HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

School of Information and communications technology

Ecobike rental

Subject: Software development ITSS

Group: 13

Phạm Thảo Nhi – 20205190

Hoàng Văn Phương – 20200478

Nguyễn Trần Minh Ngọc – 20205189

Nguyễn Hợp Phú - 20205165

Hanoi, 06/2023

Table of Contents

Table of Contents 1
1 Introduction 4
1.1 Objective 4
1.2 Scope 4
1.3 Glossary 4
1.4 References 4
2 Overall Description 5
2.1 General Overview 5
2.2 General use case diagram 6
2.3 Assumptions/Constraints/Risks 6
2.3.1 Assumptions 6
2.3.2 Risks 6
3 Detailed Requirements 7
3.1 Constraints 7
3.2 Use case Specifications 7
3.2.1 Use case Search dock 7
3.2.2 Use case Rent bike 9
3.2.3 Use case Pay deposit 11
3.2.4 Use case Return bike 15
4 System Architecture and Architecture Design 18
4.1 Architectural Patterns 18
4.2 Interaction Diagrams 19
4.2.1 Search dock 19
4.2.2 Rent bike 20
4.2.3 Pay Deposit 21
4.2.4 Return bike 22

4.3 Analysis Class Diagrams 23
4.3.1 Search dock 23
4.3.2 Rent bike 24
4.3.3 Pay Deposit 24
4.3.4 Return bike 25
4.4 Unified Analysis Class Diagram 25
5 Detailed Design 26
5.1 User Interface Design 26
5.1.1 Screen Configuration Standardization 26
5.1.2 Screen Transition Diagrams 26
5.1.3 Screen Specifications 26
5.2 Data Modeling 32
5.2.1 Conceptual Data Modeling 32
5.2.2 Database Design 32
5.3 Class Design 36
5.3.1 General Class Diagram 36
5.3.2 Class Diagrams 37
6 Design Considerations 40
6.1 Goals and Guidelines 40
6.2 Architectural Strategies 40
6.3 Coupling and Cohesion 40
6.4 Design Principles 40
7 Task Assignment 42
8 References 43

1 Introduction

1.1 Objective

Applying the knowledge from ITSS course, we develop the application from requirement step, analysis step, design to coding step and testing phase.

This document offers a thorough architectural overview of the system, using a variety of architectural viewpoints to illustrate various system components. It aims to apply the important software architecture knowledge that have been studied.

1.2 Scope

Our application provides users with bike rental service. Due to the requirement, we do not install authentication funtionality, we just focus on functionalities about watching information, renting, returning bikes, and purchasing. We also provide friendly-user interface for high experience.

1.3 Glossary

Term (mấy từ viết tắt)	Explain
ITSS	Information Technology Systems and Services
RAM	Random Access Memory
IDE	Integrated Development Environment
JRE	Java Runtime Environment

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 Requirements Document is divided into 6 sections with various subsections.

The two first sections are the introduction and the general overview of the Eco Rental Bike project.

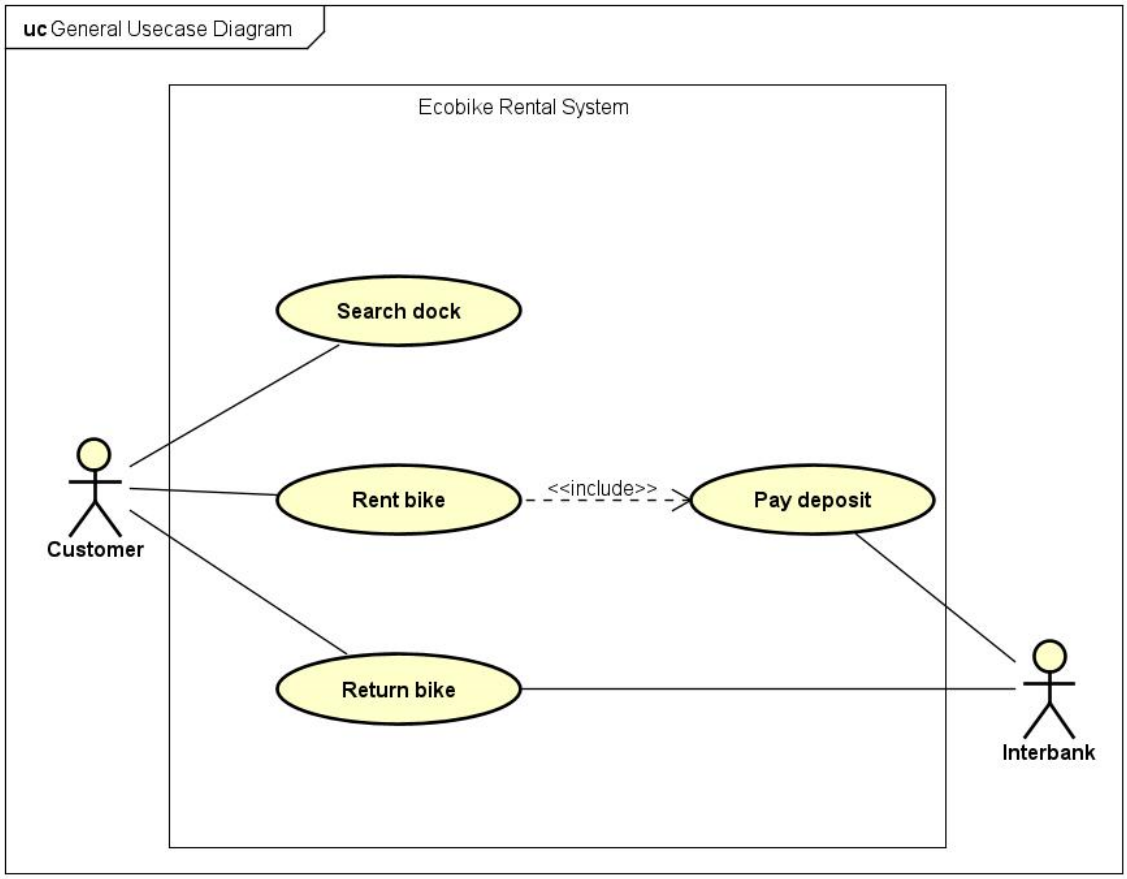
The third section is the detail requirements where we provide the various types of diagram and its specifications.

The next part is System architecture and architecture design. In this part, we describe the architecture which we choose to develop the software and some information about it. The detailed design of the architecture is also in this section.

Section 5 is detailed design part which contains the interface design, class design and so on.

The sixth section is the design consideration in which we appreciate the design pattern, design concept, ...

2.2 General use case diagram



2.3 Assumptions/Constraints/Risks

2.3.1 Assumptions

Users of the software should have a stable Internet connection. Additionally, because our program is a desktop application, the user also needs a laptop or desktop computer with an operating system (we recommend 64-Bit Microsoft Windows 10 or later, macOS 10.13 or later, or any Linux distribution that allows running applications) in order to utilize the apps. Regarding the system requirements, we would say that you need at least 2 GB of RAM and a solid-state drive with at least 5 GB of free space. The most recent version of JRE is also required, and a screen resolution of 1280 x 800 is advised. Eclipse IDE and Visual studio code also need updating.

2.3.2 Risks

3 Detailed Requirements

3.1 Constraints

3.2 Use case Specifications

3.2.1 Use case Search dock

Use Case "Search dock"

1. **Use case code**

   UC001

2. **Brief Description**

   This use case describes the interaction between customer and rental bike software when customer wishes to search a dock

3. **Actors**

   Customer

4. **Preconditions**

5. **Basic Flow of Events**

   1. The customer enters name of dock in search bar and click button search
   2. The software executes request from users
   3. The software displays list of dock

6. **Alternative flows**

No	Location	Condition	Action	Resume location
1	At Step 2	If no dock is found	System displays message 'No dock is found'	Step 1

7. **Input data**

No	Data fields	Description	Mandatory	Valid condition	Example
1.	Name	Name of the dock	Yes	A string	Park 1 station

8. **Output data**

List of docks

No	Data fields	Description	Display format	Example

1.	ID	ID of dock	Int	1
2.	Dock Name	Name of dock	String	Dock 1
3.	Address	Address of dock	String	So 1 Dai Co Viet
4.	Area	Area of the dock	Int	100
5.	Total Bike	Number of total bikes	INT	6
6.	Available Bike	Number of available bikes	INT	3

9. Postconditions

None

3.2.2 Use case Rent bike

Use Case "Rent bike"

1. Use case code

UC004

2. Brief Description

This use case describes the interaction between customer and rental bike software when customer wishes to rent a bike.

3. Actors

Customer
Interbank

4. Preconditions

None

5. Basic Flow of Events

1. Customer presses the rent bike button on screen.
2. The system requires customer to enter bike's barcode and dock's id
3. Customer enters barcode and dock id
4. The system converts barcode into bike ID in the system
5. The system queries the rental information of corresponding bike ID
6. The system displays the bike rental information on screen
7. Customer confirms bike rental information
8. The system display invoice deposite
9. Customer click confirm deposite button from the screen.
10. The system requests interbank to pay deposite
11. The system display status of rented bike

6. Alternative flows

No	Location	Condition	Action	Resume location
1.	At Step 1	If the customer wants to see the dock information first, then borrow the bike	- Customers click on the dock to see the detailed information of the dock - User enters the barcode of the bike they wants to rent in the dock	Step 6
2	At Step 6	If barcode or dock id exists or is not valid	Invalid barcode or dockid notification system	Step 3

7. Input data

No	Data fields	Description	Mandatory	Valid condition	Example
1.	Barcode	Barcode of the bike	Yes	A string	A01B0001
2.	Dock ID	ID of dock	No	Number	1

8. Output data

Information of bike rental

No	Data fields	Description	Display format	Example
1.	Bike ID	ID of the bike	A string	000111
2.	Barcode	Barcode of the bike	A string	A01B0001
3.	Type	Type of the bike (standard bicycle, standard e-bike, twin bike)	A string	standard e-bike

4.	License plate	License plate of the bike	A string	29MD100619
5.	Brand	Brand name of the bike	A string	Thong Nhat
6.	Battery Percentage	Current battery percentage of e-bike	A number followed by the percent symbol	75%
7.	Deposit	Amount of deposit	A number, followed by unit measurement 'VND'	400000 VND

Updated rental bike information

No	Data fields	Description	Display format	Example
8.	Rental time	The rental period is up to the present time.	Time format	1h20'
10.	Battery Percentage	Current battery percentage of e-bike	A number followed by the percent symbol	75%

10. Postconditions

NONE

3.2.3 Use case Pay deposit

Use Case "Pay Deposit"

1. **Use case code**

   UC004

2. **Brief Description**

   This use case happens when the customers start renting bikes. When the users rent bikes, they have to pay a deposit.

3. **Actors**

Customer, Interbank

4. **Preconditions**

Customer confirms rent bike

5. **Basic Flow of Events**

1. The System displays Credit Card form
2. User fills form and enters next button
3. The System validates data format (including expired date and security code)
4. The System displays confirmation form
5. User confirms the credit card information
6. The System requires Interbank to perform transaction
7. The System notifies successful payment

6. **Alternative flows**

No	Location	Condition	Action	Resume location
1	At Step 2	If the customer doesn't fill in enough information or there is any information in wrong format	System notifies user to fill all the field again	Step 1
2	At Step 3	If the data is in wrong format	System notifies user to fill the field again	Step 1
3	At Step 6	If the transaction failed	System notifies user that the transaction was unsuccessful, attached with related error message (incorrect security code, not enough balance)	Step 1

7. **Input data**

Credi card information form:

No	Data fields	Description	Mandatory	Valid condition	Example
1.	Cardholder name	Owner of the card	Yes	A string	Group 13
2.	Card number	Card number	Yes	A string	4242242
3.	CVV	Security code	Yes	A string of 3 numbers	123
4.	Expired date	Expired date of the card	Yes	A string in format MM/YY	03/27
5.	Content	Content of the transaction	No	A string	Deposit to Ecobike reantal company

8. Output data

Confirmation form

No	Data fields	Description	Display format	Example
1.	Cardholder name	Owner of the card	A string	Group 13
2.	Card number	Card number	A string	4242242
3.	Amount	Amount of deposit	A string of number, followed by unit mark "đ"	400.000 đ
4.	Expired date	Expired date of the card	A string in format MM/YY	03/27
5.	Content	Content of the transaction	A string	Deposit to Ecobike reantal company

6.	Command	Marks the transaction as deposit	A string	Deposit

Bike Status Information

No	Data fields	Description	Display format	Example
1.	Barcode	Barcode of the bike	String of length 12	111123456111
2.	Bike type	Type of the bike	String	E-bike
3.	Bike brand	Brand of bike	String	Thong Nhat
4.	License Plate (if needed)	License of the bike	String	29AA1999
5.	Pin (if needed)	Battery status of bike	Int	100%
6.	Stop watch	Counting rental time		

9. Postconditions

Switch to Bike Status Screen.

3.2.4 Use case Return bike

Use Case "Return bike"

1. **Use case code**

   UC005

2. **Brief Description**

   This Use Case describes the interaction between a User and the EcoBikeRental Software when the user intends to return the bike following the rental period.

3. **Actors**

   User

   Interbank

4. **Preconditions**

User must rent a bike and complete the deposit payment

5. **Basic Flow of Events**
   1. Customer clicks the "Return bike" button from the bike status screen
   2. The System displays rental time and list of dock
   3. Customer selects a dock from the list displayed on the screen
   4. The system calculates rental fee
   5. The system displays pre **invoice**
   6. Click the " Confirm payment" button from the screen (check out)
   7. The system requests interbank to refund
   8. The system display invoice

6. **Alternative flows**

Table N-Alternative flows of events for UC Return bike

No	Location	Condition	Alternative flow	Resume location
1.	At step 3	If all docks are currently occupied and unavailable If customer doesn't choose any dock	- The system displays a message informing the customer that all docks are occupied. - The customer is prompted to wait until a dock becomes	At step 3

			available or choose an alternative nearby dock.	
2.	At step 7	If service payment step has error	- The system displays an error message - The customer is prompted to retry the payment	At step 6

7. **Input data**

No	Data fields	Description	Mandatory	Valid condition	Example
1.	Bike ID	Bike ID number	Yes	A string	11
2.	nameDock	The name of the dock you will return bike	Yes	String	SO 1 DAI CO VIET
3.	depositID	Id of previous deposit transaction, get from object Invoice	Yes	String	APTX4869

8. **Output data**

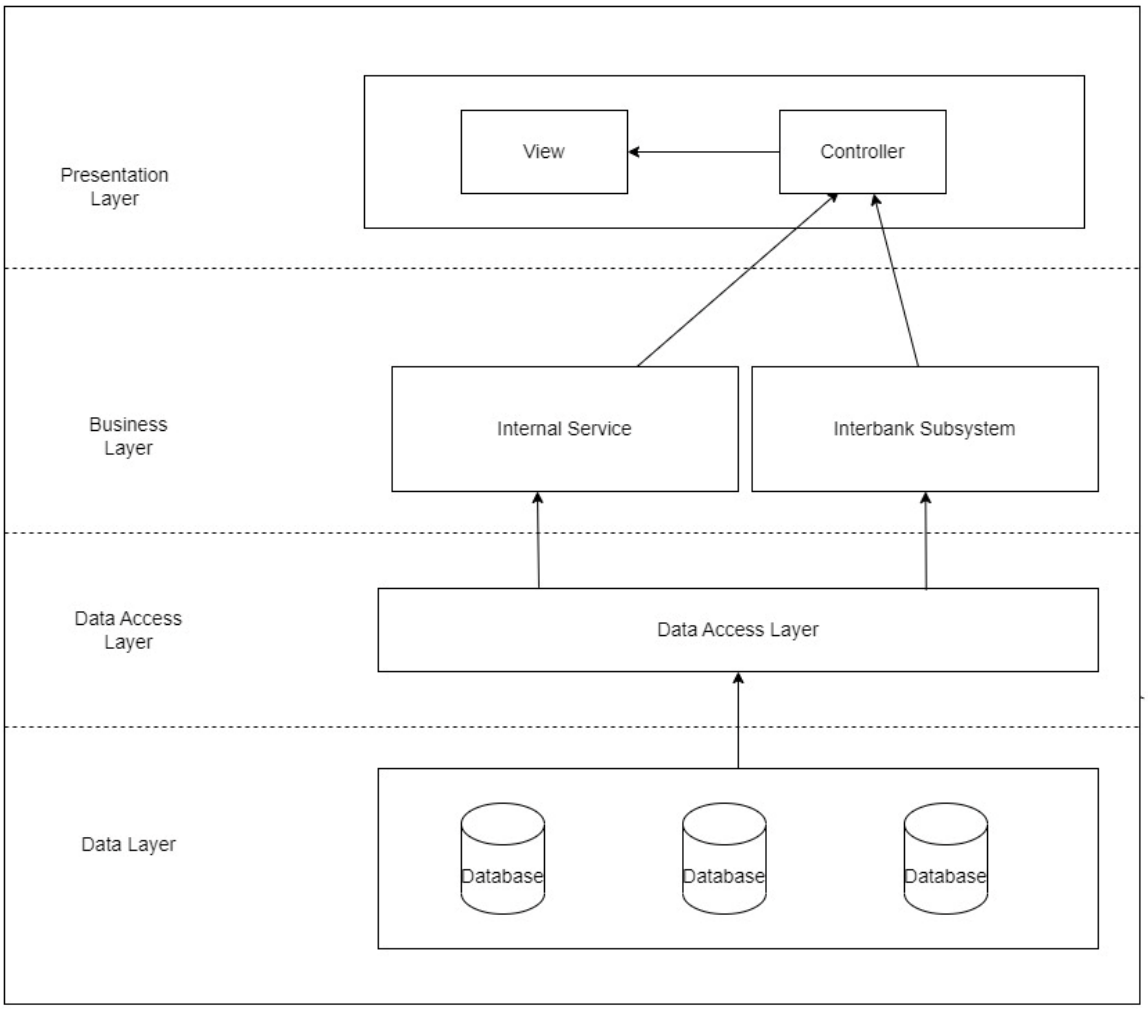
Invoice

No	Data fields	Description	Display format	Example
1.	InvoiceCode	Trading code	none	11AB2sxr
2	CardHolder Name	Name of customer from card entered	none	Nguyen Van A
3	nameBike	Type of rented bike	string	Thong Nhat 1
3	Time travel	Rental time	hh/mm/ss	01/30/00
3.	Amount deposit	Amount paid in advance	[Number]đ	400000đ
4.	Amount rental	Bike rental fee	[Number]đ	20000đ
5.	Amount refuned		[Number]đ	380000đ

9. **Postconditions**

None

4 System Architecture and Architecture Design



4.1 Architectural Patterns

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

This software is developed based on 3-tier architecture.

Presentation tier: is deployed to a computing device through a web browser or a web-based application. The presentation tier communicates with the other tiers through application program interface (API) calls.

Application tier: contains the business logic that supports the application's core functions. The underlying application tier can either be hosted on the distributed servers in the cloud or on a dedicated in-house server, depending on how much processing power the application requires.

Data tier: consists of a database and a program for managing read and write access to database. This tier may also refer to as the storage tier and can be hosted on-premises or in the cloud.

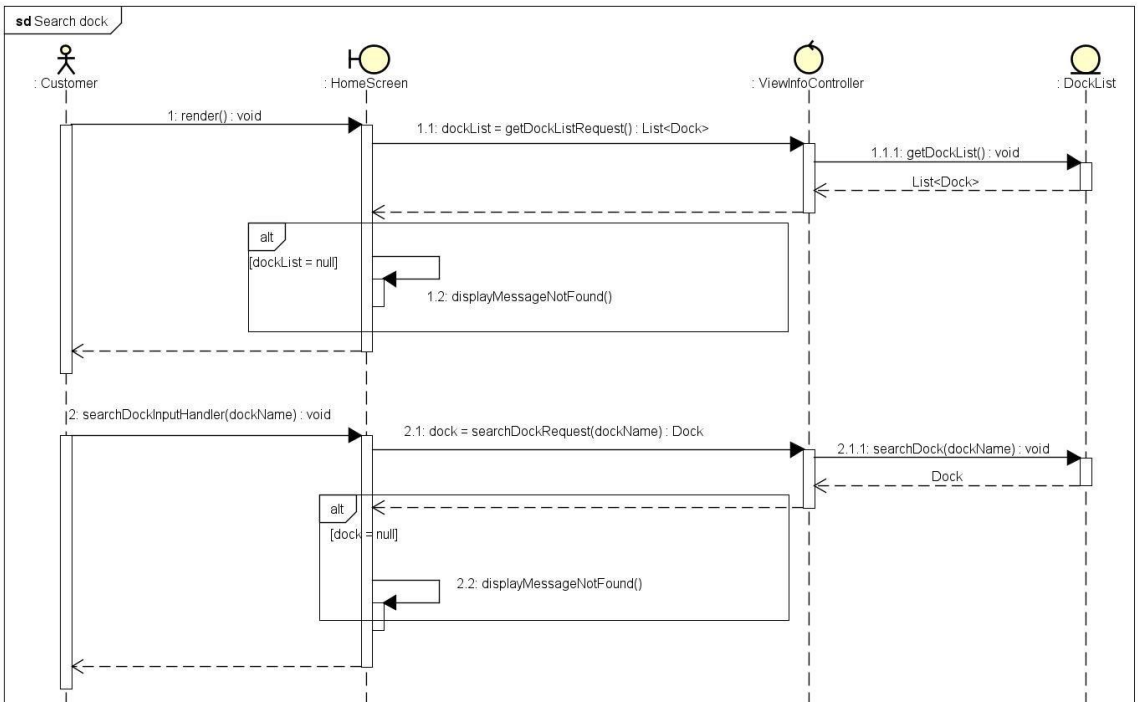
Benefits:

- Faster development: because each tier can be developed simultaneously by different teams, an organization can bring the application to market faster, and programmers can use the latest and best languages and tools for each tier.
- Improved scalability: any tier can be scaled independently of the others as needed.
- Improved reliability: an outage in one tier is less likely to impact the availability or performance of the other tiers.
- Improved security: because the presentation tier and data tier can't communicate directly, a well-designed application tier can function as a sort of interval firewall, preventing SQL injections and other malicious exploits.

4.2 Interaction Diagrams

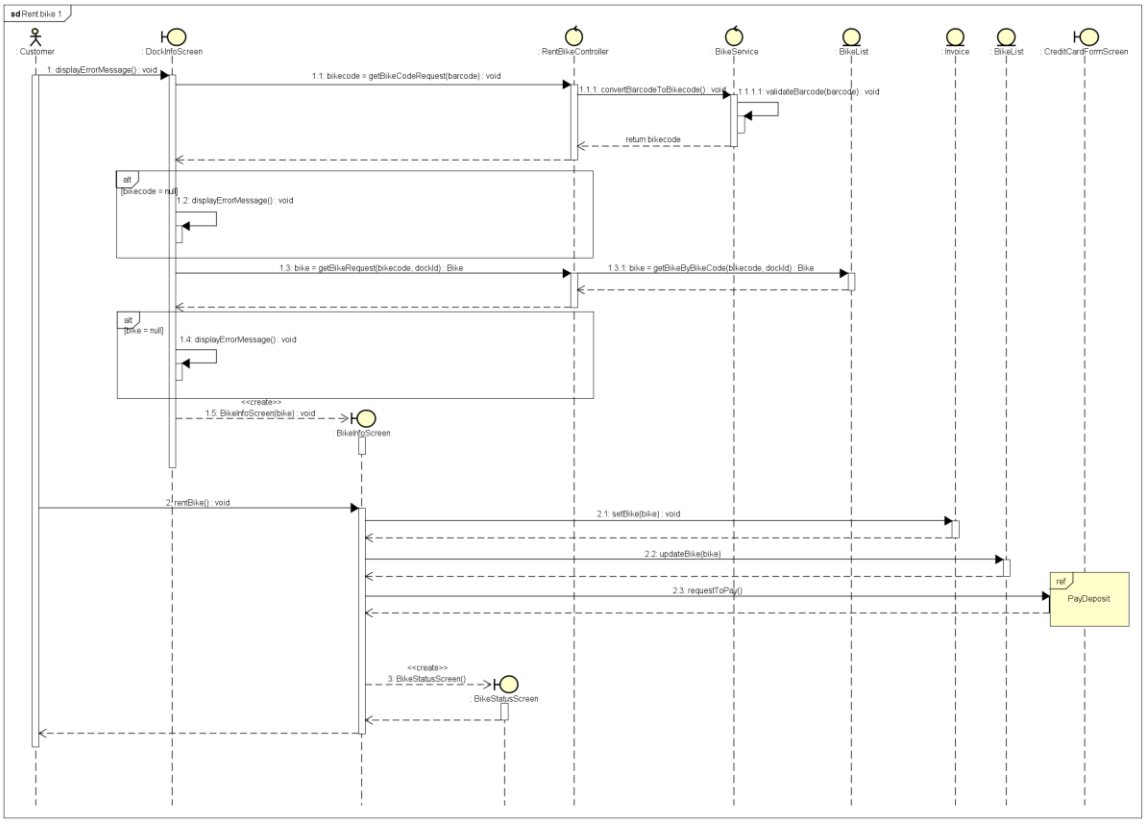
We use interaction diagram(s), i.e., sequence diagram to allocate responsibilities to classes and model analysis class interactions

4.2.1 Search dock

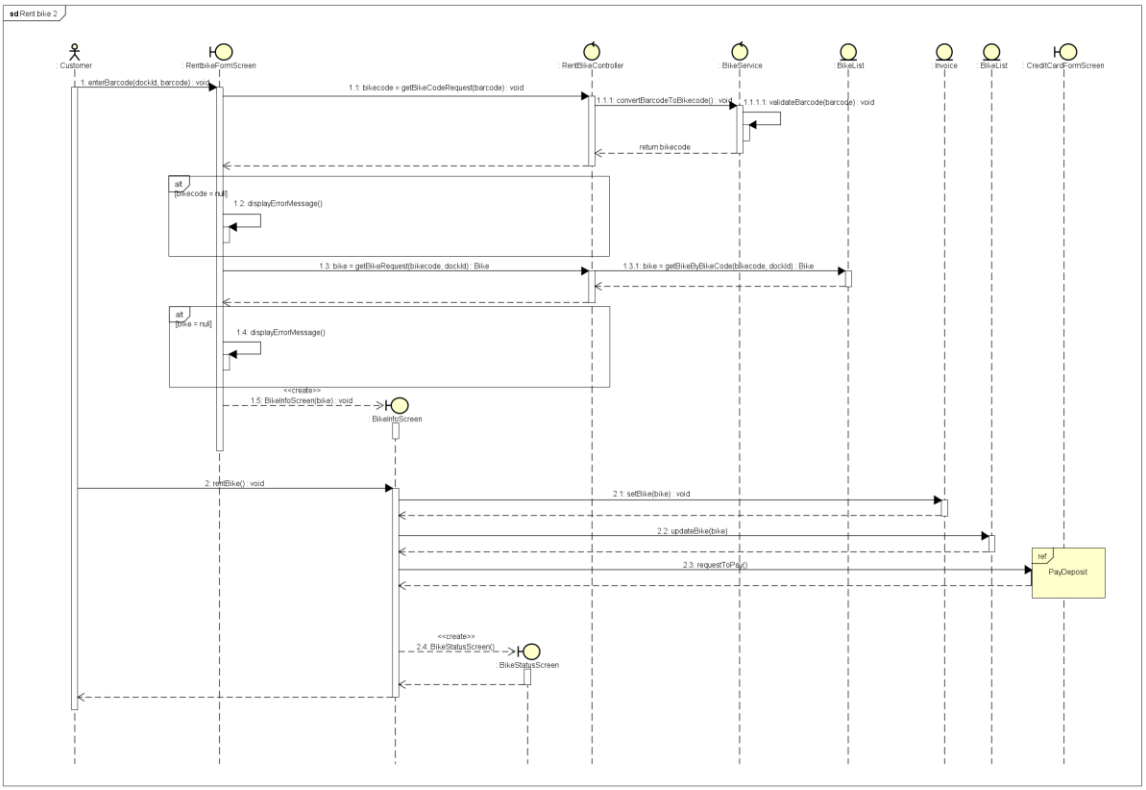


4.2.2 Rent bike

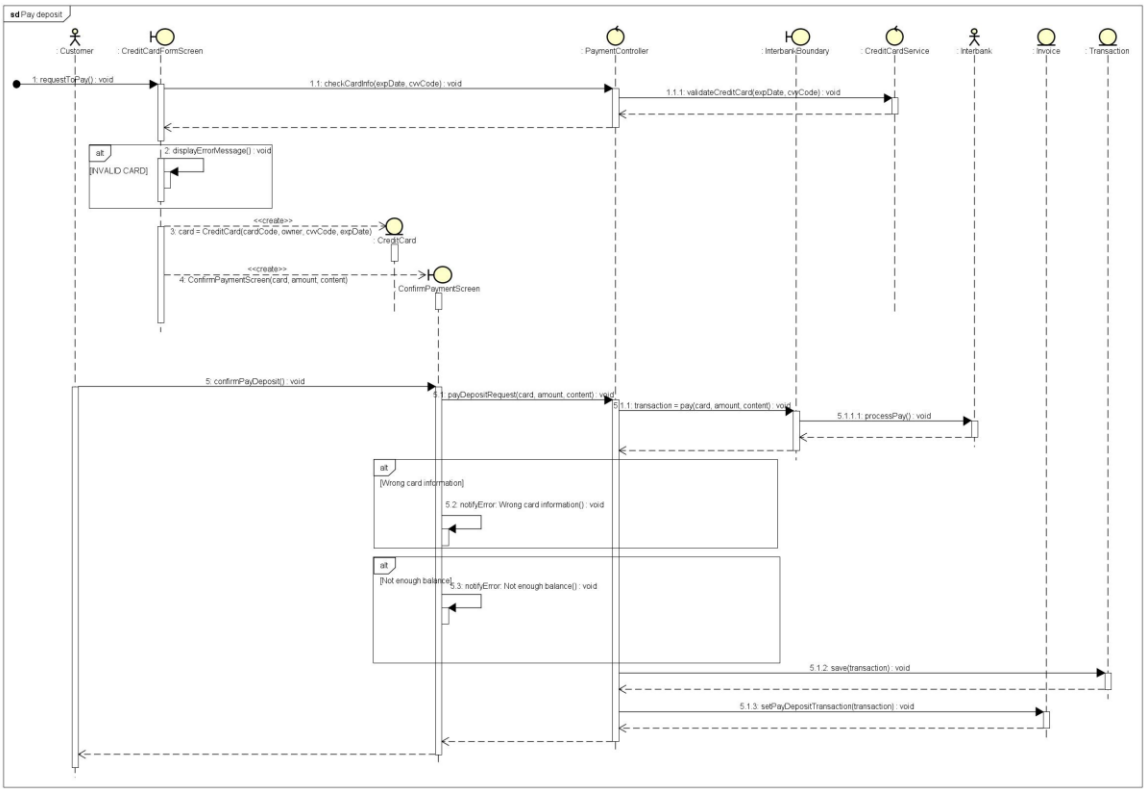
Start from DockInforScreen:



Start from RentbikeFormScreen:

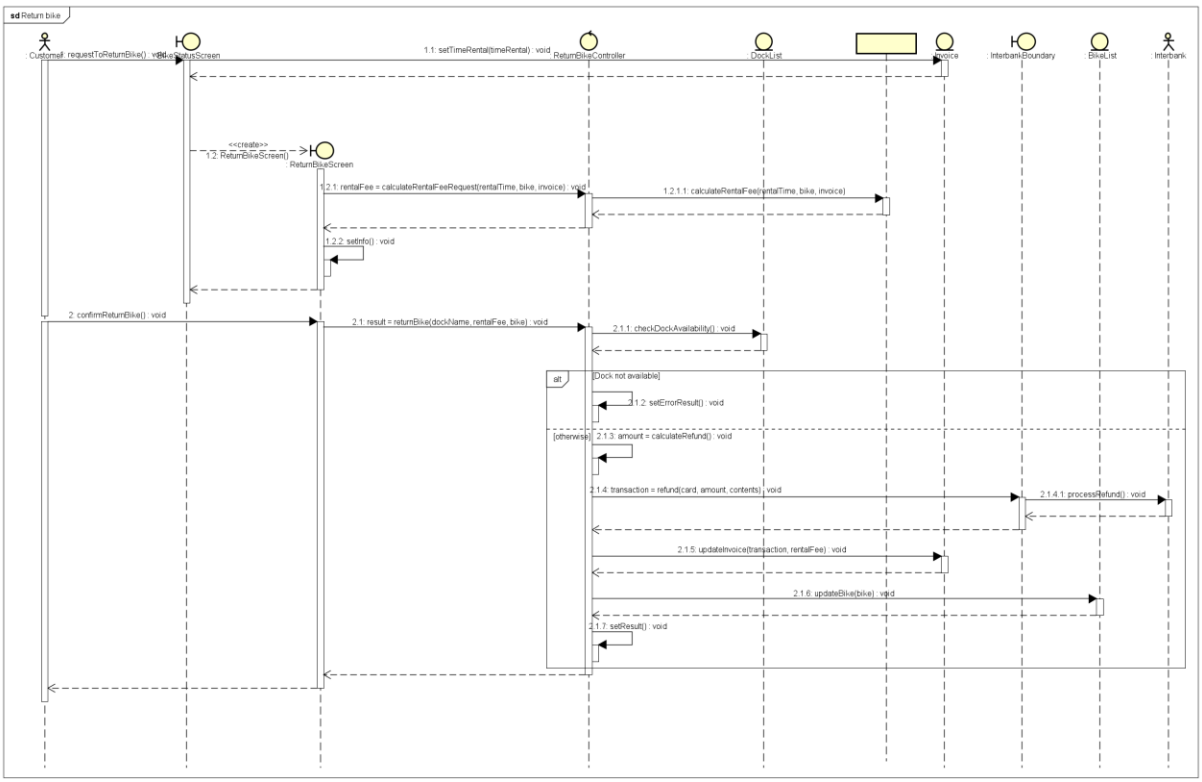


4.2.3 Pay Deposit



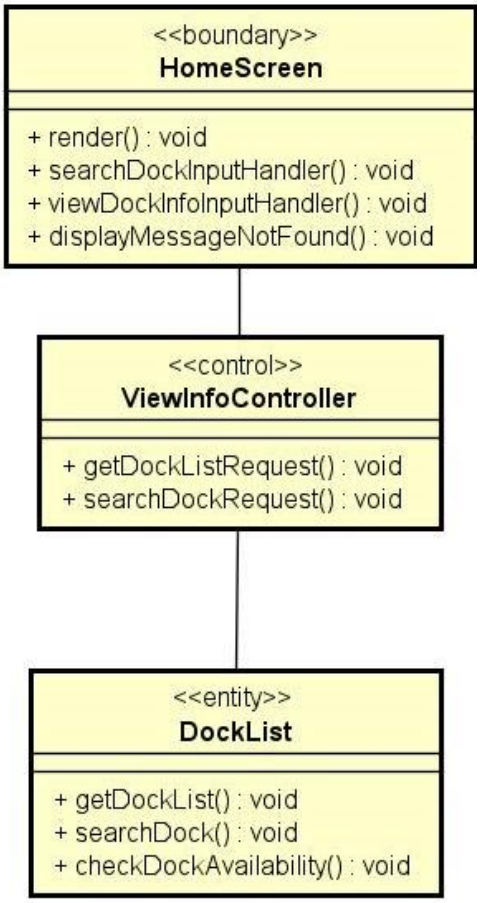
Interbank Subsystem

4.2.4 Return bike

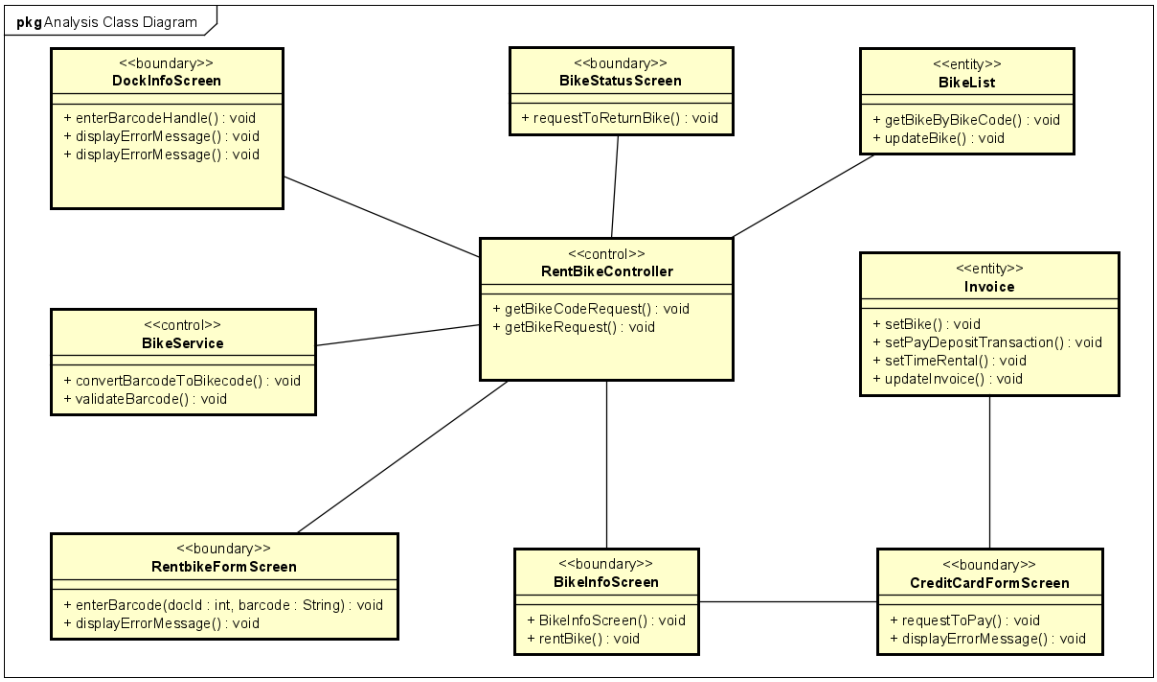


4.3 Analysis Class Diagrams

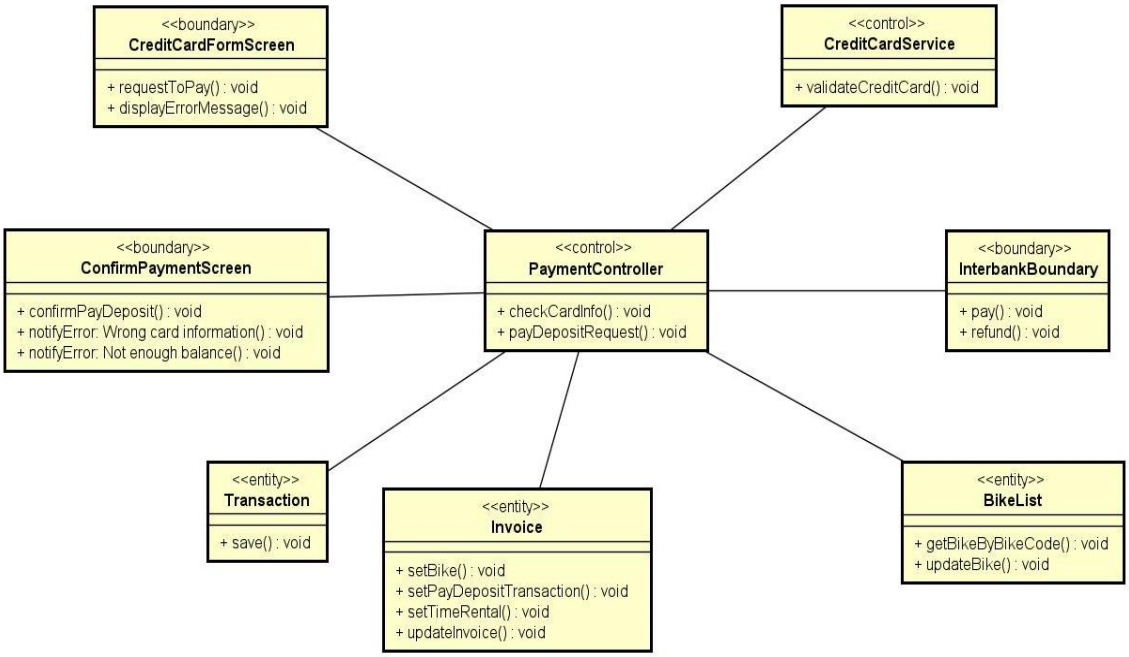
4.3.1 Search dock



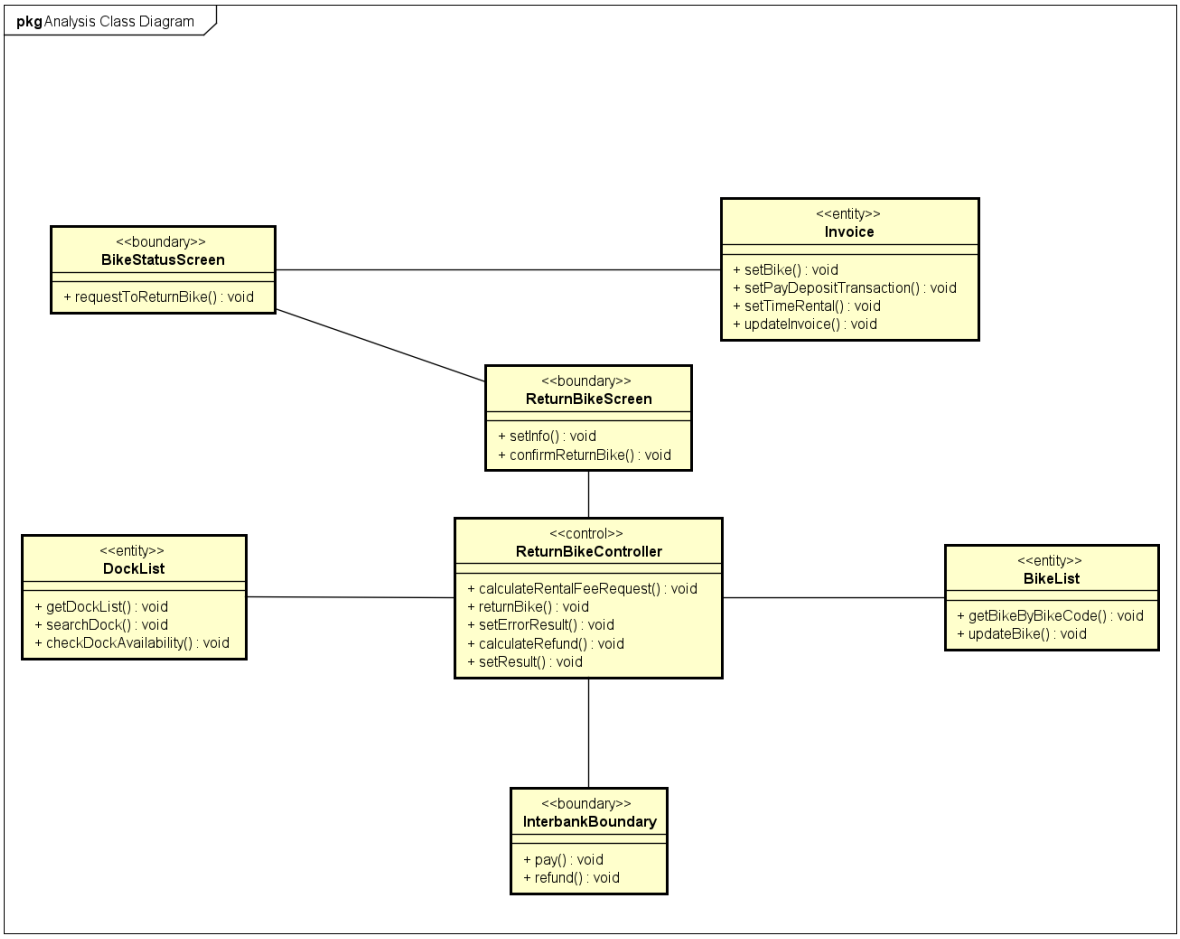
4.3.2 Rent bike



4.3.3 Pay Deposit



4.3.4 Return bike



4.4 Unified Analysis Class Diagram

5 Detailed Design

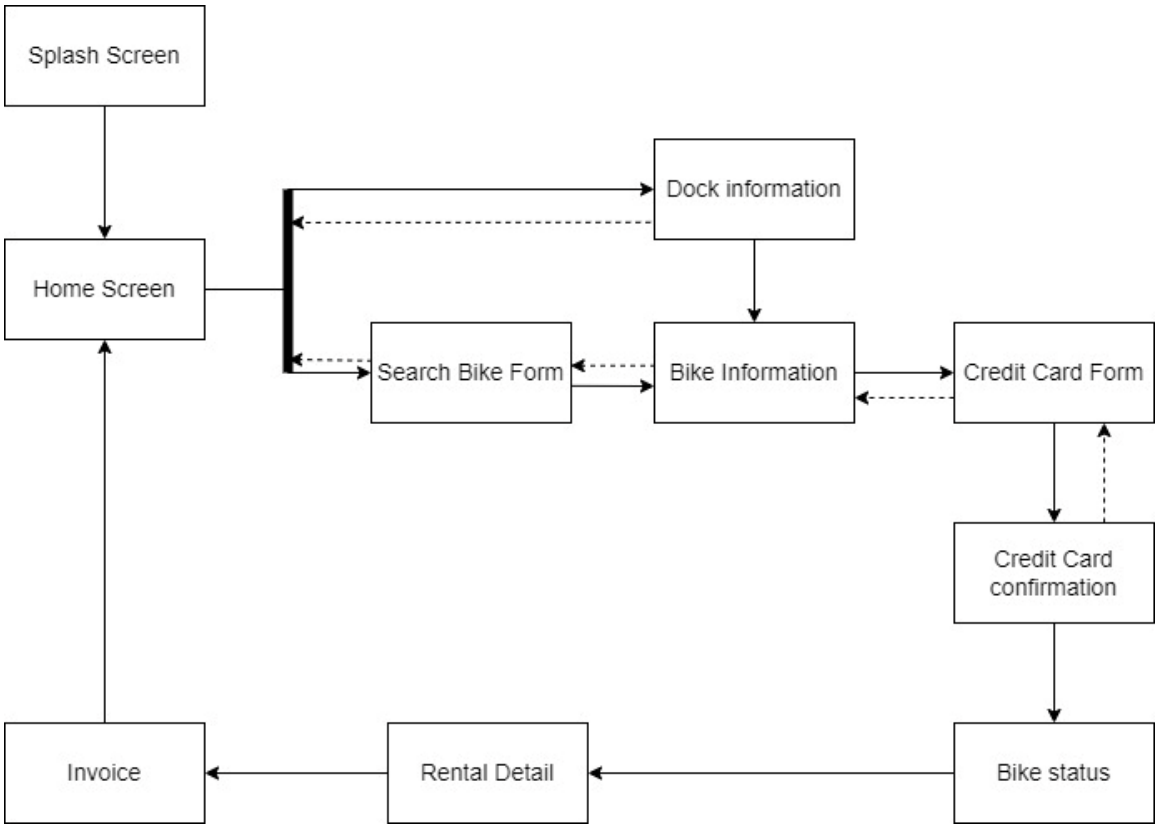
5.1 User Interface Design

5.1.1 Screen Configuration Standardization

Display screen:

- Display Resolution: Full HD (1920x1080)

5.1.2 Screen Transition Diagrams

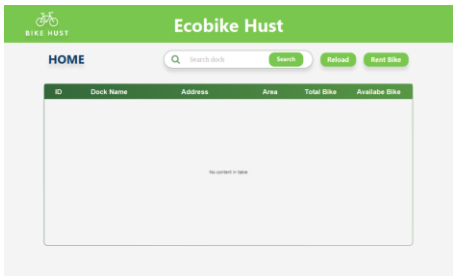


5.1.3 Screen Specifications

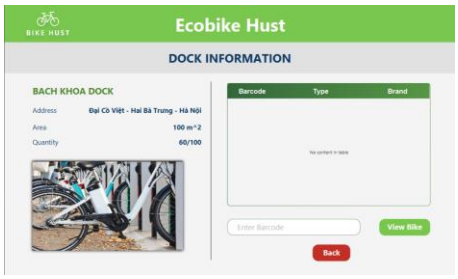
<Screen images should be included in the screen specifications>

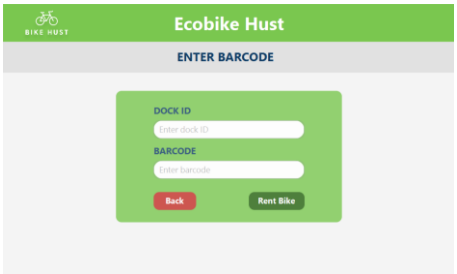
Ecobike Rental Software		**Date of creation**	**Approved by**	**Reviewed by**	**Person in change**
Screen specification	Splash screen	15/07/2023			Nguyen Tran Minh Ngoc
		Control	**Operation**	**Function**	

BIKE HUST SELECT AND GO	Display start screen	Initial	

Ecobike Rental Software		Date of creation	Approved by	Reviewed by	Person in change
Screen specification	Home screen	15/07/2023			Nguyen Tran Minh Ngoc
		**Control**	**Operation**	**Function**	
		Dock list display area	Initial	Display information about the docks in the system and the dock searched by the user	
		Search dock input field	Enter input	Allow customer to enter a name to search for dock	
		Search button	Click	Search dock	
		Rent bike button	Click	Allow customer to rent bike	
		Reload button	Click	Reload the home screen	

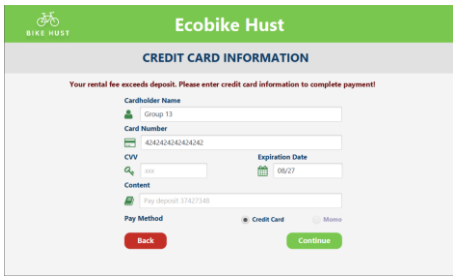
Ecobike Rental Software		Date of creation	Approved by	Reviewed by	Person in change
Screen specification	Dock information screen	15/07/2023			Nguyen Tran Minh Ngoc
		Control	**Operation**	**Function**	

	Area on the left shows dock's information	Initial	Display detailed information of the dock
	Area on the right shows information of bikes in the dock	Initial	Display detailed information of bikes in the dock
	Barcode input field	Enter input	Allow customer to enter bike barcode
	View bike button	Click	View detailed bike's information
	Back button	Click	Back to previous screen

Ecobike Rental Software		**Date of creation**	**Approved by**	**Reviewed by**	**Person in change**
Screen specification	Search Bike Form screen	15/07/2023			Nguyen Tran Minh Ngoc
		**Control**	**Operation**	**Function**	
		Dock input field	Enter input	Allow customer to enter dock ID to search bike in the dock	
		Barcode input field	Enter input	Allow customer to enter bike barcode	
		View bike button	Click	Display bike's information based on input data	
		Back button	Click	Back to previous screen	

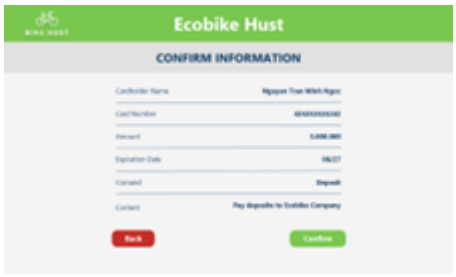
Ecobike Rental Software	**Date of creation**	**Approved by**	**Reviewed by**	**Person in change**

Screen specification	Bike information screen	15/07/2023			Nguyen Tran Minh Ngoc
		Control	**Operation**	**Function**	
		Area on the left shows bike's information	Initial	Display image of bike	
		Area on the right shows detailed bike's information	Initial	Display detailed information of bikes	
		Rent bike button	Click	Confirm to rent bike	
		Back button	Click	Back to previous screen	

Ecobike Rental Software		**Date of creation**	**Approved by**	**Reviewed by**	**Person in change**
Screen specification	Paydeposit form screen	15/07/2023			Nguyen Tran Minh Ngoc
		**Control**	**Operation**	**Function**	
		Cardholder's name input field	Enter input	Allow customer to enter cardholder name of card	
		Card number input field	Enter input	Allow customer to enter card number of credit card	
		CVV code input field	Enter input	Allow customer to enter CVV code of card	
		Expiration date input field	Enter input	Allow customer to enter expiration date of card	
		Content input field	Enter input	Allow customer to enter drilled content of card	
		Confirm button	Click	Confirm entered input data	

	Back button	Click	Back to previous screen

Ecobike Rental Software		Date of creation	Approved by	Reviewed by	Person in change
Screen specification	Confirm deposit screen	15/07/2023			Nguyen Tran Minh Ngoc

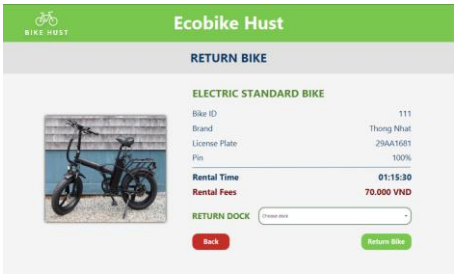


Control	Operation	Function
Invoice paydeposit display area	Initial	Display bike rental deposit invoice information
Confirm button	Click	Confirm invoice
Back button	Click	Back to previous screen

Ecobike Rental Software		Date of creation	Approved by	Reviewed by	Person in change
Screen specification	Bike status screen	15/07/2023			Nguyen Tran Minh Ngoc

Control	Operation	Function
Area on the left shows rental bike's information	Initial	Display image of rental bike
Area on the right shows detailed rental bike's information	Initial	Display detailed information of rental bikes
Return bike button	Click	Confirm to return bike

Ecobike Rental Software	Date of creation	Approved by	Reviewed by	Person in change

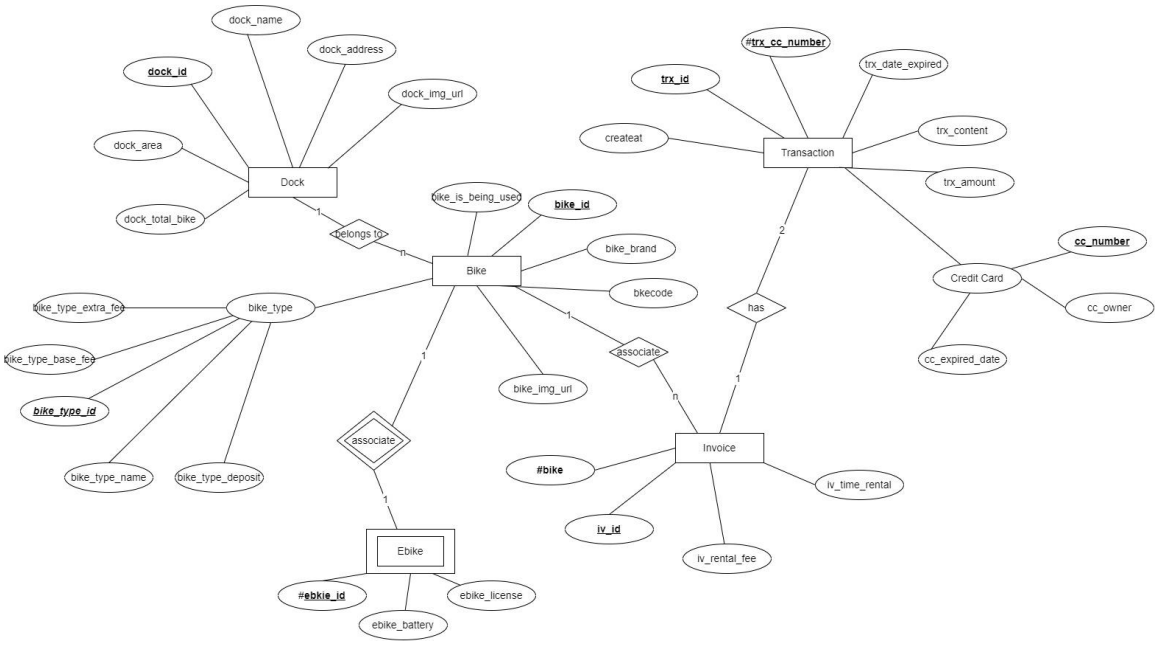
Screen specification	Return bike screen	15/07/2023			Nguyen Tran Minh Ngoc
		**Control**	**Operation**	**Function**	
		Area on the left shows returned bike's information	Initial	Display image of rental bike	
		Area on the right shows detailed returned bike's information	Initial	Display detailed information of returned bikes	
		Choose dock area	Select input	Allow customer to choose the dock to return the bike	
		Return bike button	Click	Confirm to return bike	
		Back button	Click	Back to previous screen	

Ecobike Rental Software		**Date of creation**	**Approved by**	**Reviewed by**	**Person in change**
Screen specification	Splash screen	15/07/2023			Nguyen Tran Minh Ngoc
		Control	**Operation**	**Function**	
		Invoice display area	Initial	Display payment invoice	
		Return home button	Click	Back to home screen	

5.2 Data Modeling

5.2.1 Conceptual Data Modeling

ER Diagram



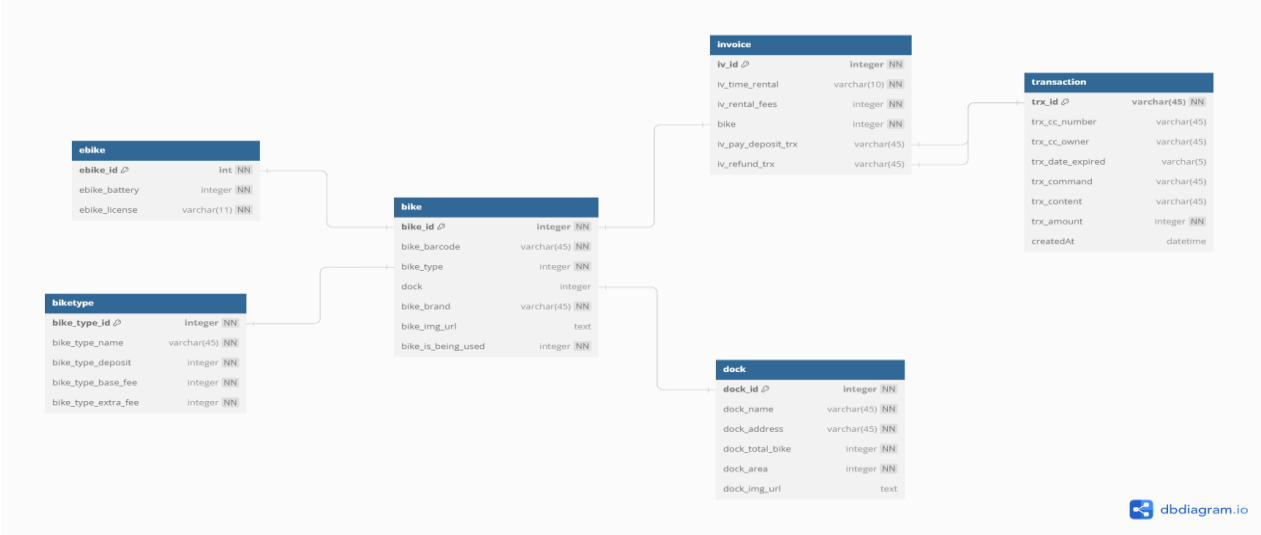
5.2.2 Database Design

5.2.2.1 Database Management System

In this project, we use MySQL as the DBMS for our system. MySQL operates on a client-server model, facilitating concurrent access to the centralized database by multiple client applications. The system's data architecture is crafted to ensure optimal storage, retrieval, and management of structured data. The employment of MySQL's robust SQL (Structured Query Language) capabilities empowers users to effectively interact with the data, employing functions, triggers, and stored procedures to manipulate and retrieve information.

5.2.2.2 Database Diagram

- Table Schema



5.2.2.3 Database Detail Design

Table "1ike". Example of table design

#	PK	FK	Column name	Data type	Default value	Mandatory	Description
1	x		bike_id	integer		x	The ID of the bike
2		x	bike_type	integer		x	The type of the bike
3			bike_barcode	varchar		x	The code of the bike
4		x	dock	integer		x	The dock where the bike is in
5			bike_brand	varchar		x	The brand of the bike
6			bike_image_url	text			The address of the photo of the bike
7			bike_is_being_used	integer		x	The bike is used or not

Table "2iketype". Example of table design

#	PK	FK	Column name	Data type	Default value	Mandatory	Description
1	x	x	bike_type_id	integer		x	The ID of the type of the bike
2			bike_type_name	varchar		x	The name of the type
3			bike_type_deposit	integer		x	The amount of money to deposit
4			bike_type_base_fee	integer		x	The fee of 30 first minutes
5			bike_type_extra_fee	integer		x	The fee after 30 minutes

Table “3”. Example of table design

#	PK	FK	Column name	Data type	Default value	Mandatory	Description
1	x	x	dock_id	integer		x	The ID of the dock
2			dock_name	varchar		x	The name of the dock
3			dock_address	varchar		x	The address of the dock
4			dock_total_bike			x	The number of bikes in the dock
5			dock_area			x	The area where the dock is in
6			dock_img_url				The address of the photo of the dock

Table “e4ike”. Example of table design

#	PK	FK	Column name	Data type	Default value	Mandatory	Description
1	x	x	ebike_id	integer		x	The ID of the ebike
2			ebike_battery	integer		x	The battery of the ebike
3			ebike_license	varchar		x	The license of the ebike

Table "5ice". Example of table design

#	PK	FK	Column name	Data type	Default value	Mandatory	Description
1	x		iv_id	integer		x	The ID of the invoice
2		x	iv_pay_deposit_trx	varchar			The ID of pay deposit transaction
3		x	iv_refund_trx	varchar			The ID of refund transaction
4			iv_time_rental	varchar		x	The amount of renting time
5			iv_rental_fees	integer		x	The rental fee of the bike
6		x	bike	integer		x	The id of the bike

Table "transaction". Example of table design

#	PK	FK	Column name	Data type	Default value	Mandatory	Description
1	x	x	trx_id	integer		x	The ID of the transaction

2			trx_cc_number	varchar		x	The credit card number of the transaction
3			trx_cc_owner	varchar		x	The owner of the credit card
4			trx_date_expired	varchar		x	The expired date of the credit card
5			trx_command	varchar			Pay or refund
6			trx_content	varchar			The content of the transaction
7			trx_amount	integer		x	The amount of money in transaction
8			createdAt	datetime			The moment the transaction is created

5.2.2.4 How to implement database

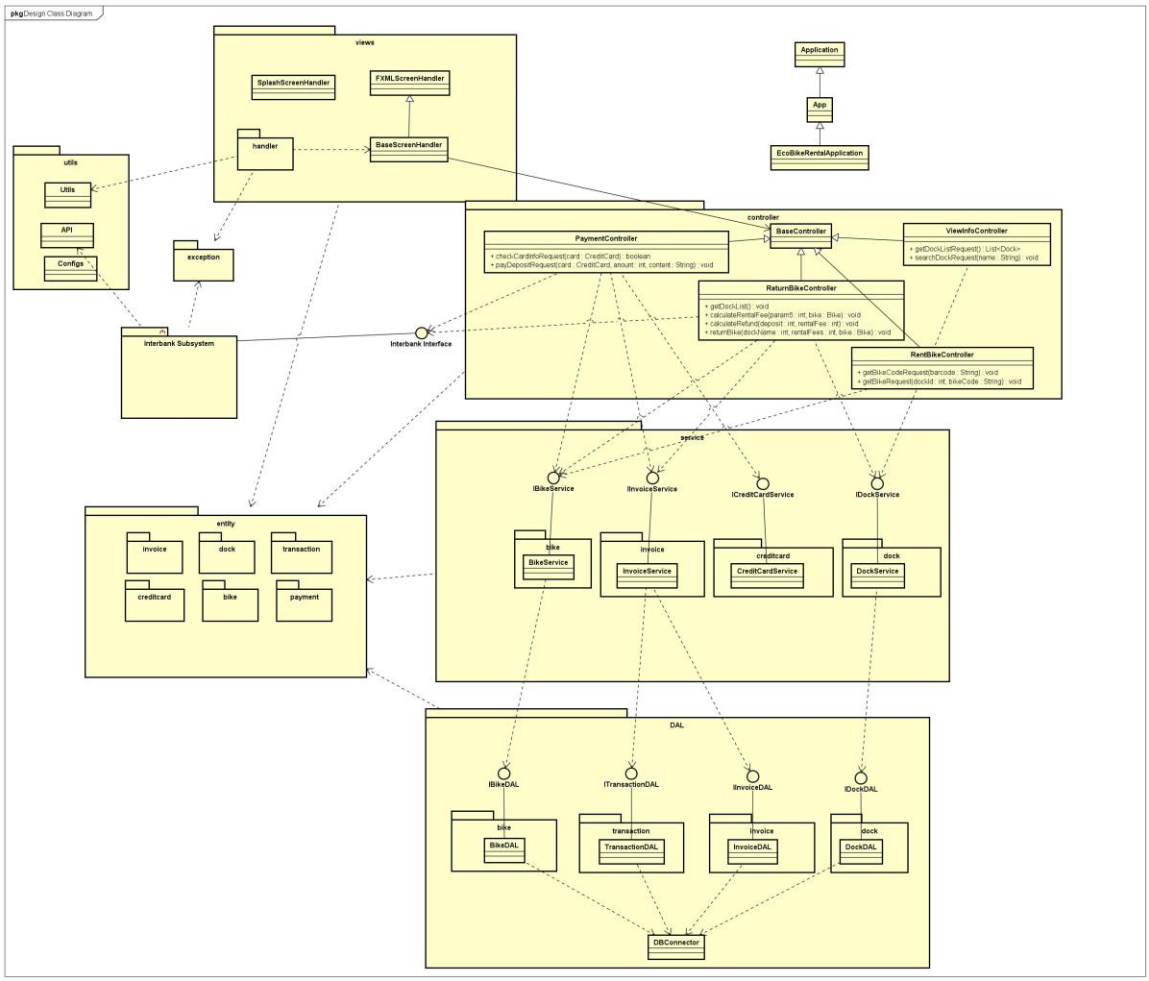
… dẫn link và hướng dẫn dump database

5.3 Class Design

5.3.1 General Class Diagram

5.3.2 Class Diagrams

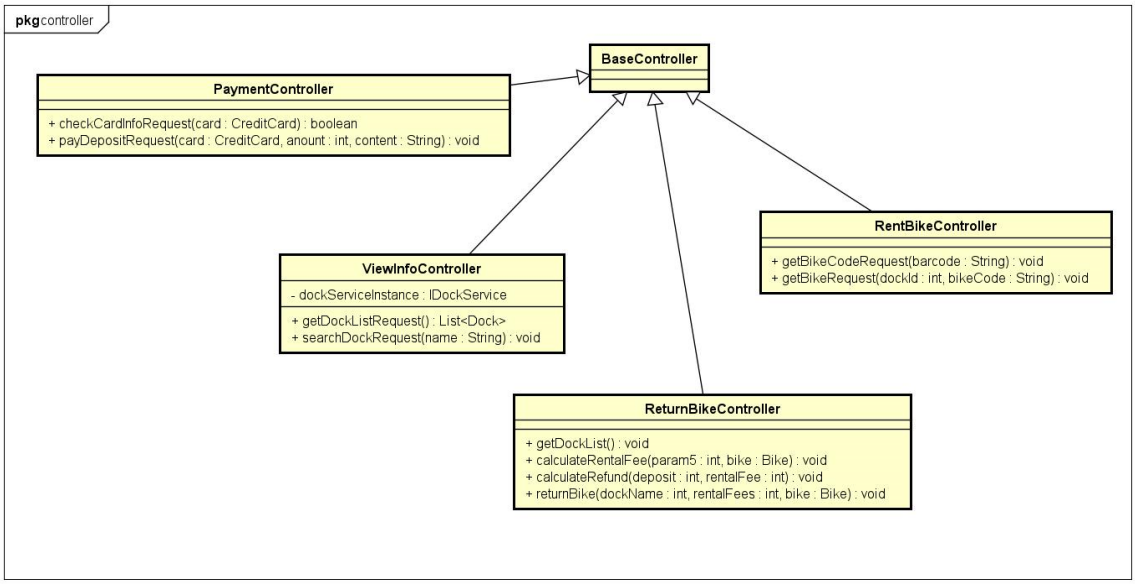
5.3.2.1 General Class Diagram



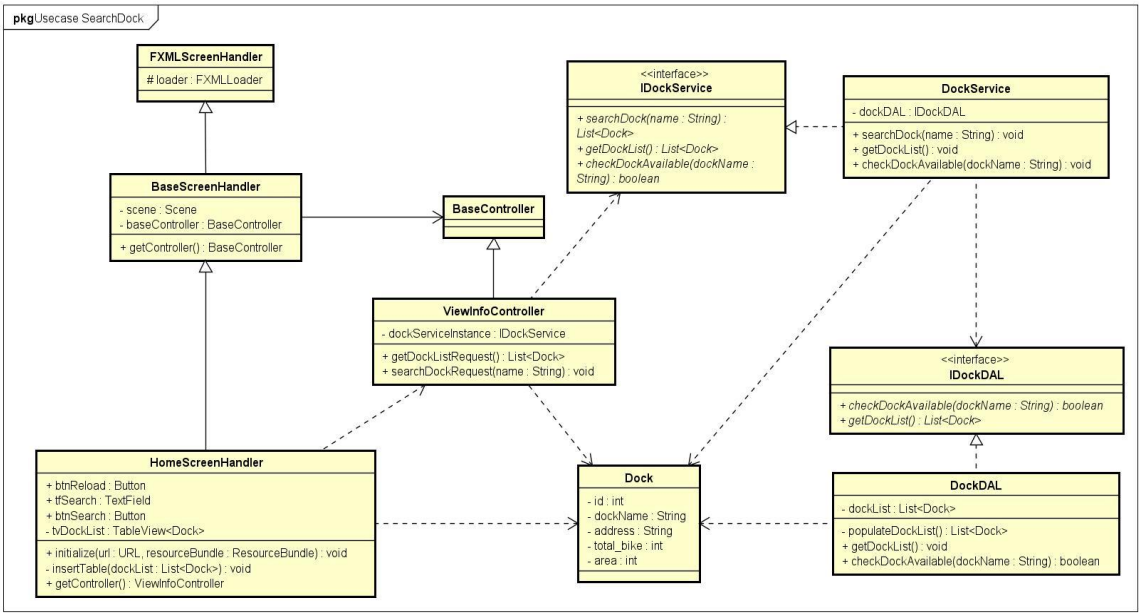
- Code organization

```
Ecobike-Rental-Project/
├── src/
│   ├── main/
│   │   ├── java/
│   │   │   ├── group13/                    // Our team
│   │   │   │   ├── ecobikerental/          // Application
│   │   │   │   │   ├── controller/         // controllers
│   │   │   │   │   ├── service/            // Business logic services
│   │   │   │   │   ├── DAL/                // Data access repositories
│   │   │   │   │   ├── entity/             // Data models and entities
│   │   │   │   │   ├── util/               // Utility classes
│   │   │   │   │   ├── views/              // Screen handler
│   │   │   │   │   ├── exception/          // Exception handler
│   │   │   │   │   ├── subsystem/          // Interbank subsystem
│   │   │   │   │   ├── App.java             // Main application class
│   │   │   │   │   ├── EcobikeRentalApplication.java
│   │   ├── resources/
│   │   │   ├── group13/                    // Our team
│   │   │   │   ├── ecobikerental/          // Application
│   │   │   │   │   ├── views/         // contains .fxml files
├── target/              // Compiled bytecode and built artifacts (ignored by version control)
├── pom.xml               // Maven project configuration
├── .gitignore            // Git ignore file
```

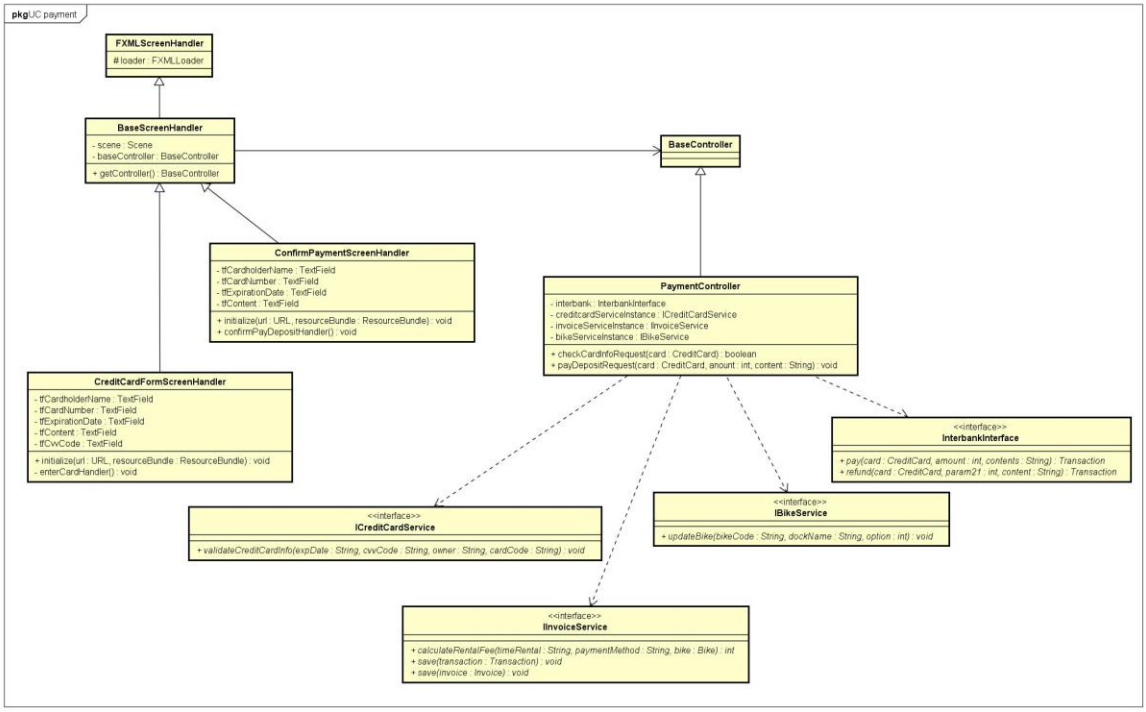
5.3.2.2 Class Diagram for Pakage controllers



5.3.2.3 Class Diagram for Usecase Search dock



5.3.2.4 Class diagram for use case Payment



6 Design Considerations

6.1 Goals and Guidelines

NONE

6.2 Architectural Strategies

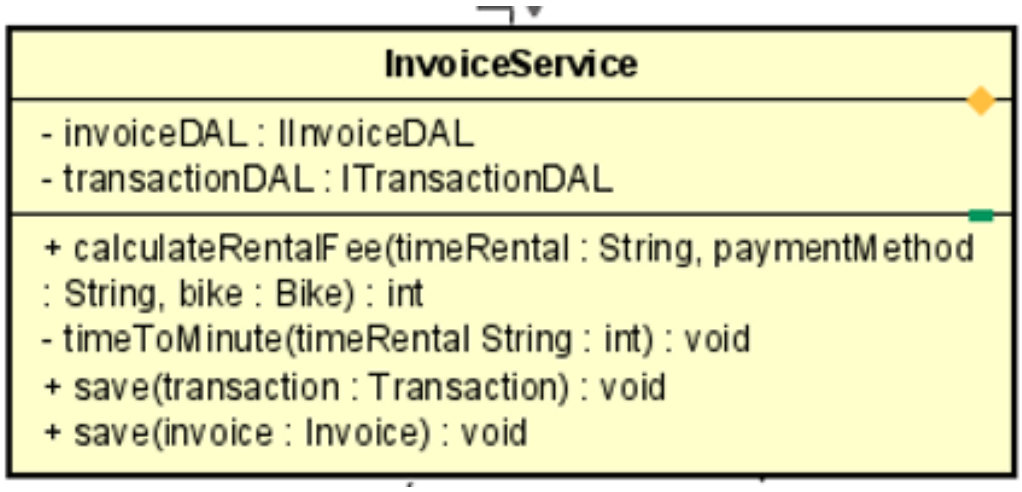
6.3 Coupling and Cohesion

Coupling

- Each layer communicates with each other through interface. This approach promotes loose coupling, which signifies that the interdependencies between different layers are minimized.
- With respect to the scope of package (if we consider each package is a module), our code organization leads to a tight coupling between these components

Cohesion

- Class "InvoiceService" has the Coincidental Cohesion: Methods were put into a class because they are related to the process of creating invoice.



6.4 Design Principles

- Single Responsibility Principle: InvoiceService class handles both the work of calculating fee and save data to database so it violates the design principles.
- Open/Closed Principle (OCP): Entity Bike and Payment can be extended without modifying the whole classes.
- Liskov Substitution Principle (LSP): class con cuar bike thay thees cho lop cha

- Interface Segregation Principle (ISP): Communication through interface between each layer
- Dependency Inversion Principle (DIP)

7 Task Assignment

Phase	Task	Search dock	Rent Bike	Pay deposit	Return Bike	Review/ Refine
1.Requirement Analysis	General Usecase Diagram	Nhi	Ngọc	Phương	Phú	All team
	Usecase specification	Nhi	Ngọc	Nhi, Phương	Phú	Nhi
2.Architectual Design						
	Architectual Patterns					Nhi
	Interaction Diagram	Nhi	Ngọc	Nhi, Phương	Phú	Nhi
	Analysis class diagram	Nhi	Ngọc	Nhi	Phú	Nhi
Detail Design						
1.Interface Design	Screen Transition diagram	Nhi	Ngọc	Nhi	Phú	Nhi
	Screen Specification	x	x	x	x	Ngọc
2.Data Modeling						
	Conceptual Data model	x	x	x	x	All team
	SQL Script					Phương
3. Class design						
	Class diagram	Nhi				Nhi
Programming						
	Code	Nhi	Ngọc	Nhi	Phú	Nhi
	Unit Test				Phú	

8 References

1. ISD.VN.20211-Group2

What we have reused/ modified	How we implemented
1. Package “View”	We reused the structure for handling input data and connecting to the .fxml file. We have designed user interface based on the origin flow of transition diagram.
2. Package “Utils”	We reused the code of API class (for creating http request to call the external Interbank API), Utils.class (for handling unit measurement used in the Interface)