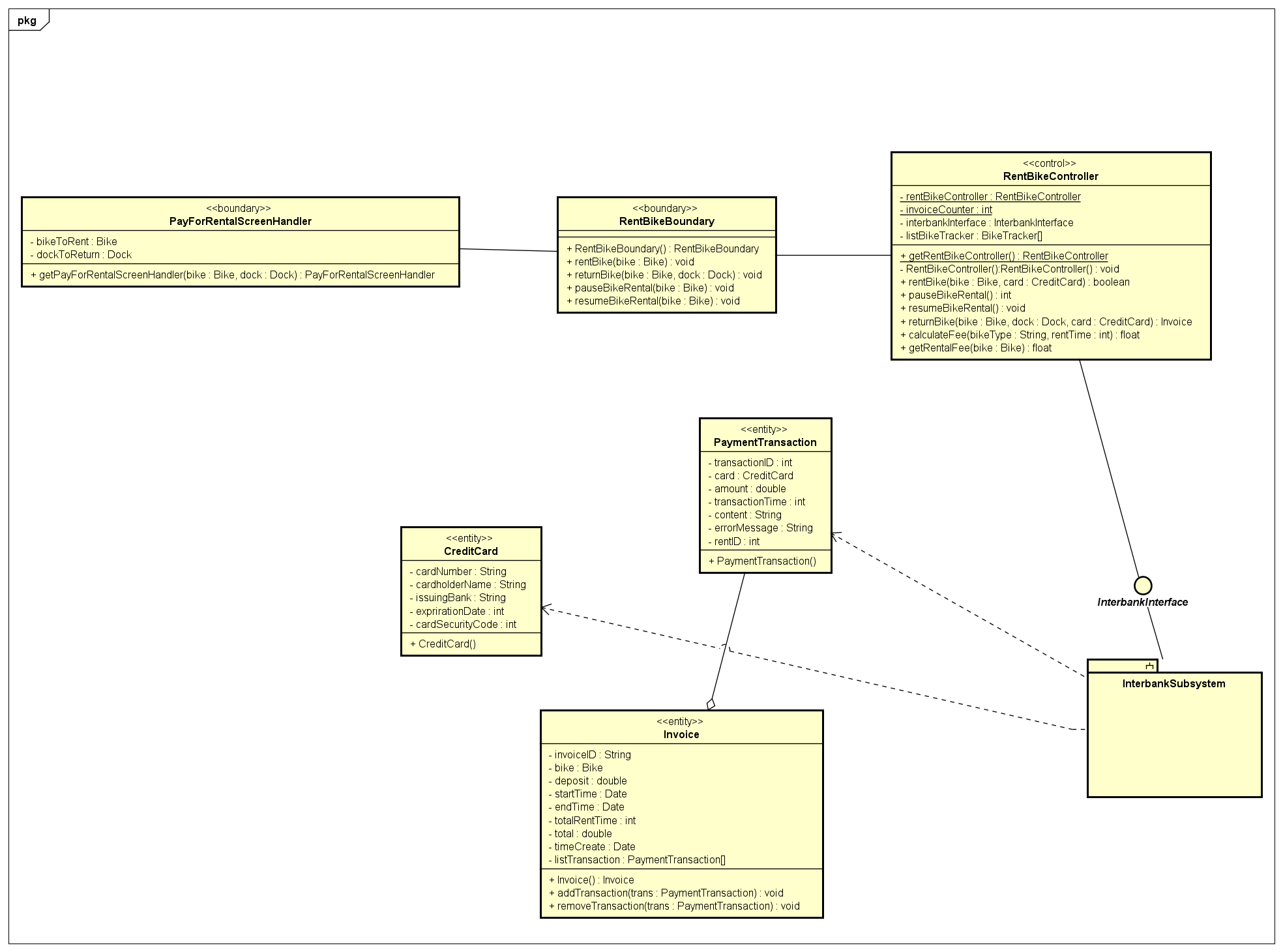
*Figure 3.14: Class Diagram for Rent Bike Use Case*

**

*Figure 3.15: Class Diagram for Deposit Use Case*

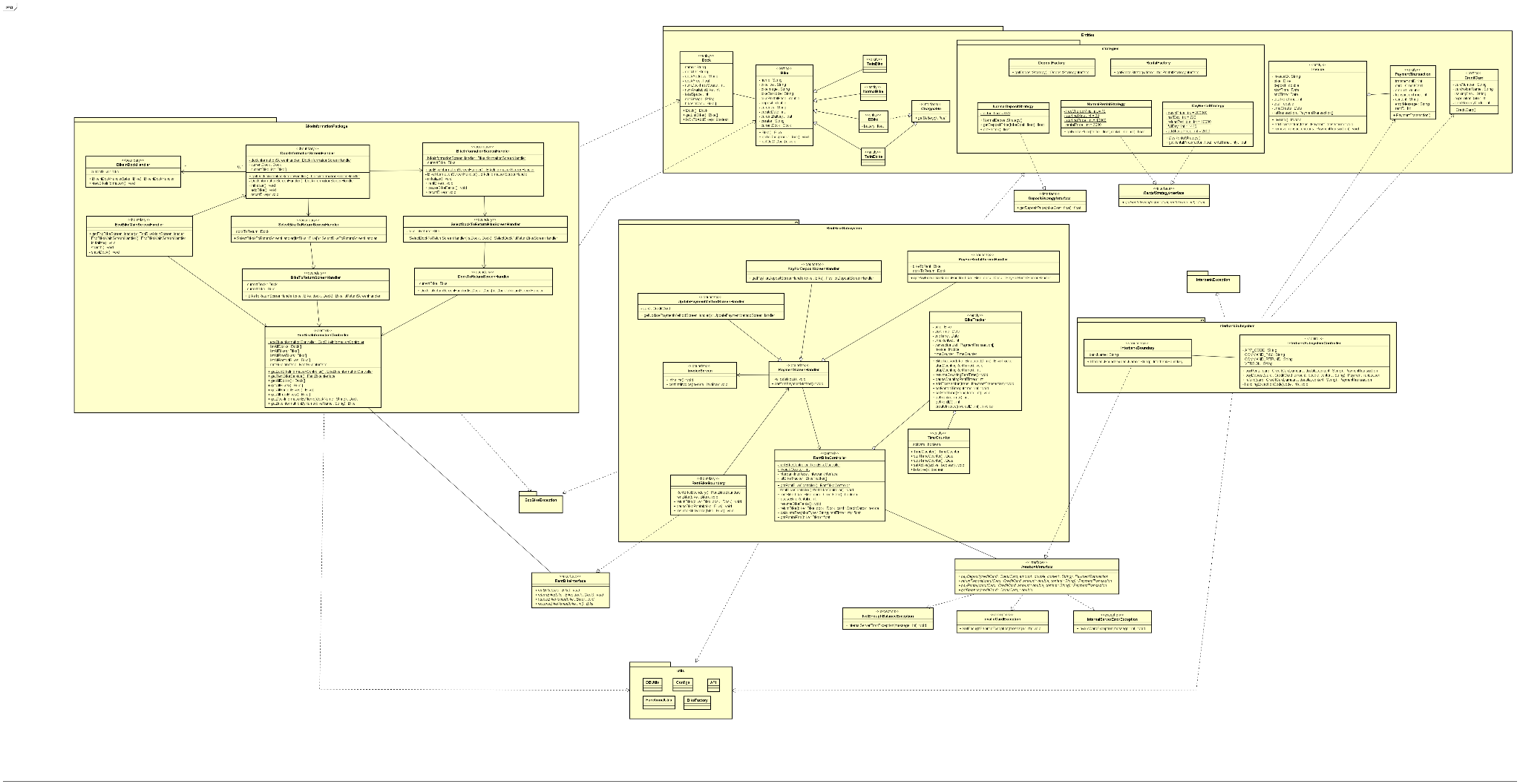
**

*Figure 3.16: Class Diagram for Return Bike Use Case*

**

*Figure 3.18: Class Diagram for Pay Rental Use Case*

## 3.4. Unified Analysis Class Diagram



*Figure 3.19: Unified Class Diagram for EcoBike Application*

## 3.5. Security Software Architecture

In this project, we will not consider features such as user authentication (e.g., sign up, sign in, sign out), we only focus on features related to rent and return bikes.

# 4. Detailed Design

## 4.1. User Interface Design

### 4.1.1. Screen Configuration Standardization

Display

***Screen resolution:*** 1366x768px

***Number of colors supported:*** 16,177,216 colors

Screen

***Size:*** 1200 x 600px

***Main background color:*** #e6ebbc (R: 230, G: 235, B: 188)

***Location of buttons:*** Bottom center of the frame

***Logo:*** 100x100 px

***Header logo:*** 100x100 px, located top left of the screen

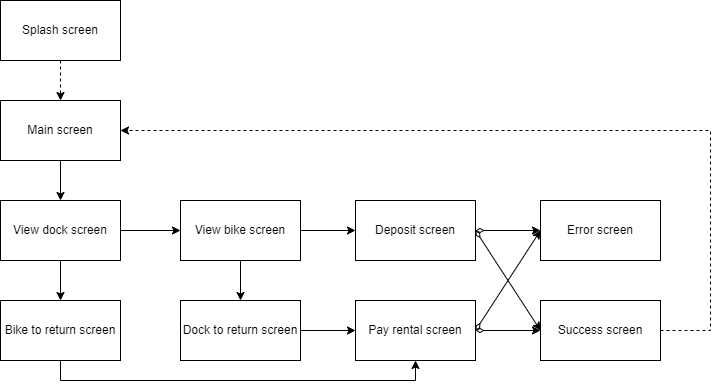
***Header/Screen title:*** Segoe UI, Bold, 24px, black

***Numbers:*** comma for thousand separation, dot for decimal separation

***Text:*** Segoe UI, size at most 24px

***Frame border (if necessary):*** bounded rectangle, dashed line with width of 3px, color #afc139 (R: 175, G:193, B:57)

### 4.1.2. Screen Transition Diagrams



*Figure 4.1: Screen Transition Diagram for EcoBike Application*

### 4.1.3. Screen Specifications

#### 4.1.3.1. Splash Screen

| ***EcoBike Software*** | | ***Date of creation*** | ***Approved by*** | ***Reviewed by*** | ***Person in charge*** |
| --- | --- | --- | --- | --- | --- |
| ***Screen specification*** | ***Splash screen*** | *28/10/2021* |  |  | *Chau* |
|  | | ***Control*** | ***Operation*** | ***Function*** | |
| *Main area* | *None* | *Introduce the application* | |

*Table 4.1. Splash Screen Specification*

#### 4.1.3.2. Main Screen

| ***EcoBike Software*** | | ***Date of creation*** | ***Approved by*** | ***Reviewed by*** | ***Person in charge*** |
| --- | --- | --- | --- | --- | --- |
| ***Screen specification*** | ***Main screen*** | *28/10/2021* |  |  | *Chau* |
|  | | ***Control*** | ***Operation*** | ***Function*** | |
| *Header logo* | *Click* | *Return immediately to main screen* | |
| *Search bar* | *Type, select & click* | *Type in information and select search type to search for docks or bikes* | |
| *Main area* | *Initial* | *Display map at current location of users and nearby docks in term. The pins of docks can be clicked to see docks details* | |

*Table 4.2. Main Screen Specification*

#### 4.1.3.3. Dock Screen

| ***EcoBike Software*** | | ***Date of creation*** | ***Approved by*** | ***Reviewed by*** | ***Person in charge*** |
| --- | --- | --- | --- | --- | --- |
| ***Screen specification*** | ***View Dock screen*** | *28/10/2021* |  |  | *Chau* |
|  | | ***Control*** | ***Operation*** | ***Function*** | |
| *Logo* | *Click* | *Return to the main screen immediately* | |
| *Header* | *Initial* | *Display title of screen* | |
| *Dock information* | *Initial* | *Display dock information* | |
| *Return bike* | *Click* | *Allow user to start return bike process at the dock* | |
| *Bike list* | *Click* | *Display brief details about bikes available in the current dock. Allow choosing each bike to see detailed information* | |

*Table 4.3. View Dock Screen Specification*

#### 4.1.3.4. Bike Screen

| ***EcoBike Software*** | | ***Date of creation*** | ***Approved by*** | ***Reviewed by*** | ***Person in charge*** |
| --- | --- | --- | --- | --- | --- |
| ***Screen specification*** | ***View bike screen*** | *28/10/2021* |  |  | *Chau* |
|  | | ***Control*** | ***Operation*** | ***Function*** | |
| *Logo* | *Click* | *Return to the main screen immediately* | |
| *Header* | *Initial* | *Display title of screen* | |
| *Bike information* | *Initial* | *Display bike information* | |
| *Option pane* | *Click* | *Allow customer to perform renting, pause or return bike* | |

*Table 4.4. View Bike Screen Specification*

#### 4.1.3.6. Deposit screen

| ***EcoBike Software*** | | ***Date of creation*** | ***Approved by*** | ***Reviewed by*** | ***Person in charge*** |
| --- | --- | --- | --- | --- | --- |
| ***Screen specification*** | ***Payment screen*** | *29/10/2021* |  |  | *Long* |
|  | | ***Control*** | ***Operation*** | ***Function*** | |
| *Logo* | *Click* | *Return to the main screen immediately* | |
| *Header* | *Initial* | *Display title of screen* | |
| *Information of payment* | *Initial* | *Display information of payment* | |
| *Button* | *Click* | *Allow customer confirm to deposit the bike* | |

*Table 4.6. Deposit Screen Specification*

#### 4.1.3.7. Payment screen

| ***EcoBike Software*** | | ***Date of creation*** | ***Approved by*** | ***Reviewed by*** | ***Person in charge*** |
| --- | --- | --- | --- | --- | --- |
| ***Screen specification*** | ***Payment screen*** | *29/10/2021* |  |  | *Duong* |
|  | | ***Control*** | ***Operation*** | ***Function*** | |
| *Logo* | *Click* | *Return to the main screen immediately* | |
| *Header* | *Initial* | *Display title of screen* | |
| *Information of payment* | *Initial* | *Display information of payment* | |
| *Buttons* | *Click* | *Allow customer confirm to pay or update card info* | |

*Table 4.7. Payment Screen Specification*

4.1.3.8 Select dock to return screen

#### 

| ***EcoBike Software*** | | ***Date of creation*** | ***Approved by*** | ***Reviewed by*** | ***Person in charge*** |
| --- | --- | --- | --- | --- | --- |
| ***Screen specification*** | ***Select dock to return screen*** | *29/10/2021* |  |  | *chauntm* |
|  | | ***Control*** | ***Operation*** | ***Function*** | |
| *Header* | *Initial* | *Explain meaning of dialog* | |
| *Information of dock* | *Initial* | *Information of docks* | |
| *Return button* | *Click* | *Allow customer to choose dock to return bike; will be disable if the dock is not available for returning bike* | |

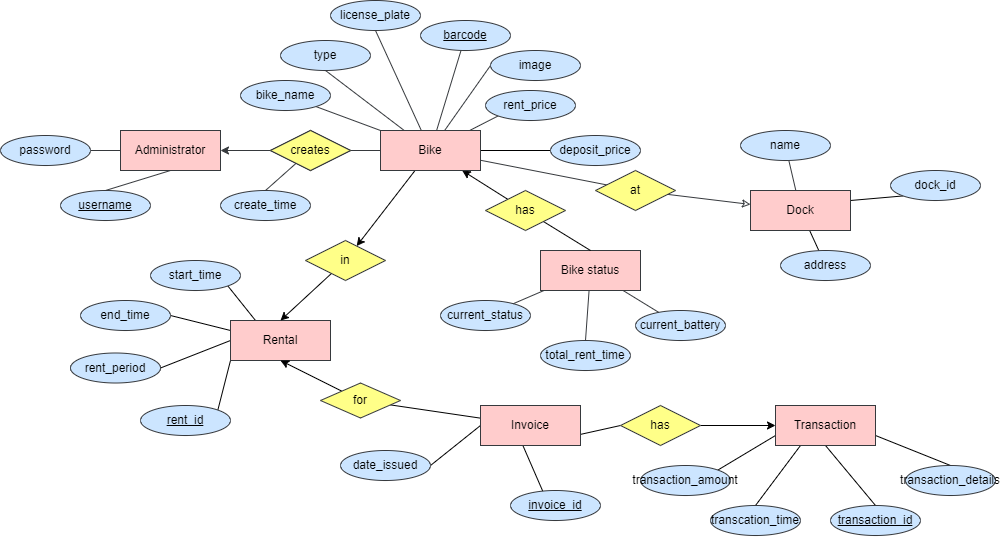
4.1.3.9 Select bike to return screen

#### 

| ***EcoBike Software*** | | ***Date of creation*** | ***Approved by*** | ***Reviewed by*** | ***Person in charge*** |
| --- | --- | --- | --- | --- | --- |
| ***Screen specification*** | ***Select bike to return screen*** | *29/10/2021* |  |  | *chauntm* |
|  | | ***Control*** | ***Operation*** | ***Function*** | |
| *Header* | *Initial* | *Explain meaning of dialog* | |
| *Information of bike* | *Initial* | *Information of bikes being rented* | |
| *Return button* | *Click* | *Allow customer to choose bike to return bike;* | |

## 4.2. Data Modeling

### 4.2.1. Conceptual Data Modeling

**

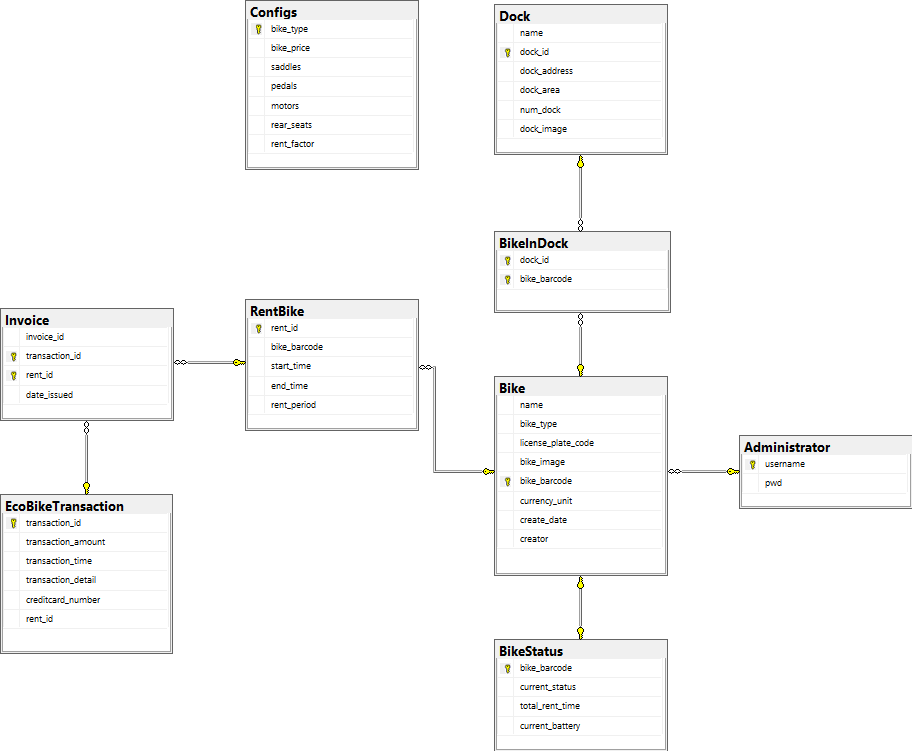
*Figure 4.2. ER Diagram for EcoBike Application*

### 4.2.2. Database Design

#### 4.2.2.1. Database Management System

Database Management System: SQLite

#### 4.2.2.2. Database Diagram



*Figure 4.3. Database Diagram for EcoBike Application*

#### 4.2.2.3. Database Detail Design

**Administrator**

| **No.** | **PK** | **FK** | **Name** | **Data type** | **Mandatory** | **Description** |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | x | x | username | varchar(256) | x | Username of the administrator |
| 2 |  |  | pwd | varchar(256) | x | Password of the administrator used to login |

**Dock**

| **No.** | **PK** | **FK** | **Name** | **Data type** | **Mandatory** | **Description** |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | x | x | dock\_id | int | x | ID of dock |
| 2 |  |  | dock\_name | varchar(256) | x | Name of the dock |
| 3 |  |  | dock\_address | varchar(256) | x | Address of the dock |
| 4 |  |  | dock\_area | float |  | Area of the dock |
| 5 |  |  | num\_available\_bike | int | x | Number of current available bike in dock |
| 6 |  |  | num\_free\_dock | int | x | Number of current available bike slot in dock for returning bike |

**Bike**

| **No.** | **PK** | **FK** | **Name** | **Data type** | **Mandatory** | **Description** |
| --- | --- | --- | --- | --- | --- | --- |
| 1 |  |  | bike\_name | varchar(256) | x | Name of the bike |
| 2 |  |  | bike\_type | varchar(16) | x | Type of bike |
| 3 |  |  | license\_plate\_code | varchar(32) | x | Code of the license plate of the bike |
| 4 |  |  | bike\_image | varchar(256) |  | Path to image of the bike |
| 5 | x | x | bike\_barcode | int | x | Barcode of the bike |
| 6 |  |  | bike\_rental\_price | float | x | Price to rent the bike |
| 7 |  |  | deposit\_price | float | x | Deposit cost to rent the bike |
| 8 |  |  | currency\_unit | varchar(3) | x | Currency unit used to calculate rental fee and deposit fee |
| 9 |  |  | create\_date | date | x | Day imported bike data |
| 10 |  |  | creator | varchar(256) | x | The administrator who create data for the bike |

**Bike in dock**

| **No.** | **PK** | **FK** | **Name** | **Data type** | **Mandatory** | **Description** |
| --- | --- | --- | --- | --- | --- | --- |
| 1 |  |  | dock\_id | int | x | Id of the dock |
| 2 |  |  | bike\_barcode | int | x | Barcode of the bike in dock |

**Bike status**

| **No.** | **PK** | **FK** | **Name** | **Data type** | **Mandatory** | **Description** |
| --- | --- | --- | --- | --- | --- | --- |
| 1 |  |  | bike\_barcode | int | x | Barcode of the bike |
| 2 |  |  | current\_status | varchar(4) | x | ‘free’/’rent’ |
| 3 |  |  | total\_rent\_time | int | x | Total time that the bike is rented (in minute) |
| 4 |  |  | current battery | float | x | Current battery status of the bike |

**Invoice**

| **No.** | **PK** | **FK** | **Name** | **Data type** | **Mandatory** | **Description** |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | x |  | invoice\_id | int | X | ID of the invoice |
| 2 |  | x | transaction\_id | int | x | ID of the transaction |
| 3 |  | x | customer\_id | int | x | ID of the customer |

**EcoBikeTransaction**

| **No.** | **PK** | **FK** | **Name** | **Data type** | **Mandatory** | **Description** |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | x |  | transaction\_id | int | x | ID of transaction |
| 2 |  |  | transaction\_amount | int | x | The amount of money for the transaction |
| 3 |  |  | transaction\_time | String | x | Time the transaction is made |
| 4 |  |  | transaction\_detail | varchar(256) |  | The content of the transaction |
| 5 |  | x | creditcard\_number | int | x | The number of the credit card |

**RentBike**

| **No.** | **PK** | **FK** | **Name** | **Data type** | **Mandatory** | **Description** |
| --- | --- | --- | --- | --- | --- | --- |
| 1 |  |  | bike\_barcode | int | x | Barcode of the bike being rented |
| 2 |  |  | start\_time | time | x | Time start renting |
| 3 |  |  | end\_time | time |  | Time end renting (null if the bike is currently being rented) |
| 4 |  |  | rent\_period | int |  | Total time renting the bike, in terms of minutes (null if the bike is currently being rented) |
| 5 | x |  | rent\_id | int |  | ID of the rental |

**Invoice**

| **No.** | **PK** | **FK** | **Name** | **Data type** | **Mandatory** | **Description** |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | x |  | invoice\_id | int | x | ID of the invoice |
| 2 | x |  | transaction\_id | int | x | Id of the transaction related to the invoice |
| 3 | x |  | rent\_id | int |  | ID of the rental |
| 4 |  |  | date\_issued | String |  | String representation of the date that the invoice is issued |

#### 

#### Database script:

--

-- PRAGMA foreign\_keys = off;

BEGIN TRANSACTION;

-- Table: Administrator

-- DROP TABLE IF EXISTS Administrator;

CREATE TABLE Administrator (

username VARCHAR (256) NOT NULL

PRIMARY KEY,

pwd VARCHAR (256) NOT NULL

);

-- Table: Bike

-- DROP TABLE IF EXISTS Bike;

CREATE TABLE Bike (

name VARCHAR (256) NOT NULL,

bike\_type VARCHAR (16) NOT NULL,

license\_plate\_code VARCHAR (32) NOT NULL,

bike\_image VARCHAR (256),

bike\_barcode VARCHAR (10) PRIMARY KEY,

currency\_unit VARCHAR (3) NOT NULL,

create\_date DATE,

creator VARCHAR (256),

CONSTRAINT FK\_Bike\_Creator FOREIGN KEY (

creator

)

REFERENCES Administrator (username)

);

-- Table: Dock

-- DROP TABLE IF EXISTS Dock;

CREATE TABLE Dock (

name VARCHAR (256),

dock\_id INTEGER PRIMARY KEY IDENTITY,

dock\_address VARCHAR (256),

dock\_area FLOAT,

num\_dock INTEGER,

dock\_image VARCHAR (256)

);

-- Table: BikeInDock

-- DROP TABLE IF EXISTS BikeInDock;

CREATE TABLE BikeInDock (

dock\_id INTEGER,

bike\_barcode VARCHAR (10) NOT NULL,

CONSTRAINT PK\_Bike\_In\_Dock PRIMARY KEY (

dock\_id,

bike\_barcode

),

CONSTRAINT FK\_BikeInDock\_Dock FOREIGN KEY (

dock\_id

)

REFERENCES Dock (dock\_id),

CONSTRAINT FK\_BikeInDock\_Bike FOREIGN KEY (

bike\_barcode

)

REFERENCES Bike (bike\_barcode)

);

-- Table: BikeStatus

-- DROP TABLE IF EXISTS BikeStatus;

CREATE TABLE BikeStatus (

bike\_barcode VARCHAR (10) NOT NULL

PRIMARY KEY,

current\_status VARCHAR (4),

total\_rent\_time INTEGER,

current\_battery FLOAT,

CONSTRAINT FK\_BikeStatus\_Barcode FOREIGN KEY (

bike\_barcode

)

REFERENCES Bike (bike\_barcode),

CONSTRAINT Check\_BikeStatus\_Total\_Rent\_Time CHECK (total\_rent\_time >= 0),

CONSTRAINT Check\_BikeStatus\_Battery CHECK (current\_battery >= 0)

);

-- Table: Configs

-- DROP TABLE IF EXISTS Configs;

CREATE TABLE Configs (

bike\_type VARCHAR (256) PRIMARY KEY,

bike\_price FLOAT,

saddles INTEGER,

pedals INTEGER,

motors INTEGER,

rear\_seats INTEGER,

rent\_factor FLOAT

);

-- Table: EcoBikeTransaction

-- DROP TABLE IF EXISTS EcoBikeTransaction;

CREATE TABLE EcoBikeTransaction (

transaction\_id INTEGER NOT NULL

PRIMARY KEY IDENTITY,

transaction\_amount FLOAT NOT NULL,

transaction\_time VARCHAR (256) NOT NULL,

transaction\_detail VARCHAR (256),

creditcard\_number VARCHAR (25) NOT NULL,

rent\_id INTEGER

);

-- Table: RentBike

-- DROP TABLE IF EXISTS RentBike;

CREATE TABLE RentBike (

rent\_id INTEGER NOT NULL

PRIMARY KEY IDENTITY,

bike\_barcode VARCHAR (10) NOT NULL,

start\_time VARCHAR (256) NOT NULL,

end\_time VARCHAR (256),

rent\_period INTEGER,

CONSTRAINT FK\_RentBike\_Bike FOREIGN KEY (

bike\_barcode

)

REFERENCES Bike (bike\_barcode),

);

-- Table: Invoice

-- DROP TABLE IF EXISTS Invoice;

CREATE TABLE Invoice (

invoice\_id INTEGER NOT NULL,

transaction\_id INTEGER NOT NULL,

rent\_id INTEGER NOT NULL,

date\_issued VARCHAR (256),

CONSTRAINT PK\_Rent\_Trans\_ID PRIMARY KEY(transaction\_id, rent\_id),

CONSTRAINT FK\_Invoice\_Transaction FOREIGN KEY (transaction\_id) REFERENCES EcoBikeTransaction(transaction\_id),

CONSTRAINT FK\_Invoice\_Customer FOREIGN KEY (rent\_id) REFERENCES RentBike(rent\_id),

);

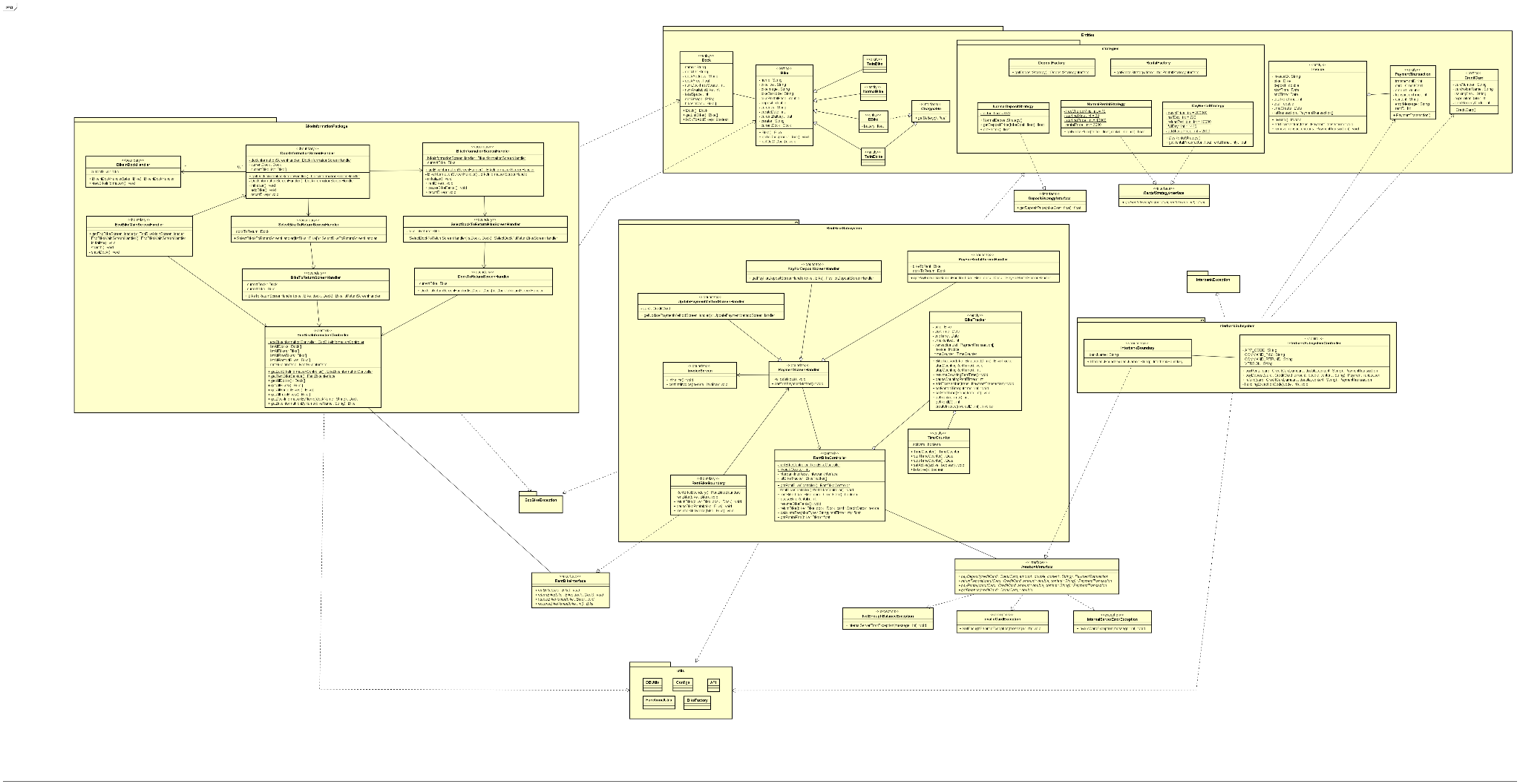
COMMIT TRANSACTION;

-- PRAGMA foreign\_keys = on;

## 4.3. Non-Database Management System Files

## 4.4. Class Design

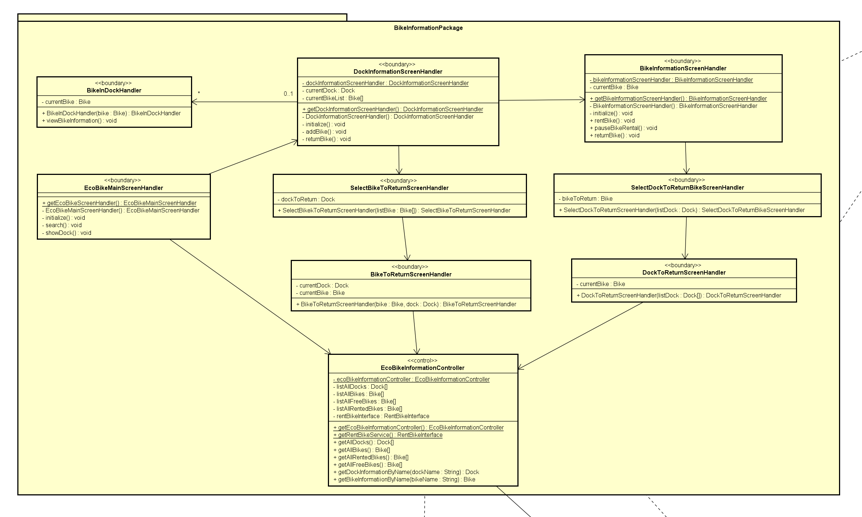
### 4.4.1. General Class Diagram



*Figure 4.4. General Class Diagram for EcoBike Application*

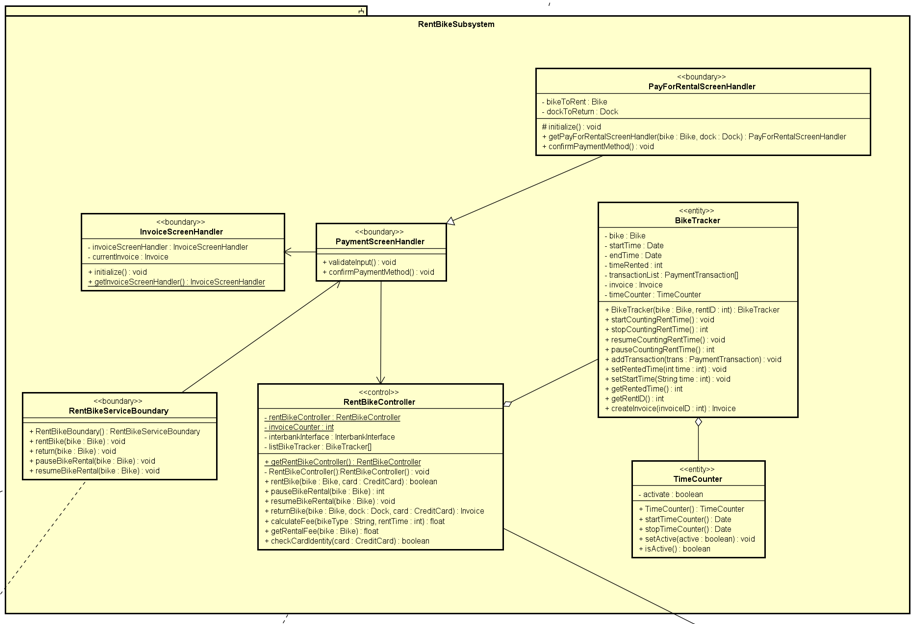
### 4.4.2. Class Diagrams

#### 4.4.2.1. Class Diagram for Subsystem BikeInformation



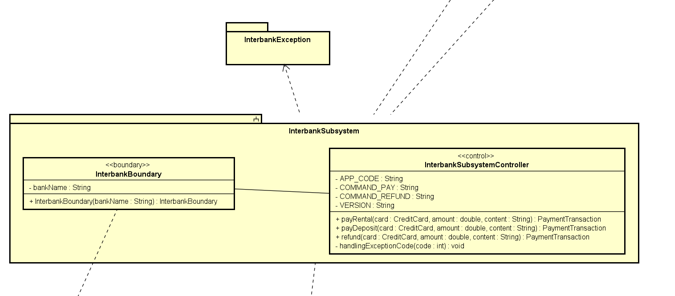
*Figure 4.5. Class Diagram for Package BikeInformation*

#### 4.4.2.2. Class Diagram for Subsystem RentBike



*Figure 4.6. Class Diagram for Subsystem RentBike*

#### 4.4.2.3. Class Diagram for Subsystem InterBank



*Figure 4.7. Class Diagram for Subsystem InterBank*