

HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY
School of Information and communications technology

Software Requirement Specification

Version 1.0

EcoBikeRental

Subject: IT Software Development

Group 10

Vu Trung Dung
Nguyen Xuan Hoang
Nguyen Trung Nghia
Nguyen Ngoc Quy

Hanoi, 12/2020

Table of contents

Table of contents	2
Introduction	6
1.1. Objective	6
1.2. Scope	6
1.3. Glossary	6
1.4. References	6
Overall Description	7
2.1. Actors	7
2.2. Use case diagrams	7
2.3. Business processes	7
Detailed Requirements	8
3.1. Use case specification for “Renting Bikes”	8
3.1.1. Use case code	8
3.1.2. Brief description	8
3.1.3. Actors	8
3.1.4. Preconditions	8
3.1.5. Basic flow of events	8
3.1.6. Alternative flows	9
3.1.7. Activity diagrams	9
3.1.8. Input data	10
3.1.9. Output data	10
3.1.10. Post conditions	10
3.2. Use case specification for “Choose Payment Method”	10
3.2.1. Use case code	10
3.2.2. Brief Description	11
3.2.3. Actors	11
3.2.4. Preconditions	11
3.2.5. Basic Flow of Events	11
3.2.6. Alternative flows	11
3.2.7. Activity diagrams	12
3.2.8. Input data	12
3.2.9. Output data	12
3.2.10. Postconditions	13
3.3. Use case specification for “Deduct money”	13
3.3.1. Use case code	13

3.3.2. Brief Description	13
3.3.3. Actors	13
3.3.4. Preconditions	13
3.3.5. Basic Flow of Events	13
3.3.6. Alternative flows	13
3.3.7. Activity diagrams	14
3.3.8. Input data	14
3.3.9. Output data	15
3.3.10. Postconditions	15
3.4. Use case specification for “Return Bikes”	15
3.4.1. Use case code	15
3.4.2. Brief Description	15
3.4.3. Actors	15
3.4.4. Preconditions	16
3.4.5. Basic Flow of Events	16
3.4.6. Alternative flows	16
3.4.7. Activity diagrams	17
3.4.8. Input data	17
3.4.9. Output data	17
3.4.10. Postconditions	18
3.5. Use case specification for “Return Deposit Money”	18
3.5.1. Use case code	18
3.5.2. Brief Description	19
3.5.3. Actors	19
3.5.4. Preconditions	19
3.5.5. Basic Flow of Events	19
3.5.6. Alternative flows	19
3.5.7. Activity diagrams	20
3.5.8. Input data	20
3.5.9. Output data	20
3.5.10. Postconditions	21
3.6. Use case specification for “View Rented Bikes Information”	21
3.6.1. Use case code	21
3.6.2. Brief Description	21
3.6.3. Actors	21
3.6.4. Preconditions	22
3.6.5. Basic Flow of Events	22
3.6.6. Alternative flows	22
3.6.7. Activity diagrams	23

3.6.8. Input data	23
3.6.9. Output data	23
3.6.10. Postconditions	24
3.7. Use case specification for “View All Bikes In Station”	24
3.7.1. Use case code	24
3.7.2. Brief Description	24
3.7.3. Actors	24
3.7.4. Preconditions	24
3.7.5. Basic Flow of Events	25
3.7.6. Alternative flows	25
3.7.7. Activity diagrams	26
3.7.8. Input data	26
3.7.9. Output data	27
3.7.10. Postconditions	27
3.8. Use case specification for “View Bike Details”	27
3.8.1. Use case code	27
3.8.2. Brief Description	27
3.8.3. Actors	27
3.8.4. Preconditions	27
3.8.5. Basic Flow of Events	28
3.8.6. Alternative flows	28
3.8.7. Activity diagrams	28
3.8.8. Input data	28
3.8.9. Output data	28
3.8.10. Postconditions	29
3.9. Use case specification for “View All Dock Stations”	29
3.9.1. Use case code	29
3.9.2. Brief Description	29
3.9.3. Actors	29
3.9.4. Preconditions	29
3.9.5. Basic Flow of Events	30
3.9.6. Alternative flows	30
3.9.7. Activity diagrams	30
3.9.8. Input data	30
3.9.9. Output data	31
3.9.10. Postconditions	31
Supplementary specification	31
4.1. Functionality	31

4.2. Usability	31
4.3. Reliability	32
4.4. Performance	32
4.5. Supportability	32
4.6. Other requirements	32

1. Introduction

1.1. Objective

The purpose of this document is to present a detailed description of the Eco Bike Rental 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 and will be proposed to the Eco Park Township Management Department for its approval.

1.2. Scope

This software system will be a Eco Park Bike Rental System for everyone including novice users to use without any training. This system will be designed to allow for approximately 100 average concurrent users with no perceivable performance difference and can be operated upto 200 hours continuously. The system is also very responsive with typical response time around 1 second and only requires 2 hours of downtime for maintenance.

1.3. Glossary

Term	Definition
User	Main actor of the system
Map	The entire area of Eco Park, with detailed location of all docking stations
Docking station	The area to store all bikes available to the user
E-bike	Standard bike with an integrated electric motor for assisted propulsion
Twin bike	Standard bike with 2 saddles, 2 pedal and no electric motor

1.4. References

- IEEE. IEEE Std 830–1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.
- [IEEE] The applicable IEEE standards are published in “IEEE Standards Collection,” 2001 edition.
- [Bruade] The principal source of textbook material is “Software Engineering: An Object-Oriented Perspective” by Eric J. Bruade (Wiley 2001).

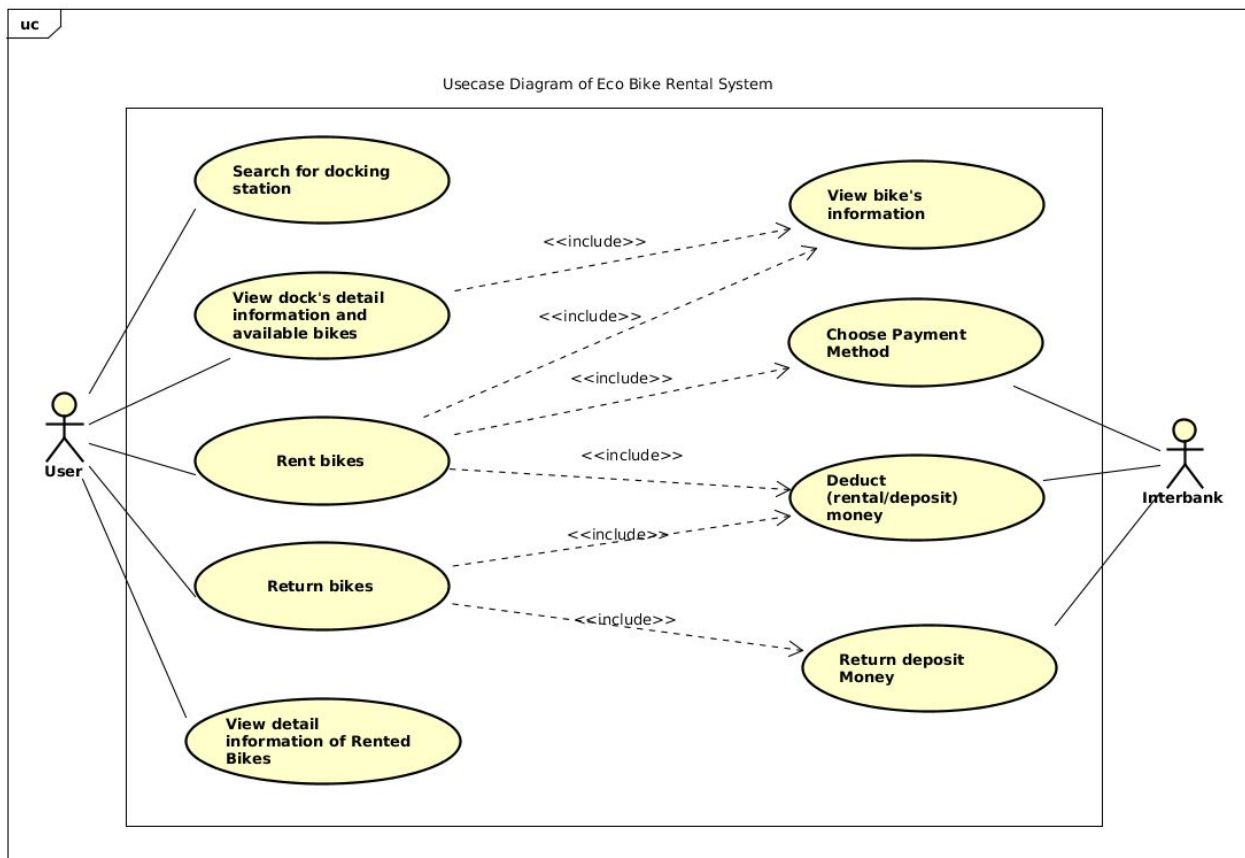
- [Reaves SPMP] "Software Project Management Plan Jacksonville State University of Computing and Information Sciences Web Accessible Alumni Database." Jacksonville State University, 2003.

2. Overall Description

2.1. Actors

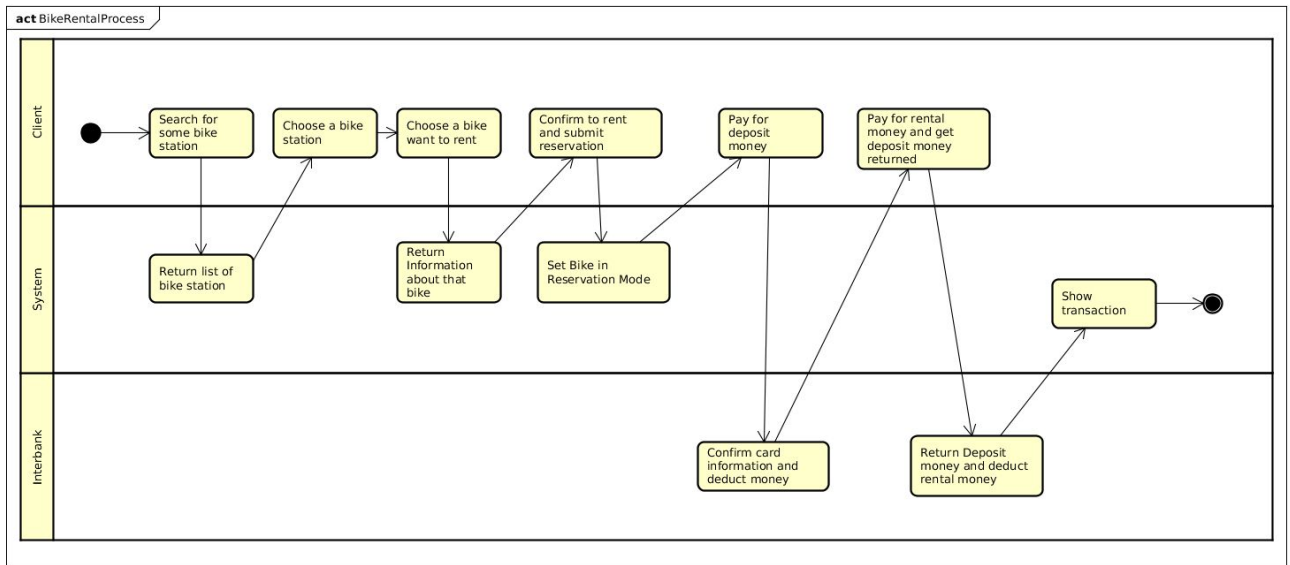
There are 2 main actors in the system: Customer and Interbank.

2.2. Use case diagrams



2.3. Business processes

In our EcoBike System, we will have the general business processes for renting bike as following diagram:



3. Detailed Requirements

3.1. Use case specification for “Renting Bikes”

3.1.1. Use case code

UC001

3.1.2. Brief description

This user case describes the interaction between Customer and Software when Customer wishes to rent a bike

3.1.3. Actors

- Customer

3.1.4. Preconditions

None

3.1.5. Basic flow of events

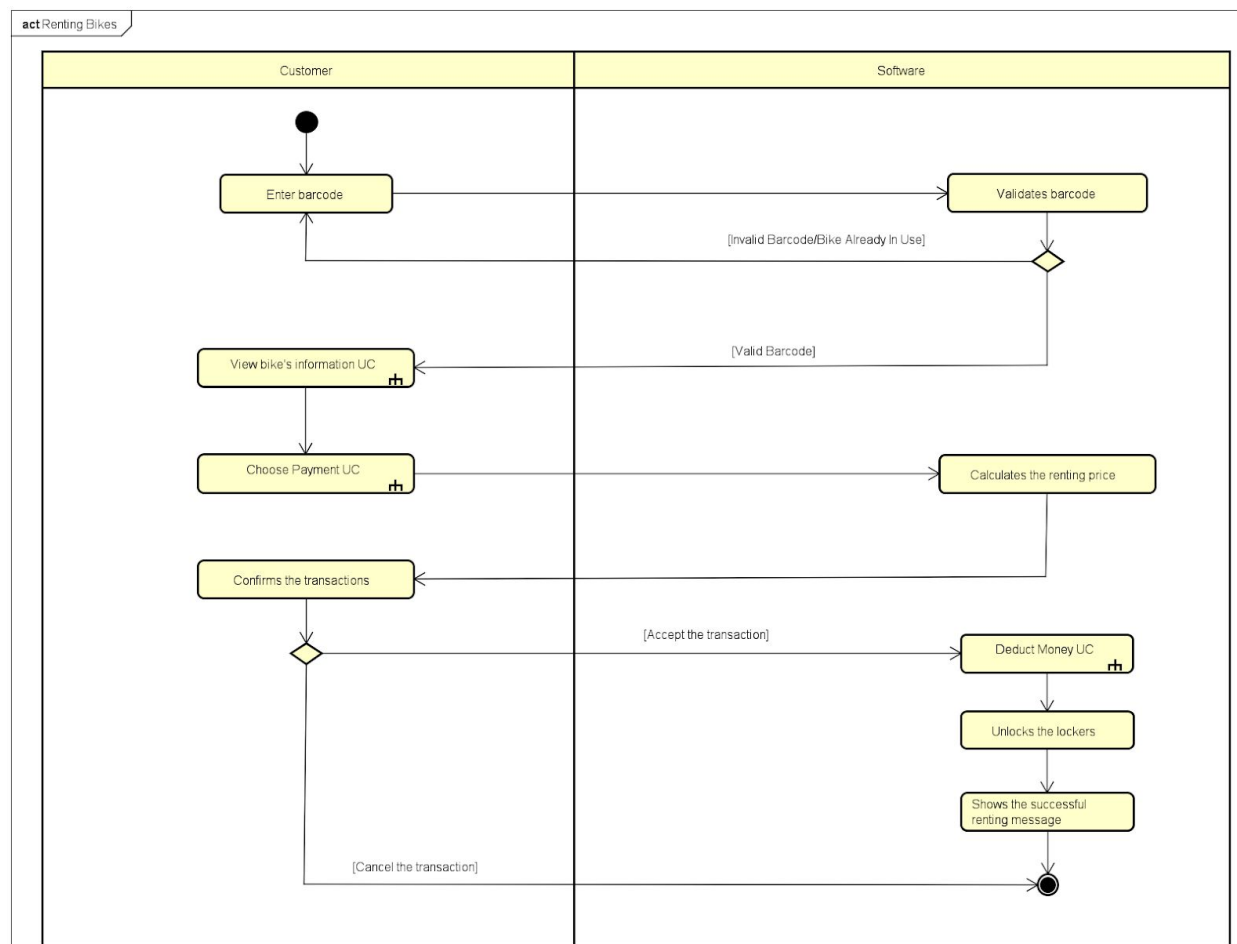
- Step 1. Customer enter the barcode and request to rent a bike
- Step 2. Software validates the barcode
- Step 3. Software forwards to View bikes’s information Use Case
- Step 4. Software forwards to choose Payment Method Use Case
- Step 5. Software calculates the renting price and asks Customer to confirm
- Step 6. Customer confirms the transactions
- Step 7. Software forwards to Deduct money Use Case
- Step 8. Software saves the transactions
- Step 9. Software unlocks the locker, allow customer to use the bike
- Step 10. Shows the successful renting message

3.1.6. Alternative flows

Table 1-Alternative flows of events for UC Renting Bikes

No	Location	Condition	Action	Resume location
1.	At Step 2	If the barcode which customer entered is invalid or bike is already in use	<ul style="list-style-type: none"> ▪ Notify invalid barcode or the bike is unavailable to Customer 	Resumes at Step 1
2.	At Step 6	If Customer cancel the confirmation	<ul style="list-style-type: none"> ▪ Cancel all the transactions ▪ Notify that all transactions was cancels 	Use case ends

3.1.7. Activity diagrams



3.1.8. Input data

Table 2-Input data of Barcode

No	Data fields	Description	Mandatory	Valid condition	Example
1.	Barcode		Yes		4fy7tvi7

3.1.9 Output data

Table 3-Output data of Transaction Status

No	Data fields	Description	Display format	Example
1	Transaction id	The id of the transactions which customer paid for renting		ID121
2	Start Time	Time when customer start renting	Hh:mm dd/mm/yyyy	09:00 19/09/2020
3	Deposit	Amount of money which customer paid for renting	- Comma for thousands separator - Positive integer - Right alignment	430,000
4	Barcode of the rented bike			4fy7tvi7

3.1.10. Post conditions

None

3.2. Use case specification for “Choose Payment Method”

3.2.1. Use case code

UC002

3.2.2. Brief Description

This user case describes the interaction between Customer, Software and Interbank when customer wishes to choose a payment method.

3.2.3. Actors

- Customer
- Interbank

3.2.4. Preconditions

Customer had logged in

3.2.5. Basic Flow of Events

Step 1. Customer choose payment methods and payment information

Step 2. Software call Validating Information API

Step 3. Interbank validates payment information and sends status to Software

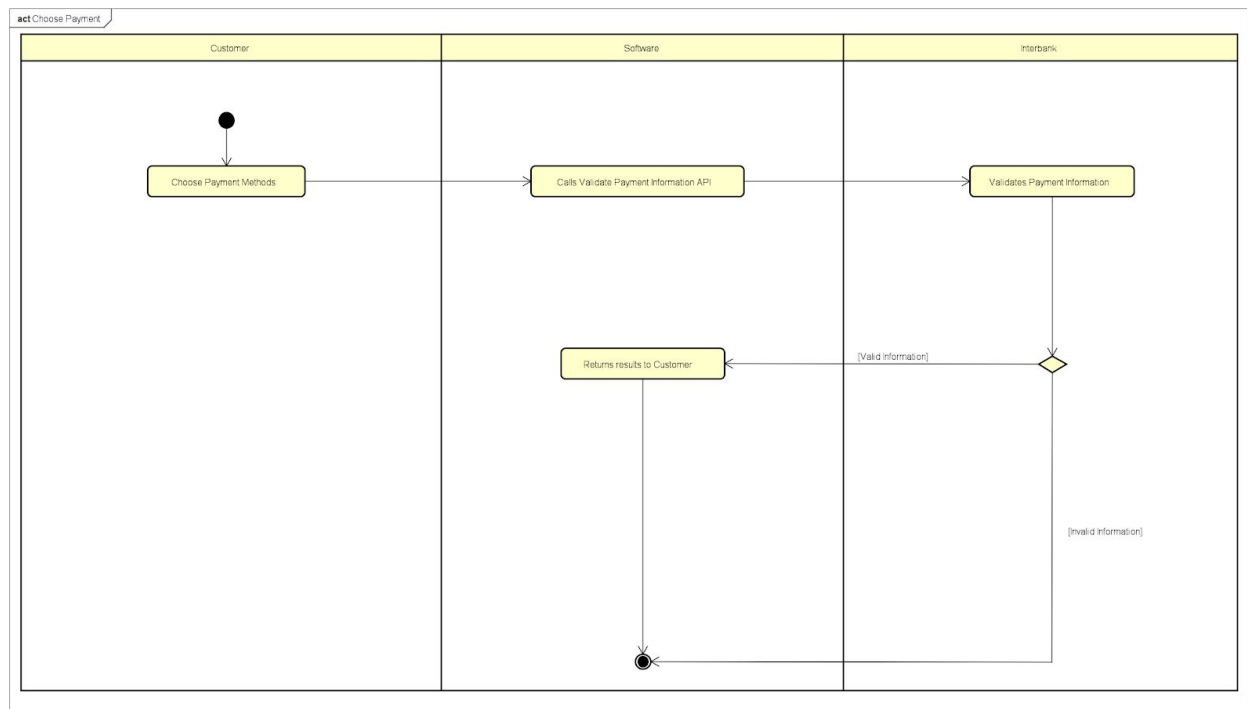
Step 4. Software returns result to customer

3.2.6. Alternative flows

Table 4-Alternative flows of events for UC Choose Payment Method

No	Location	Condition	Action	Resume location
1.	At Step 3	Error when validating payment information	<ul style="list-style-type: none">▪ Sends the failure message to Software▪ Notify that wrong payment method to customer	The use case ends

3.2.7. Activity diagrams



3.2.8. Input data

Table 5-Input data of payment information

No	Data fields	Description	Mandatory	Valid condition	Example
1.	Bank's Name	Choose from a list	Yes		Vietcombank
2.	Card's Number		Yes		4283 1001 2345 6789
3.	Cardholder Name		Yes		Nguyen Xuan Hoang
4.	Card expire Date		Yes	Time in future	12/22

3.2.9. Output data

None

3.2.10. Postconditions

None

3.3. Use case specification for “Deduct money”

3.3.1. Use case code

UC003

3.3.2. Brief Description

This use case can be included whenever the software needs to deduct an amount of money from the user by using the interbank API.

3.3.3. Actors

- Customer

3.3.4. Preconditions

None

3.3.5. Basic Flow of Events

Step 1. The software calls check account API

Step 2. The interbank return the balance on the customer’s credit card

Step 3. The software compares amount of money need to deduct with the balance

Step 4. The software call “deduct money” API

Step 5. The interbank deduct money from customer’s account

Step 6. The software returns status of transactions

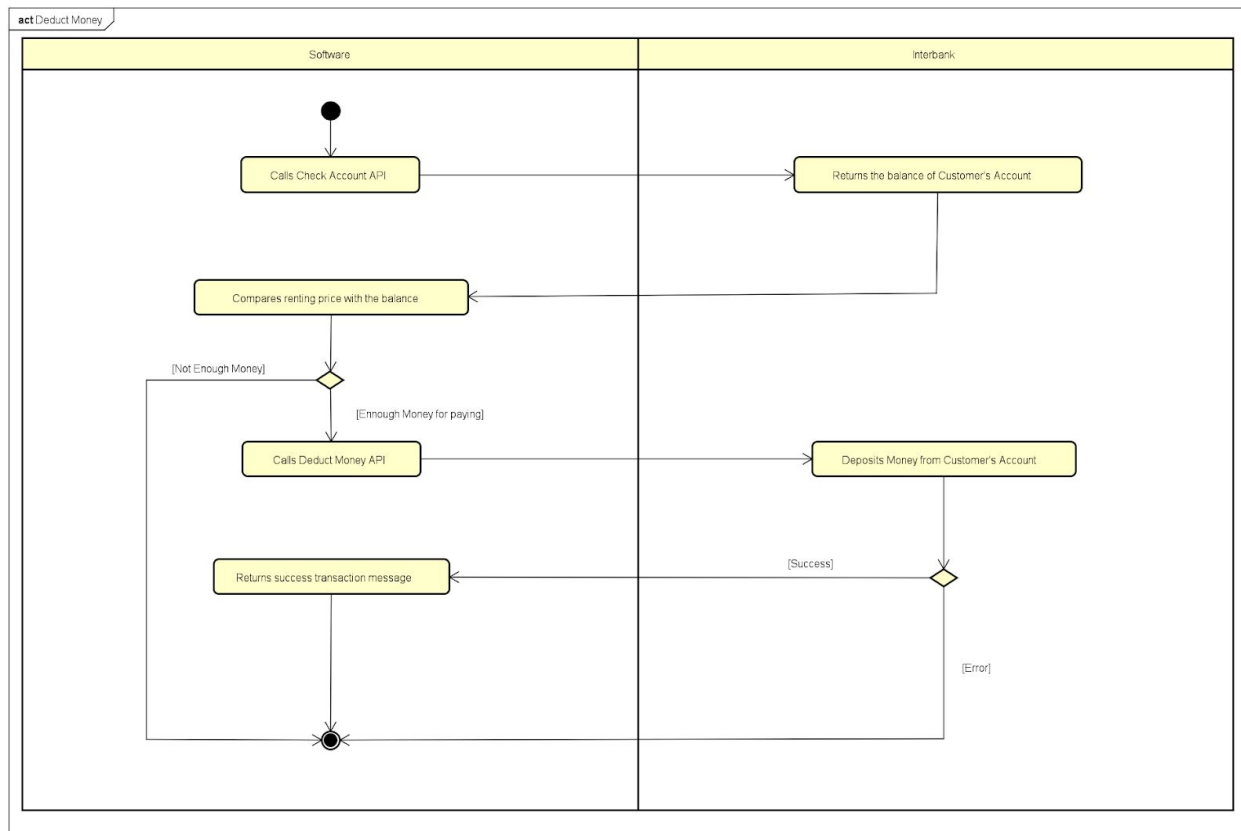
3.3.6. Alternative flows

Table N-Alternative flows of events for UC Deduct Money

No	Location	Condition	Action	Resume location
1.	At Step 2	InterbankError	▪ Sends the failure message to Software	The use case ends
2.	At Step 3	If not enough money for renting	▪ Sends the failure message to software	Use case ends

	At step 5	Interbank Error	<ul style="list-style-type: none"> Sends the failure message to software 	The use case ends

3.3.7. Activity diagrams



3.3.8. Input data

Table A- Input data for calling deduct money API

No	Data fields	Description	Mandatory	Valid condition	Example
1.	Current balance		Yes	Must not contain words	

2.	Account number		Yes	Must not contain words	
3.	Bank's name	Choose from a list	Yes		

3.3.9. Output data

No	Data fields	Description	Display format	Example
1	Owner's balance	Balance	- Number, separated comma for big number	2,000,000 VND
2	deduct amount	Amount of deducted money	- Number, separated comma for big number	2,000,000 VND
3	Transaction status	status	- Success or failed	Successfully transaction
4	Card's owner information(owner's name, ...)	Information of card's owner	- Words	12310243324 Nguyen Trung Nghia

3.3.10. Postconditions

None

3.4. Use case specification for "Return Bikes"

3.4.1. Use case code

UC004

3.4.2. Brief Description

This Use case describes the interaction between Customer and EcoBikeRental System where Customers wish to return their bike which they had already rented before.

3.4.3. Actors

- Customer
- Interbank

3.4.4. Preconditions

Customers had already rented bikes in EcoBikeRental System

3.4.5. Basic Flow of Events

Step 1. The customer requests to return bikes.

Step 2. The software sends the rental bike information.

Step 3. The customer chooses an available dock station.

Step 4. The software validates the chosen dock station's status.

Step 5. The software forwards to Return Deposit Money Use Case.

Step 6. The software forwards to Deduct Money Use Case.

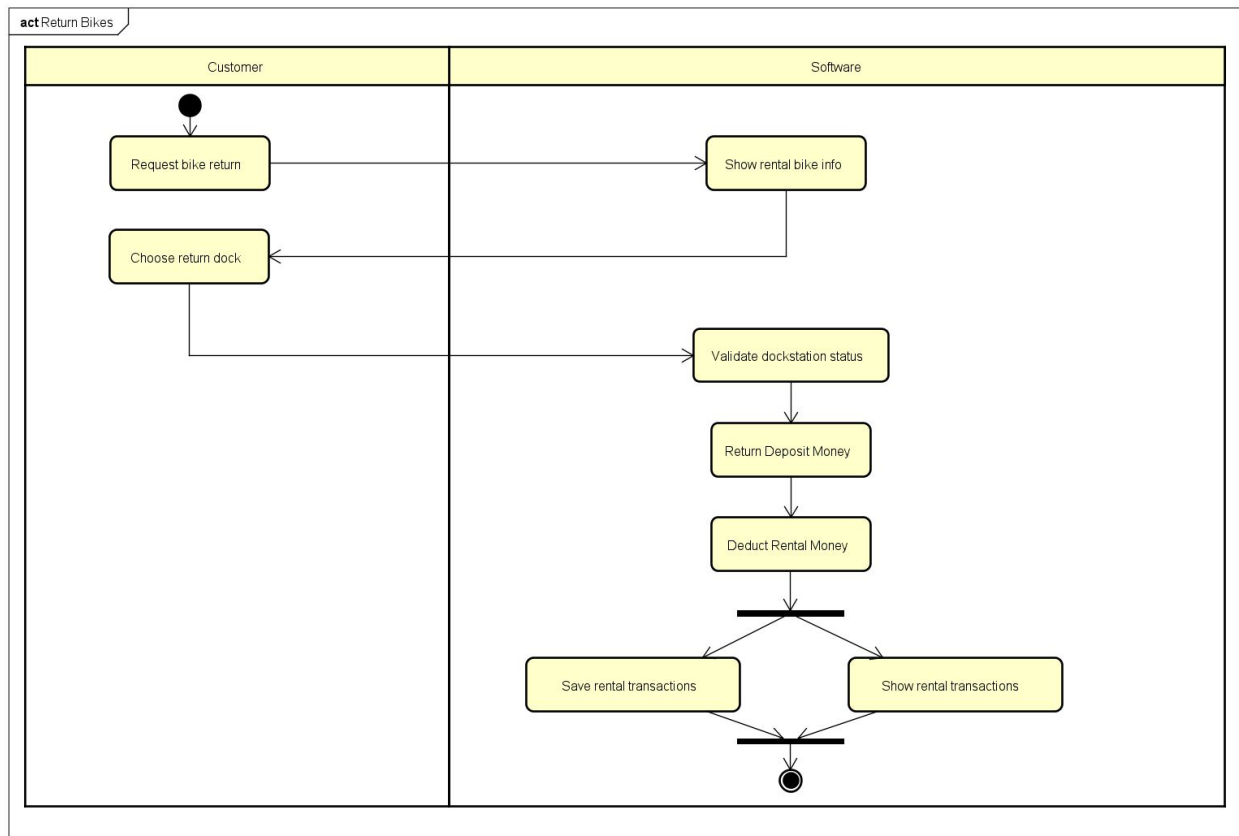
Step 7. The software shows and saves rental transactions

3.4.6. Alternative flows

Table N-Alternative flows of events for UC Return Bike

No	Location	Condition	Action	Resume location
1.	At Step 2	If the barcode was invalid	▪ Notifies barcode error to customer	The use case ends

3.4.7. Activity diagrams



3.4.8. Input data

Table A- Input data for Barcode Scanning

No	Data fields	Description	Mandatory	Valid condition	Example
1.	Barcode number	Barcode of the rented bike	Yes		3jk6ad2

3.4.9. Output data

Table B- Output data for Barcode Scanning

No	Data fields	Description	Display format	Example
1	Customer name	The name of the customer		Vu Trung Dung
2	Barcode of the rented bike			3jk6ad2

3	Rented Bike Information	Information of rented bike		E-bike, Battery 80%
4	Deduct Amount	Total money for the rented bike	Comma for thousands separator Have positive or negative sign Right alignment	- 246,000
5	Deposit Amount	Total money for depositing the bike	Comma for thousands separator Have positive or negative sign Right alignment	+ 200,000
6	Currency			VND
8	Subtotal	Total money the customer have to pay, include returned deposit amount	Comma for thousands separator Have positive or negative sign Right alignment	- 46,000
7	Transaction status	The status of the transaction	Success - Failed	Success

3.4.10. Postconditions

None

3.5. Use case specification for “Return Deposit Money”

3.5.1. Use case code

UC005

3.5.2. Brief Description

This Use case describes the interaction between Customer and EcoBikeRental System where Customers wish to take back the deposit money they had already deposited before.

3.5.3. Actors

- Customer
- Interbank

3.5.4. Preconditions

Customer had already deposit money for rented bike in EcoBikeRental System

3.5.5. Basic Flow of Events

Step 1. The Software calls Add API from the bank .

Step 2. The Interbank shows the card information of the customer.

Step 3. The Customer confirms that the card information is correct.

Step 4. The Interbank adds deposit money to customer's credit card

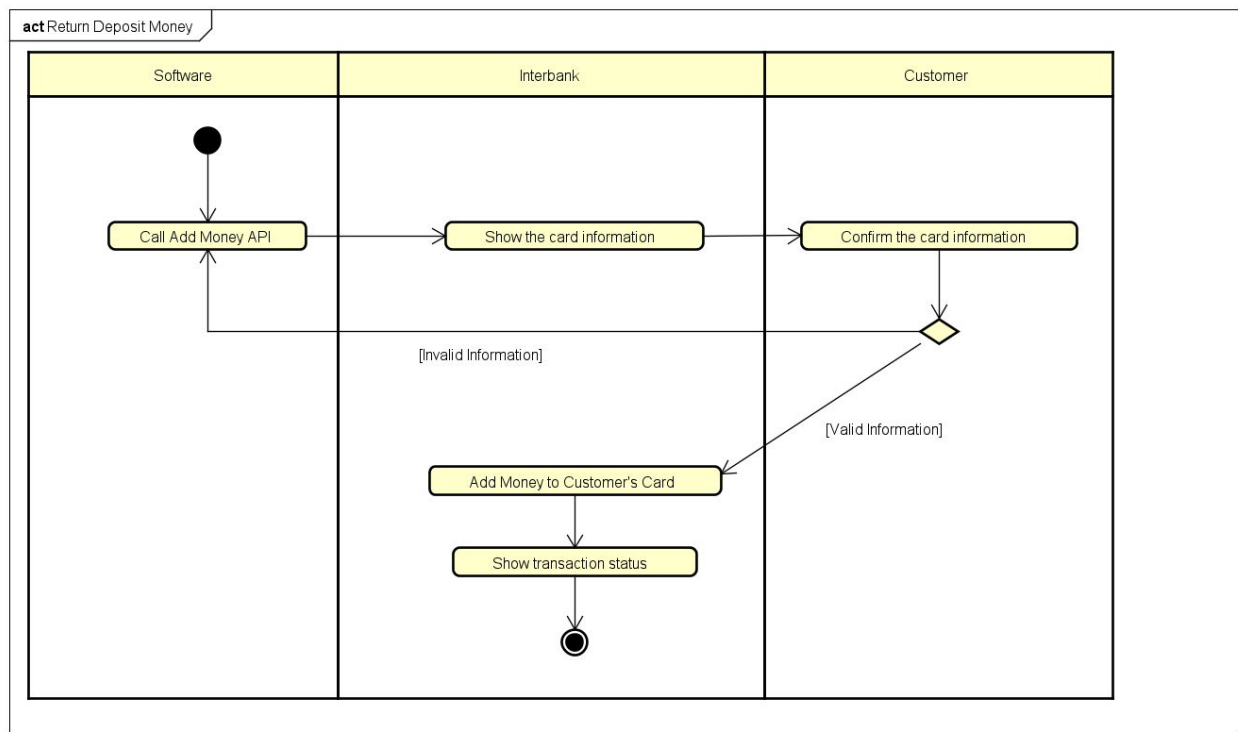
Step 5. The Software displays the success transaction.

3.5.6. Alternative flows

Table N-Alternative flows of events for UC Return Bike

No	Location	Condition	Action	Resume location
1.	At Step 3	If the card information is invalid	▪ Software notifies that the interbank return the wrong card information	Resumes at Step 1

3.5.7. Activity diagrams



3.5.8. Input data

None

3.5.9. Output data

Table B1-Output data of Displaying card information

No	Data fields	Description	Display format	Example
1	Bank's name	The name of the card's bank		Techcombank Hang Dau
2	Card's information	Information of the card holder, includes card number and card holder's name		1902470153012 Chu Viet Dung

Table B2-Output data of Displaying transactions status

No	Data fields	Description	Display format	Example
----	-------------	-------------	----------------	---------

1	Bank's name	The name of the card's bank		Techcombank Hang Dau
2	Card's information	Information of the card holder, includes card number and card holder's name		1902470153012 Chu Viet Dung
4	Return Deposit Amount	Total deposit money to be returned	Comma for thousands separator Positive integer Right alignment	200,000
5	Currency			VND
6	Transaction status	The status of the transaction	Success - Failed	Success
8	Transaction instructions			Return Deposit Money for Bike rented in EcoBikeRental System

3.5.10. Postconditions

None

3.6. Use case specification for “View Rented Bikes Information”

3.6.1. Use case code

UC006

3.6.2. Brief Description

UC “View Rented Bike Information” allows users to see all the information of each bike they're renting, including bike type, renting time, the amount to be paid up to now, and bike status (e.g current battery percentage of e-bike).

3.6.3. Actors

Customer

3.6.4. Preconditions

Successfully login

3.6.5. Basic Flow of Events

Step 1: Customer click on View Rented Bike tab

Step 2: Software check the user's rental code and collect the bike information

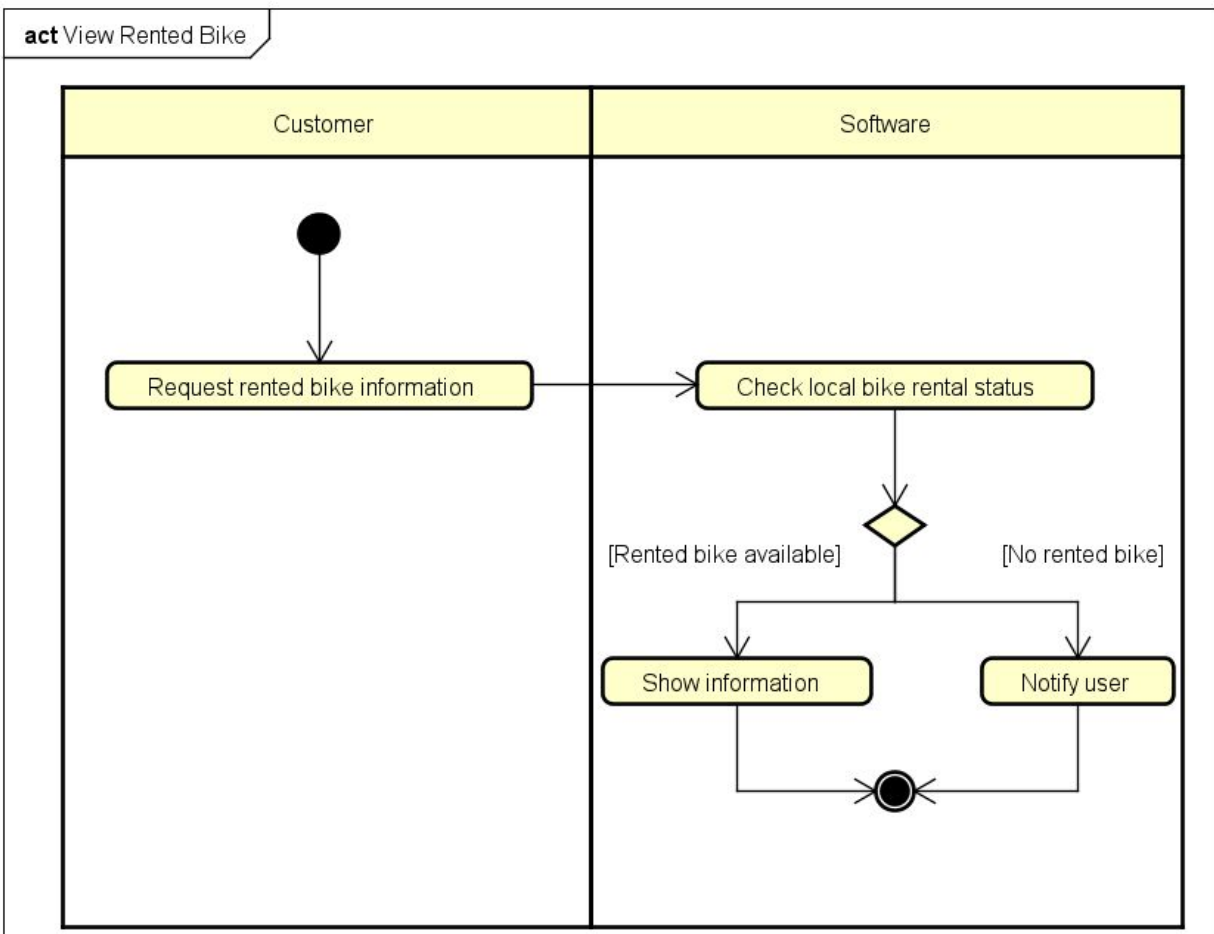
Step 3: Software display the view of bike information currently rented by the user

3.6.6. Alternative flows

Table N-Alternative flows of events for UC View Bike Information

No	Location	Condition	Action	Resume location
1.	At Step 2	in case of failed querying, notify to the user, end of use case	<ul style="list-style-type: none">▪ The software notify for the error to the user	The use case ends

3.6.7. Activity diagrams



3.6.8. Input data

Table A- Input data for View Rented Bike Information

No	Data fields	Description	Mandatory	Valid condition	Example
1.	Rental Code	Stored rental code on rented bike	Yes		

3.6.9. Output data

Table B- Output data for View Rented Bike Information

No	Data fields	Description	Display format	Example
1	Bike type	Type	Code	1/2/3

2	Renting time	Renting time	Time	24m42s
3	Deposited amount	Money	Currency	200000 VND
4	Current amount to be paid	Money	Currency	15000 VND

3.6.10. Postconditions

None

3.7. Use case specification for “View All Bikes In Station”

3.7.1. Use case code

UC007

3.7.2. Brief Description

UC “View All Bike in Station” allows the user to see all the bike in a specified station including: the bike category, the battery of e-bikes and availability status

3.7.3. Actors

- Customer

3.7.4. Preconditions

Successfully login

3.7.5. Basic Flow of Events

Step 1: Customer click on a dock to see information of bikes in that station

Step 2: Software check the station code from that click and collect the information in database

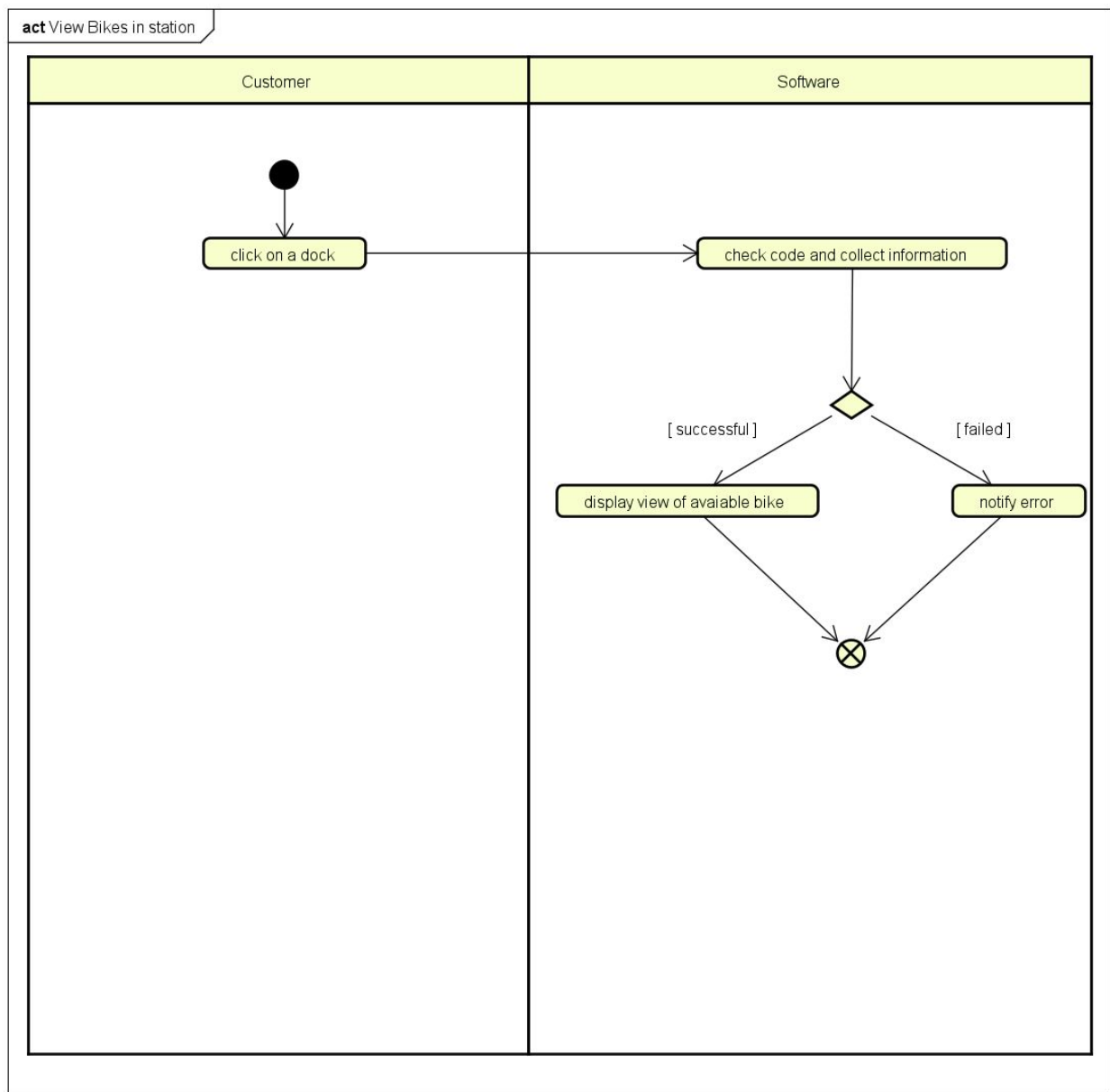
Step 3: Software display the view of available bike in a list/table for customer to choose

3.7.6. Alternative flows

Table N-Alternative flows of events for UC View Bike Information

No	Location	Condition	Action	Resume location
1.	At Step 2	in case of failed querying, notify to the user, end of use case	<ul style="list-style-type: none">▪ The software notify for the error to the user	The use case ends

3.7.7. Activity diagrams



3.7.8. Input data

Table A- Input data for View Bike In Station

No	Data fields	Description	Mandatory	Valid condition	Example
1.	Station code	Choose by clicking	Yes		Station1

3.7.9. Output data

Table B- Output data for View Bike In Station

No	Data fields	Description	Display format	Example
1	Type	Bike's type	<ul style="list-style-type: none">• Word	Standard
2	Battery	Current amount of battery (e-bikes)	<ul style="list-style-type: none">• Percentage	58%
3	Status	Bike's availability	<ul style="list-style-type: none">• Word	In Used

3.7.10. Postconditions

Being able to choose and view details

3.8. Use case specification for “View Bike Details”

3.8.1. Use case code

UC008

3.8.2. Brief Description

UC “View Bike Details” allows the user to see all detailed information of each bike in a specified station including: the battery of e-bikes and the maximum amount of time that customer is able to use.

3.8.3. Actors

- Customer

3.8.4. Preconditions

Selected a bike from the bike list in each docking station

3.8.5. Basic Flow of Events

Step 1: Customer click on a bike to see detailed information

Step 2: Software get the detailed information from database query

Step 3: Software display the detailed information to the customer and provider renting bike option if possible

3.8.6. Alternative flows

Table N-Alternative flows of events for UC View Bike Details

No	Location	Condition	Action	Resume location
1.	At Step 2	in case of failed querying, notify to the user, end of use case	<ul style="list-style-type: none">The software notify for the error to the user	The use case ends

3.8.7. Activity diagrams

3.8.8. Input data

Table A- Input data for View Bike Details

No	Data fields	Description	Mandatory	Valid condition	Example
1.	Bike code	Choose by clicking	Yes		Bike1

3.8.9. Output data

Table B- Output data for View Bike Details

No	Data fields	Description	Display format	Example
1	Type	Bike's type	<ul style="list-style-type: none">Word	Standard
2	Battery	Current amount of battery (e-bikes)	<ul style="list-style-type: none">Percentage	58%

3	Color	Bike's color	<ul style="list-style-type: none"> Color 	Blue
4	Barcode	Bike's barcode	<ul style="list-style-type: none"> Numbers 	#12424
5	Value	Bike's value	<ul style="list-style-type: none"> Numbers 	40000 VND
6	Deposit Charges	All of bike's different deposit charges	<ul style="list-style-type: none"> Numbers 	3000 VND

3.8.10. Postconditions

Being able to choose renting option

3.9. Use case specification for “View All Dock Stations”

3.9.1. Use case code

UC009

3.9.2. Brief Description

UC “View All Dock Station” allows the user to see all the available dock stations along with their details like location, area size and number of bikes available in each dock

3.9.3. Actors

- Customer

3.9.4. Preconditions

Successfully login

3.9.5. Basic Flow of Events

Step 1: Customer start the app

Step 2: Software check the collect the information in database

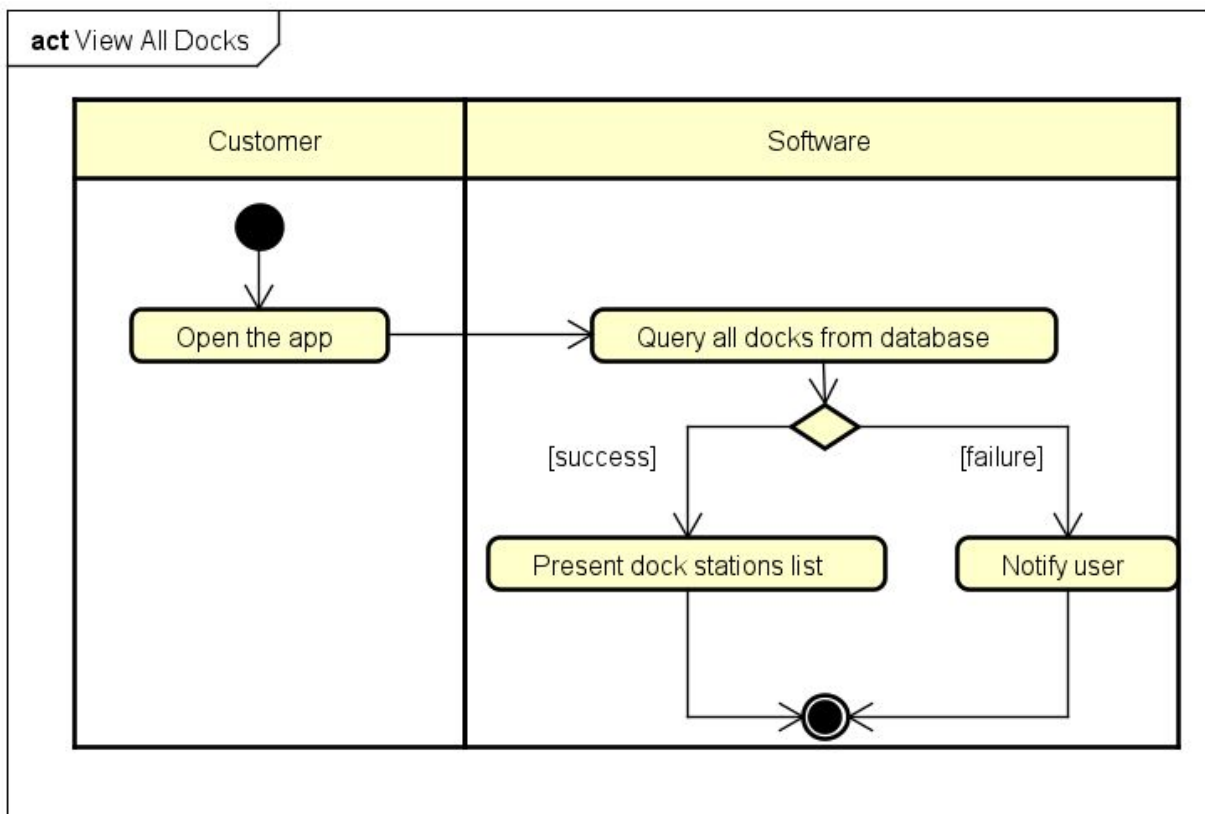
Step 3: Software display the list of available dock stations in a list for customer to choose

3.9.6. Alternative flows

Table N-Alternative flows of events for UC View All Dock Station

No	Location	Condition	Action	Resume location
1.	At Step 2	in case of failed querying, notify to the user, end of use case	<ul style="list-style-type: none">The software notify for the error to the user	The use case ends

3.9.7. Activity diagrams



3.9.8. Input data

None

3.9.9. Output data

Table B- Output data for View Bike In Station

No	Data fields	Description	Display format	Example
1	Name	Dock's name	• Words	Thanh Xuan
2	Address	Dock's address	• Numbers and words	Hanoi
3	Area	Dock's area size	• Number	252x252
4	Available bike	Number of available bikes in dock	• Number	3//30

3.9.10. Postconditions

Being able to choose a bike

4. Supplementary specification

4.1. Functionality

- EcoBikeRental System allows Customers to use the System for their need to rent bikes.
- EcoBikeRental System allows Customers to choose credit cards from different banks to pay for their renting charge.
- Platform of using: Website

4.2. Usability

- The software doesn't require any specific skill to use and is appropriate for all ages.

- The GUI is totally clear and responsive for web app or mobile app so that the users don't have any problem in seeing the desired result

4.3. Reliability

- Mean time between failures is roughly 1 year
- Mean time to repair is 24 hours
- There are critical errors such as database crash or heavy traffic

4.4. Performance

- The average response time is 3 second after users perform an action to a software feature. There are notifications in case the system has errors or crashes.

4.5. Supportability

- The software can be accessed and installed only through android system/ website

4.6. Other requirements

None