HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

School of Information and Communications Technology

Software Requirement Specification

Version 1.0

EcoBikeRental

Subject: Software Development

Group 3:

Nguyễn Gia Thanh -20184195

Dương Văn Bách-20184045

Nguyễn Minh Dũng-20184079

Hanoi, October 2020

Table of contents

1 Introduction

1.1 Objective

This document is to provide information about EcoBikeRental system, users and services that system provided

1.2 Scope

<In this subsection:

(1) Identify the software product(s) to be produced by name
(2) Explain what the software product(s) will, and, if necessary, will not do
(3) Describe the application of the software being specified, including relevant benefits, objectives, and goals
(4) Be consistent with similar statements in higher-level specifications if they exist

This should be an executive-level summary. Do not enumerate the whole requirements list here>

1.2.1 Product name: **EcoBikeRental software**

1.2.2 Explain:

Software is for users to rent and return bikes automatically. EcoBikeRental is a 24/7 platform-independent system which allows novice users to user without any training. Users must have account to enter to system. Software allow user to enter barcode to rent bike and return bike, use credit card for payment, show information of dock and bike.

1.2.3 Application:

Software helps to reduce employees, saves money and time. It satisfies needs on bike rental service especially in Ecopark Township. It is expected to serve 100 users at the same time without noticeable loss of performance and to operate in an average of 200 hours without failure. The system also can be repaired within 2 hours after any typical failure. The response time for the system is 1 second or 2 seconds during a peak load if it is not explicitly stated.

1.3 Glossary

1.4 References

2 Overall Description

2.1 Actors

2.1.1 Customer

2.1.2 Interbank

2.2 Use case diagrams

2.3 Business processes

3 Detailed Requirements

3.1 Use case specification for "Return Bike"

Use case "Return Bike"

1. Use case code:

UC001

2. Brief Description:

This use case describes the interaction between Customer and ECOBIKERENTAL Software when customer wishes to return a bike

3. Actors

3.1 Customer

4. Preconditions

Preconditions of this use case is that customer can return a bike if only if he/she is renting a bike.

5. Basic Flow of Events

Step 1. The customer send request to return bike

Step 2. The ECOBIKERENTAL software prompt a form for entering barcode of bike

Step 3. The customer enters barcode of return bike

Step 4. The ECOBIKERENTAL software check if barcode had been entered is valid

Step 5. The ECOBIKERENTAL software notifies that bike's barcode is a valid barcode

Step 6. The customer submits request to return bike

Step 7. The ECOBIKERENTAL software checks if that bike is valid for the customer to return

Step 8. The ECOBIKERENTAL software calculates amount of money corresponding to the rental period with deposit

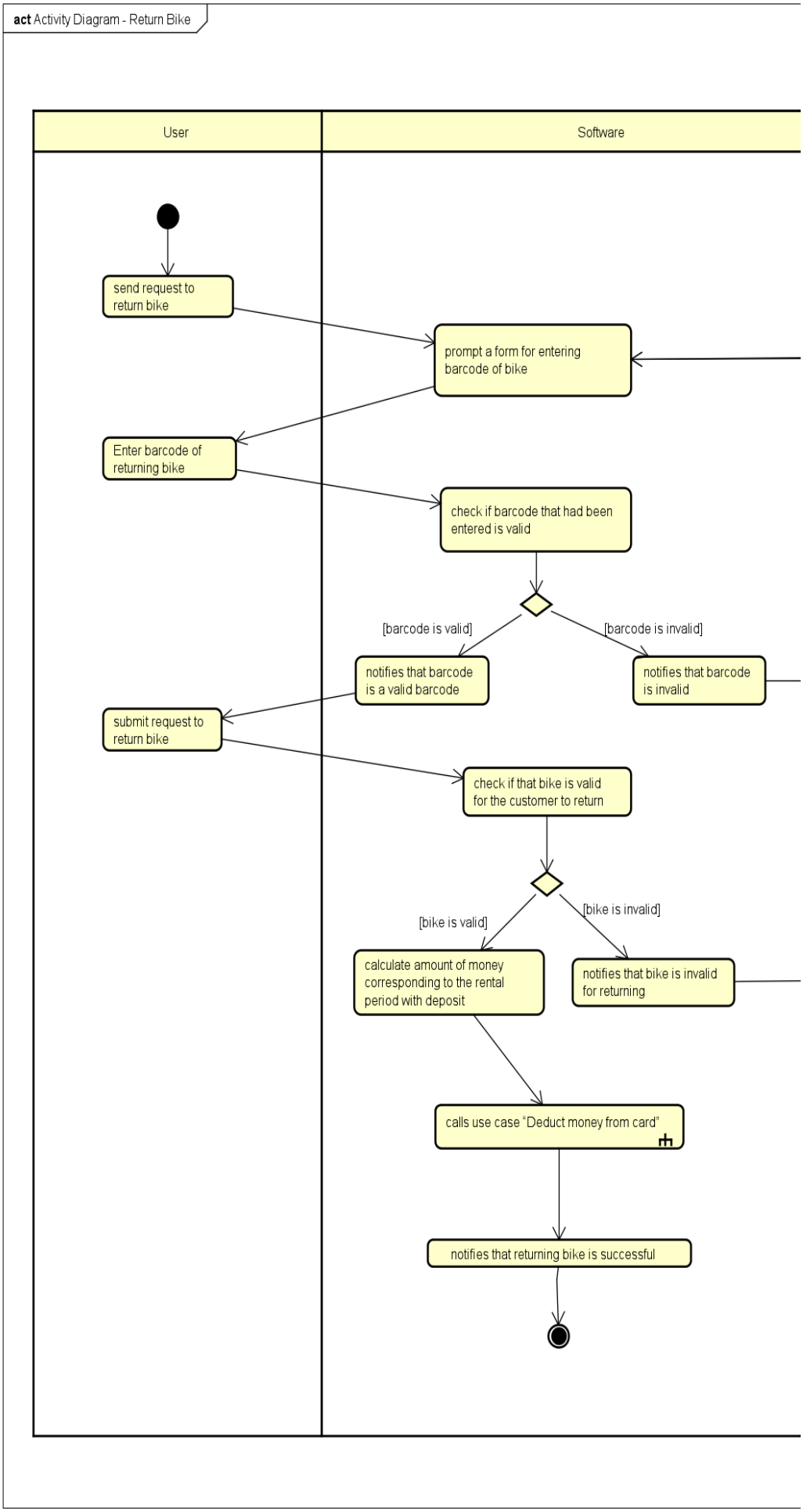
Step 9. The ECOBIKERENTAL software calls use case "Deduct money from card"

Step 10. The ECOBIKERENTAL software notifies that returning bike is successful

6. Alternative flows

No	Location	Condition	Action	Resume location
1	At Step 5	If the barcode had been entered is not a valid bike code	The ECOBIKERENTAL software notifies that barcode is invalid and asks the customer to enter other barcode.	Resumes at Step 2
2	At Step 8	If that bike is not rented by the customer who sent the request returning bike	The ECOBIKERENTAL software notifies that bike is not belong to the customer and asks the customer to enter other barcode.	Resumes at Step 2
3	At Step 8	If that bike had not been rented yet	The ECOBIKERENTAL software notifies that bike is still free, invalid for returning right now and asks the customer to enter other barcode.	Resumes at Step 2

7. Activity diagram


act Activity Diagram - Return Bike
User
Software
send request to return bike
prompt a form for entering barcode of bike
Enter barcode of returning bike
check if barcode that had been entered is valid
[barcode is valid]
[barcode is invalid]
notifies that barcode is a valid barcode
notifies that barcode is invalid
submit request to return bike
check if that bike is valid for the customer to return
[bike is valid]
[bike is invalid]
calculate amount of money corresponding to the rental period with deposit
notifies that bike is invalid for returning
calls use case "Deduct money from card"
notifies that returning bike is successful

8. Input data

Table1 - Input data of returning a bike

No	Data fields	Description	Mandatory	Valid condition	Example
1	Renter Name		Yes		Tran Thi Hang
2	Phone number				0396652104
3	Bike barcode	Each bike has a code to distinguish	Yes	A string contains numbers and letters only	XD12134

9. Output data

Table2 - Output data of returning a bike

No	Data fields	Description	Display format	Example
1	Type	Type of rented bike. Can be 1 of 3 types: * Standard bike * Standard e-bike * Twin bike		Twin bike
2	Bike barcode	Each bike has a code to distinguish	A string contains numbers and letters only	XD12134
3	Start	Time that customer starts renting that bike	hh:mm:ss, dd/mm/yy	09:00:00, 12/9/2020

4	End	Time that customer returns bike	hh:mm:ss, dd/mm/yy	10:10:00, 12/9/2020
5	Rental Period	Amount of time that customer rented that bike	X days, y hours, z minutes	70 minutes
5	Deposit	Deposit amount that the customer paid when renting that bike	* Comma for thousands separator * Positive integer * Right alignment	550,000
6	Subtotal	Amount of money for that rental period		19,000
8	Total	Amount of money that the customer has to pay after subtracting deposit (negative means the customer will receive money back)		- 531,000
9	Currency			VND
10	Name			Tran Thi Hang
11	Phone number			0396652104

10. Postconditions

*3.2 Use case specification for "*Deduct money from card*"*

Use case "Deduct money from card"

1. Use case code:

UC002

2. Brief Description:

This use case describes the interaction between Customer, Interbank and ECOBIKERENTAL Software when the customer wishes to pay for transaction of renting a bike.

3. Actors

3.1 Customer

3.2 Interbank

4. Preconditions

5. Basic Flow of Events

Step 1. The ECOBIKERENTAL software displays the payment screen

Step 2. The customer enters card information and confirm transaction

Step 3. The ECOBIKERENTAL software asks the Interbank to process the transaction

Step 4. The Interbank processes the transaction

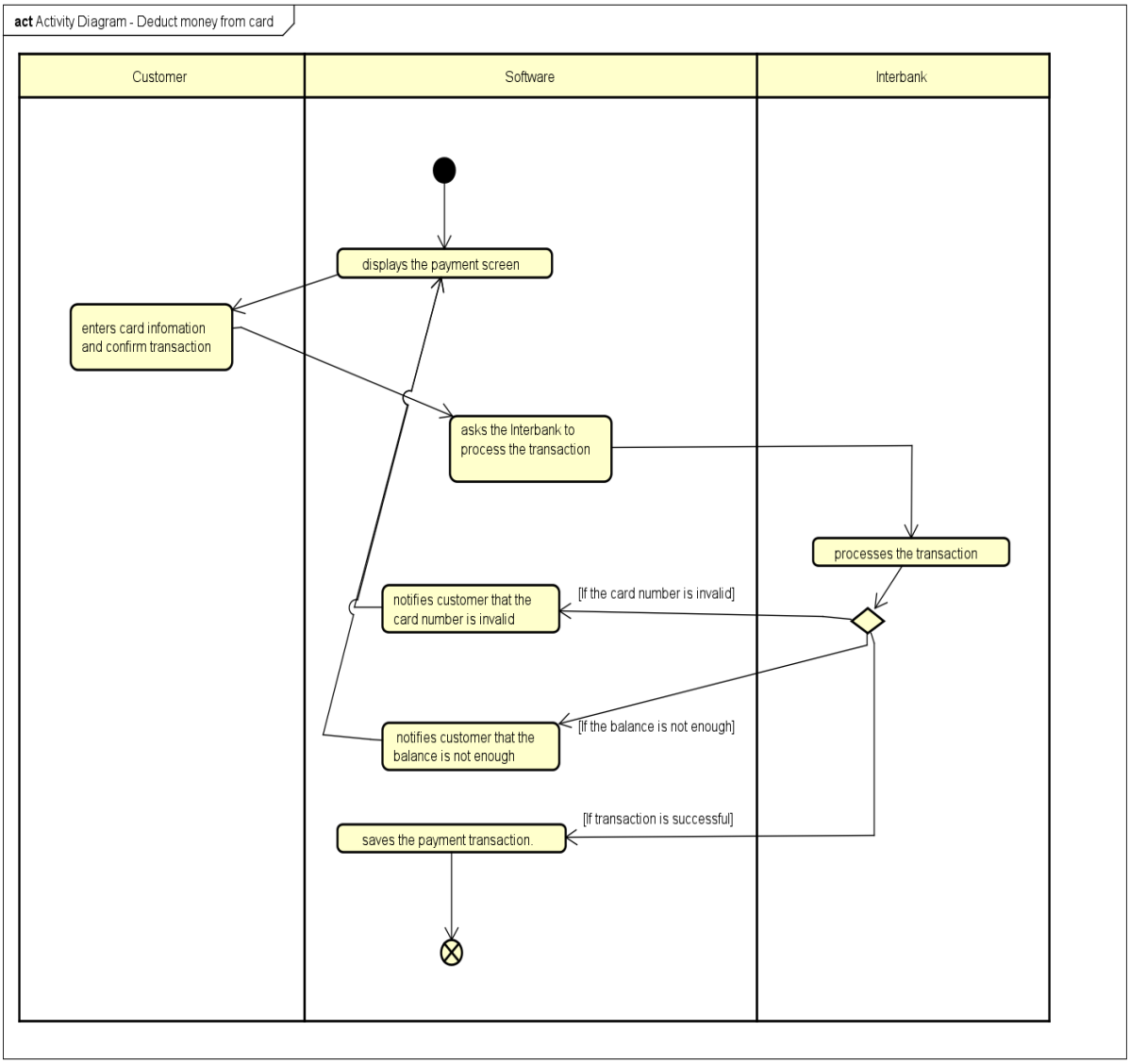
Step 5. The ECOBIKERENTAL software saves the payment transaction.

6. Alternative flows

No	Location	Condition	Action	Resume location
1	At Step 5	If the card number is invalid	* Interbank respond to ECOBIKERENTAL software that card number is invalid. * The ECOBIKERENTAL software notifies customer that the card number is invalid.	Resumes at Step 1

2	At Step 5	If the balance is not enough	* Interbank respond to ECOBIKERENTAL software that the balance is not enough. * The ECOBIKERENTAL software notifies customer that the balance is not enough.	Resumes at Step 1

7. Activity diagram



8. Input data

Table1 - Input data of delivery information

No	Data fields	Description	Mandatory	Valid condition	Example
1	Card holder Name		Yes		Tran Thi Hang
2	Card number		Yes	16 digits	1111111111111111
3	Expiration date		Yes	Expire after 1 day of transaction	31/12/2022
4	Security code		Yes		111111

9. Output data

10. Postconditions

3.3 Use case specification for "Select a dock marker on list"

Use case "Select a dock marker on list"

1. Use case code

UC008

2. Brief description

This use case describes the interaction between Customer and ECOBIKERENTAL Software when customer wishes to select a dock marker on map

3. Actors

3.1 Customer

4. Preconditions

Preconditions of this use case is that customer can select a dock marker if only if he/she searched dock markers.

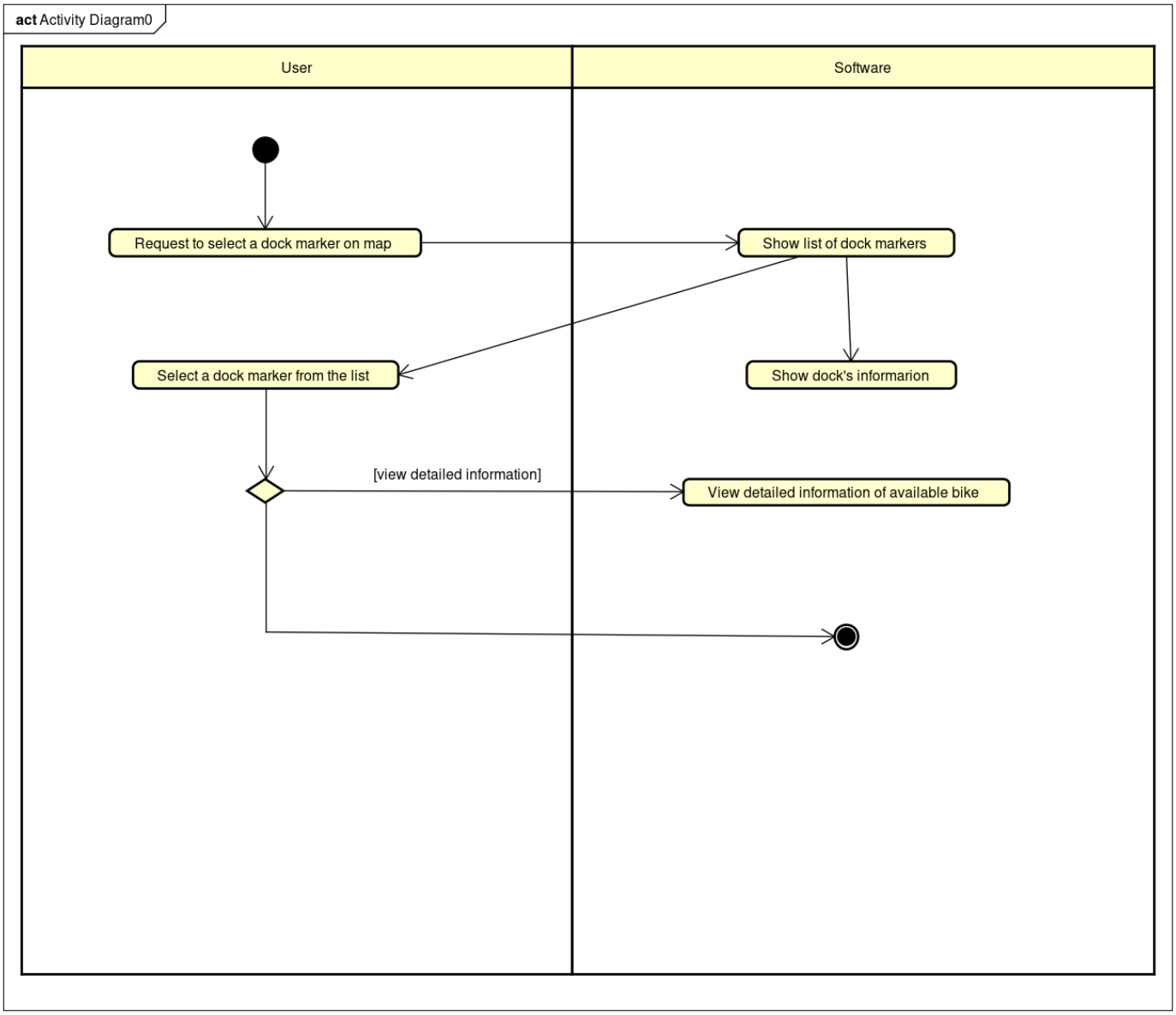
5. Basic flow of events:

Step 1. The customer request to select a dock marker on map
Step 2. The software shows list of dock markers according to search result
Step 3. The software calls use case "View dock's information"
Step 4. The customer selects a dock marker from the list
Step 5. The software redirects to dock marker detail section

6. Alternative flows

No	Location	Condition	Action	Resume location
1	At step 4	User chooses a dock marker	Insert usecase "View of available bike"	Resumes at step 5

7. Activity diagram



8. Input data

9. Output data

Table1- Output data of dock's information

No	Data fields	Description	Display format	Example
1	Name of the dock			EcoBikeRental Hai Ba Trung
2	Address			No.1, Dai Co Viet Street
3	Dock area			Hai Ba Trung

4	Number of available bikes	Number of unrented bikes in the dock		67 bikes are available
5	Number of empty docking points	Number of available slots to return bikes		25 empty docking point
6	Distance	Show the distance from customer's current position to the selected dock		2km away
7	Walking time	Show the calculated time for customer to walk to the selected dock	x hours, y minutes	30 minutes

Table2- Output data of available bikes

No	Data fields	Description	Display format	Example
1	Type	Type of the bike		Standard bike
2	Number of saddle		Number	01
3	Number of pedal		Number	01
4	Number of seat		Number	01
5	Cost coefficient	Cost coefficient fee to rent the bike	Number	1.5
6	Electric motor's battery	Show the battery percentage of e-bikes		78%
7	Time remain	Show the time to use the e-bike before it runs out of battery		45 minutes left

10. Postconditions

*3.4 Use case specification for "*View station information*"*

Use Case "View station information"

1. Use case code

UC004

2. Brief Description

This use case describes the interaction between user and EcoBikeRental software when user wishes to view information of the dock in detail

3. Actors

3.1 User

3.2 Software

4. Preconditions:

User signed in software

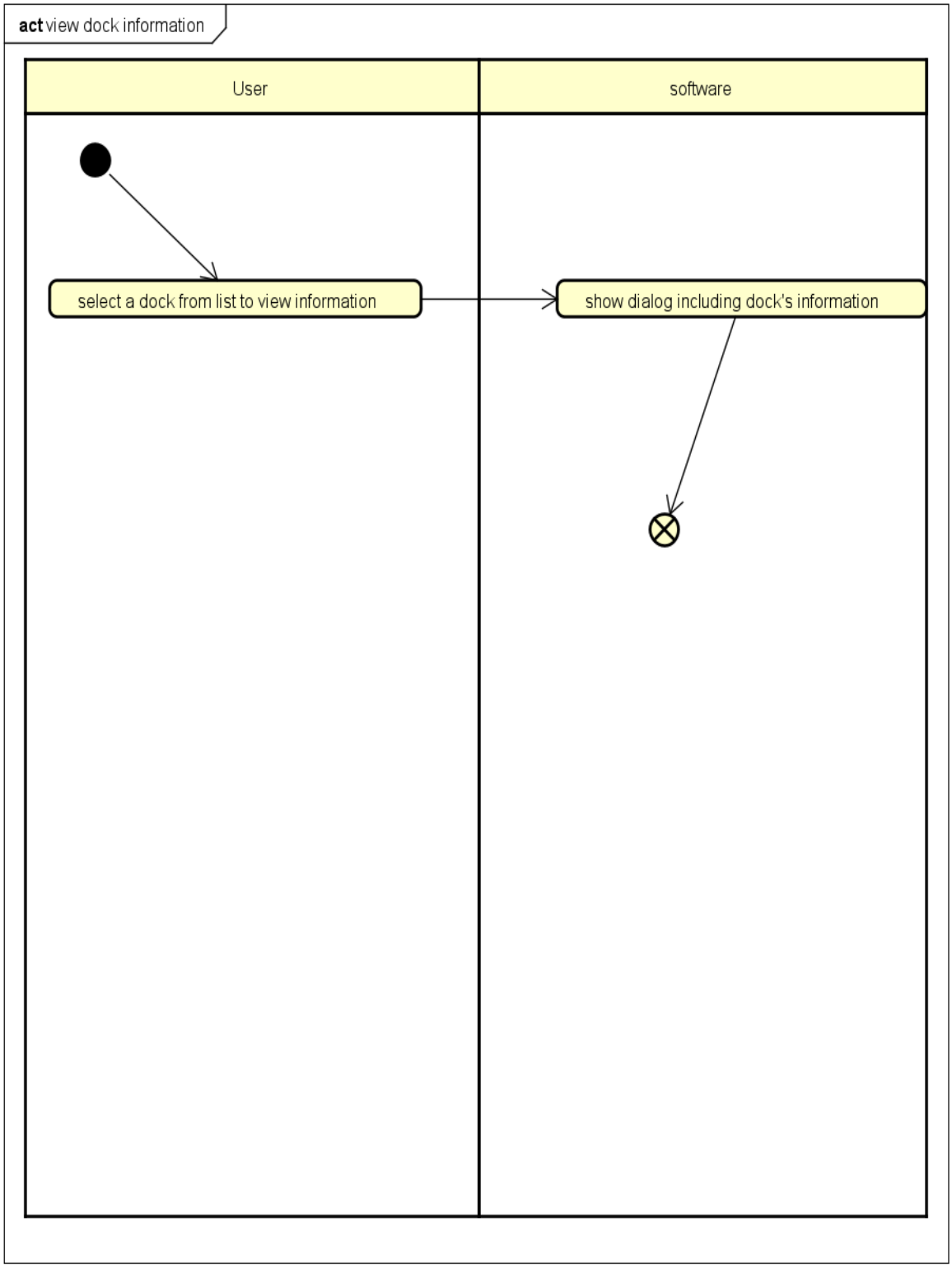
5. Basic Flow of Events

Step 1. User select a dock from list to view information

Step 2. Software show information of the dock and end use case

6. Alternative flows

7. Activity diagram


act view dock information
User
software
select a dock from list to view information
show dialog including dock's information

8. Input data

Table A-Input data of View station information

No	Data fields	Description	Mandatory	Valid condition	Example
1.	Dock id	Id of the dock in database	yes	Id is in database	DOCK01

9. Output data

Table B-Output data of View dock‘s information

No	Data fields	Description	Display format	Example
	Dock ‘s name		Bold text Middle alignment	**Trương Định**
	Address	Address of the dock		1 Trương Định street, Hoang Mai district
	Dock area			700m2
	Number of available bikes			60
	Number of empty docking			50
	Distance	Distance from current user ‘s location to dock		500m
	Walking time	Walking time from user ‘s location to this dock		10 minutes
	List available bike	List available bikes: type and quantity coresponding to these bike		Standard bike: 10

10. Postconditions

User has viewed dock ‘s information

*3.5 Use case specification for "*View bike information*"*

Use Case "View bike information"

1. Use case code

UC005

2. Brief Description

This use case describes the interaction between user and EcoRentalbike software when user wishes to view bike's information

3. Actors

3.1 User

3.2 Software

4. Preconditions

User selected a dock

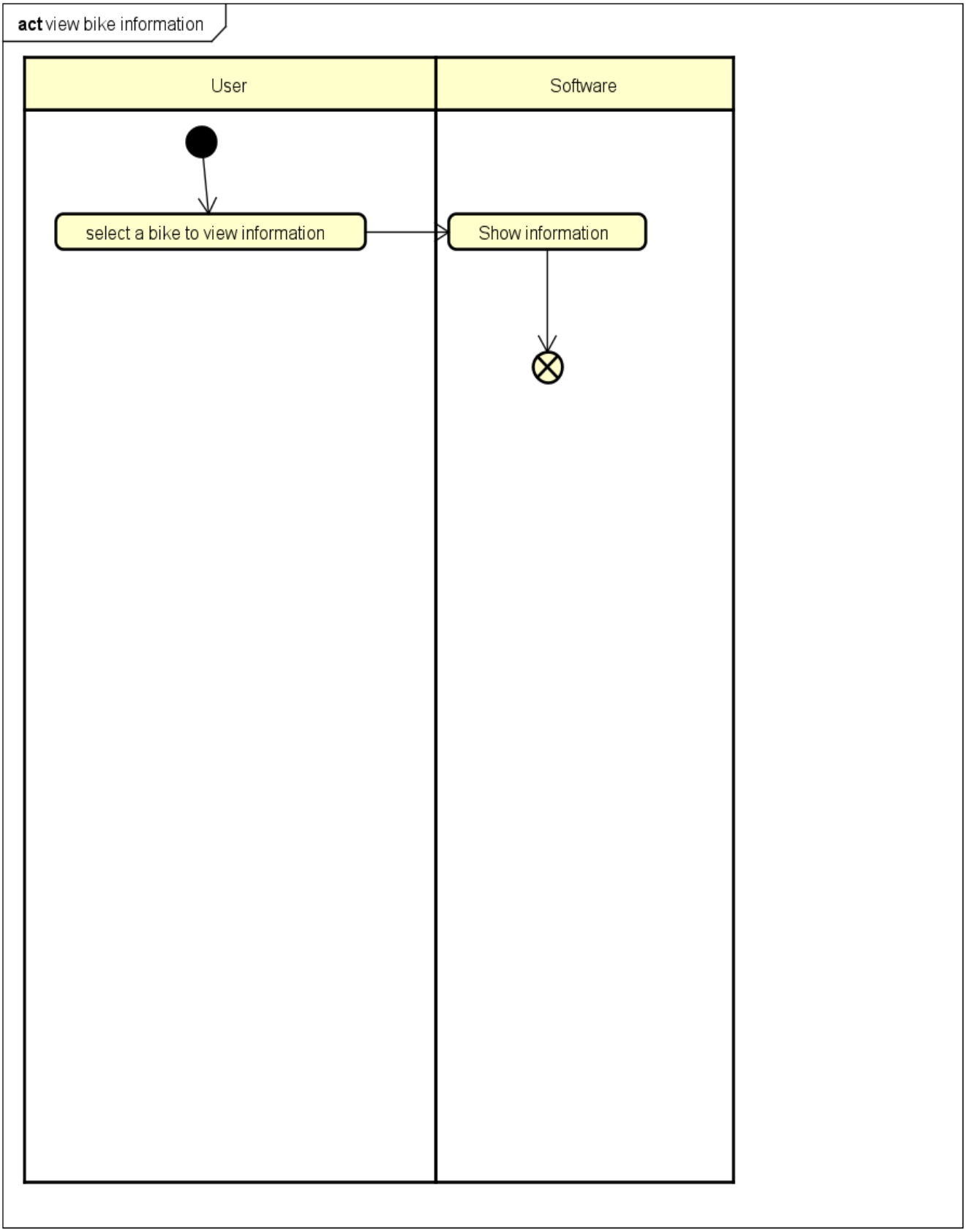
5. Basic Flow of Events

Step 1. User select bike to view information

Step 2. Software show information of that bike and end use case

6. Alternative flows

7. Activity diagram


act view bike information
User
Software
select a bike to view information
Show information

8. Input data

Table A-Input data of View station information

No	Data fields	Description	Mandatory	Valid condition	Example
1.	bike id	Id of the bike in database	yes	Id is in database	STB01

9. Output data

Table B-Output data of View bike information

No	Data fields	Description	Display format	Example
1.	Bike ‘s id			STB01
2.	Bike type			Standard bike
3.	Bike ‘s coeficient price			1
4.	Electric motor ‘s battery percentage	If bike type is e-bike show this information		50%
5.	Estimating time	How much time is left if this bike is e-bike		1 hour 15 minutes

10. Postconditions

User has viewed bike information

*3.6 Use case specification for "*Rent bike*"*

Use Case "Rent Bike"

1. Use case code

UC006

2. Brief Description

In the ECOBIKERENTAL software project, UC "Rent Bike" describes the interaction between user and ECOBIKERENTAL software when the user wishes to rent a bike.

3. Actor

3.1 User

4. Preconditions

There is an active network connection to the Internet

5. Basic Flow of Events

Step 1. The user requests to rent a bike.

Step 2. The ECOBIKERENTAL software displays a form for user to enter the barcode of the bike.

Step 3. The user enters the barcode of the bike he/she wants to rent.

Step 4. The ECOBIKERENTAL software displays the current information of the rented bike.

Step 5. The ECOBIKERENTAL software calls an API to convert the barcode into a rental code.

Step 6. The ECOBIKERENTAL software asks the user to choose a payment method to make transactions by display a list of available options (in this simulation, there is only one option which is paying via credit card).

Step 7. The user chooses a payment method.

Step 8. The ECOBIKERENTAL software calculates the deposit amount.

Step 9. The ECOBIKERENTAL software displays the transaction information.

Step 10. The user confirms the transaction.

Step 11. The ECOBIKERENTAL software calls UC "Deduct money from card".

Step 12. The ECOBIKERENTAL software saves the transaction.

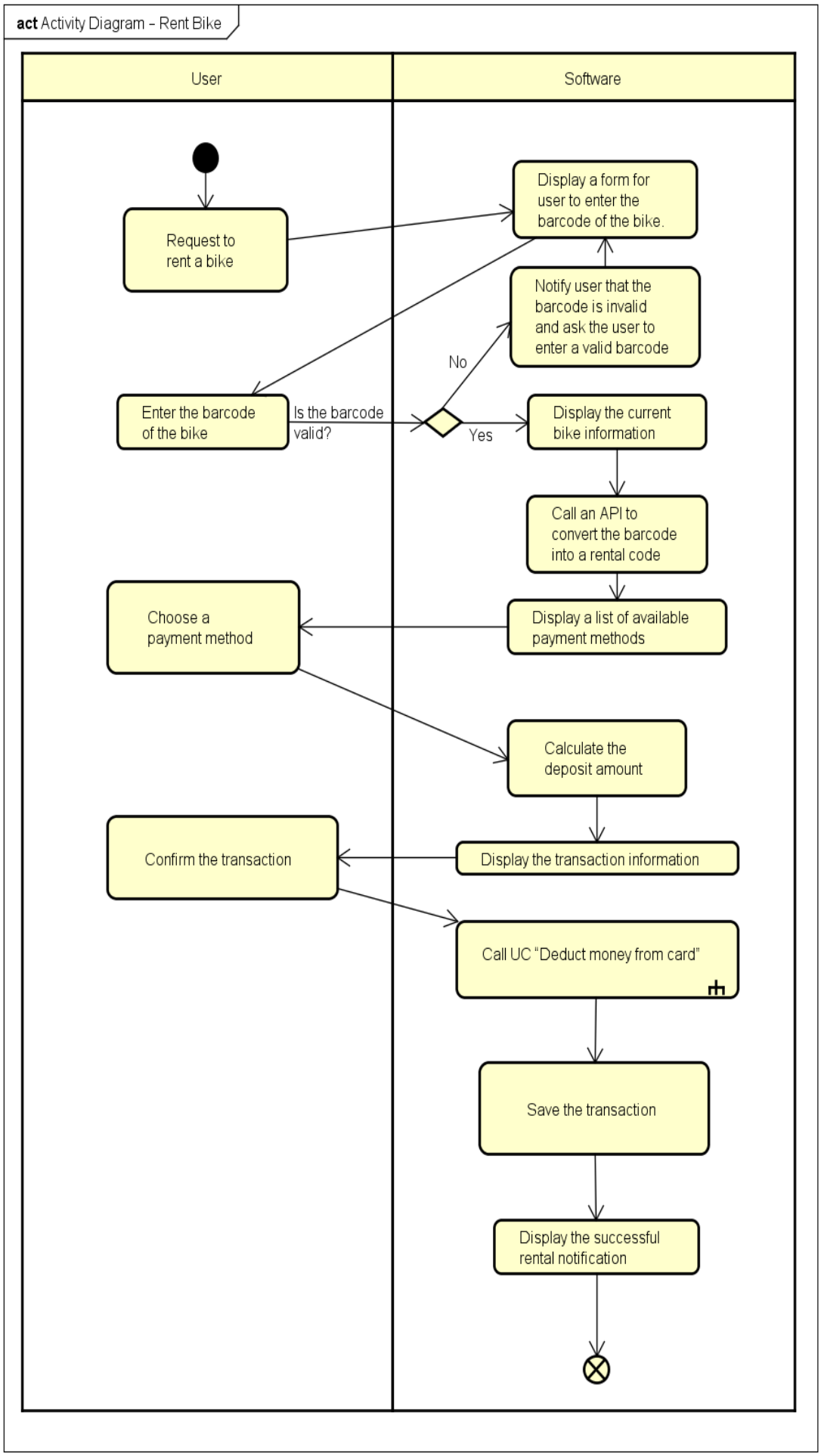
Step 13. The ECOBIKERENTAL software displays the successful rental notification.

6. Alternative Flows

Table 1-Alternative flows of events for UC Rent Bike

No	Location	Condition	Action	Resume location
1	At Step 4	If the barcode entered is invalid	- The ECOBIKERENTAL software notifies user that the barcode is invalid and asks the user to enter a valid barcode.	Resumes at Step 2

7. Activity diagram


act Activity Diagram – Rent Bike
User
Software
Request to rent a bike
Display a form for user to enter the barcode of the bike.
Notify user that the barcode is invalid and ask the user to enter a valid barcode
No
Enter the barcode of the bike
Is the barcode valid?
Yes
Display the current bike information
Call an API to convert the barcode into a rental code
Choose a payment method
Display a list of available payment methods
Calculate the deposit amount
Confirm the transaction
Display the transaction information
Call UC "Deduct money from card"
Save the transaction
Display the successful rental notification

8. Input data

Table 2- Input data of bike barcode

No	Data fields	Description	Mandatory	Valid condition	Example
1	Barcode	Each bike has a different barcode	Yes	A string contains 12-13 digits	1-234567-890128

Table 3- Input data of payment method

No	Data fields	Description	Mandatory	Valid condition	Example
1	Payment Method	Choose from a list	Yes		Credit Card

9. Output data

Table 4-Output data of displaying rented bike's current information

No	Data fields	Description	Display Format	Example
1	Bike Type	Can be one in three following types: - standard bike - standard e-bike - twin bike		Standard Bike
2	Barcode	Each bike has a different barcode.	A string contains 12-13 digits	1-234567-890128
3	License Plate	License plate of the rented bike	A string of uppercase letters and digits	69NO420
4	Current Battery	Show the current battery of the rented bike, only when the bike type is standard e-bike.	Positive double with 2 decimal places. Period for decimal point.	42.69%

Table 5-Output data of displaying transaction information

No	Data fields	Description	Display Format	Example
1	Deposit Amount	The deposit amount that the user has to pay when renting bike.	- Comma for thousands separator - Positive integer - Right alignment	400,000
2	Currency			VND
3	Payment Method			Credit Card
4	Rental Code	Rental code converted from the barcode of the rented bike	A string of digits	1234567890

10. Postconditions

The logs have been updated accordingly.

One barcode corresponding to a bike is now unavailable.

4 Supplementary specification

4.1 Functionality

- In some steps of many use cases, if we have to work with the database and there is an error related to database connecting or database operating, a corresponding message must be displayed in order to distinguish between database-related error and user's error.
- General displaying format:
    - For integer number, comma for thousands separator
    - For number, right alignment
    - For message, left alignment
    - Font: Arial 14, black
    - White background

4.2 Usability

- Functions and features are designed to optimize user's experience and can be operated simply.
- The novice user should not need to be trained in order to use the software.
- Need a detailed guide for user's error so that he/she knows how to navigate when an error is met

4.3 Reliability

- The system should be able to serve a good number of users in discrete time spans.
- In the case of simultaneous users, the system is expected to serve up to 100 requests.
- The system should run smoothly, consecutively, automatically and reliably. Ideally, a time span of more than 200 hours operating without failure is acceptable.

4.4 Performance

- The software should always operate correctly, responsively in any general cases. In some special cases, a slight drop in performance, response time is allowable.

- Implicitly stated, ideally, the response time for any tasks, with a moderate load, within the system is 1 second. But in case of peak load, a response time in the interval of 2 seconds is admissible.

4.5 Supportability

- Any typical failure causing to the system should be quickly noticed and fixed.
- Preferably, after any kind of failure, the system is repaired within the time interval of 2 hours.

4.6 Other requirements

- No other requirements are needed, yet.