

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	3
List of Figures	6
List of Tables	7
1. Introduction	8
1.1. Objective	8
1.2. Scope	8
1.3. Glossary	8
1.4. References	9
2. Overall Description	10
2.1. General Overview	10
2.2. Assumptions/Constraints/Risks	11
2.2.1. Assumptions	11
2.2.2. Constraints	11
2.2.3. Risks	11
3. System Architecture and Architecture Design	12
3.1. Architectural Patterns	12
3.2. Interaction Diagrams	12
3.2.1. Communication Diagrams	12
3.2.2. Sequence Diagrams	15
3.3. Analysis Class Diagrams	18
3.4. Unified Analysis Class Diagram	21
3.5. Security Software Architecture	21
4. Detailed Design	22
4.1. User Interface Design	22
4.1.1. Screen Configuration Standardization	22
4.1.2. Screen Transition Diagrams	22
4.1.3. Screen Specifications	22
4.1.3.1. Splash Screen	22
4.1.3.2. Main Screen	23
4.1.3.3. Dock Screen	23
4.1.3.4. Bike Screen	24
4.1.3.5. Payment Method Screen	25
4.1.3.6. Deposit screen	25
4.1.3.7. Payment screen	25
4.2. Data Modeling	27
4.2.1. Conceptual Data Modeling	27
4.2.2. Database Design	27

4.2.2.1. Database Management System	27
4.2.2.2. Database Diagram	28
4.2.2.3. Database Detail Design	28
4.3. Non-Database Management System Files	35
4.4. Class Design	35
4.4.1. General Class Diagram	35
4.4.2. Class Diagrams	36
4.4.2.1. Class Diagram for Package BikeInformation	36
4.4.2.2. Class Diagram for Subsystem RentBike	37
4.4.2.3. Class Diagram for Subsystem InterBank	37
4.4.3. Class Design	38
4.4.3.1. Class RentBikeController	38
4.4.3.2. Class ReturnBikeController	39
4.4.3.3. Class PaymentController	41
4.4.3.4. Class BikeInformationScreenHandler	43
5. Design Considerations	45
5.1. Goals and Guidelines	45
5.2. Architectural Strategies	45
5.3. Coupling and Cohesion	45
5.3.1. Coupling	45
5.3.1.1. Content coupling	45
5.3.1.2. Common coupling	46
5.3.1.3. Control coupling	46
5.3.1.4. Stamp coupling	46
5.3.1.5. Data coupling	46
5.3.2. Cohesion	47
5.3.2.1. Coincidental cohesion	47
5.3.2.2. Logical cohesion	47
5.3.2.3. Temporal cohesion	47
5.3.2.4. Procedure cohesion	47
5.3.2.5. Communicational cohesion	47
5.3.2.6. Sequential cohesion	48
5.3.2.7. Information cohesion	48
5.3.2.8. Functional cohesion	48
5.4. Design Principles	48
5.5. Design Patterns	48

List of Figures

<i>Figure 1.1: General use case diagram</i>	11
<i>Figure 3.1: Communication Diagram for Rent Bike Use Case</i>	13
<i>Figure 3.2: Communication Diagram for Deposit Use Case</i>	13
<i>Figure 3.3: Communication Diagram for Update Payment Method Use Case</i>	14
<i>Figure 3.4: Communication Diagram for Return Bike Use Case</i>	14
<i>Figure 3.5: Communication Diagram for Return Deposit Use Case</i>	15
<i>Figure 3.6: Communication Diagram for Pay For Rental Use Case</i>	15
<i>Figure 3.7: Sequence Diagram for Rent Bike Use Case</i>	16
<i>Figure 3.8: Sequence Diagram for Deposit Use Case</i>	16
<i>Figure 3.9: Sequence Diagram for Update Payment Method Use Case</i>	17
<i>Figure 3.10: Sequence Diagram for Return Bike Use Case</i>	18
<i>Figure 3.11: Sequence Diagram for Return Deposit Use Case</i>	18
<i>Figure 3.12: Sequence Diagram for Pay For Rental Use Case</i>	19
<i>Figure 3.13: Class Diagram for View Bike Use Case</i>	20
<i>Figure 3.14: Class Diagram for View Bike Use Case</i>	21
<i>Figure 3.15: Class Diagram for Deposit Use Case</i>	21
<i>Figure 3.16: Class Diagram for Return Bike Use Case</i>	22
<i>Figure 3.17: Class Diagram for Return Deposit Use Case</i>	22
<i>Figure 3.18: Class Diagram for Pay Rental Use Case</i>	23
<i>Figure 3.19: Unified Class Diagram for EcoBike Application</i>	24
<i>Figure 4.1: Screen Transition Diagram for EcoBike Application</i>	25
<i>Figure 4.2. ER Diagram for EcoBike Application</i>	29
<i>Figure 4.3. Database Diagram for EcoBike Application</i>	30
<i>Figure 4.4. General Class Diagram for EcoBike Application</i>	37
<i>Figure 4.5. Class Diagram for Package BikeInformation</i>	38
<i>Figure 4.6. Class Diagram for Subsystem RentBike</i>	39

<i>Figure 4.7. Class Diagram for Subsystem InterBank</i>	39
<i>Figure 4.8. RentBikeController Class Diagram</i>	40
<i>Figure 4.9. Payment ControllerClass Diagram</i>	42
<i>Figure 4.10. BikeInformationController Class Diagram</i>	45

List of Tables

<i>Table 1.1: Terms used in the document</i>	10
<i>Table 4.1. Splash Screen Specification</i>	24
<i>Table 4.2. Main Screen Specification</i>	24
<i>Table 4.3. View Dock Screen Specification</i>	25
<i>Table 4.4. View Bike Screen Specification</i>	25
<i>Table 4.5. Payment Method Screen Specification</i>	26
<i>Table 4.6. Deposit Screen Specification</i>	26
<i>Table 4.7. Payment Screen Specification</i>	27
<i>Table 4.8. Customer table design</i>	29
<i>Table 4.9. Administrator table design</i>	30
<i>Table 4.10. Dock table design</i>	30
<i>Table 4.11. Bike table design</i>	31
<i>Table 4.12. Bike In Dock table design</i>	31
<i>Table 4.13. Bike Status table design</i>	32
<i>Table 4.14. Invoice table design</i>	32
<i>Table 4.15. Transaction table design</i>	33
<i>Table 4.16. Rent Bike table design</i>	33
<i>Table 4.17. Credit Card table design</i>	34
<i>Table 4.18. RentBikeController attributes</i>	39
<i>Table 4.19. RentBikeController operations</i>	40
<i>Table 4.20. BikeTracker attributes</i>	41
<i>Table 4.21. BikeTracker operations</i>	41
<i>Table 4.24. BikeInformationScreenHandler operations</i>	45
<i>Table 4.25. BikeInformationScreenHandler operations</i>	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/XL C/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:

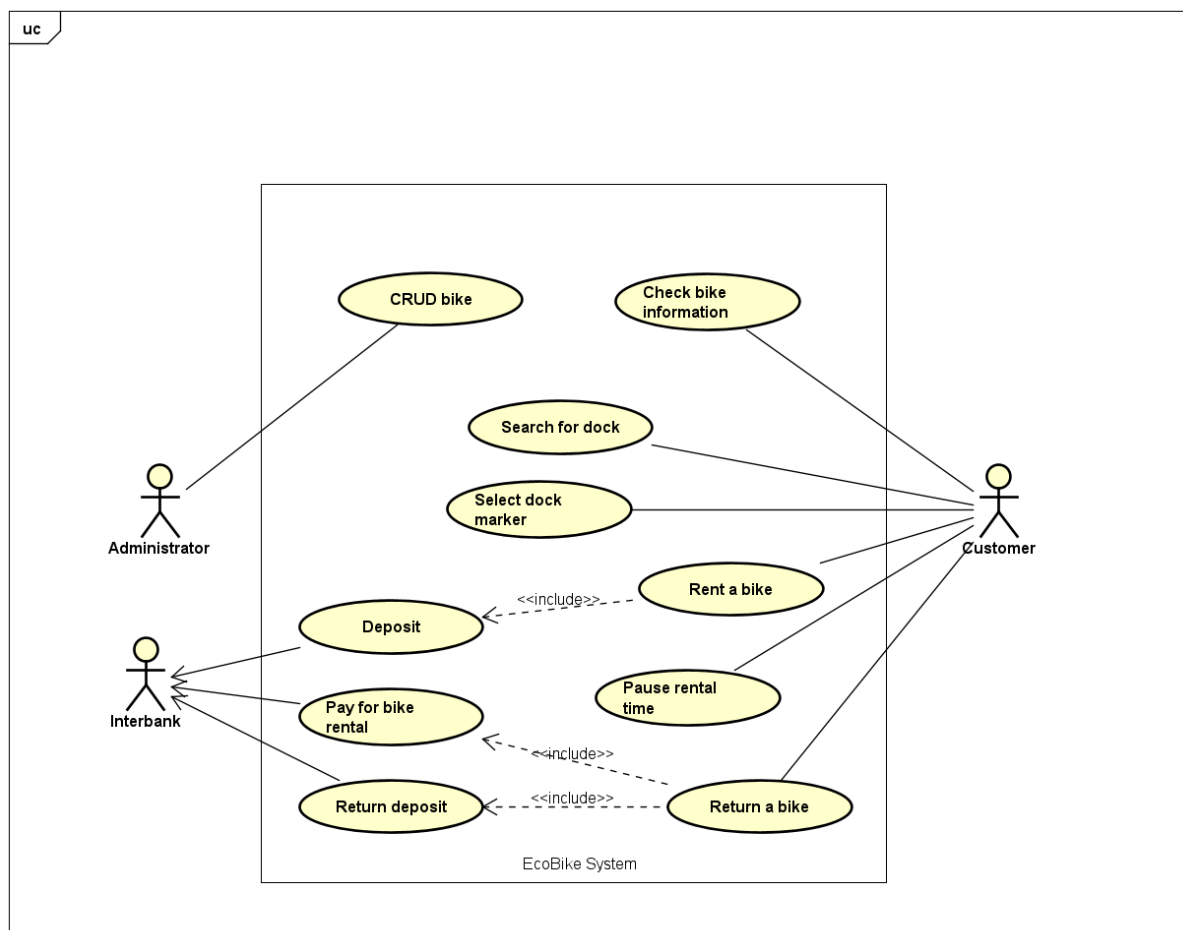


Figure 1.1: General use case diagram

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

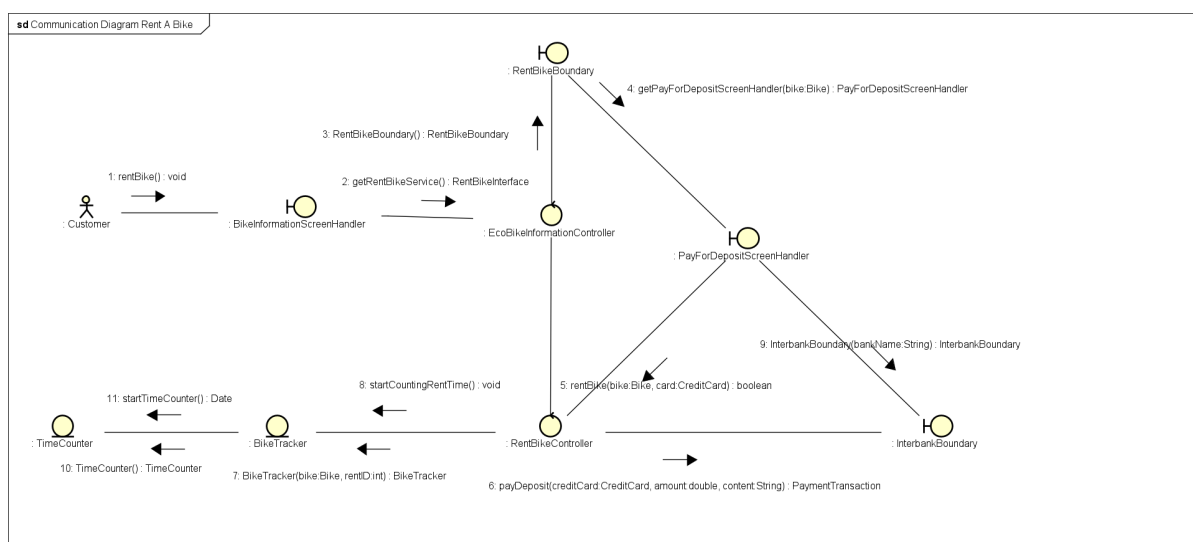


Figure 3.1: Communication Diagram for Rent Bike Use Case

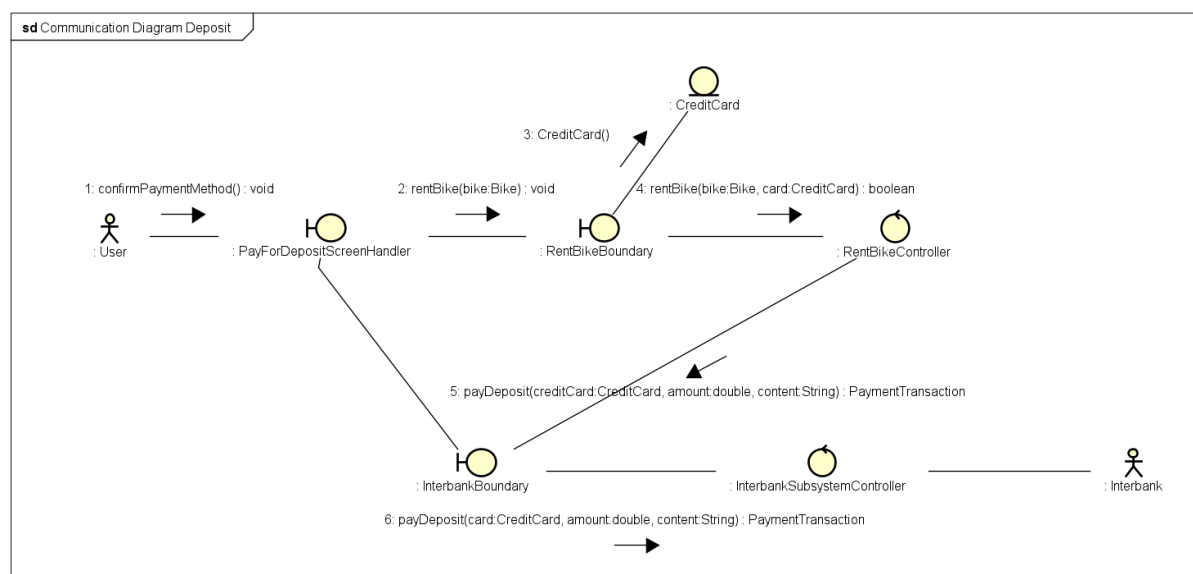


Figure 3.2: Communication Diagram for Deposit Use Case

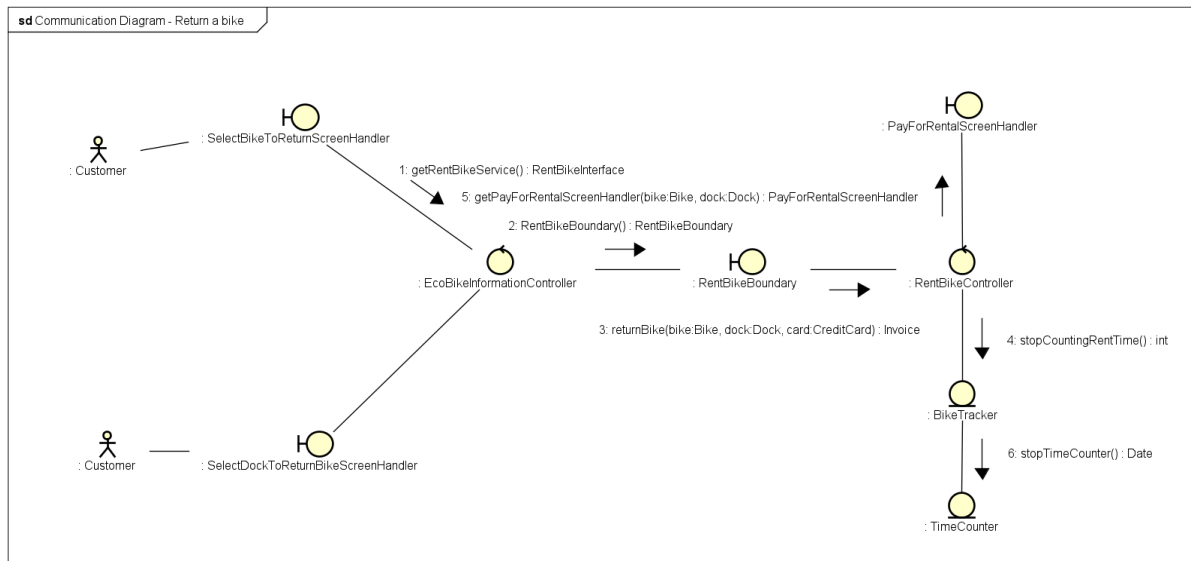


Figure 3.4: Communication Diagram for Return Bike Use Case

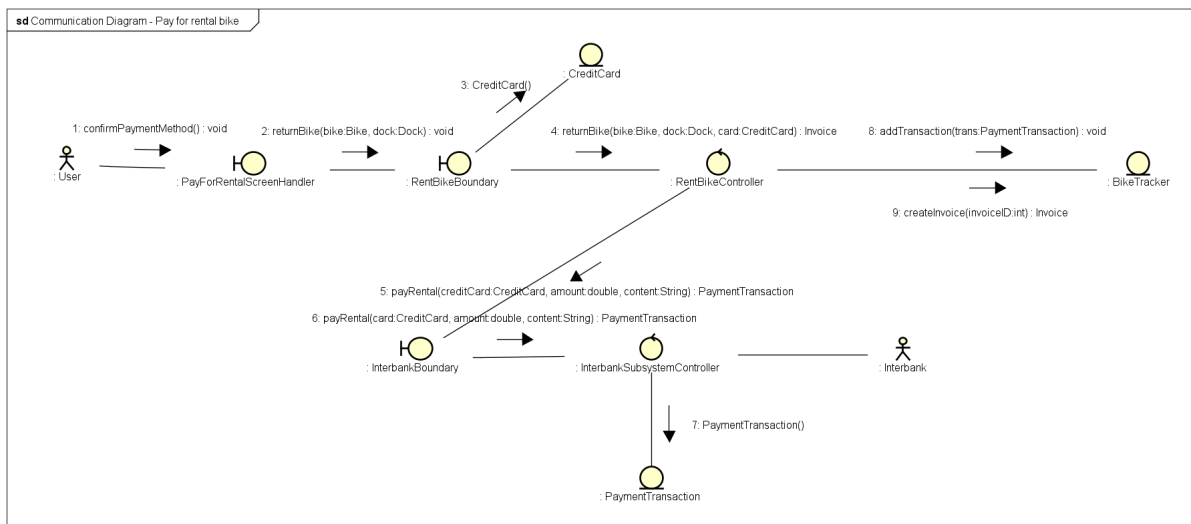


Figure 3.6: Communication Diagram for Pay For Rental Use Case

3.2.2. Sequence Diagrams

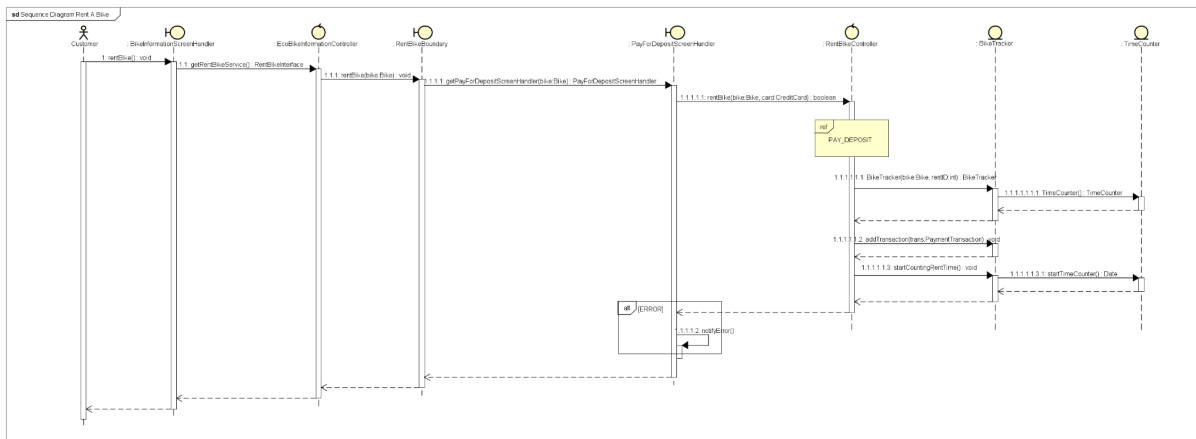


Figure 3.7: Sequence Diagram for Rent Bike Use Case

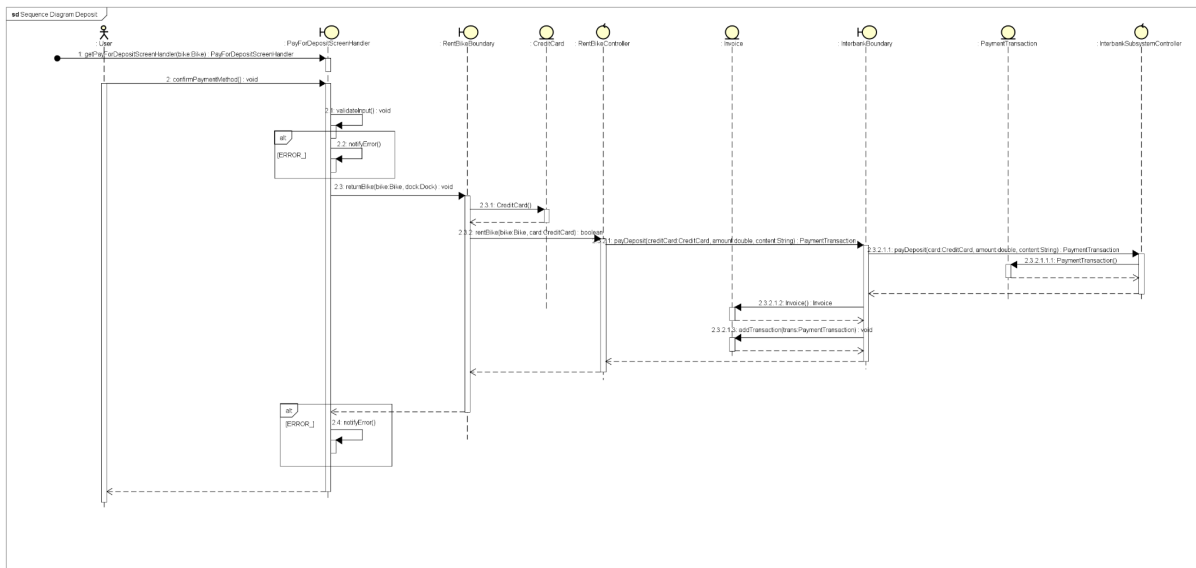


Figure 3.8: Sequence Diagram for Deposit Use Case

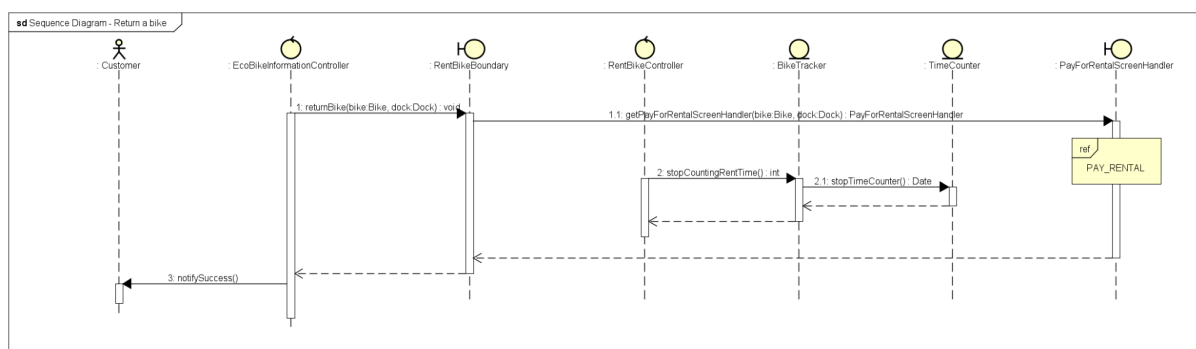


Figure 3.10: Sequence Diagram for Return Bike Use Case

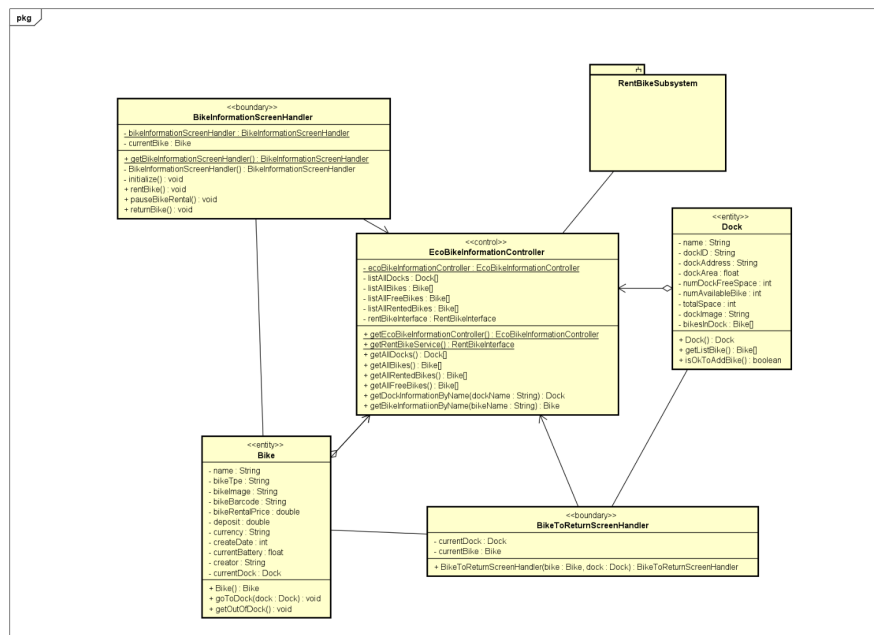


Figure 3.14: Class Diagram for Rent Bike Use Case

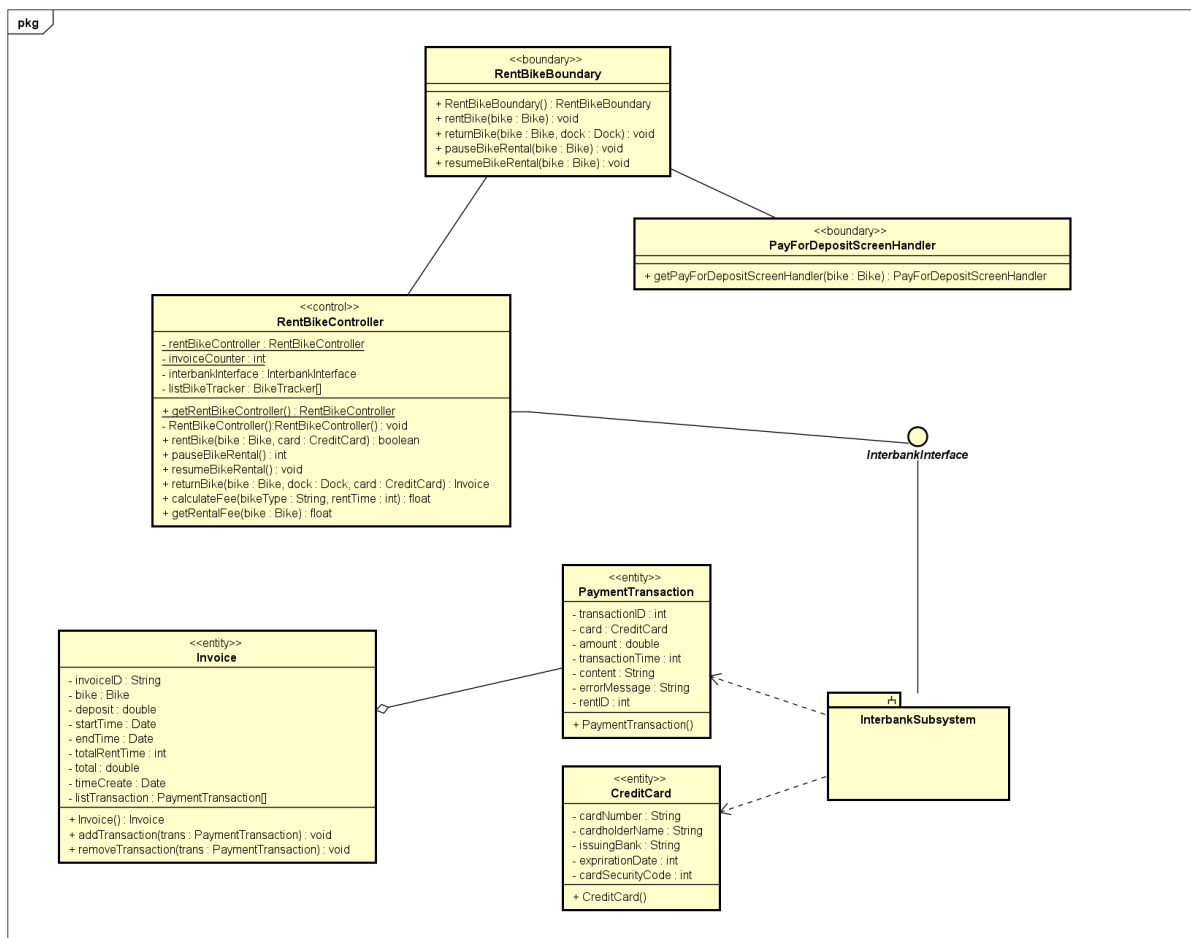


Figure 3.15: Class Diagram for Deposit Use Case

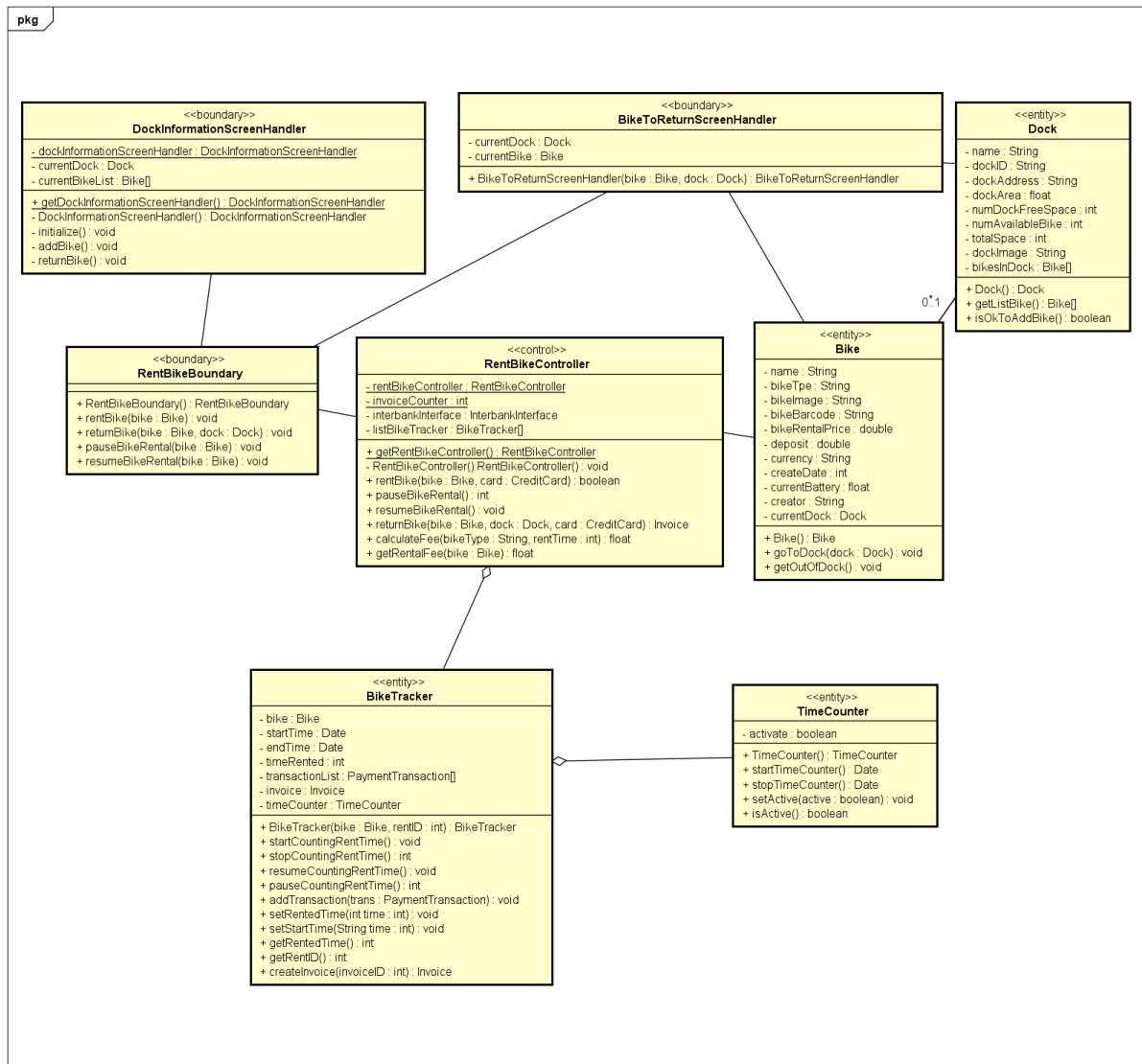


Figure 3.16: Class Diagram for Return Bike Use Case

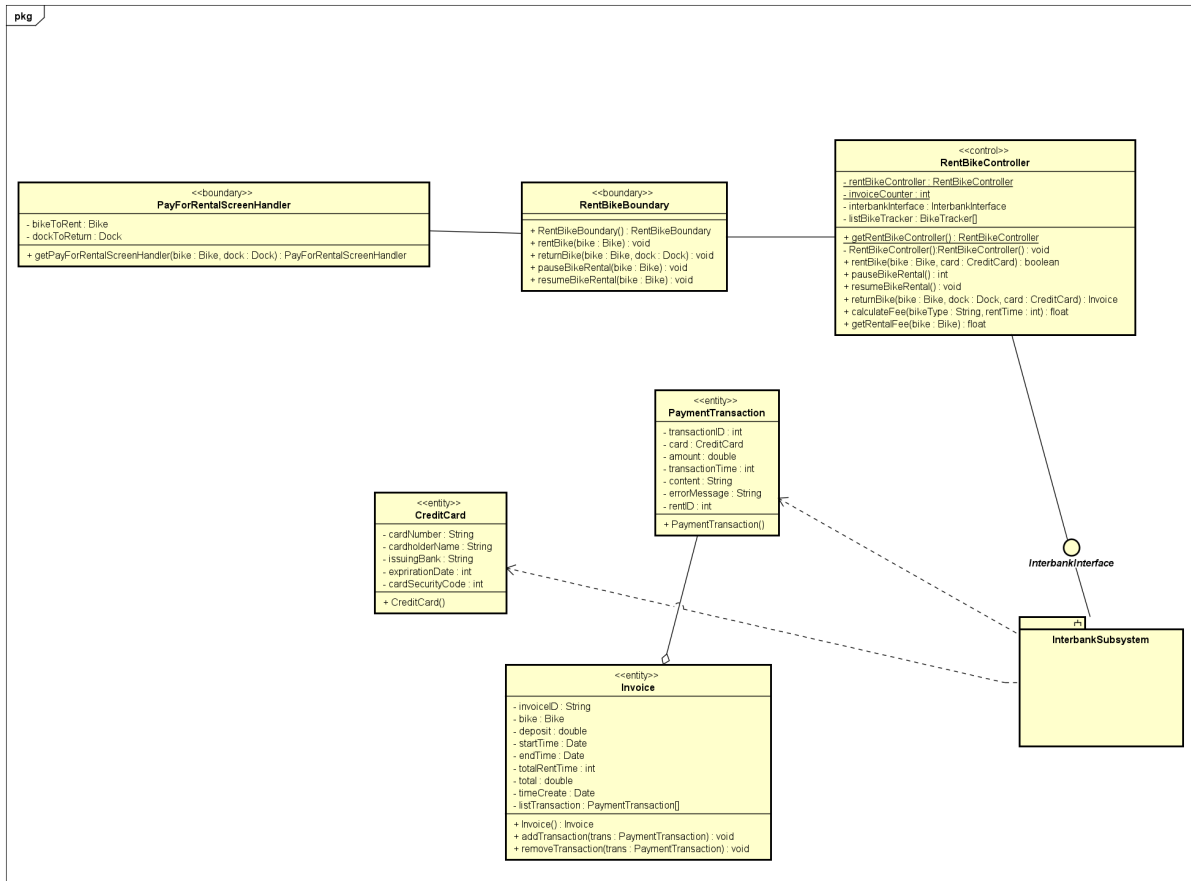


Figure 3.18: Class Diagram for Pay Rental Use Case

3.4. Unified Analysis Class Diagram

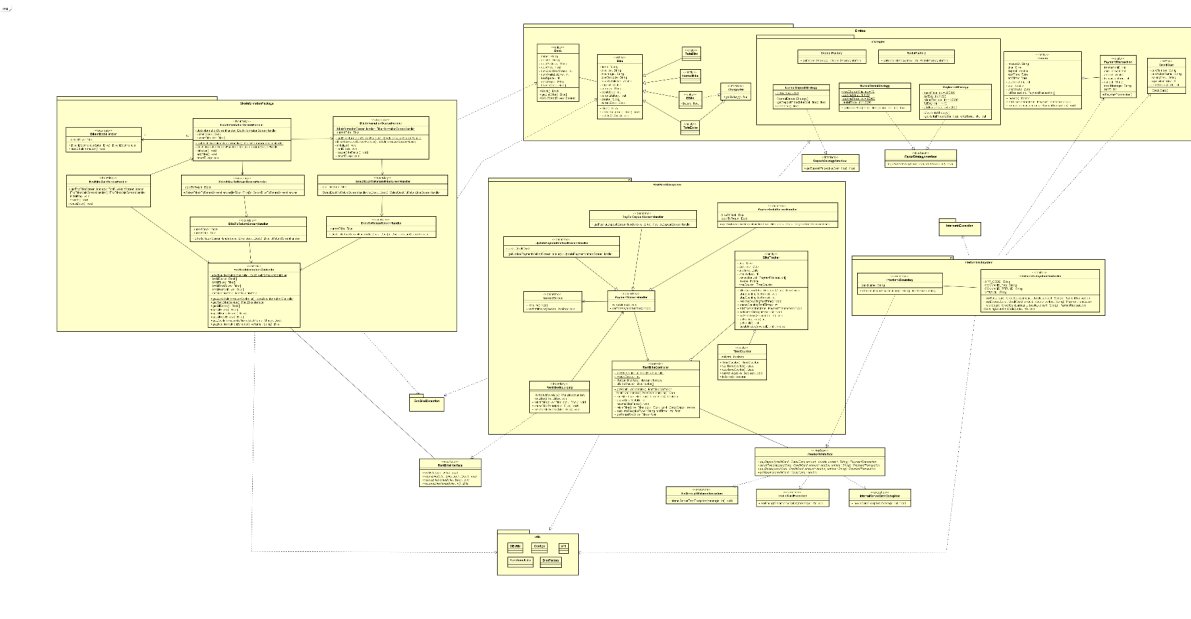


Figure 3.19: Unified Class Diagram for EcoBike Application

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

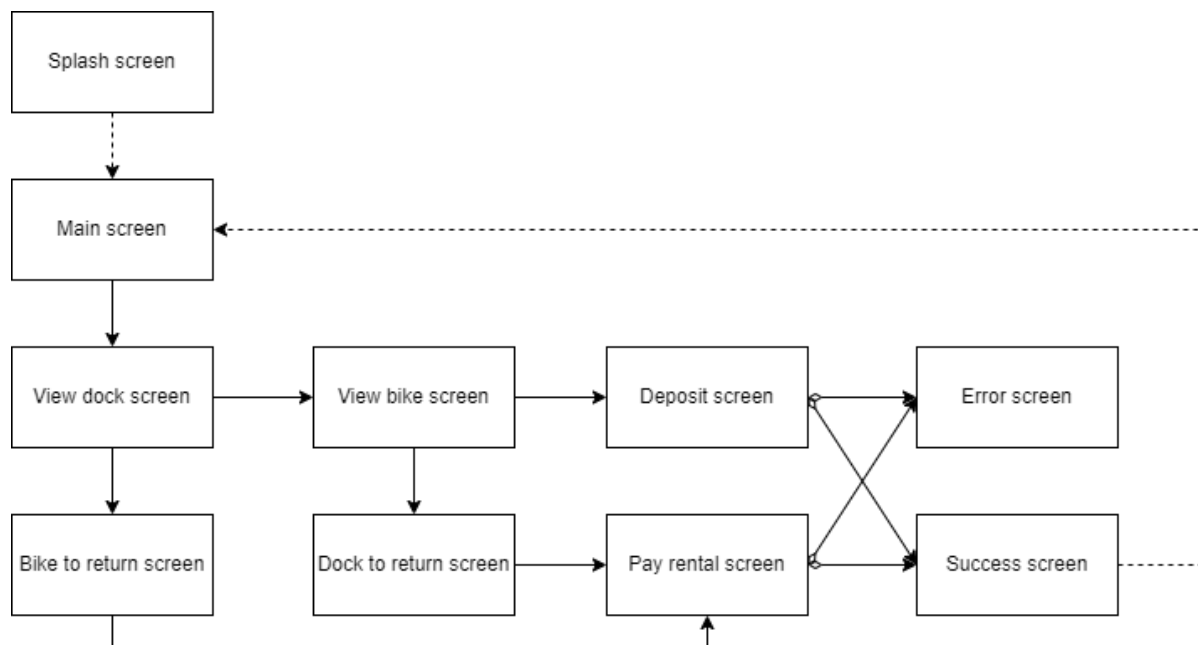


Figure 4.1: Screen Transition Diagram for EcoBike Application

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

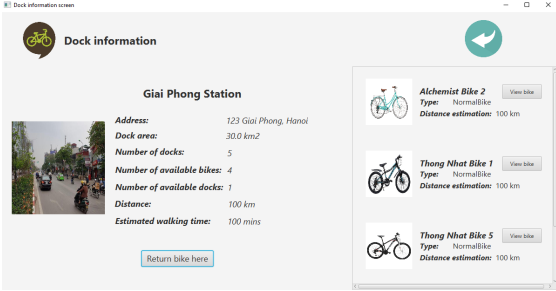
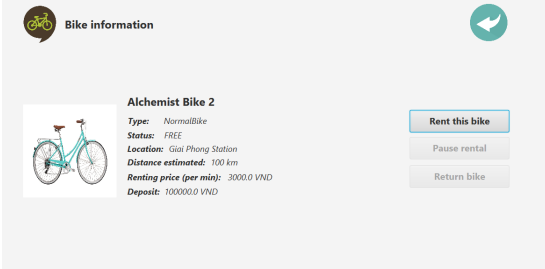
EcoBike Software		Date of creation	Approved by	Reviewed by	Person in charge
Screen specification	View Dock screen	28/10/2021			Chau
		Control	Operation	Function	
		Logo	Click	Return to the main screen immediately	
		Header	Initial	Display title of screen	
		Dock information	Initial	Display dock information	
		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	

	<i>Option pane</i>	<i>Click</i>	<i>Allow customer to perform renting, pause or return bike</i>
--	--------------------	--------------	--

Table 4.4. View Bike Screen Specification

4.1.3.6. Deposit screen

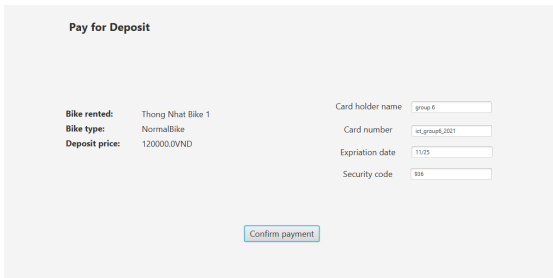
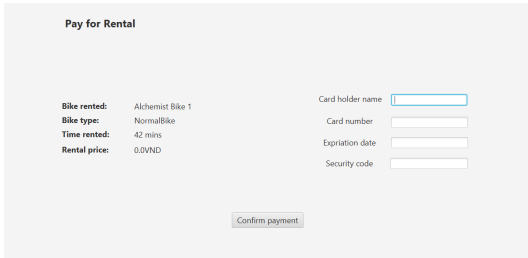
EcoBike Software		Date of creation	Approved by	Reviewed by	Person in charge
Screen specification	Payment screen	29/10/2021			Long
		Control	Operation	Function	
		Logo	Click	Return to the main screen immediately	
		Header	Initial	Display title of screen	
		Information of payment	Initial	Display information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

Table 4.6. Deposit Screen Specification

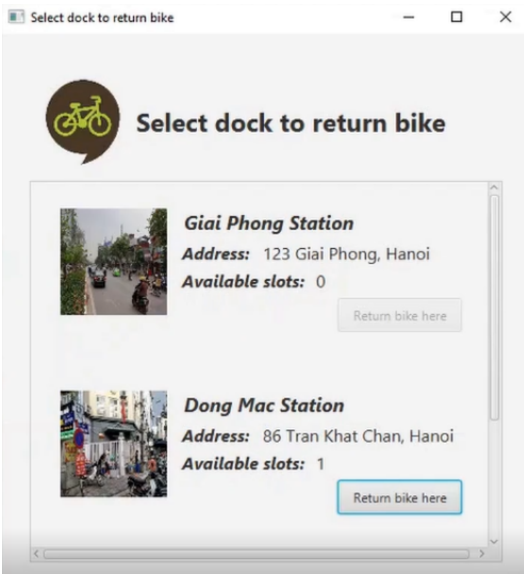
4.1.3.7. Payment screen

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	

	<i>Buttons</i>	<i>Click</i>	<i>Allow customer confirm to pay or update card info</i>
--	----------------	--------------	--

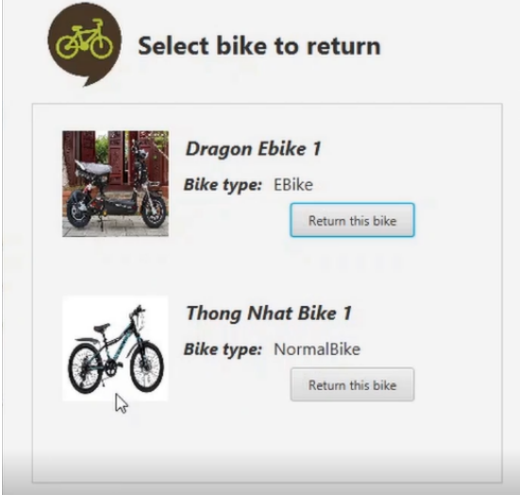
Table 4.7. Payment Screen Specification

4.1.3.8 Select dock to return screen

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

4.1.3.9 Select bike to return screen

<i>EcoBike Software</i>		<i>Date of creation</i>	<i>Approved by</i>	<i>Reviewed by</i>	<i>Person in charge</i>
<i>Screen specification</i>	<i>Select bike to return screen</i>	29/10/2021			chauntm

	Control	Operation	Function
	<i>Header</i>	<i>Initial</i>	<i>Explain meaning of dialog</i>
	<i>Information of bike</i>	<i>Initial</i>	<i>Information of bikes being rented</i>
	<i>Return button</i>	<i>Click</i>	<i>Allow customer to choose bike to return bike;</i>

4.2. Data Modeling

4.2.1. Conceptual Data Modeling

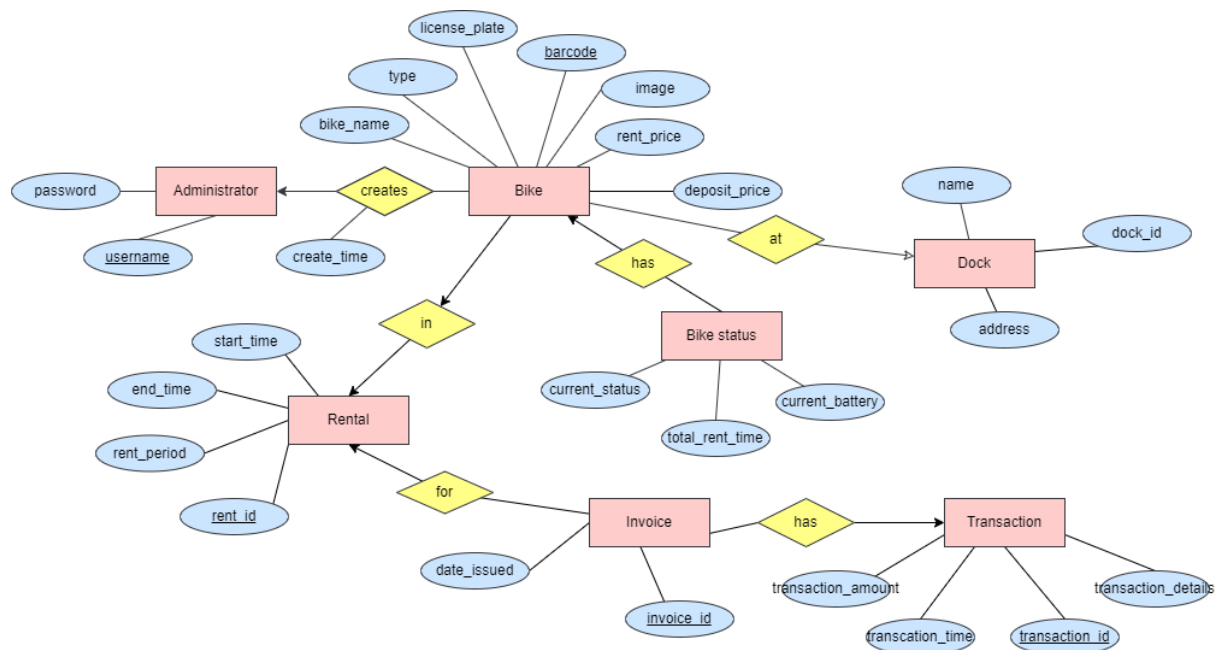


Figure 4.2. ER Diagram for EcoBike Application

4.2.2. Database Design

4.2.2.1. Database Management System

Database Management System: SQLite

4.2.2.2. Database Diagram

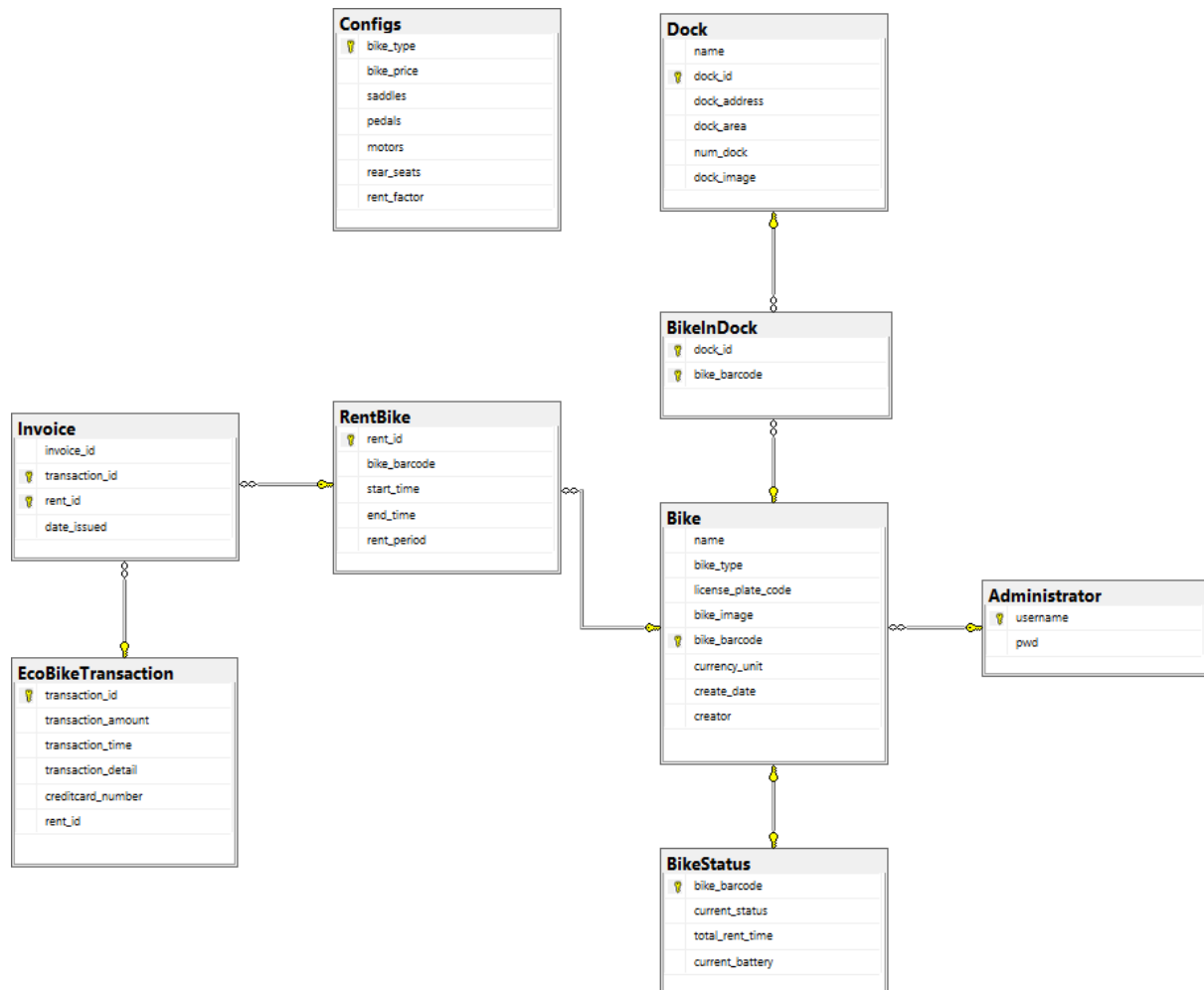


Figure 4.3. Database Diagram for EcoBike Application

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

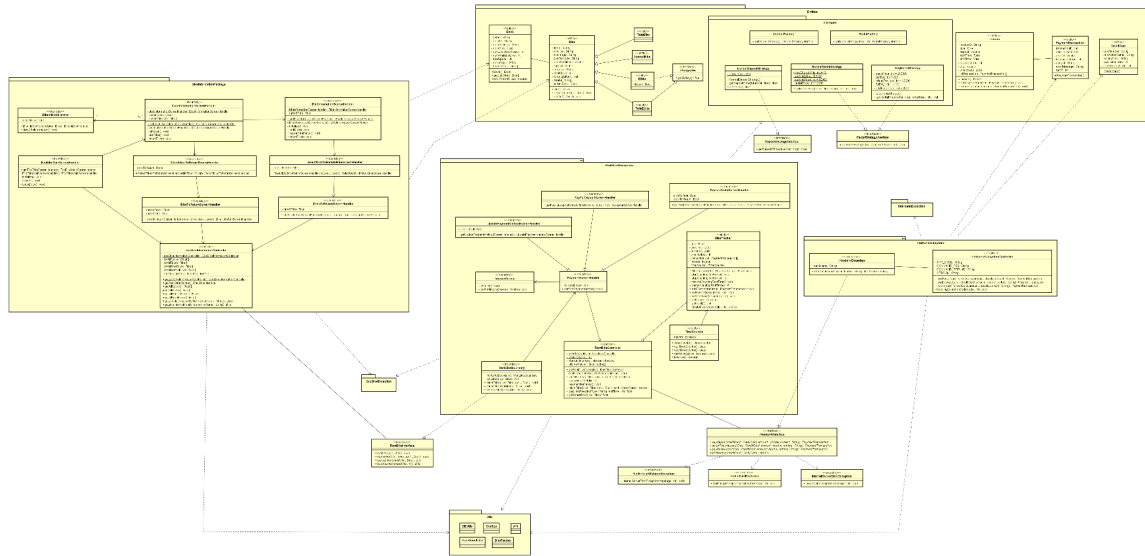


Figure 4.4. General Class Diagram for EcoBike Application

4.4.2. Class Diagrams

4.4.2.1. Class Diagram for Subsystem BikeInformation

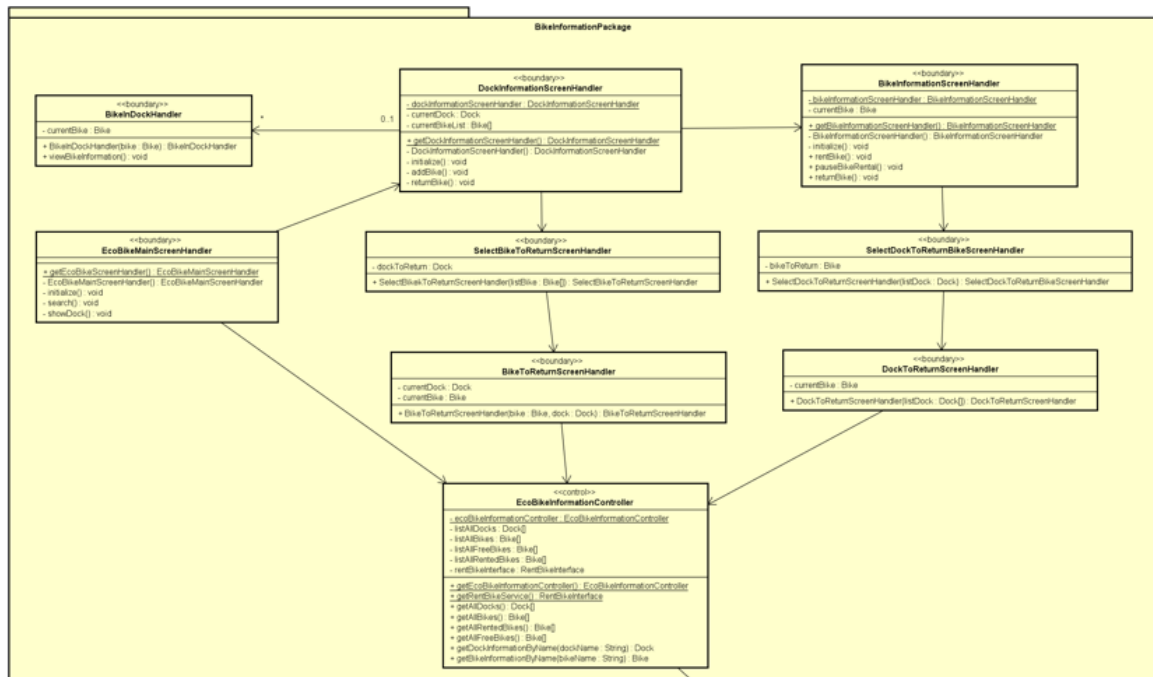


Figure 4.5. Class Diagram for Package BikeInformation

4.4.2.2. Class Diagram for Subsystem RentBike

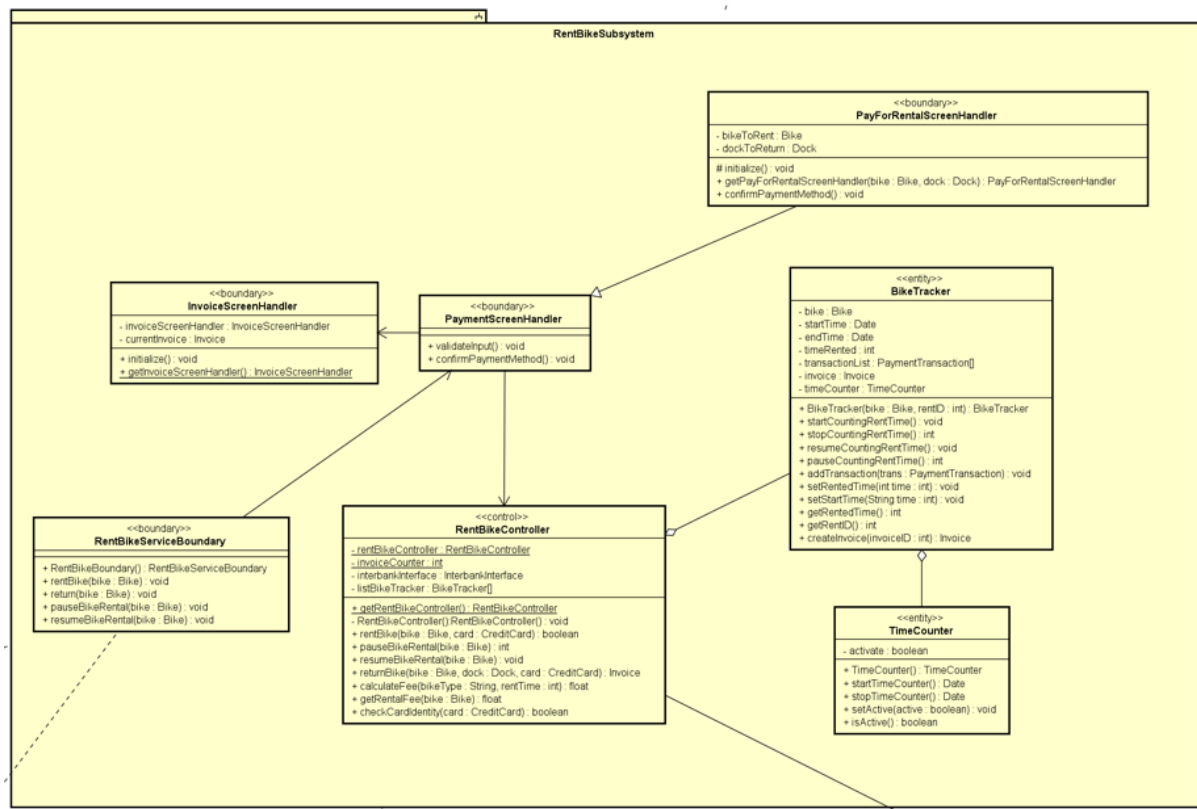


Figure 4.6. Class Diagram for Subsystem RentBike

4.4.2.3. Class Diagram for Subsystem InterBank

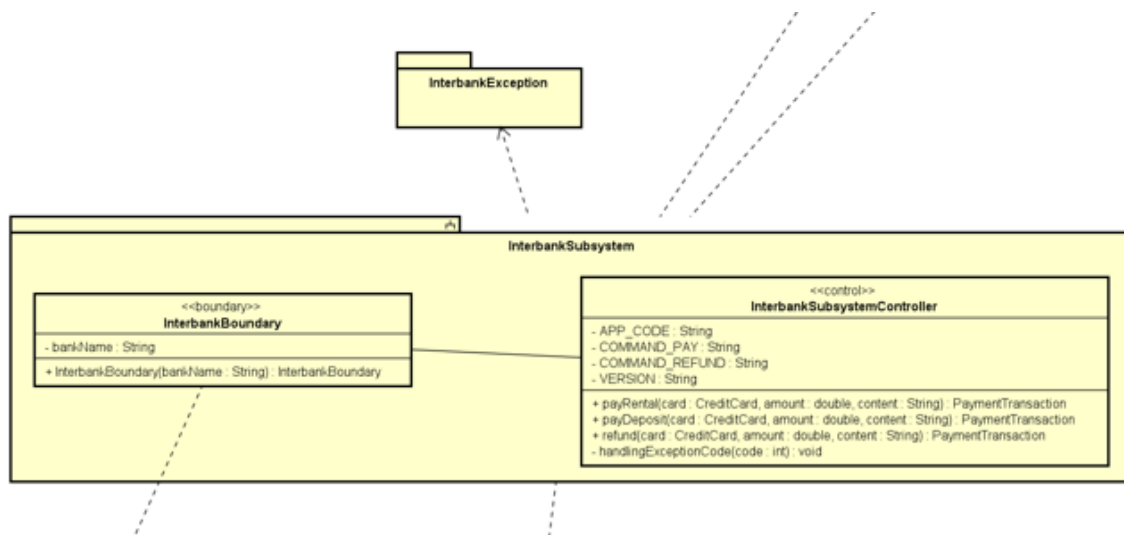


Figure 4.7. Class Diagram for Subsystem InterBank