HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

School of Information and communications technology

Software Requirement Specification

Version 1.0

EcoBikeRental (EBR)

Subject: ITSS Software Development

Group 03

Trịnh Thu Hải - 20184255

Nguyễn Huy Hoàng - 20184265

Bùi Thanh Tùng - 20184324

Hanoi, Oct 2021

Table of contents

1 Introduction

1.1 Objective

The objective of this document is to present the detailed descriptions for the EcoBikeRental application. It will define and describe the requirements of the application and spell out its functions and constraints. purpose and the features of the system, user group and their usable function at run time. The intended audience of this document includes the stakeholders and the developers of the application.

1.2 Scope

This software will be an EcoBikeRental (EBR) application for customers of Ecopark's bike renting service. The software's goal is to facilitate the hourly bike rental service by providing the means to automate the renting and charging process, which helps eliminate the need for human workers. By fully atomizing the key steps in the rental service, the operational cost of the bike rental service is reduced and the chances of human-made errors are minimized.

More precisely, this software is designed to help customers navigate the many docking stations to find a bicycle of choice among them as well as returning a bicycle one to the nearest station when they finish. Additionally, it also tracks the usage time and supports the customer to make online payment accordingly. The customer initiates the renting and returning process by requesting through their application using the identifier of the bike. The application will send control signals to the locker on the bike to lock and unlock it accordingly. The system offers three kinds of bicycles, namely standard bike, twin bike and standard e-bike, which have different rental costs. The payment method supported is credit card by linking to Interbank.

1.3 Glossary

Term	**Definition**
Barcode	A machine-readable code in the form of numbers and a pattern of parallel lines of varying widths, printed on lock for identifying bikes

Customer	Users of the EBR
Docking station	Place where customers go to rent bikes and return bikes
Locker	A device on each bicycle which contains a barcode to identify that bicycle. It is unlocked upon renting the bicycle and locked again upon returning it.
Interbank	A bank to pay for transactions made by the customer
Software Requirements 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.
Stakeholder	Any person with an interest in the project who is not a developer
Standard bike	One of the three kinds of bicycle available for rent, which has 01 saddle, 01 pedal, and 01 rear seat in the back
Standard e-bike	One of the three kinds of bicycle available for rent, which is built like a standard bike and has an integrated electric motor for assist propulsion and rental fee costs 1.5 times more expensive than the fee of standard bike
Twin bike	One of the three kinds of bicycle available for rent, which has 02 saddles, 02 pedal, and 01 rear seat with no integrated electric motor with rental fee costs 1.5 times more expensive than the fee of standard bike

1.4 References

IEEE. ISO/IEC/IEEE 29148:2011 ISO/IEC/IEEE International Standard - Systems and software engineering -- Life cycle processes --Requirements engineering. IEEE Computer Society, 2011.

2 Overall Description

2.1 Actors

There are three main actors in the system: Customer, Interbank and Locker. Each contributes in different parts of the system.

2.2 Use case diagrams

The project comprises three main use cases: “Check dock”, “Rent bike” and “Return bike”. Within this very section a brief summary of each use case will be present. Note that the above list has been sorted, in chronological order, which the creators of this document have envisioned, for the customer to go through in their pursuit of a ride in the park.

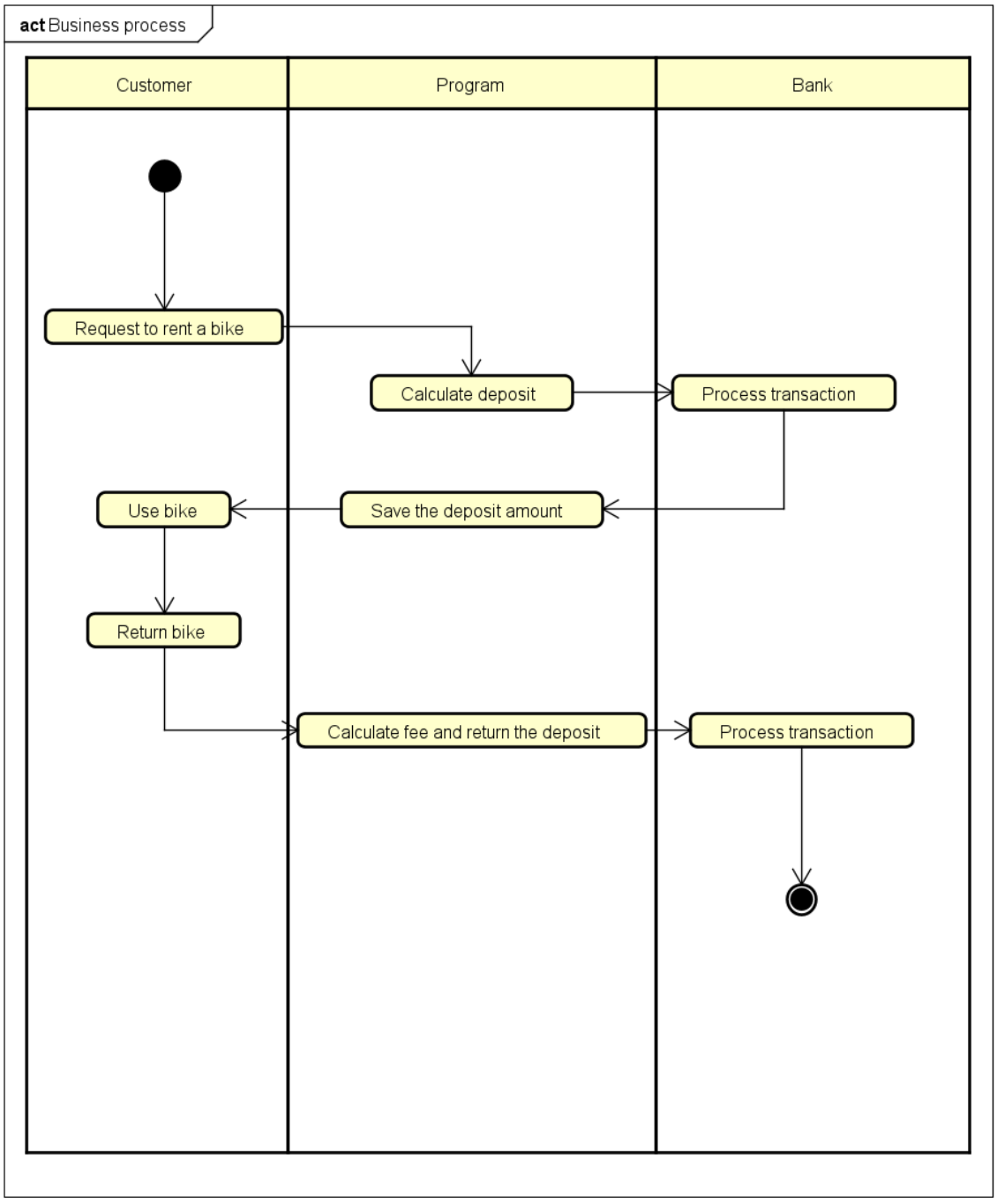
First is the “View dock” use case. The customer can choose a dock from a list of docks shown on the screen instead of a map. They can check the dock’s information, including the name, address, dock area, the number of available bikes, the number of empty docking points, distance, and walking time from the customer’s location to this dock. The detailed information of available bikes in a dock is also viewable, such as battery level, type of bike, top speed, rent rate, deposit rate,…

Next is the “Rent bike” use case. The customer can view the information of any bike (mentioned in the View dock use case) that catches their eyes. Afterward, they enter the barcode of the bike they want to rent into the software. After the customer pays the necessary deposit to rent the bike, the Locker will unlock the bike for a ride within EcoPark. If the customer fails to pay for the depositor, if the Inter-bank cannot process the transaction, the customer won’t be able to use the service.

Finally, we have the “Return bike” use case. At any point during their time, the customer can see the amount of money they would have to pay. When the customer wishes to return the bike, they need to push it into an empty locker and close the lock. the system will automatically return the deposit and deduct the amount of money corresponding to the rental period. If this transaction with the Inter-bank fails to go through, the Customer can try again or link to another bank account. If no transaction is made, or if the Customer doesn’t have enough money in their balance to pay, a notification will be sent to the EcoPark administration to figure out an alternative payment method.


EcoBikeRental
Search dock
View list of docks
View dock
View Bike Info
View rent
Rent bike
Deposit rent
Pay rent
Return bike
<<include>>
<<include>>
<<include>>
<<include>>
<<include>>
Customer
Interbank
Locker

2.3 Business process



3 Detailed Requirements

Details of the use cases given in the following sections are specified below.

3.1 Use case specification for UC001 - "View bike info"

Use case "View bike info"

1. Use case code

UC001

2. Brief Description

This use case describes the interaction between Customer and EBR when Customer wants to see bike info.

3. Actors

3.1 Customer

4. Preconditions

- Customers need to use the app to scan the barcode on the lock, select a dock, or request to view the bike info when renting a bike.

5. Basic Flow of Events

1. The customer requests to view the bike info.
2. Software checks the bike identity.
3. The software displays the information about the bike.

…

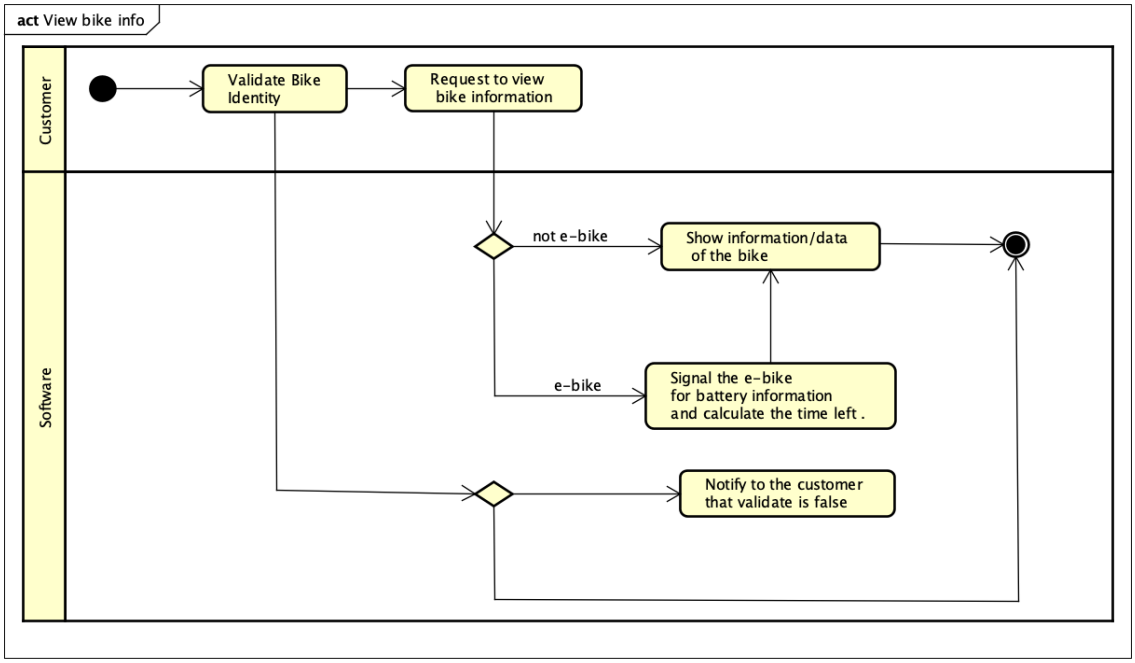
6. Alternative flows

Table N-Alternative flows of events for UC View bike info

No	Location	Condition	Action	Resume location
1.	At Step 2	If the check fails	▪ Notify the customer	Use case ends
2.	At Step 3	If the bike is e-bike	▪ Signal the e-bike for battery information	Step 3

			and calculate the time left	

7. Activity diagrams



8. Input data

Table A-Input data of Lock barcode

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

9. Output data

Table B-Output data of bike information screen

No	Data fields	Description	Display format	Example
1	Bike type			E-bike
2	Deposit amount		-Comma for thousand separator - Positive integer - Right alignment	100,000
3	License plate			29H1-10053
4	Number of saddle			2
5	Number of pedal			2
6	Number of rear seat			1

7	Current battery percentage of electric bicycle		Optional, only display in case bike type is e-bike	100%

10. Postconditions

The customer can know about the information of the bike.

3.2 Use case specification for UC002 - "Return bike"

Use case "Return bike"

1. Use case code

UC002

2. Brief Description

This use case describes the interaction between Customer, Locker, and EBR when Customer wants to return the bike.

3. Actors

3.1 Customer

3.2 Locker

4. Preconditions

- Customers need internet access.

5. Basic Flow of Events

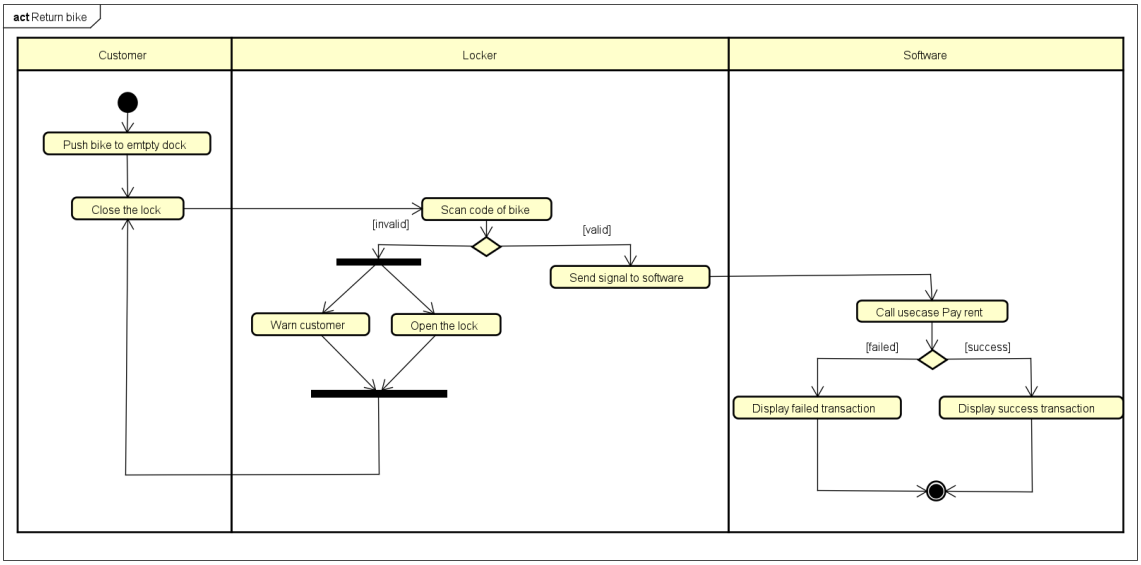
1. Customer pushes bike to an empty lock
2. Customer closes the lock.
3. Locker scans the barcode of the bike.
4. Locker sends signal to the software
5. Software calls use case "Pay rent"
6. Software displays success notification.

6. Alternative flows

Table N-Alternative flows of events for UC Return bike

No	Location	Condition	Action	Resume location
3.	At Step 3	If the locker cannot scan the code.	▪ Locker warns customer by alarming and automatically re-opens the lock	Step 2
4.	At Step 5	If the use case "Pay rent" failed	▪ Software notifies customer that the transaction failed	The use case ends

7. Activity diagrams



8. Input data

Table A-Input data of barcode scanner

No	Data fields	Description	Mandatory	Valid condition	Example
1	Code of bike		Yes		

9. Output data

Table A-Output data of success transaction

No	Data fields	Description	Display format	Example

1	Response code	The result of the transaction	Boolean	True

Table B-Output data of failed transaction

No	Data fields	Description	Display format	Example
1	Response code	The result of the transaction	Boolean	False
2	Reason	The reason explaining the failure of the transaction	String	Transaction failed

10. Postconditions

At the end of this use case, either one of the following cases occurs:
- The transaction is successfully processed and the balance is updated.
- The transaction failed and the message was returned with an error code.

3.3 Use case specification for UC003 - "Rent bike"

Use case "Rent bike"

1. Use case code

UC003

2. Brief Description

This use case describes the interaction between Customer, Locker, and EBR when Customer wants to rent a bike.

3. Actors

3.1 Customer

3.2 Locker

3.3 Inter-bank

4. Preconditions

- Customers need internet access.
- The camera on the EBR device is working.

5. Basic Flow of Events

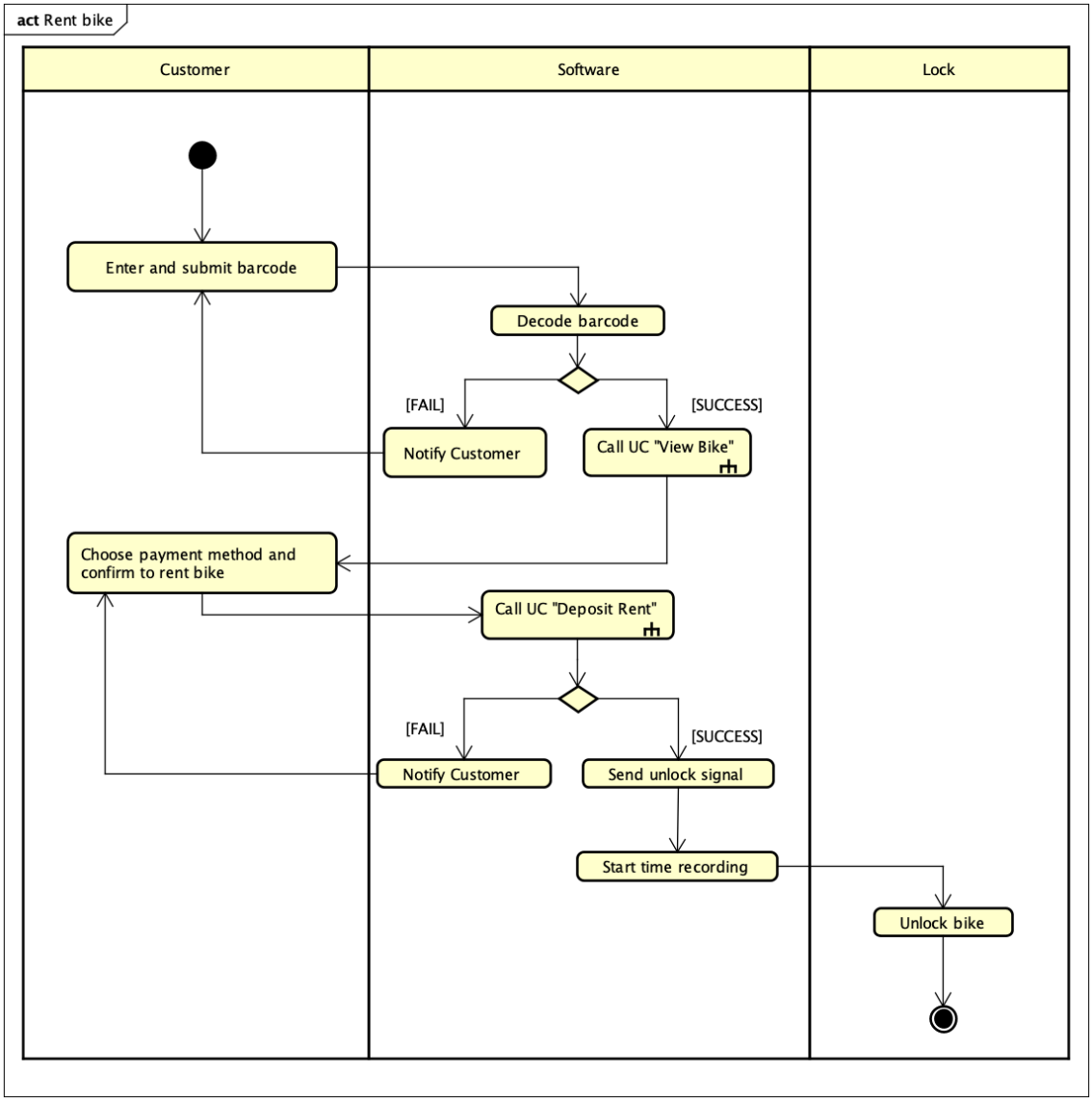
1. Customer opens the scanner and focuses on the lock.
2. Software scans and decodes the barcode on the lock.
3. Calls use case “View bike”
4. Customer chooses the payment method and confirms renting the bike.
5. Call use case “deposit rent”.
6. Software sends an unlock signal to the lock.
7. Software starts the timer.
8. Lock unlocks.

6. Alternative flows

Table N-Alternative flows of events for UC Rent bike

No	Location	Condition	Action	Resume location
5.	At step 2	The barcode is invalid or unclear	▪ Notify the customer to keep the scanner more steady or re-check the bar-code.	Step 1
6.	At step 5	If the use case “Deposit rent” fails	▪ Notify the customer	Step 4

7. Activity diagrams



8. Input data

Table A-Input data of lock barcode

No	Data fields	Description	Mandatory	Valid condition	Example
1	Lock barcode		Yes	The code must exists in database	

Table B-Input data of rental form

No	Data fields	Description	Mandatory	Valid condition	Example

1	License plate	Bike identity	Yes		29M1-75244
2	Deposit	The necessary amount to rent the bike (with currency)	Yes		400.000 VND
3	Payment method	Choose from list	Yes		4221 XXXX – VISA – Luu Duc Thanh

9. Output data

Table B-Output data of …

No	Data fields	Description	Display format	Example	

10. Postconditions

Either one of the following results occurs:

- The customer deposits to rent the bike, the lock is unlocked and the system starts recording the rental.

- The transaction fails, and the customer can retry or edit the payment method to continue.

3.4 Use case specification for UC004 - "Pay rent"

Use case "Pay rent"

1. Use case code

UC004

2. Brief Description

This use case describes the interaction between Customer and Software when customer wants to pay rent.

3. Actors

3.1 Inter-bank

4. Preconditions

- Customers need internet access.

5. Basic Flow of Events

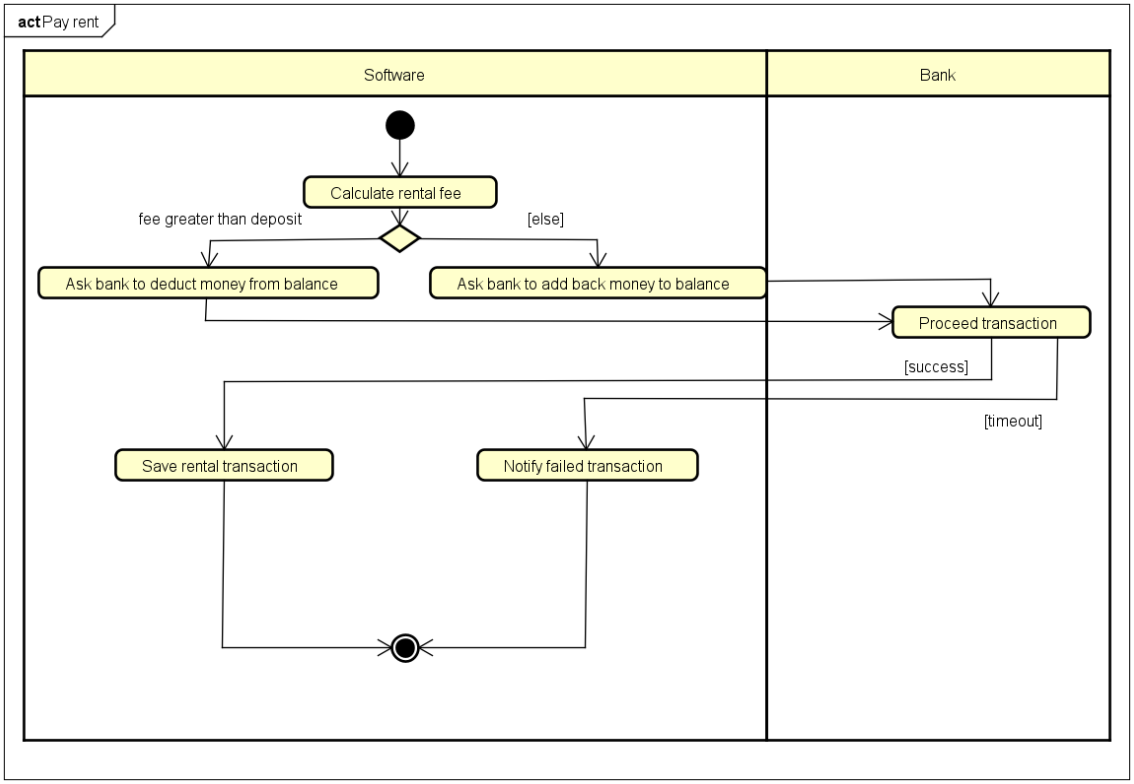
1. Software calculates the rental fee.
2. Software asks the inter-bank to add money to the bank balance.
3. Interbank proceeds the transaction.
4. Software saves the transaction.

6. Alternative flows

Table N-Alternative flows of events for UC Pay rent

No	Location	Condition	Action	Resume location
7.	At Step 1	If rental fee is larger than the deposit money	▪ Software asks inter-bank to deduct money from bank balance	Step 3
8.	At Step 3	If response time from inter-bank exceeds 1 second or transaction fails	▪ Software notifies customer that transaction failed	The use case ends

7. Activity diagrams



8. Input data

No	Data fields	Description	Mandatory	Valid condition	Example

9. Output data

Table B-Output data of success transaction

No	Data fields	Description	Display format	Example
1	Response code	The result of the transaction	Boolean	True

Table B-Output data of failed transaction

No	Data fields	Description	Display format	Example
1	Respons e code	The result of the transaction	Boolean	False
2	Reason	The reason explaining the failure of the transaction	String	Timeout

10. Postconditions

At the end of this use case, either software saves the transaction as well as the balance is updated or the failure is logged and the account balance remains.

3.5 Use case specification for UC005 - "Deposit rent"

Use case "Deposit rent"

1. Use case code

UC005

2. Brief Description

This use case describes the interaction between interbank and Software when a customer returns the bike and therefore triggers automatic payment.

3. Actors

3.1 Inter-bank

4. Preconditions

- Customers need internet access.

5. Basic Flow of Events

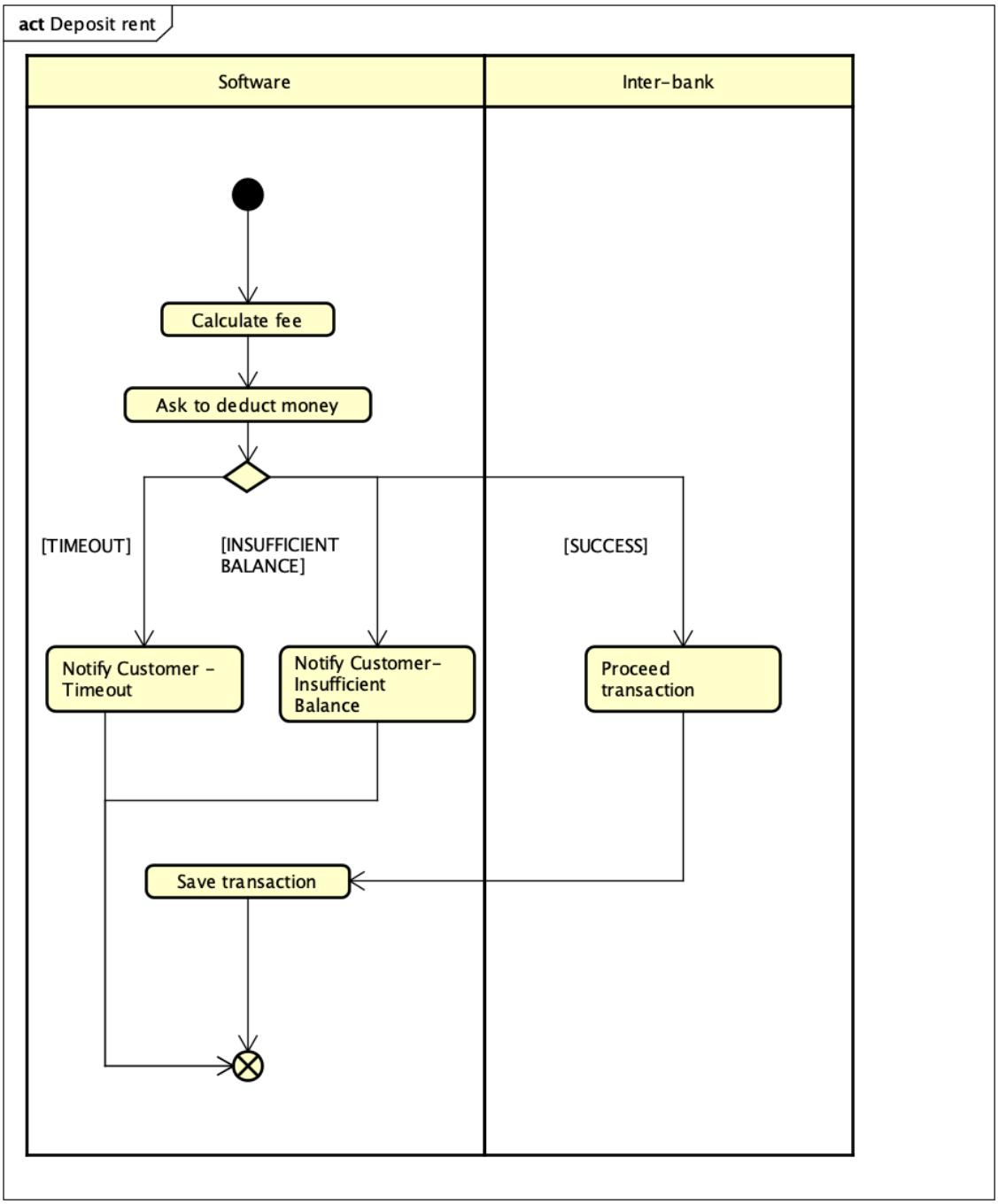
1. Software calculates the rental fee.
2. Software asks the inter-bank to deduct money to the bank balance.
3. Interbank proceeds the transaction.
4. Software saves the transaction.

6. Alternative flows

Table N-Alternative flows of events for UC Deposit rent

No	Location	Condition	Action	Resume location
9.	At Step 3	If response time from inter-bank exceeds 01 second or transaction fails	▪ Software notifies customer that transaction failed	The use case ends
10.	At Step 3	If the balance is not sufficient to pay the fee	▪ Software notifies customer that the balance is insufficient to pay the fee	The use case ends

7. Activity diagrams



8. Input data

Table A-Input data of rental form

No	Data fields	Description	Mandatory	Valid condition	Example
1	License plate	Bike identity	Yes		29M1-75244

2	Deposit	The necessary amount to rent the bike (with currency)	Yes		400.000 VND
3	Payment method	Choose from list	Yes		4221 XXXX – VISA – Luu Duc Thanh

9. Output data

Table B-Output data of success transaction

No	Data fields	Description	Display format	Example
1	Respons e code	The result of the transaction	Boolean	True

Table B-Output data of failed transaction

No	Data fields	Description	Display format	Example
1	Respons e code	The result of the transaction	Boolean	False
2	Reason	The reason explaining the failure of the transaction	String	Not enough balance

10. Postconditions

At the end of this use case, either software saves the transaction as well as the balance is updated or the failure is logged and the account balance remains.

3.6 Use case specification for UC006 - "Check dock"

Use case "Check dock"

1. Use case code

UC006

2. Brief Description

This use case describes the interaction between customer and Software when a customer wants to see dock information.

3. Actors

3.1 Customer

4. Preconditions

- Customers need internet access.

- The camera on the machine running EBR is active.

5. Basic Flow of Events

1. Customer opens the app with a new session, or navigates to the main view.
2. Software calls the api Locate and Query nearby stations.
3. Software display current location and nearby docking stations
4. Customer selects a dock.
5. Software queries the dock as well as the bikes parked there.
6. Software displays the information.

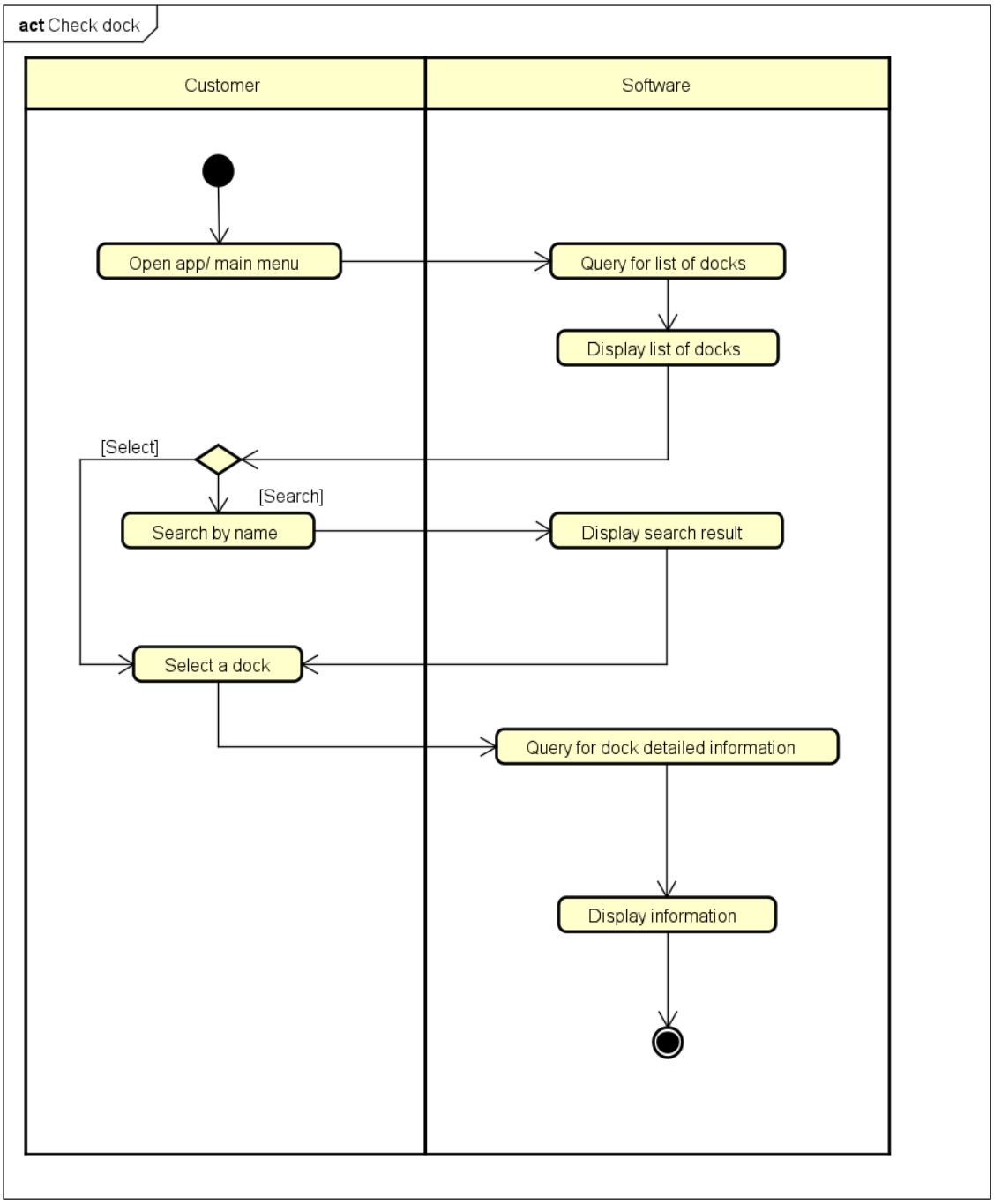
6. Alternative flows

Table N-Alternative flows of events for UC Check dock

No	Location	Condition	Action	Resume location
1	At step 4	If the customer searches the dock by name/address	▪ - Software queries the list of dock stations matched with keyword	Step 4

			▪ - Software display the result list	

7. Activity diagrams



8. Input data

Table A-Input data of lock barcode

No	Data fields	Description	Mandatory	Valid condition	Example
1	Keyword	Name/Address of dock station that customer searches for	No		No.1 Block A

9. Output data

Table B-Output data of result list

No	Data fields	Description	Display format	Example
1	Map	Geographical depiction of current location and accessible docking points	Canvas	
2	Overall Description	Explanation of the view	String	Result: Default
3	Name	Name of dock stations	String	Dock 1A
4	Address	Address of dock stations	String	No.1 Block A
5	No. of e-bike	Available quantity of e-bike in the dock in the moment	- Positive integer - Right aligned	7

6	No. of single-bike	Available quantity of single bike in the dock in the moment	- Positive integer - Right aligned	7
7	No. of single-bike	Available quantity of single bike in the dock in the moment	- Positive integer - Right aligned	7
8	Distance	Distance between current location and the dock	- Comma for thousand separator - Positive integer - Right alignment - Meter unit	1.723

Table B-Output data of dock detailed info

No	Data fields	Description	Display format	Example
1	Name	Name of dock stations	String	Dock 1A
2	Address	Address of dock stations	String	No.1 Block A
3	No. of e-bike	Available quantity of e-bike in the dock in the moment	- Positive integer - Right aligned	7
4	No. of single-bike	Available quantity of single bike in the dock in the moment	- Positive integer - Right aligned	7

5	No. of single-bike	Available quantity of single bike in the dock in the moment	- Positive integer - Right aligned	7
6	Distance	Distance between current location and the dock	- Comma for thousand separator - Positive integer - Right alignment - Meter unit	1.723
7	Bike type			E-bike
8	License plate			29H1-10053
9	Current battery percentage of electric bicycle	Optional, Only displayed if the bike type is e-bike		100%
11	Number of rear seat			1
12	Number of saddle			2
13	Number of pedal			2

10. Postconditions

The customer can find a suitable dock station for either renting a bike or returning a bike

4 Supplementary specification

4.1 Functionality

- While renting, customers can always use the app to view the information about the bike he/she is renting.
- The number of docks is relatively changed when the customer changes the zoom level of the map.
- The system must be able to detect the signal when the customer returns the bike.
- A credit is used to rent one bike only.

4.2 Usability

- Design system for user-friendliness and ease to understand and use.
- Need to have a step-by-step tutorial to follow through for new users.

4.3 Reliability

- The system is expected to operate on average 200 hours without failure.
- The system must notify the customer of system errors as opposed to customer errors.

4.4 Performance

- The system is expected to serve 100 users at the same time without noticeable loss of performance.
- The response time is 1 second at normal or 2 seconds during a peak load if it is not explicitly stated.

4.5 Supportability

- The system can be repaired within 2 hours after any typical failure.

4.6 Other requirements

- The system must be able to prevent cyberattacks related to the credit card API.