Hanoi University of Science and Technology
School of Information and Communication Technology

EcoBike Application

Subject: ITSS Software Development

Supervisor: Asst. Prof. Nguyen Thi Thu Trang

Group number: 4

Group participants

Name	Student ID
Vu Thanh Hai	20194756
Hoang Thi Hang	20194757
Nguyen Thu Hieu	20194761

Table of Contents

List of Figures

List of Tables

1. Introduction

1.1 Objective

This document is to present a detailed description of the EcoBike system. 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. This document is intended for both the stakeholders and the developers of the system.

1.2 Scope

This EcoBike system is designed for EcoPark, which has an hourly bike rental service with docking stations for customers to rent or return bikes automatically. This system provides convenience for customers to finding a dock of bike, signing up for renting a bike, paying renting fees and returning a bike. EcoBike also assists administrators in monitoring bikes in the system, which includes adding, viewing, updating and removing bike to/from the system.

In details, when the customer starts EcoBike, the application displays the map of the region around the current location of him. The customers can then select or search for a bike station to view its information, including the distance and estimated walking time to the selected dock. At the same time, EcoBike provides a list of available bikes at the dock at the meantime. Customer can also select a bike in order to get its information and current states before deciding to rent it.

To rent a bike, customer uses the application to scan the barcode on the lock and open the bike. Details about the selected bike is shown, and if he wants to rent it, the application calculates the deposit, and transactions is performed by using credit cards connected to an interbank. When being in rent, the state of the bike is always updated so that customers can have a better look on his current renting bike.

To return a bike, customer firmly pushes the bike to an empty dock point and closes the lock. EcoBike then calculates the total charges corresponding to renting time. At the same time, the system saves the rental transaction and sends an email of transactions to customers.

The interbank, which will be in connection with EcoBike system, will receive payment request from EcoBike after the customer has confirmed the rental. It will help the customer to pay deposit before renting a bike, and finish the fees after he is done with renting the bike. Also, if the customer rents a bike for less than a predefined time period, he will have his deposit returned. This return will also be issued by EcoBike and be performed by the interbank.

For the administrator side, he can manipulate the list of bikes in system for rental at any time. To add a new bike, he provides information for the application so that EcoBike can validates it and create a new bike profile for tracking its states. Administrator can choose to view lists of bikes, as well as detail information of these bike. Also, while being in the monitor screen, he can choose to update bike information or delete it from the list if it is not being used in the future anymore.

1.3 Glossary

Table 1: Terms used in the document

Term	Definition
Bike	The transportation means to be rent in this application system
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 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 total time 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.
Transaction	The action of paying for bike deposit, bike rental or returning deposit

2. Overall description

2.1 Actors

2.1.1 Customer

The customer is the actor who interacts directly with the system for renting a bike. The activities of the customers with EcoBike system includes viewing and selecting dock, viewing bike information, renting, returning bike and performing payment.

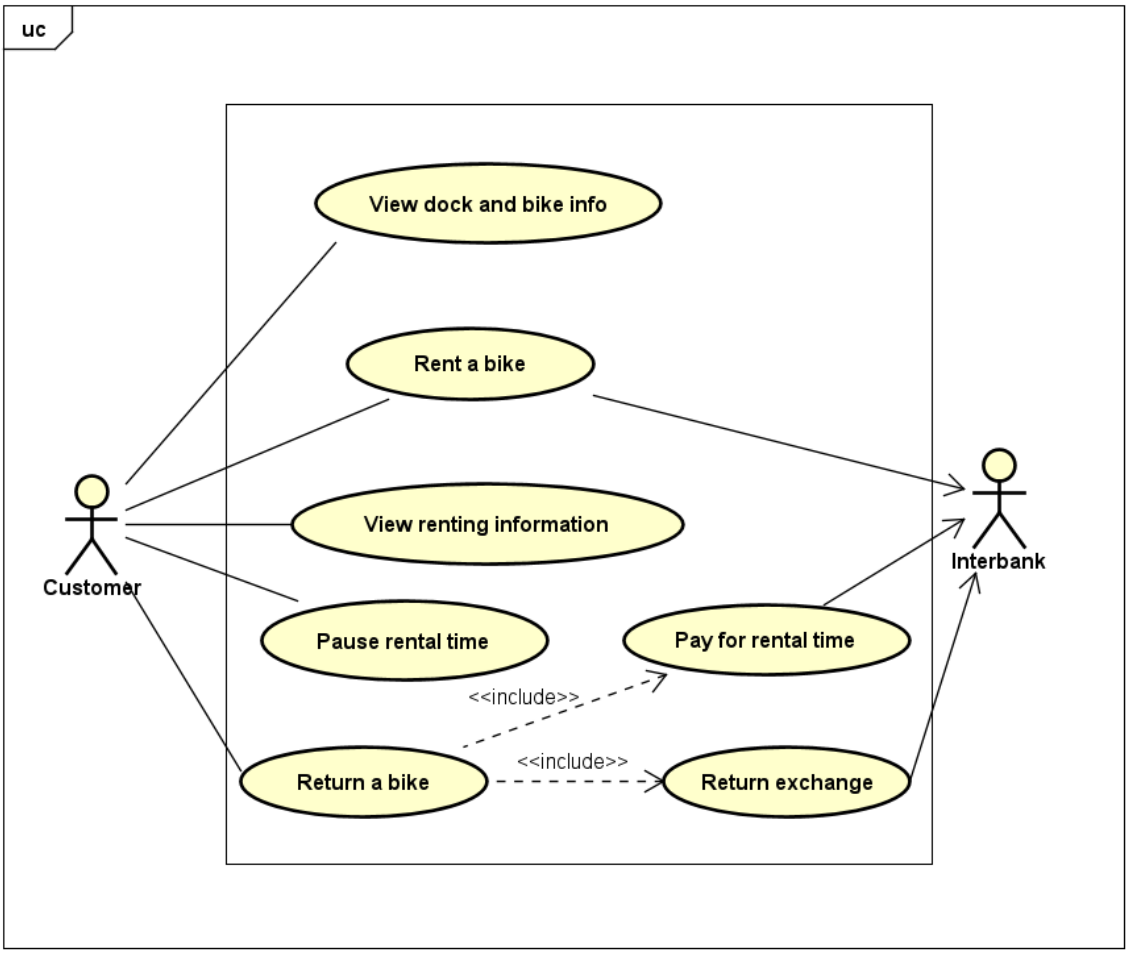
2.1.2 Interbank

The interbank is the actor who interacts with the system for the purpose of performing transactions issued by the customers. The activities includes paying deposit, paying rental fees and returning deposit

2.2 Usecase diagram

The below diagrams illustrate the overall usecase of the actors on the EcoBike system

Figure 1: *Usecase diagram of EcoBike system*



3. Detailed Design

3.1 Usecase specifications for "View parking dock and vehicle information"

3.1.1 Usecase Code: UC-001

3.1.2 Brief description

This use case describes the interaction between user and EcoBikeRental app when userwants to view information of bike and parking dock.

3.1.3 Actors

- User

3.1.4 Preconditions

- User succesfully logged in.

3.1.5 Basic flow of events

Step 1: User view dock marker

Step 2: EcoBikeRental app shows information of the dock and available bikes

Step 3: The user choose a bike

Step 4: EcoBikeRental app displays detail information about the bike

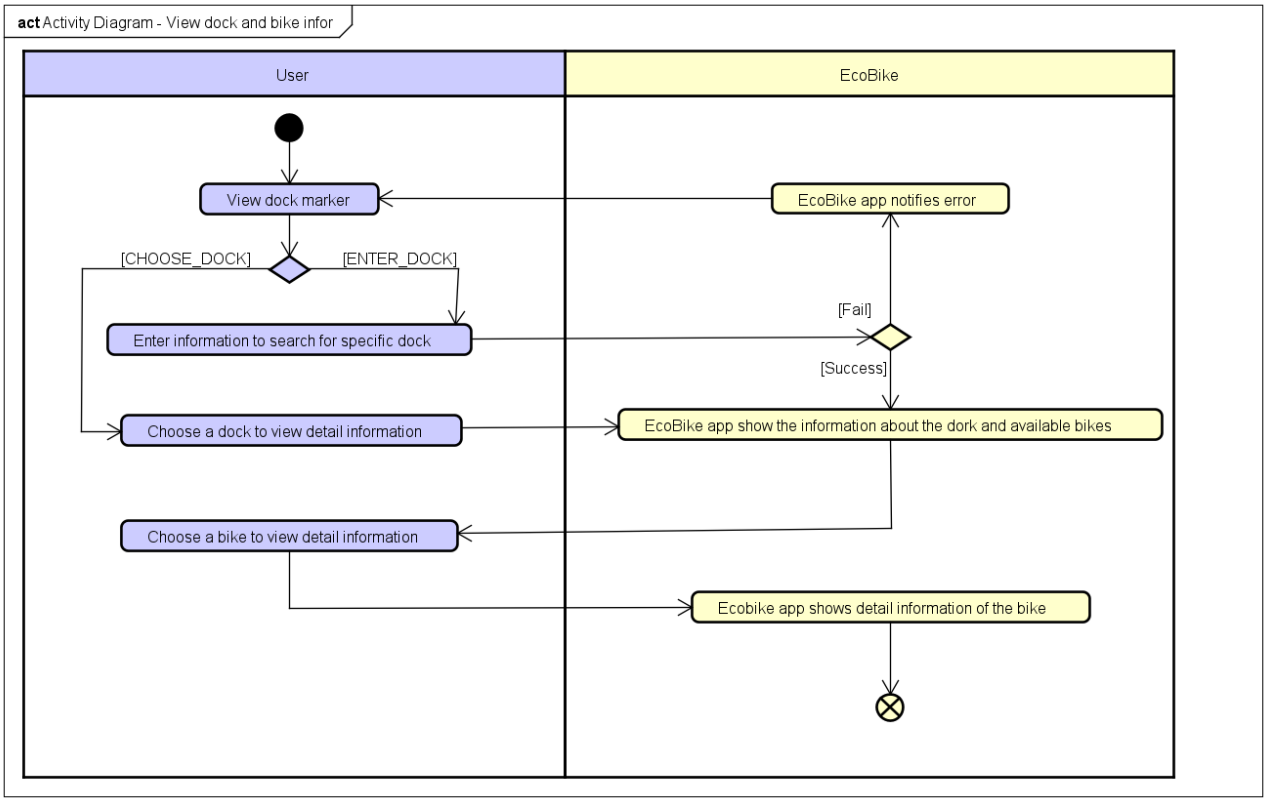
3.1.6 Alternative flows of events

Table 2: *Alternative flow of usecase*

No	Location	Conditions	Actions	Resume location
1	At step 2	User searches for a specific dock	User search for a dock by name or address	
1a		If no dock matches	EcoBikeRental app notifies error	At step 2
1b		If there is a dock meets the given infor	EcoBikeRental app fetches the matching data and display result	At step 3

3.1.7 Activity diagram

Figure 2: *Activity diagram of "View bike information" usecase*



3.1.8 Input data

None

3.1.9 Output data

Table 3: *Output data for dock marker and available bikes information*

No	Data fields	Description	Display format	Example
1	Name	Name of dock marker	- String	Bach Khoa
2	Address	Address of dock marker	- String	22 Le Thanh Nghi, Dong Tam precint, Hai Ba Trung district, Ha Noi
3	Area	Parking area	- Float - Along with m2	500 m2
4	Number of available bikes		- Integer	100
5	Number of empty docking points		- Integer	4

6	Distance	Distance from user's location to this dock	- Float - Along with m	50 m
7	Time to dock	Walking time from user's location to this dock	- Integer - Along with minutes	15 minutes

Table 4: *Output data for detail information of bike*

No	Data fields	Description	Display format	Example
1	Name		- String - All caps	VINFAST
2	Type of bike		- String	Twin bike
3	Image		- Image	
4	License plate		- String	29-MĐ1000.20
5	Status		- String	Free
6	Location	Name of dock marker	- String	Bach Khoa
7	Current battery	Electric bike	- Percentage - Integer	80%
8	Distance estimated	Distance bike can go with current battery	- Integer - Along with km	1 km
9	Rental price	Price to rent the bike	- Float - Dot as decimal separator - 2 digits after decimal separator - Along with currency	12.00 VND
10	Deposit	The deposit customer have to pay to rent the bike	- Float - Dot as decimal separator	700000.00 VND

			- 2 digits after decimal separator - Along with currency	

3.1.10 Postconditions

None

3.1.10 Interaction Diagram

Figure 3.1: *Sequence diagram of "View bike information" usecase*

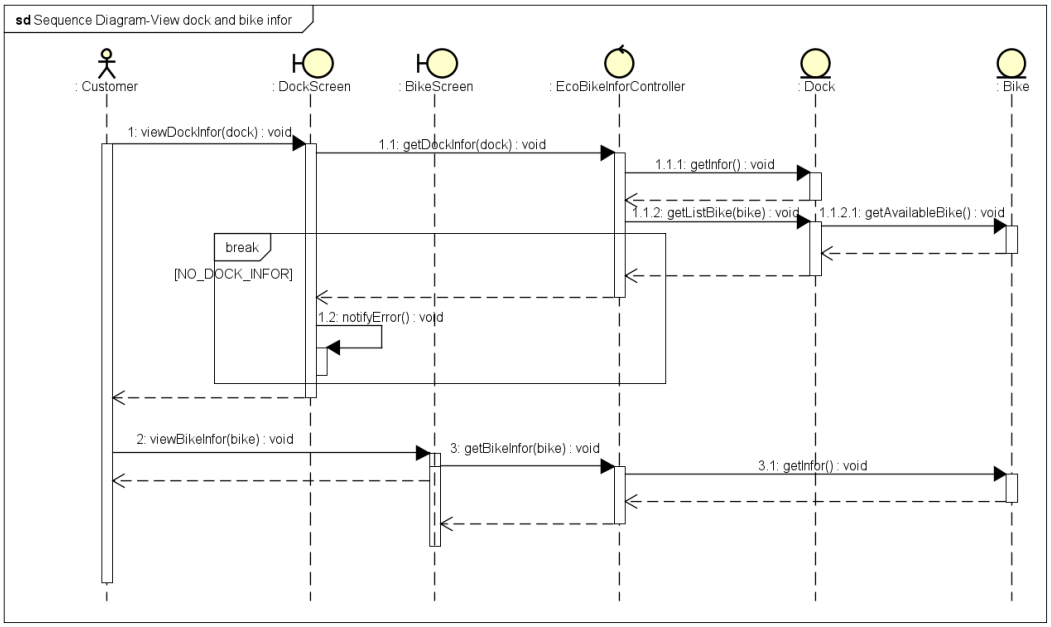
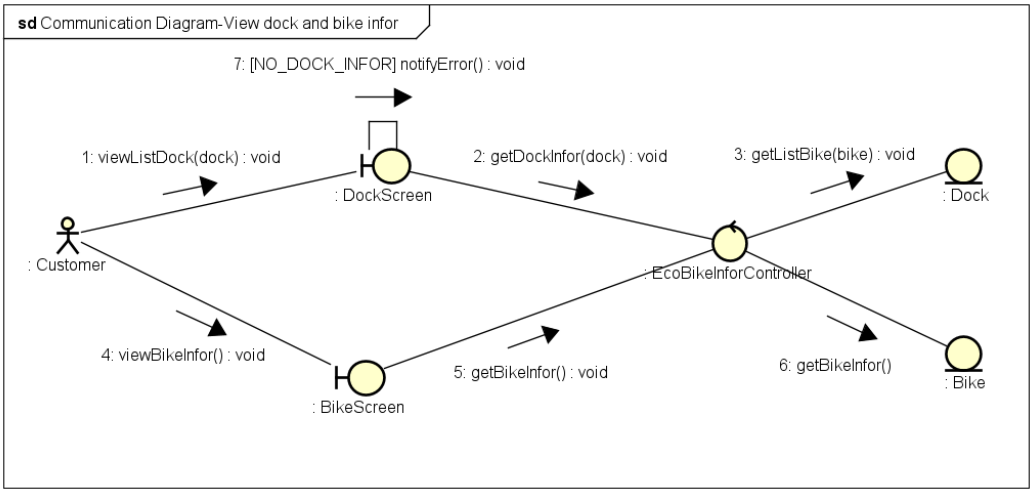


Figure 3.2: *Communication diagram of "View bike information" usecase*



3.2 Usecase specifications for “Rent a bike”

3.2.1 Use case code: UC-002

3.2.2 Brief description

Customer rents a bike via Eco App.

3.2.3 Actor

- Customer

3.2.4 Preconditions

- There is an active network connection to the internet
- User login successfully to app
- User find a dock on site and there is still available bikes for renting

3.2.5 Basic Flow of Event

Step 1: Customer scans the barcode on the lock via app

Step 2: The software shows the information of that bike

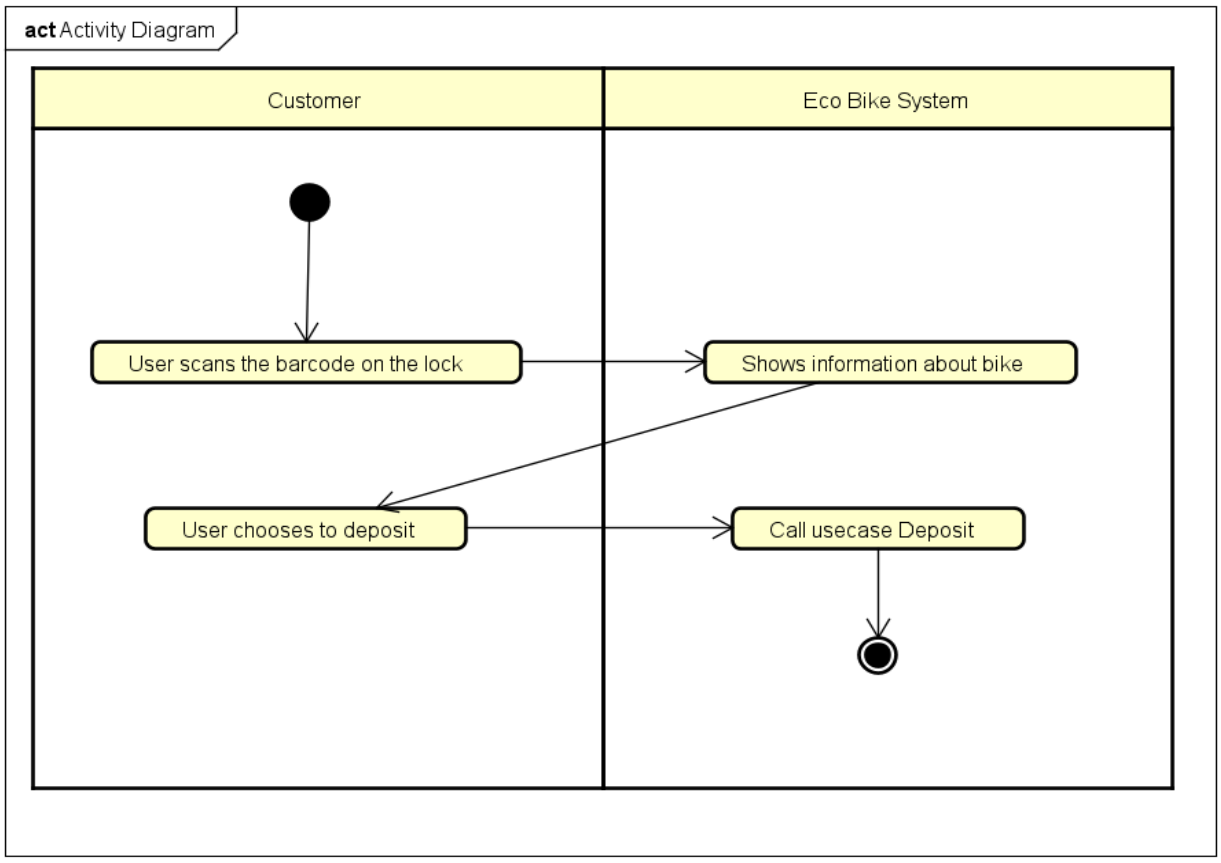
Step 3: Software call the usecase "Deposit"

3.2.6 Alternative flows of events

None

3.2.7 Activity diagram

Figure 4: *Activity diagram of “Rent a bike” usecase*



3.2.8 Input data

Table 5*: Input of scanner to rent bike*

No	Data field	Description	Mandatory	Valid condition	Example
1	Barcode		Yes	Exist barcode	123

3.2.9 Output data

The same as the usecase “Deposit”

3.2.10 Post condition

- None

3.2.11 Interaction Diagram

Figure 5.1: *Sequence diagram of “Rent bike” usecase*

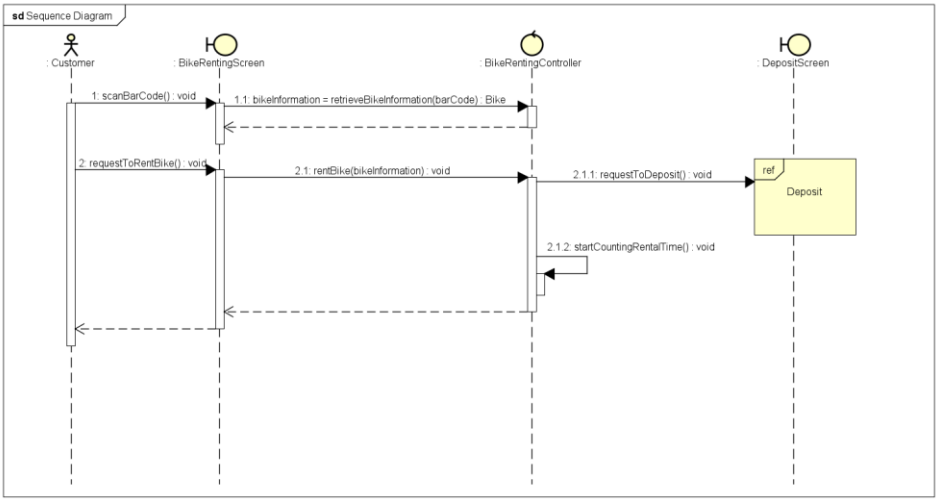
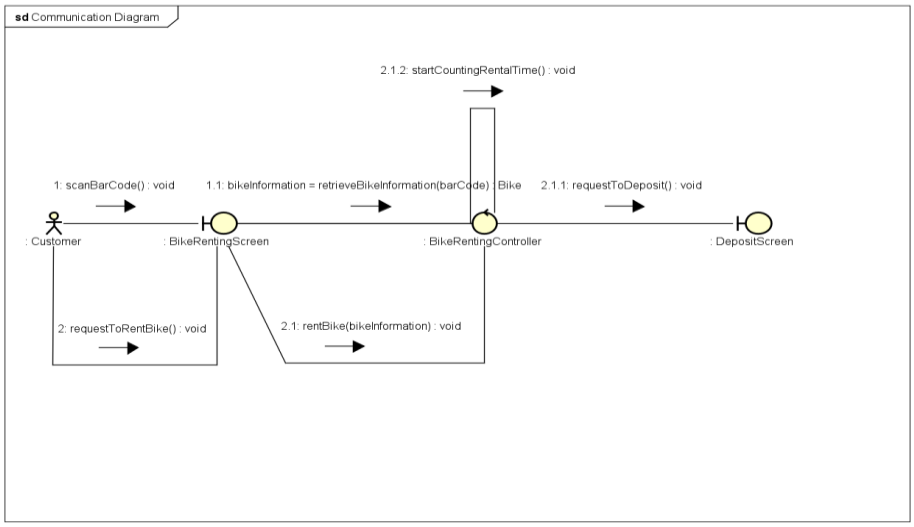


Figure 5.2: *Communcation diagram of “Rent bike” usecase*



3.3 Usecase specifications for “Deposit”

3.3.1 Usecase code: UC-003

3.3.2 Brief description

Customer deposit money in order to rent a bike

3.3.3 Actors

- Customer

3.3.4 Preconditions

- There is an active network connection to the internet
- User scanned the barcode and ready to deposit for that bike

3.3.5 Basic flow of events

Step 1: Customer chooses payment method, default is pay by credit card

Step 2: Customer inputs card information and transaction content

Step 3: Software calculates the deposit fee

Step 4: Customer confirms transaction

Step 5: Software deducts money from customer account

Step 6: Software shows the information of transaction to screen and save that transaction to the system

Step 7: Software sends email informing customer about the status of transaction

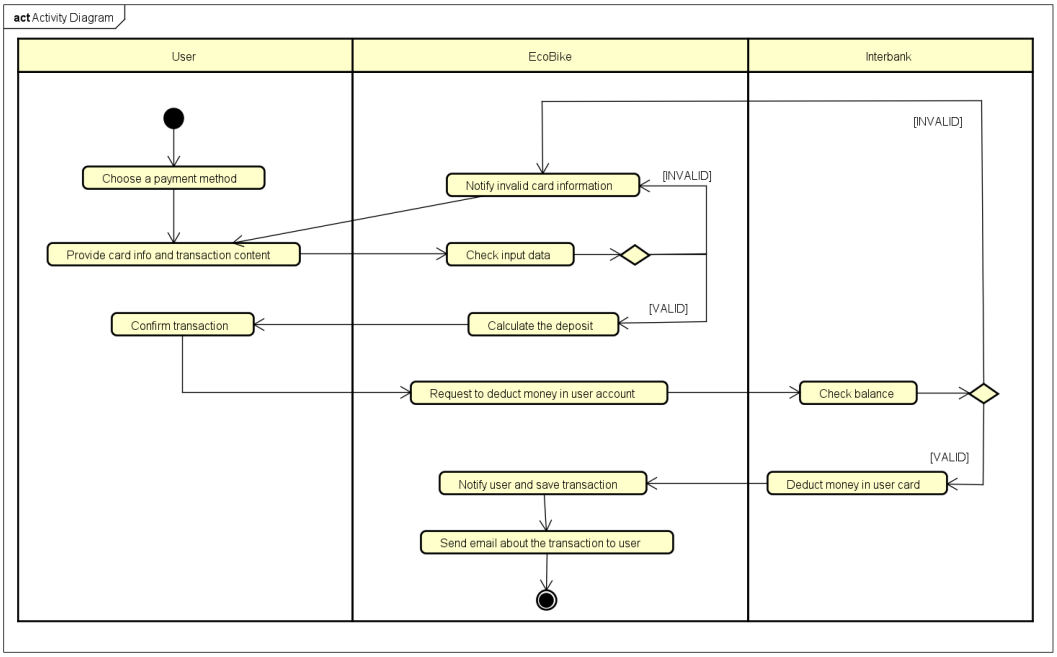
3.3.6 Alternative flow of events

Table 6*: Alternative flow of events input card information*

No	Location	Condition	Action	Resume location
1	Step 2	Invalid card information	Show error	Previous step
2	Step 2	Not enough balance	Notify customer	Previous step

3.3.7 Activity diagram

Figure 6: *Activity diagram of “Deposit” usecase*



3.3.8 Input data

Table 7: *Input data of card and transaction*

No	Data field	Description	Mandatory	Valid condition	Example
1	Card holder name		Yes	String	Nguyen Van Trang
2	Card number		Yes	String	031201002654
3	Issuing bank		Yes	String	Viettinkbank
4	Expiration date		Yes	Date time	15/12/2022
5	Security code		Yes	String	IT1897
6	Transaction content		Yes	Multiline sentence	Pay for e-bike

3.3.9 Output data

Table 8: *Output to screen after depositing successfully*

No	Data field	Description	Display format	Example

1	Card holder name		String	Nguyen Van Trang
2	Card number		String	031201002654
3	Transaction content		Multiline sentence	Pay for e-bike
4	Amount to be deposited		Positive integer Commas for thousand separators	100,000 VNĐ
5	Bike code		String	123
6	Bike type		One among: E-bike, Normal bike, Twin bike, Twin E-bike	E-bike
7	Current battery		Positive integer	10%

3.3.10 Post conditions

Transaction is saved to the system and the logs update accordingly.

3.3.11 Interaction Diagram

***Figure 7.1**: Sequence diagram of "Deposit" usecase*

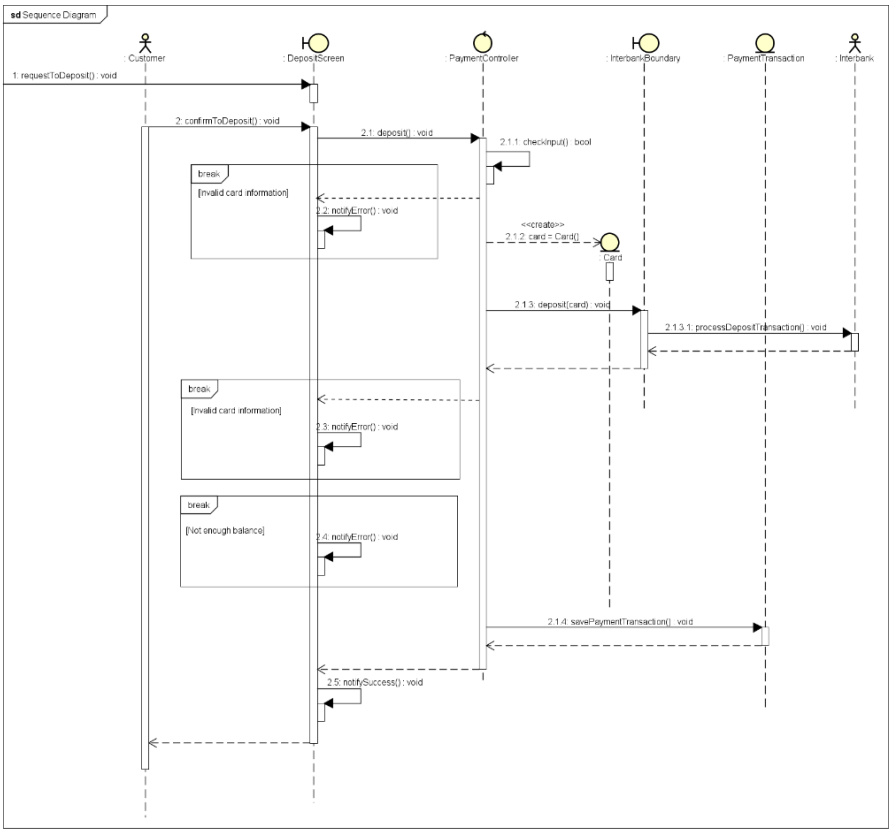
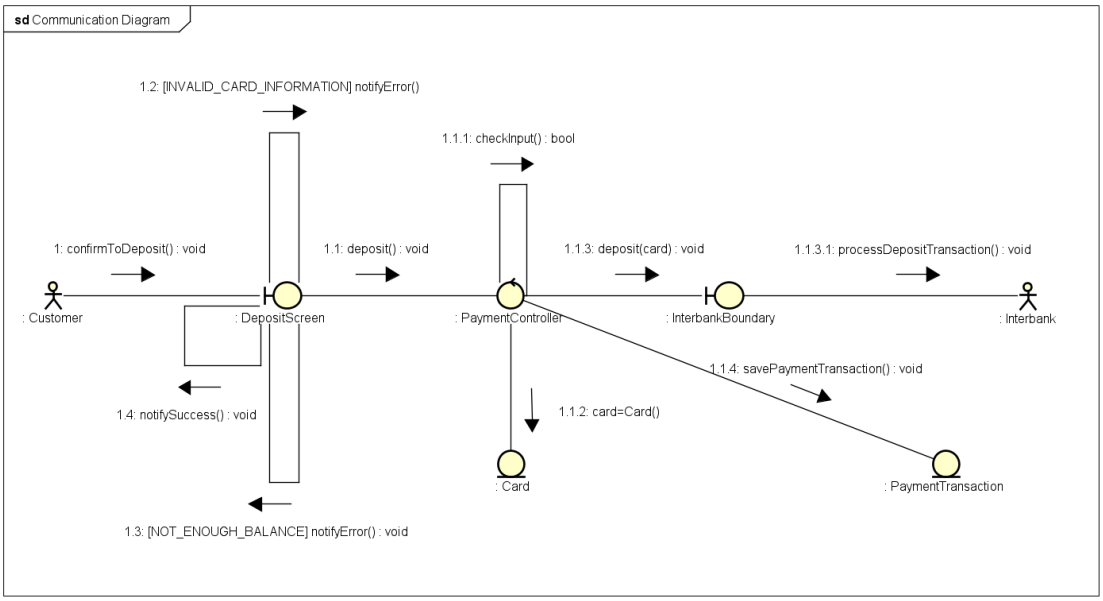


***Figure 7.2**: Communication diagram of "Deposit" usecase*



3.4 Usecase specifications for "View renting information"

3.4.1 Usecase code: UC-004

3.4.2 Brief description

User view information while renting

3.4.3 Actors

- Customer

3.4.4 Preconditions

- There is an active network connection to the internet

- User rented a bike and is using it

3.4.5 Basic flow of events

Step 1: Customer requests to check the information of the bike.

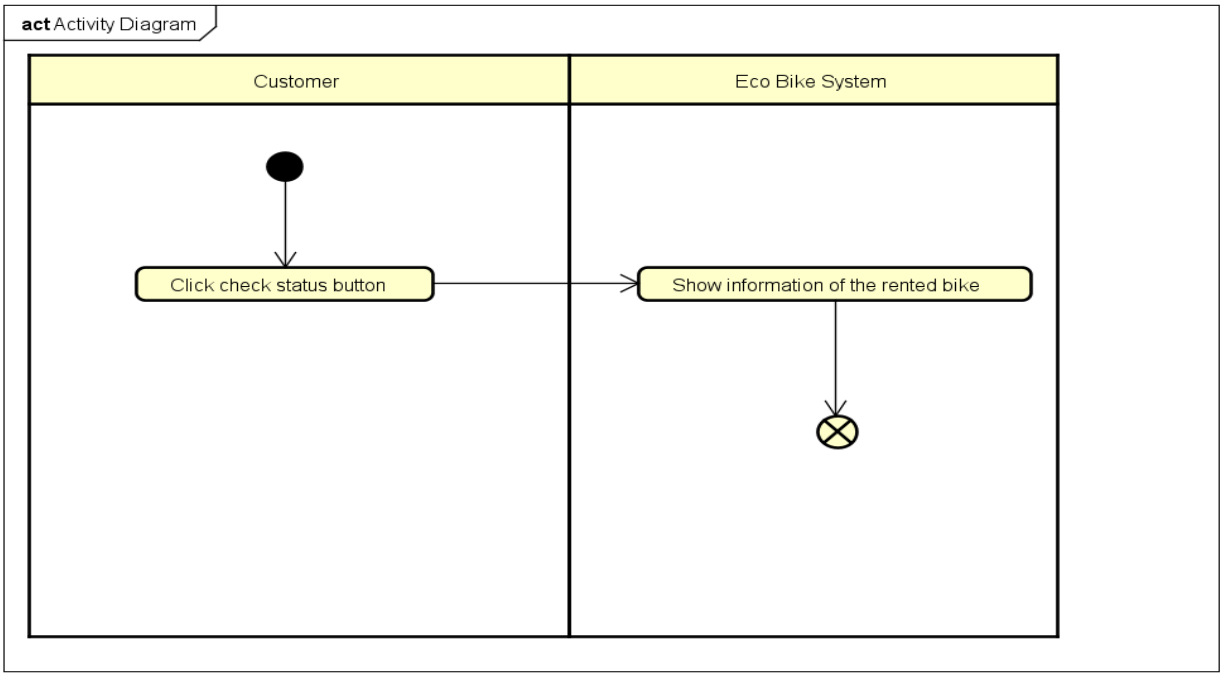
Step 2: The software shows the information of that bike, including: bike type, time of renting, amount to be paid, current battery

3.4.6 Alternative flow

None

3.4.7 Activity diagram

Figure 8: *Activity diagram of "View renting information" usecase*



3.4.8 Input data

Customer clicks the "Check Bike Status" button.

3.4.9 Output data

Table 9: *Output data of bike*

No	Data field	Description	Display format	Example
1	Bike code	ID of bike	String	123
2	Bike type	Type of bike	One among: E-bike, Normal bike, Twin bike, Twin E-bike	E-bike
3	Time of renting		Positive integer	2h
4	Amount to be paid	Depend on time of renting	Positive integer Commas for thousand separators	100,000 VNĐ
5	Current battery		Positive integer	10%

3.4.10 Postconditions

None

3.4.11 Interaction Diagram

***Figure 9.1**: Sequence diagram of "View renting information" usecase*

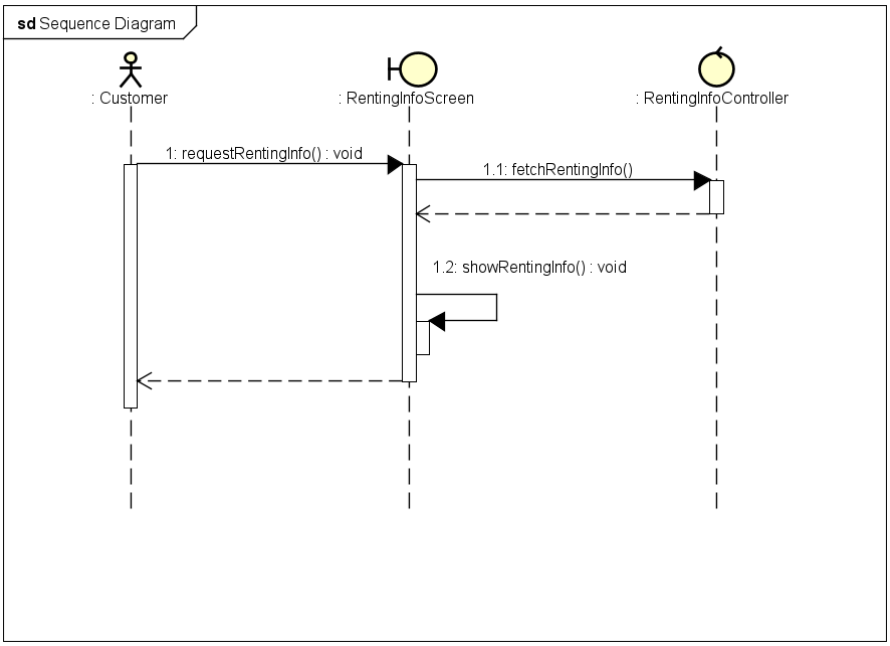
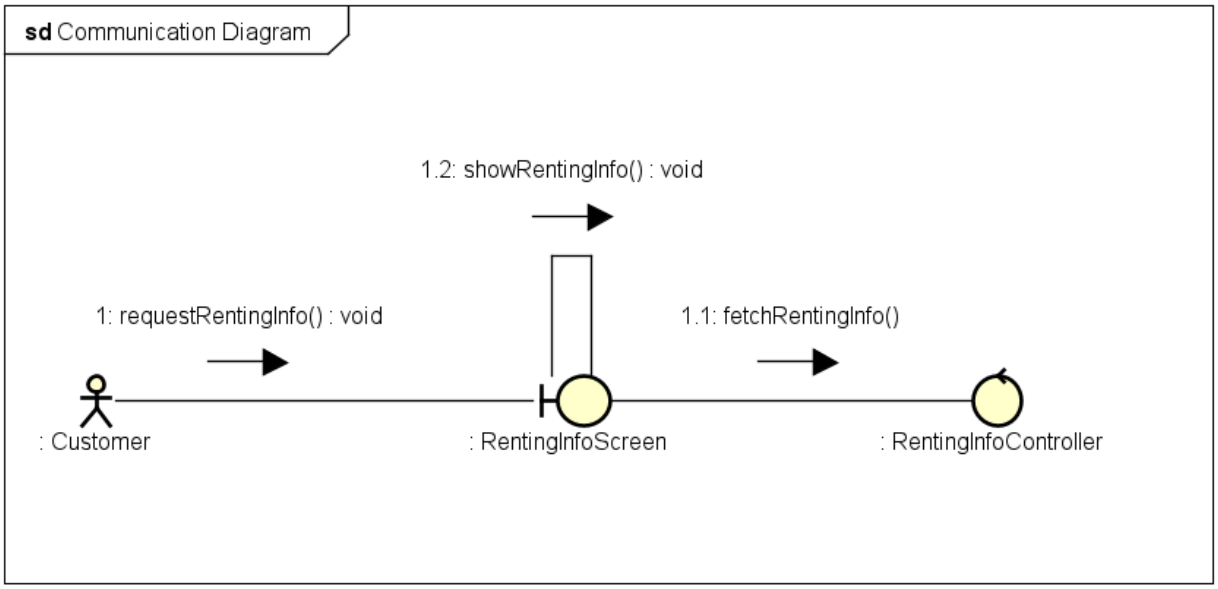


***Figure 9.2**: Communication diagram of "View renting information" usecase*



3.5 Usecase specifications for "Return A Bike"

3.5.1 Usecase code: UC-005

3.5.2 Brief description

This use case describes the interaction between the EcoBike software with the customer and the Interbank when the customer wants to return the bike they rented.

3.5.3 Actors

- Customer
- Eco-Bike software
- Interbank

3.5.4 Preconditions

- There is Internet connection
- The user enters the app

3.5.5 Basic flow of events

Step 1: The customer requests to return the bike
Step 2: The software displays the list of docks to choose
Step 3: The customer chooses a dock to return bike
Step 4: The software calls use case "**Return exchange**"
Step 5: The software calls use case "**Pay for rental time**"

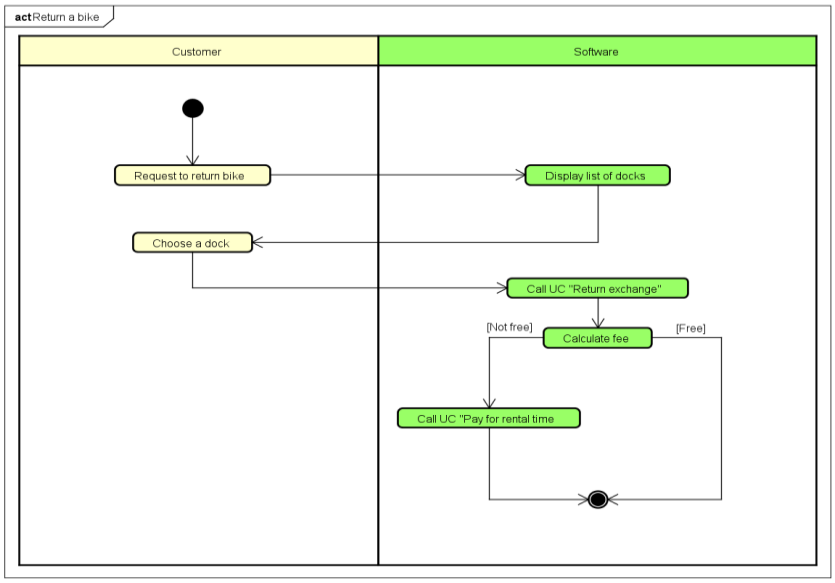
3.5.6 Alternative flow

Table 10: *Alternative flows of events for "Return a bike"*

No	Location	Condition	Action	Resume location
1	At Step 4	If the customer rent for less than 10 minutes	The software notifies and ends the use case	

3.5.7 Activity diagram

Figure 10: *Activity diagram of "Return a bike" usecase*



3.5.8 Input data

None

3.5.9 Output data

None

3.5.10 Postconditions

None

3.5.11 Interaction Diagram

Figure 11.1: *Sequence diagram of "Return a bike" usecase*

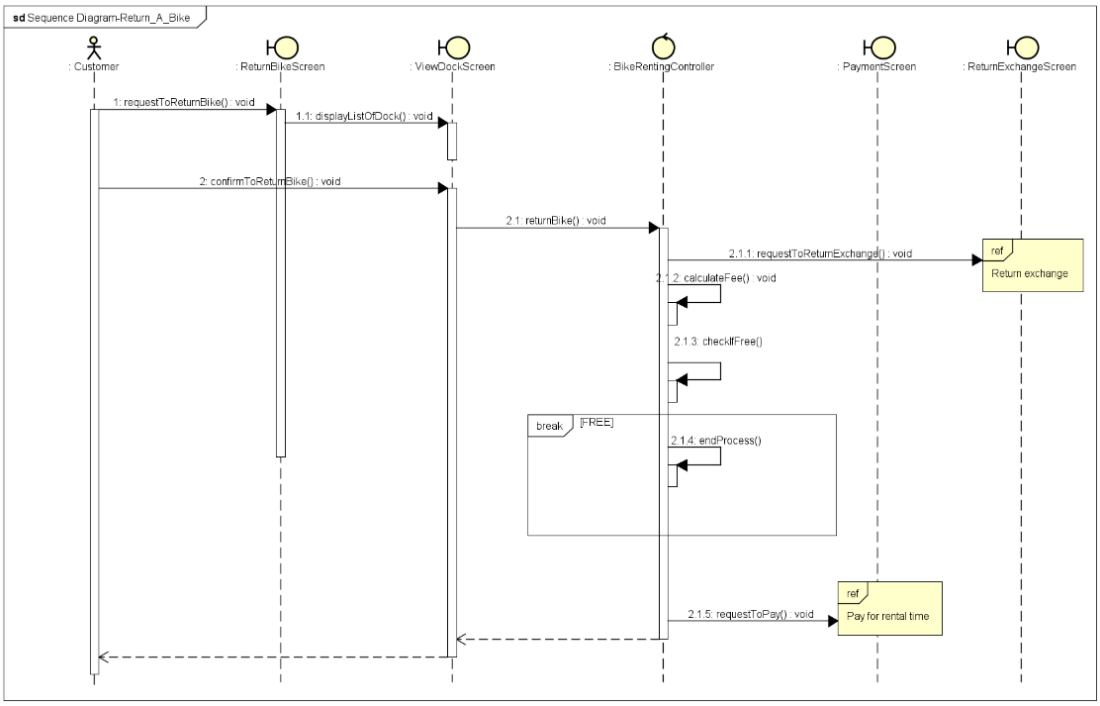
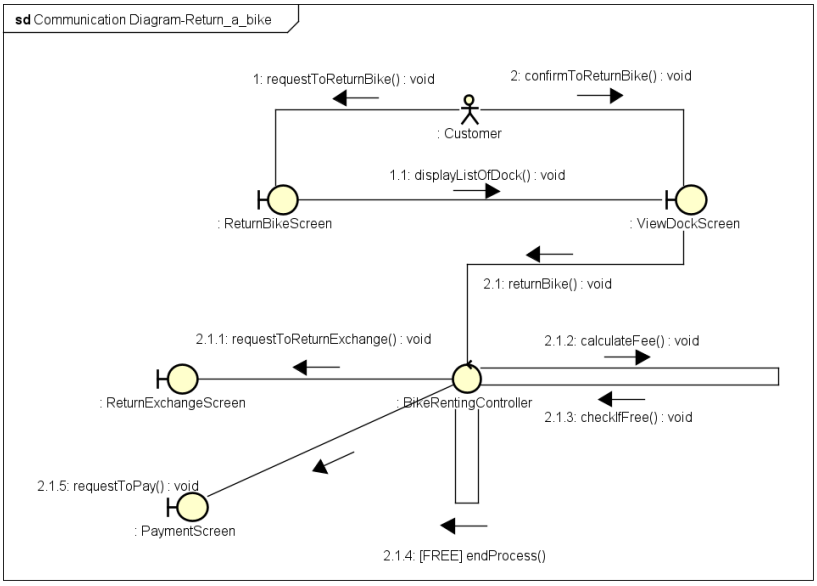


Figure 11.2: *Communication diagram of "Return a bike" usecase*



3.6 Usecase specifications for "Return exchange"

3.6.1 Usecase code: UC-006

3.6.2 Brief description

This use case describes the interaction between the EcoBike software and the Interbank when software returns exchange

3.6.3 Actors

- Interbank

- Eco-Bike software

3.6.4 Preconditions

- There is internet connection from EcoBike system to the interbank

- The user confirms to return bike successfully

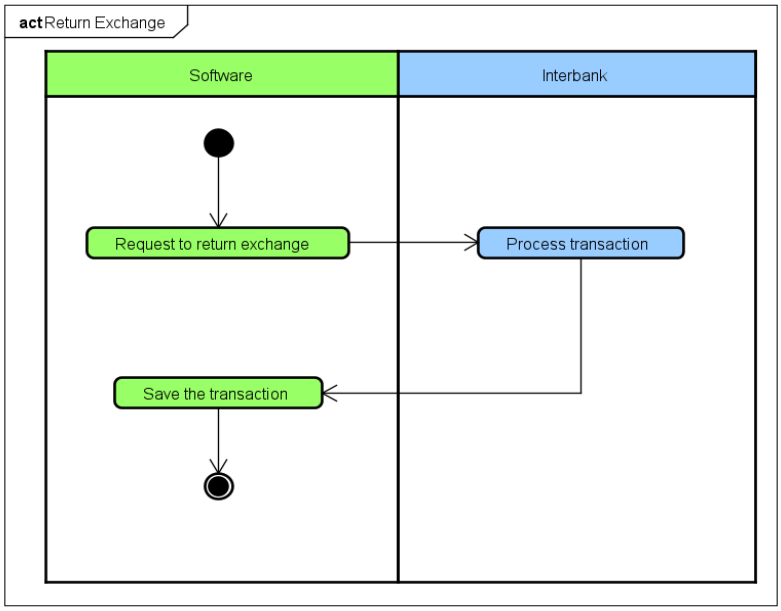
3.6.5 Basic flow of events

Step 1: The software asks the bank to return exchange
Step 2: The interbank processes the transaction
Step 3: The software saves the transaction

3.6.6 Alternative flow

3.6.7 Activity diagram

Figure 12: *Activity diagram of "Return exchange" usecase*



3.6.8 Input data

None

3.6.9 Output data

None

3.6.10 Postconditions

None

3.6.11 Interaction Diagram

Figure 13.1: *Sequence diagram of "Return exchange" usecase*

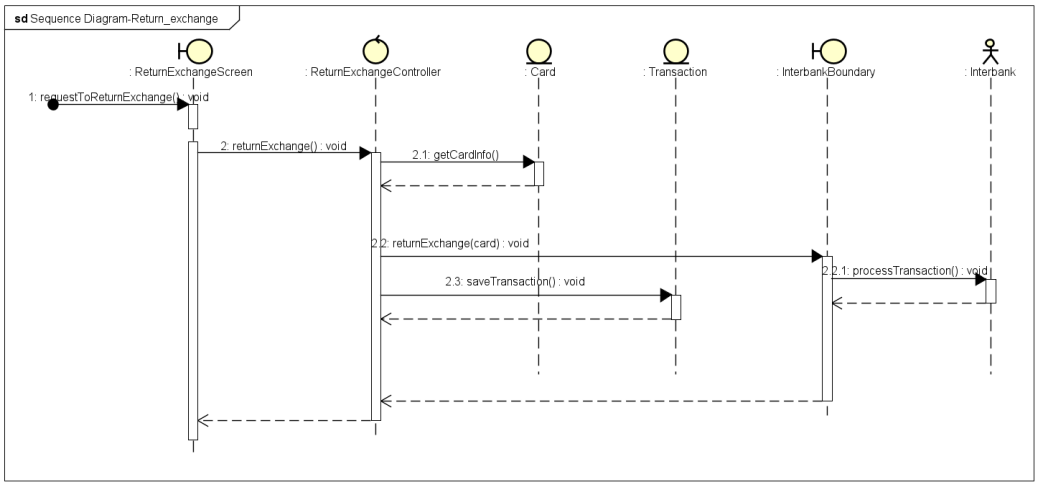
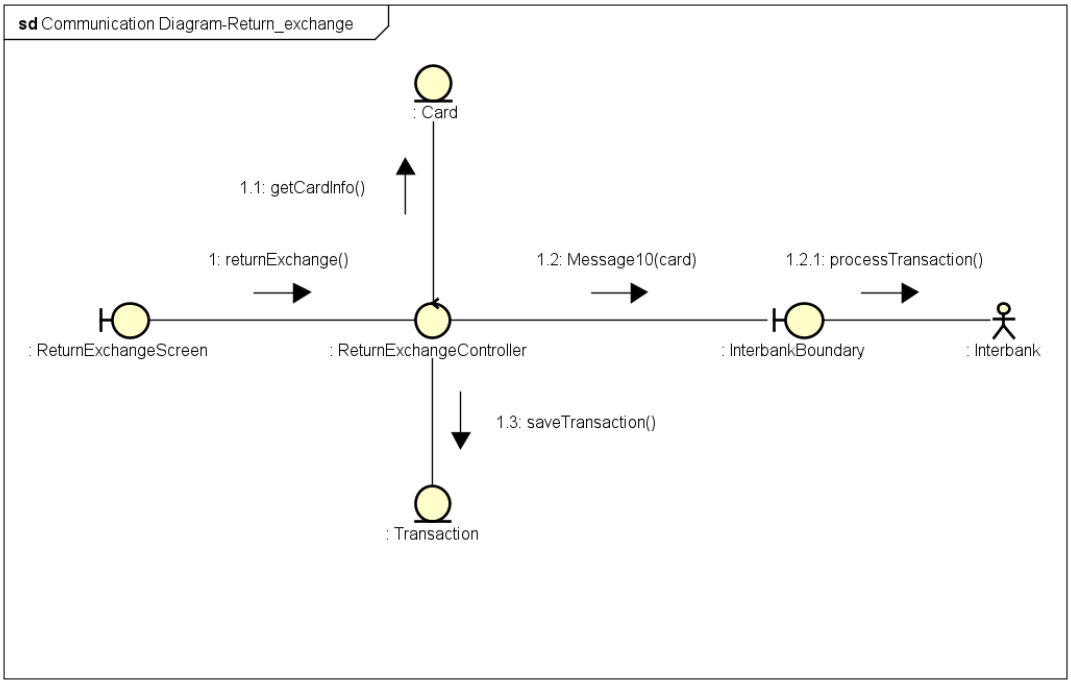


Figure 13.2: *Communication diagram of "Return exchange" usecase*



3.7 Usecase specifications for "Pay for Rental Time"

3.7.1 Usecase code: UC-007

3.7.2 Brief description

This use case describes the interaction between the EcoBike software with the customer and the Interbank when the customer pays for the rental time

3.7.3 Actors

- Customer

- Eco-Bike software

- Interbank

3.7.4 Preconditions

- There is Internet connection
- The software has returned exchange successfully

3.7.5 Basic flow of events

Step 1: The software calculates the rental fee
Step 2: The software asks the bank to pay the fee
Step 3: The interbank processes the transaction
Step 4: The software saves the transaction info
Step 5: The software displays the successful transaction notification
Step 6: The software sends an emails of transaction info to the customer

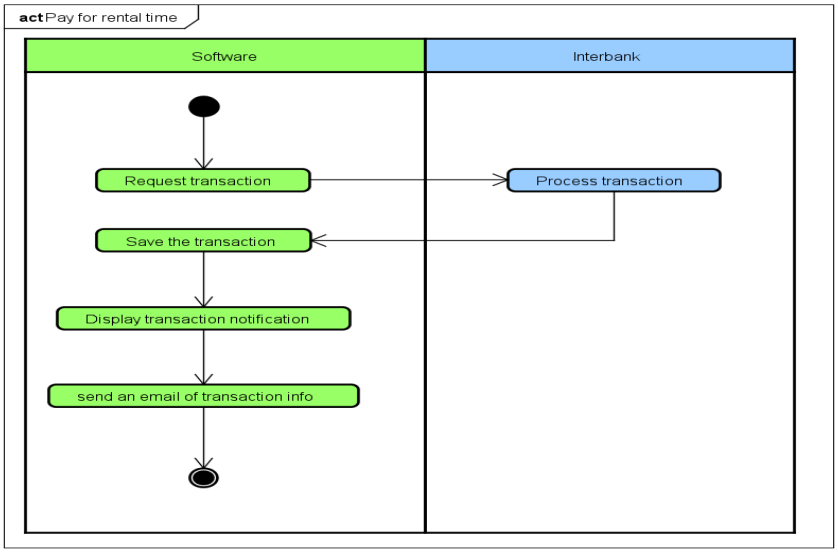
3.7.6 Alternative flow

Table 11: *Alternative flows of events for "Pay for Rental Bike"*

No	Location	Condition	Action	Resume location
1	At step 3	The balance is not enough	The software notifies that the balance is not enough	Step 2

3.7.7 Activity diagram

Figure 14: *Activity diagram of "Pay for rental time" usecase*



3.7.8 Input data

Table 12: *Input cart infomation*

No	Data fields	Description	Display format	Example
1	Card Holder Name	Name of Card Owner	-Text	VU THANH HAI
2	Card Number		-Text	987152_group06_2020
3	Issuing Bank		-Text	VietTTinBank
4	Expiration date		-Date	23/12/2026
5	Security Code		-Text	123456
6	Transaction description		-Text	

3.7.9 Output data

Table 13: *Transaction output info*

No	Data fields	Description	Display format	Example
1	Invoice ID		- Number	1
2	Time issued		-Time	5:30 16-12-2022
3	Rent ID		- Number	2
4	Bike Name		-Text	Ebike-01
5	Bike Type		-Text	EBike
6	Total rent time		- Number	2
7	Exchange ID		- Number	3
8	Exchange amount		- Float - Dot as decimal separator - 2 digits after decimal separator - along with currency	20000.00 VND
9	Rental Payment ID		- Number	4
10	Rental Payment Amount		- Float - Dot as decimal separator - 2 digits after decimal separator - along with currency	10000.00 VND
11	Total	Total amount of money corresponding the rental time interval	- Float - Dot as decimal separator - 2 digits after decimal separator - along with currency	20000.00 VND

3.7.10 Postconditions

None

3.7.11 Interaction Diagram

Figure 15.1: *Sequence diagram of "Pay for rental time" usecase*

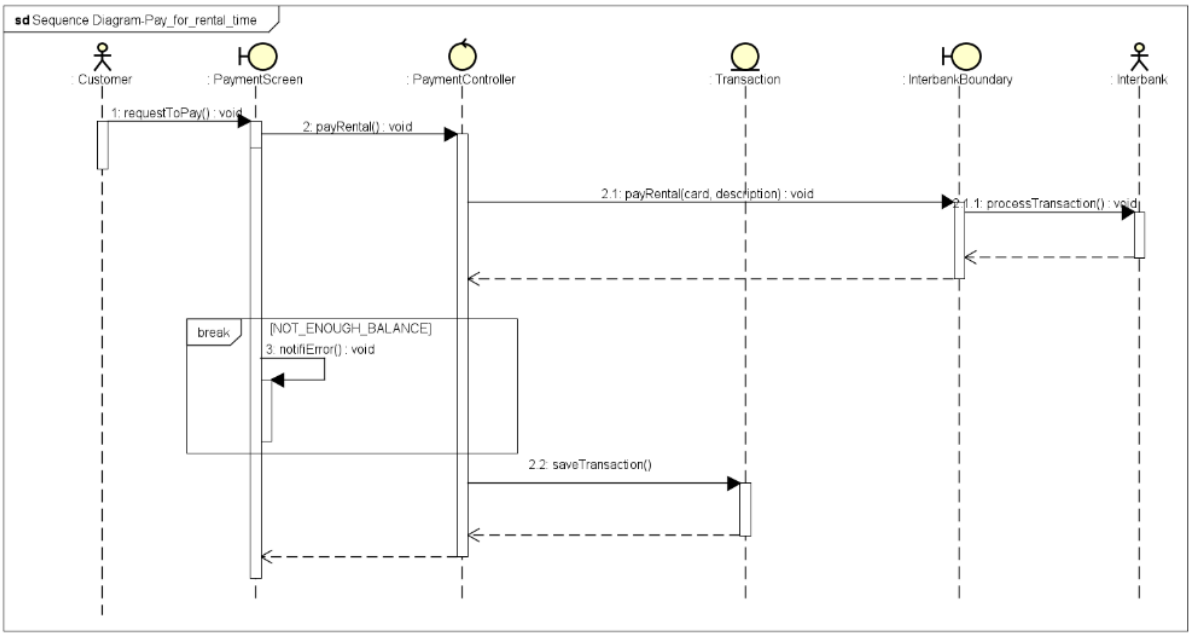
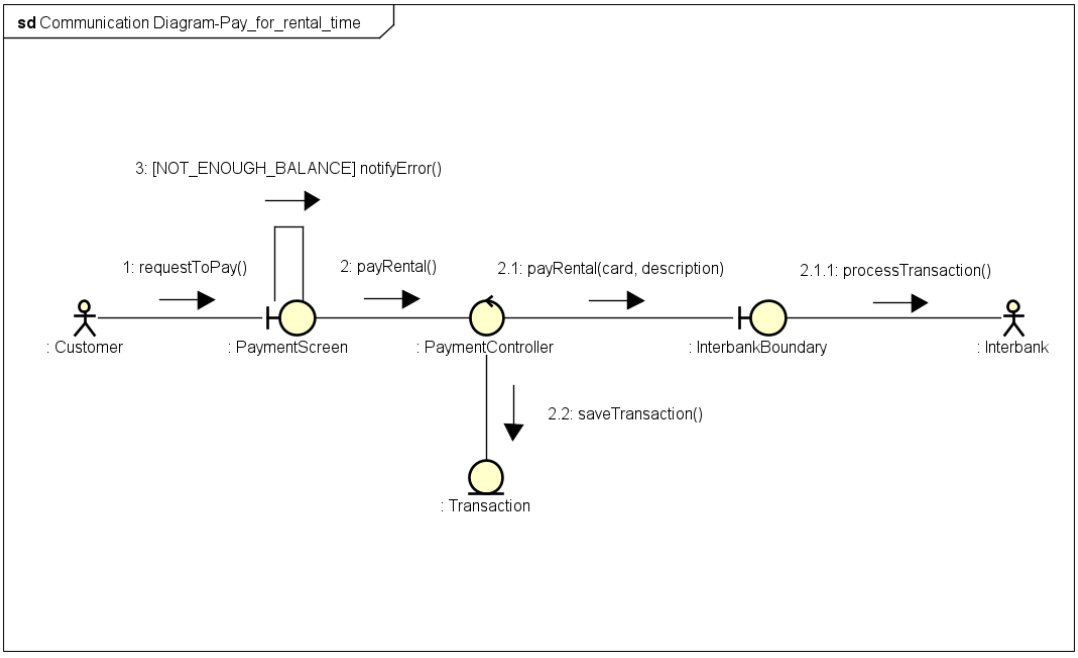


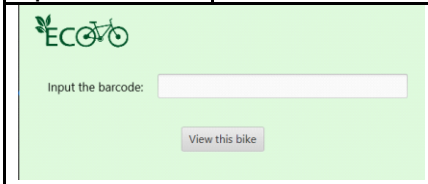
Figure 15.2: *Communication diagram of "Pay for rental time" usecase*



4. Interface Design

4.1 Scan barcode screen

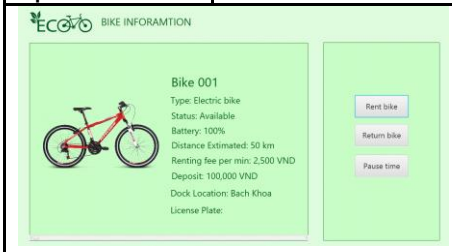
Table 14: *Scan barcode screen specification*

Ecobike Software		**Date of creation**	**Approved by**	**Reviewed by**	**Person in charge**
Screen specification	Scan barcode screen	3/1/2023			Nguyễn Thụ Hiếu
		Control	Operation	Function	
		Input scan barcode	Enter Text	Search for bike via barcode	
		View bike button	Click	View the bike attached to this barcode	

Screen name	Scan barcode			
Item name	Number of digits (bytes)	Type	Field attribute	Remarks
Barcode	50	String	Blue	Left-justified

4.2 View Bike Screen

Table 15: *View Bike screen specification*

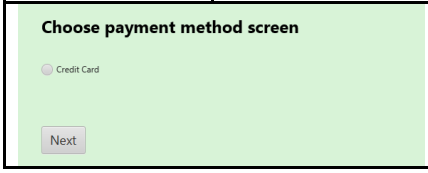
Ecobike Software		**Date of creation**	**Approved by**	**Reviewed by**	**Person in charge**
Screen specification	View Bike Screen	3/1/2023			Nguyễn Thụ Hiếu
		Control	Operation	Function	
		Area for displaying bike info	Initial	Display bike info	
		Area for action buttons	Click	Perform action rent bike, return bike or pause time	

Screen name	View bike screen			
Item name	Number of digits (bytes)	Type	Field attribute	Remarks
Bike name	50	String	Blue	Left-justified
Type	20	Type	Blue	Left justified
Status	20	String	Blue	Left justified
Battery	3	Numeric	Blue	Left justified
Distance Estimated	7	Numeric	Blue	Left justified

Renting fee per min	20	Numeric	Blue	Left justified
Deposit	20	Numeric	Blue	Left justified
Dock Location	100	String	Blue	Left justified
License Plate	100	String	Blue	Left justified

4.3 Choose payment method screen

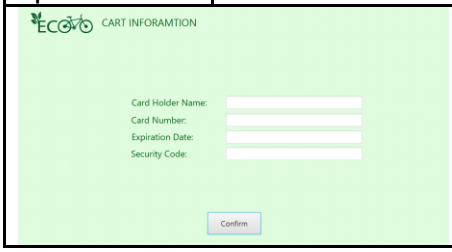
Table 16: *Choose payment screen specification*

Ecobike Software		**Date of creation**	**Approved by**	**Reviewed by**	**Person in charge**
Screen specification	Choose payment method screen	3/1/2023			Nguyễn Thụ Hiếu
		Control	Operation	Function	
		Radio button	Click	Choose payment method	

Screen name	Choose payment method			
Item name	Number of digits (bytes)	Type	Field attribute	Remarks
Payment method	1	bool	Blue	Left-justified

4.4 Cart Info Screen

Table 17: *Cart Info screen specification*

Ecobike Software		**Date of creation**	**Approved by**	**Reviewed by**	**Person in charge**
Screen specification	Cart Info Screen	3/1/2023			Nguyễn Thụ Hiếu
		Control	Operation	Function	
		Area for input card info	Enter text	Input card data	
		Confirm button	Click	Submit card info to system	

Screen name	Cart Info screen			

Item name	Number of digits (bytes)	Type	Field attribute	Remarks
Card holder name	50	String	Blue	Left-justified
Card Number	50	String	Blue	Left justified
Expiration Date	20	Date	Blue	Left justified
Security Code	10	String	Blue	Left justified

4.5 Payment Screen

Table 18: *Payment screen specification*

Ecobike Software		**Date of creation**	**Approved by**	**Reviewed by**	**Person in charge**
Screen specification	Payment Screen	3/1/2023			Nguyễn Thụ Hiếu
		Control	Operation	Function	
		Area for displaying transaction info	Initial	Display transaction info	
		Confirm button	Click	Confirm the transaction	

Screen name	Payment screen			
Item name	**Number of digits (bytes)**	**Type**	**Field attribute**	**Remarks**
Customer name	50	String	Blue	Left-justified
Bike name	20	String	Blue	Left justified
Bike Type	20	Type	Blue	Left justified
Rental Time	2	Numeric	Blue	Left justified
Cost	20	Numeric	Blue	Left justified
Exchange	20	Numeric	Blue	Left justified
Total	20	Numeric	Blue	Left justified

4.6 List dock screen

Table 19: *List Dock screen specification*

EcoBike Software		**Date of creation**	**Approved by**	**Reviewed by**	**Person in charge**
Screen specification	List dock screen	02/01/2023			

	Control	Operation	Function
	Logo	Click	Return to the main screen immediately
	Header	Initial	Display title of screen
	List of dock	Initial	Display list of docks
	Button	Click	Allow customer to return bike and view dock info

Screen	**Normal order invoice & Rush order invoice screen**			
Field name	**Type**	**Limitation**	**Attribute**	**Remarks**
Dock name	Text	100 characters	Bold	Left-justified
Distance	Digits	4 digits	Normal	In terms of kms

4.7 Splash screen

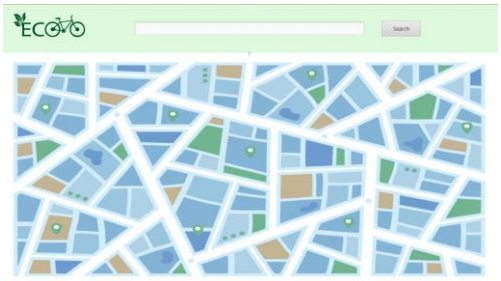
Table 20: *Splash screen specification*

EcoBike Software		**Date of creation**	**Approved by**	**Reviewed by**	**Person in charge**
Screen specification	**Splash screen**	02/01/2023			
		**Control**	**Operation**	**Function**	
		Main area	None	Introduce the application	

4.8 Main screen

Table 21: *Main screen specification*

EcoBike Software		**Date of creation**	**Approved by**	**Reviewed by**	**Person in charge**
Screen specification	**Main screen**	02/01/2023			

	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

4.9 View Dock Screen

Table 22: *View Dock screen specification*

EcoBike Software		Date of creation	Approved by	Reviewed by	Person in charge
Screen specification	**View Dock screen**	02/01/2023			

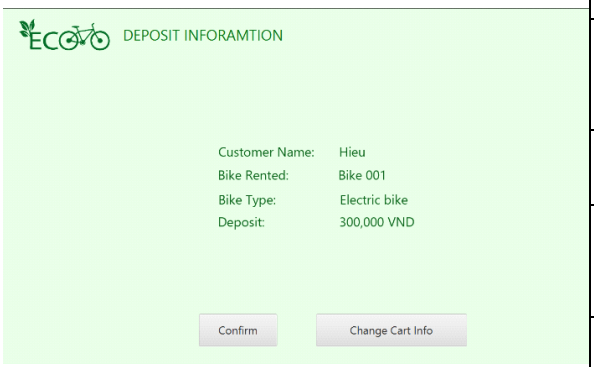
	Control	Operation	Function
	Logo	Click	Return to the main screen immediately
	Header	Initial	Display title of screen
	Dock information	Initial	Display dock information
	Return bike	Click	Allow user to start return bike process at the dock
	Bike list	Click	Display brief details about bikes available in current dock. Allow choosing each bike to see detailed information

Screen	View dock Screen			
Field name	**Type**	**Limitation**	**Attribute**	**Remarks**
Dock name	Text	50 characters	None	Left-justified
Bike name				

Address	Text	50 characters	None	Left-justified
Area	Digits	None	None	Comma for thousand separation
Total of docks	Digits	None	None	None
Bikes available				
Docks available				
Distance				In terms of kms
Battery				In terms of percentage Comma for decimal separation Ends with percentage notation
Time to dock				In terms of minutes

4.10 Deposit screen

Table 23: *Deposit screen specification*

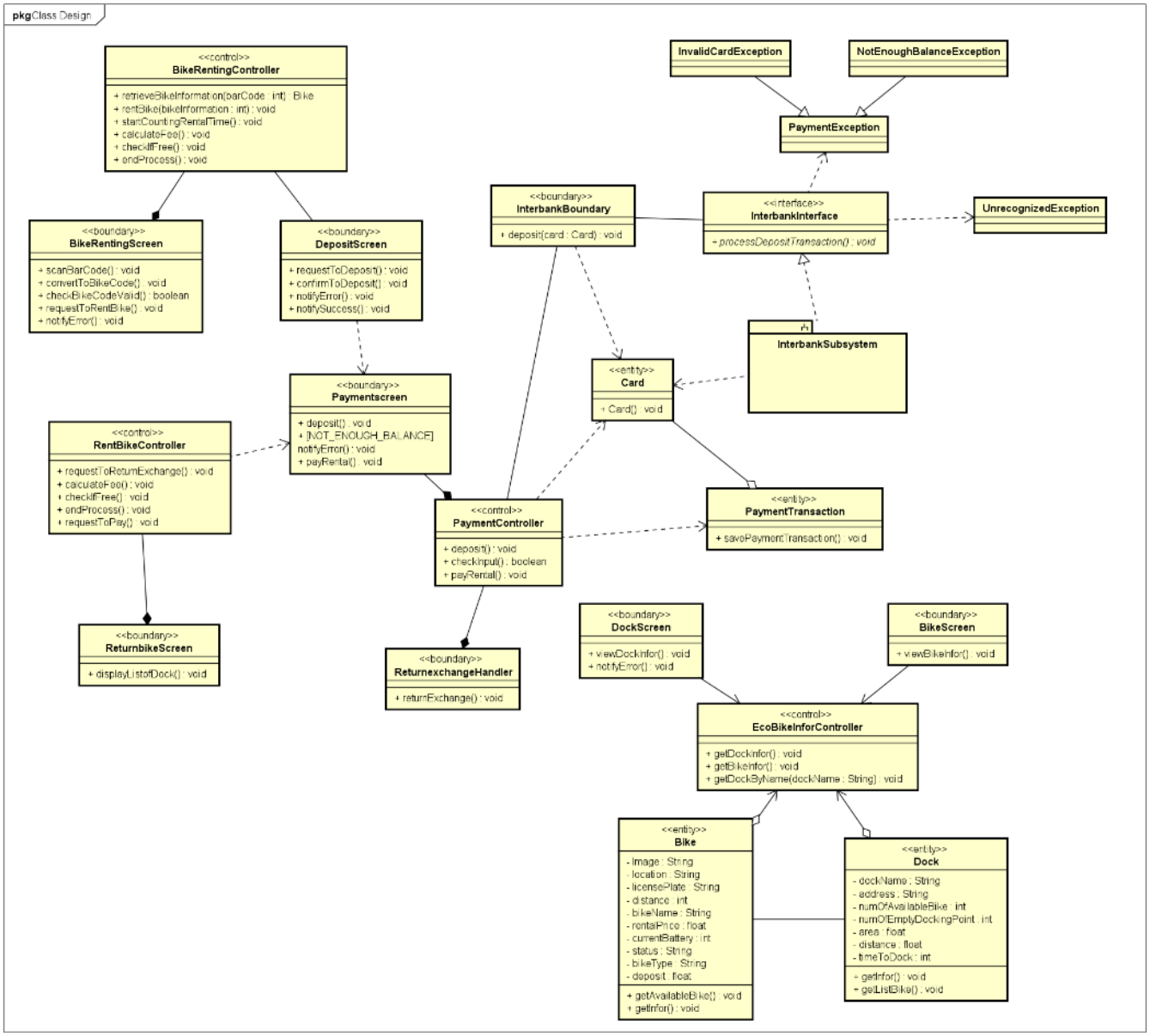
EcoBike Software		**Date of creation**	**Approved by**	**Reviewed by**	**Person in charge**
Screen specification	**Deposit screen**	02/01/2023			
		**Control**	**Operation**	**Function**	
		Logo	Click	Return to the main screen immediately	
		Header	Initial	Display title of screen	
		Information of payment	Initial	Display information of payment	
		Button	Click	Allow customer confirm to deposit the bike	

Screen	**Normal order invoice & Rush order invoice screen**

Field name	Type	Limitation	Attribute	Remarks
Customer name	Text	50 characters	None	Left-justified
Bike rented	Text	50 characters	None	None
Type of bike	Text	16 characters	None	None
Deposit	Digits	None	None	Including subtotal and VAT

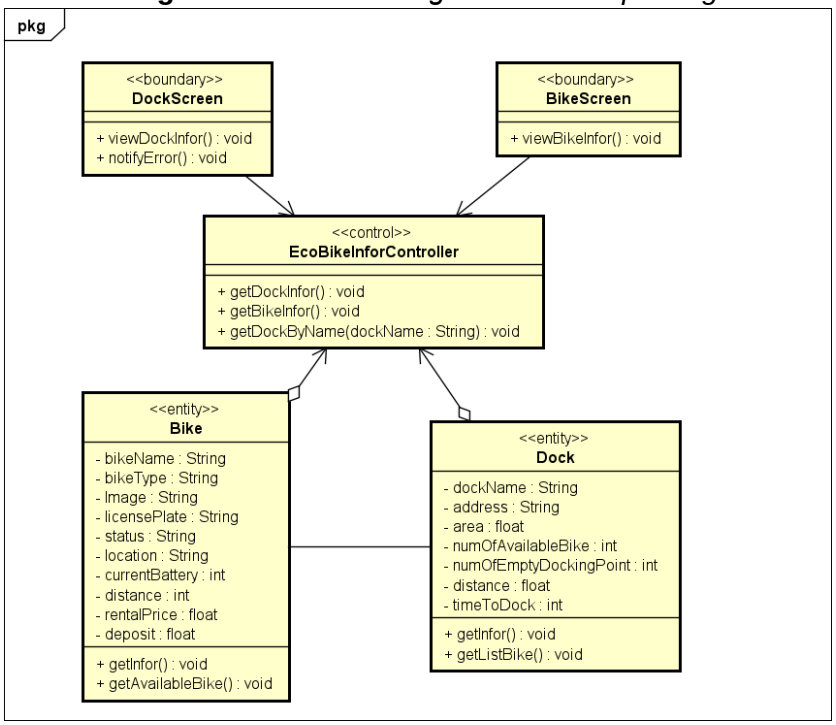
5. Class Design

Figure 16: *Full Class Diagram of EcoBike System*



5.1 CLASS DESIGN FOR UC “View parking dock and vehicle information”

Figure 17.1**:** *Class Diagram of "View parking dock and vehicle information"*



5.1.1 Analysis class diagram

I. Class "Bike"

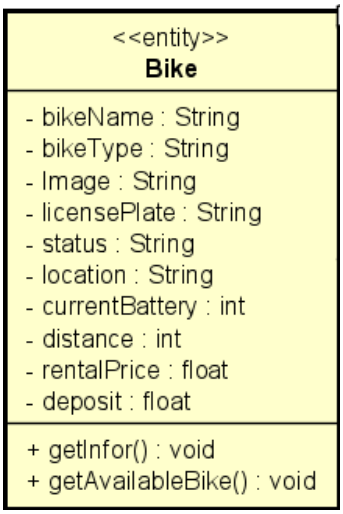


Figure 17.2: *Class Bike*

Attribute

Table 24: *Attribute of Class Bike*

#	*Name*	*Data type*	*Default value*	*Description*

1	bikeName	String	NULL	Represent name of the bike
2	bikeType	String	NULL	Represent type of the bike
3	image	String	NULL	Represent image of the bike
4	licensePlate	String	NULL	Represent license plate of thebike
5	status	String	NULL	Represent status of the bike
6	location	String	NULL	Represent location of the bike
7	currentBattery	int	NULL	Represent current battery of thebike
8	distance	int	NULL	Represent distance bike can gowith current battery
9	rentalPrice	float	NULL	Represent price to rent the bike
10	deposit	float	NULL	Repesent deposit user has to payto rent the bike

Operation

Table 25: *Operation of Class Bike*

#	*Name*	*Return type*	*Description (purpose)*
1	getInfor	void	Get detail information of the bike, and then return EcoBikeInforController
2	getAvailableBike	void	Get number of available bike, and then returnEcoBikeInforController

Parameter : None

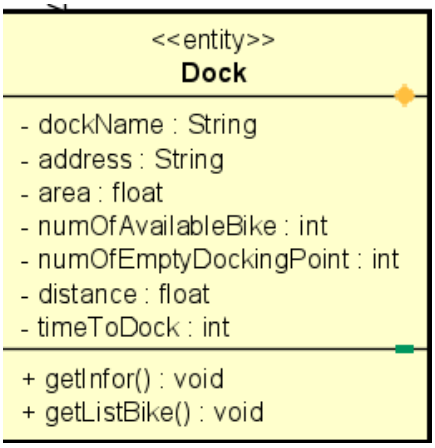
Exception: None

Method: None

State: None

II. Class “Dock”

Figure 17.3: *Class Dock*



Attribute

Table 26: *Attribute of Class Dock*

#	*Name*	*Data type*	*Default value*	*Description*
1	dockName	String	NULL	Represent name of the dock
2	address	String	NULL	Represent address of the dock
3	area	float	NULL	Represent parking area of thedock
4	numOfAvailable	int	NULL	Represent number of availablebikes in the dock
5	numOfEmptyDocingPoint	int	NULL	Represent number of emptydocking point in the dock
6	distance	float	NULL	Represent distance from user’slocation to the dock
7	timeToDock	int	NULL	Represent walking time from user’s location to the dock

Operation

Table 27: *Operation of Class Dock*

#	*Name*	*Return type*	*Description (purpose)*
1	getInfor	void	Get detail information of the dock, and thenreturn EcoBikeInforController
2	getListBike	void	Get list of available bike, and then returnEcoBikeInforController

Parameter : None

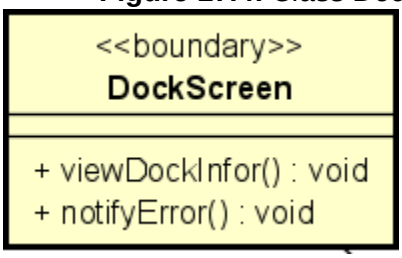
Exception: None

Method: None

State: None

III. Class "DockScreen"

Figure 17.4: *Class DockScreen*



Attribute

None

Operation

Table 28: *Operation of Class DockScreen*

#	*Name*	*Return type*	*Description (purpose)*
1	viewDockInfor	void	Get detail information of the dock
2	notifyError	void	Notify error when no dock match the given infotmation, and then return DockScreen

Parameter : None

Exception: None

Method: None

State: None

IV. Class “BikeScreen”

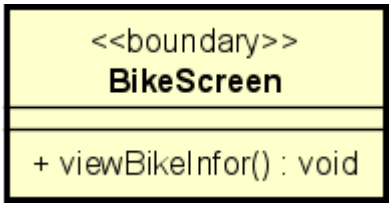


Figure 17.5: *Class BikeScreen*

Attribute

None

Operation

Table 29: *Operation of Class BikeScreen*

#	*Name*	*Return type*	*Description (purpose)*
1	viewBikeInfor	void	Display detail information of the bike

Parameter : None

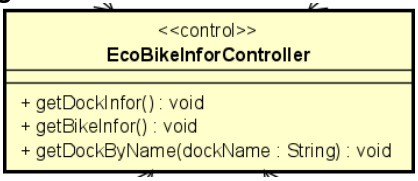
Exception: None

Method: None

State: None

V. Class “EcoBikeInforController”

Figure 17.6: *Class EcoBikeInfoController*



Attribute

None

Operation

Table 30: *Operation of Class EcoBikeInfoController*

#	*Name*	*Return type*	*Description (purpose)*
1	getDockInfor	void	Get detail information of the dock, and thenreturn DockScreen

2	getBikeInfor	void	Get detail information of the bike, and then return BikeScreen
3	getDockByName	void	Get dock by name

Parameter

- dockName: name of the dock

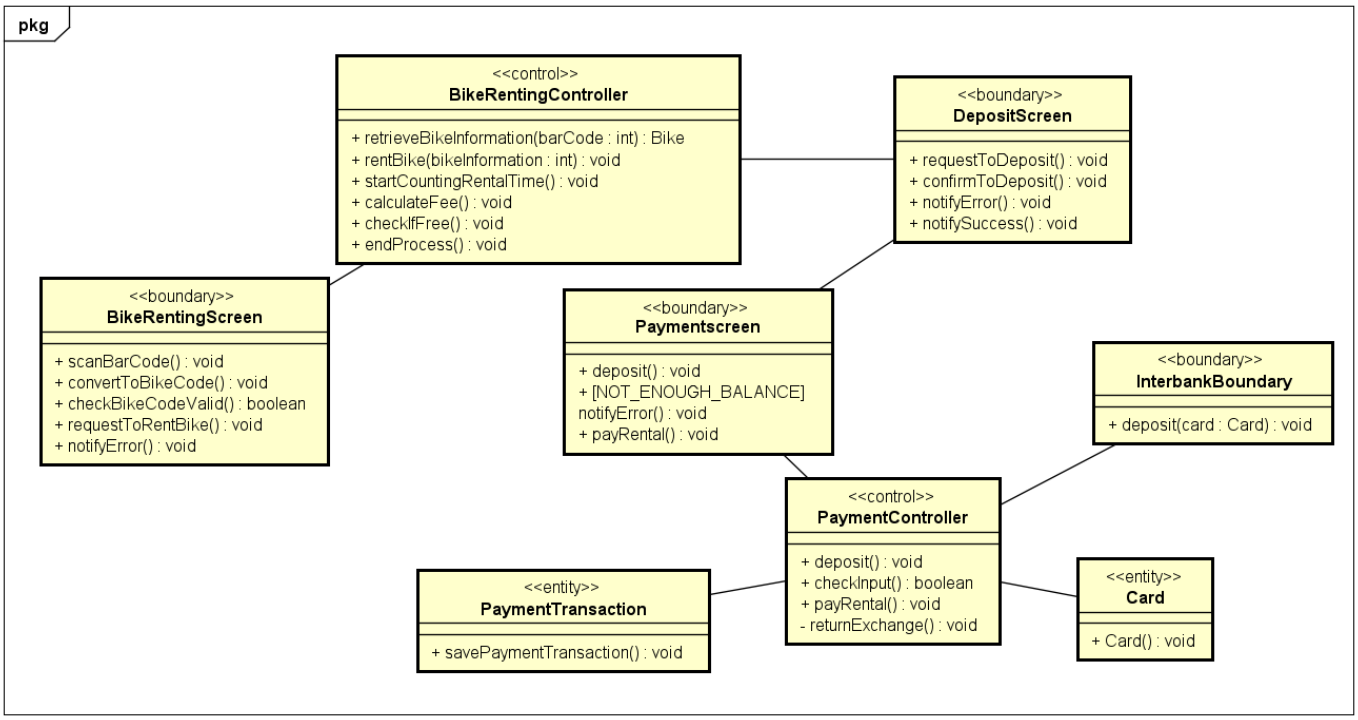
Exception: None

Method: None

State: None

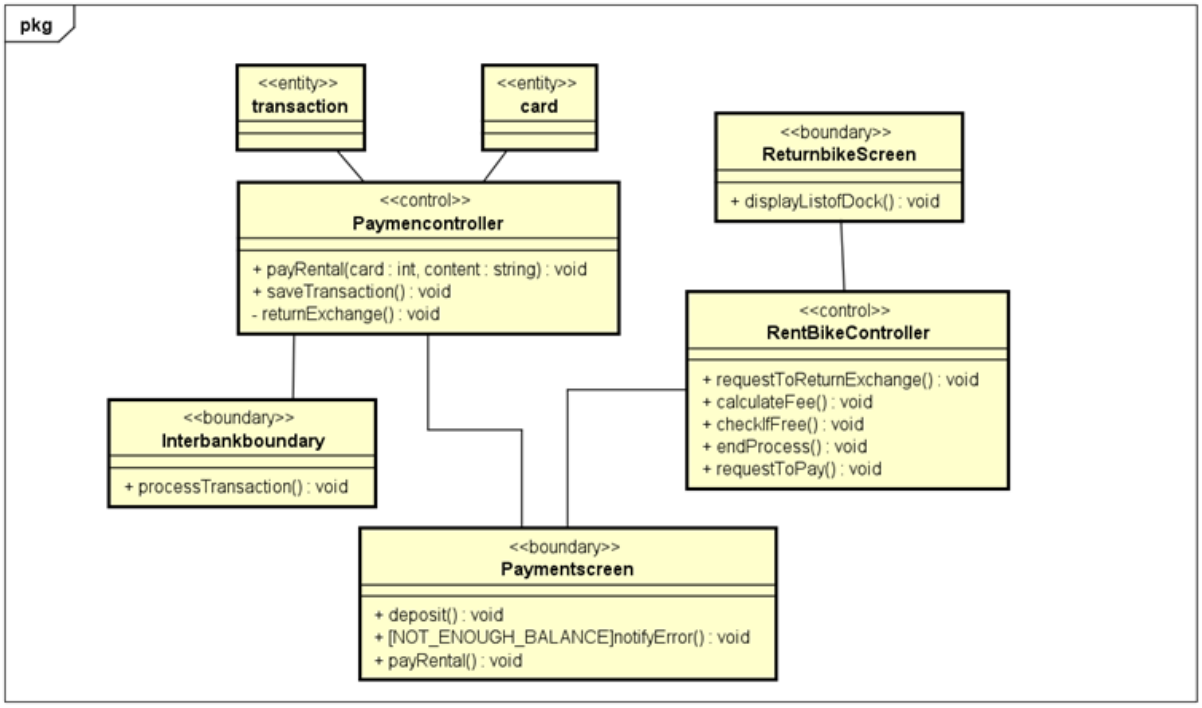
5.2 CLASS DESIGN FOR UC "Rent a Bike"

Figure 18: *Class Diagram of "Rent a Bike"*



5.3 CLASS DESIGN FOR UC "Return a Bike"

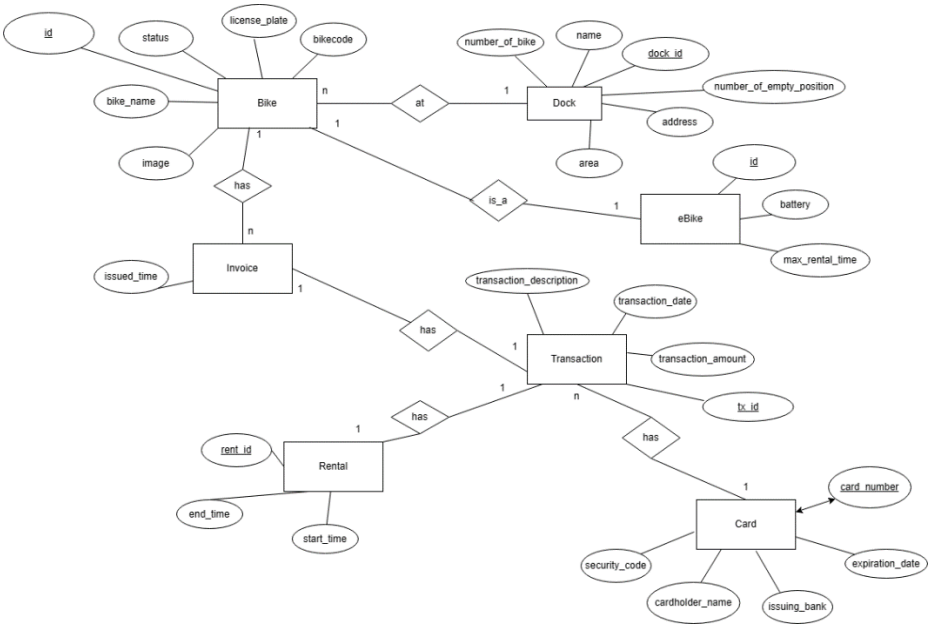
Figure 19: *Class Diagram of "Return a Bike"*



6. Data Modeling

6.1 ER Diagram

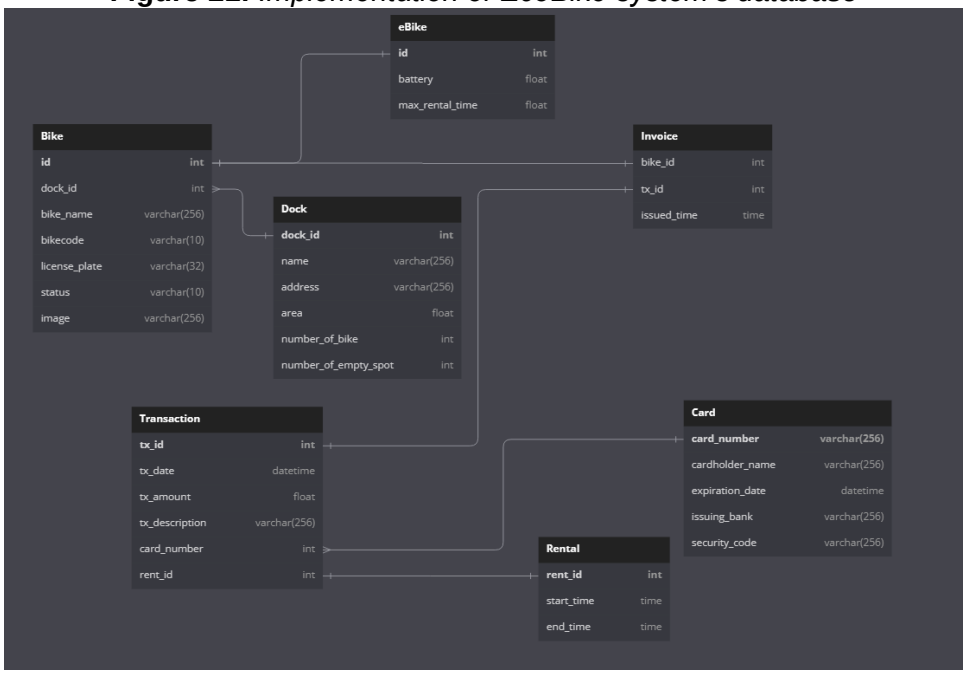
Figure 20: *ER-Diagram for EcoBike system's database*



6.2 Database design

6.2.1 Logical data model

Figure 21: *Implementation of EcoBike system's database*



6.2.2 Physical data model

Dock

Table 31: *Dock table design*

No.	PK	FK	Name	Data type	Mandatory	Description
1	x		dock_id	int	x	ID of dock
2			name	varchar(256)	x	Name of the dock
3			address	varchar(256)	x	Address of the dock
4			area	float		Area of the dock
5			num_of_bike	int	x	Number of current available bike in dock
6			num_of_empty_spot	int	x	Number of current available bike slot in dock for returning bike

Bike

Table 32: *Bike table design*

No.	PK	FK	Name	Data type	Mandatory	Description
1	x		id	int	x	ID of the bike
2		x	dock_id	int	x	ID of the dock
3			bike_name	varchar(256)	x	Name of the bike
4			lisence_plate	varchar(32)		License plate of the bike
5			bike_code	varchar(32)	x	Barcode of the bike

6			status	varchar(10)	x	Available/rented
7			image	varchar(256)		Path to image of the bike

eBike

Table 33: *eBike table design*

No.	PK	FK	Name	Data type	Mandatory	Description
1	x	x	id	int	x	ID of the invoice
2			battery	float	x	Current battery of the bike
3			max_rental_time	float	x	Time bike can go with current battery

Invoice

Table 34: *Invoice table design*

No.	PK	FK	Name	Data type	Mandatory	Description
1		x	bike_id	int	x	ID of the invoice
2	x	x	transaction_id	int	x	ID of the transaction
3			issued_time	DateTime	x	String representation of the time that the invoice is issued

Card

Table 35: *Card table design*

No.	PK	FK	Name	Data type	Mandatory	Description
1	x		card_number	varchar(256)	x	Number of the card
2			cardholder_name	varchar(256)	x	Name of cardholder
3			expiration_date	datetime	x	Expiration date of the card
4			issuing_bank	varchar(256)	x	Issuing bank of the card
5			security_code	varchar(256)	x	Security code

Transaction

Table 36: *Transaction table design*

No.	PK	FK	Name	Data type	Mandatory	Description
1	x		transaction_id	int	x	ID of transaction
2			transaction_date	datetime	x	Time the transaction is made
3			transaction_amount	float	x	The amount of money for the transaction
4			transaction_desciption	varchar(256)		The content of the transaction
5		x	rent_id	int	x	ID of

Rental

Table 37: *Rental table design*

No.	PK	FK	Name	Data type	Mandatory	Description
1	x		rent_id	int	x	ID of the rental
2			start_time	time	x	Time start renting
3			end_time	time		Time end renting (null if the bike is currently being rented)

6.2.3 Database script

```
Create table Bike(
  id int primary key,
  dock_id int NOT NULL FOREIGN KEY REFERENCES Dock(dock_id),
  bike_name varchar(256),
  bike_code varchar(10),
  license_plate varchar(32),
  status varchar(10),
  image varchar(256)
)
Create table Dock (
  dock_id int primary key,
  name varchar(256),
  address varchar(256) NOT NULL,
  area float,
  number_of_bike int,
  number_of_empty_spot int
)
Create table eBike (
  id int primary key,
```

```
  battery int NOT NULL,
  max_rental_time float,
  FOREIGN KEY id REFERENCES Bike(id)
)
Create table Invoice (
  bike_id int NOT NULL FOREIGN KEY REFERENCES Bike(id),
  tx_id int NOT NULL FOREIGN KEY REFERENCES Transaction(tx_id),
  issued_time Datetime NOT NULL,
)
Create table Transaction (
  tx_id int primary key,
  tx_date Datetime,
  tx_description varchar(256),
  tx_amount float NOT NULL,
  rent_id int NOT NULL FOREIGN KEY REFERENCES Rental(rent_id),
  card_number int NOT NULL FOREIGN KEY REFERENCES Card(card_number)
)
Create table Rental (
  rent_id int primary key,
  start_time Datetime NOT NULL,
  end_time Datetime NOT NULL
)
Create table Card (
  card_number int primary key,
  cardholder_name varchar(256) NOT NULL,
  expiration_date Datetime,
  issuing_bank varchar(256) NOT NULL,
  security_code varchar(256)
)
```

7. Design Considerations

7.1 Goals

- Provide a friendly application for customer
- Provide an eye-catching interface for users

7.2 Architectural Strategies

- Programming Language: Java
- Database: MySQL
- UML: Astah UML
- GUI: Scene Builder JavaFx