

bHANOI 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, 10/2020*

# Table of contents

|   |   |
|---|---|
| Table of contents                                       | 2 |
| 1 Introduction  | 3 |
| 1.1 Objective   | 3 |
| 1.2 Scope   | 3 |
| 1.3 Glossary  | 3 |
| 1.4 References  | 3 |
| 2 Overall Description                                   | 3 |
| 2.1 Actors  | 3 |
| 2.2 Use case diagrams                                   | 3 |
| 2.3 Business processes                                  | 3 |
| 3 Detailed Requirements                                 | 4 |
| 3.1 Use case specification for “Rent Bike”              | 5 |
| 3.2 Use case specification for “Choose Payment Method”  | 6 |
| 3.3 Use case specification for “Deduct money”           | 6 |
| 3.4 Use case specification for “Return Bikes”           | 6 |
| 3.5 Use case specification for “Return Deposit Money”   | 6 |
| 3.6 Use case specification for “View Bikes Information” | 6 |
| 3.7 Use case specification for “View Bikes In Station”  | 6 |
| 4 Supplementary specification                           | 8 |
| 4.1 Functionality                                       | 8 |
| 4.2 Usability   | 8 |
| 4.3 Reliability   | 8 |
| 4.4 Performance   | 8 |
| 4.5 Supportability                                      | 8 |
| 4.6 Other requirements                                  | 8 |

# 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).

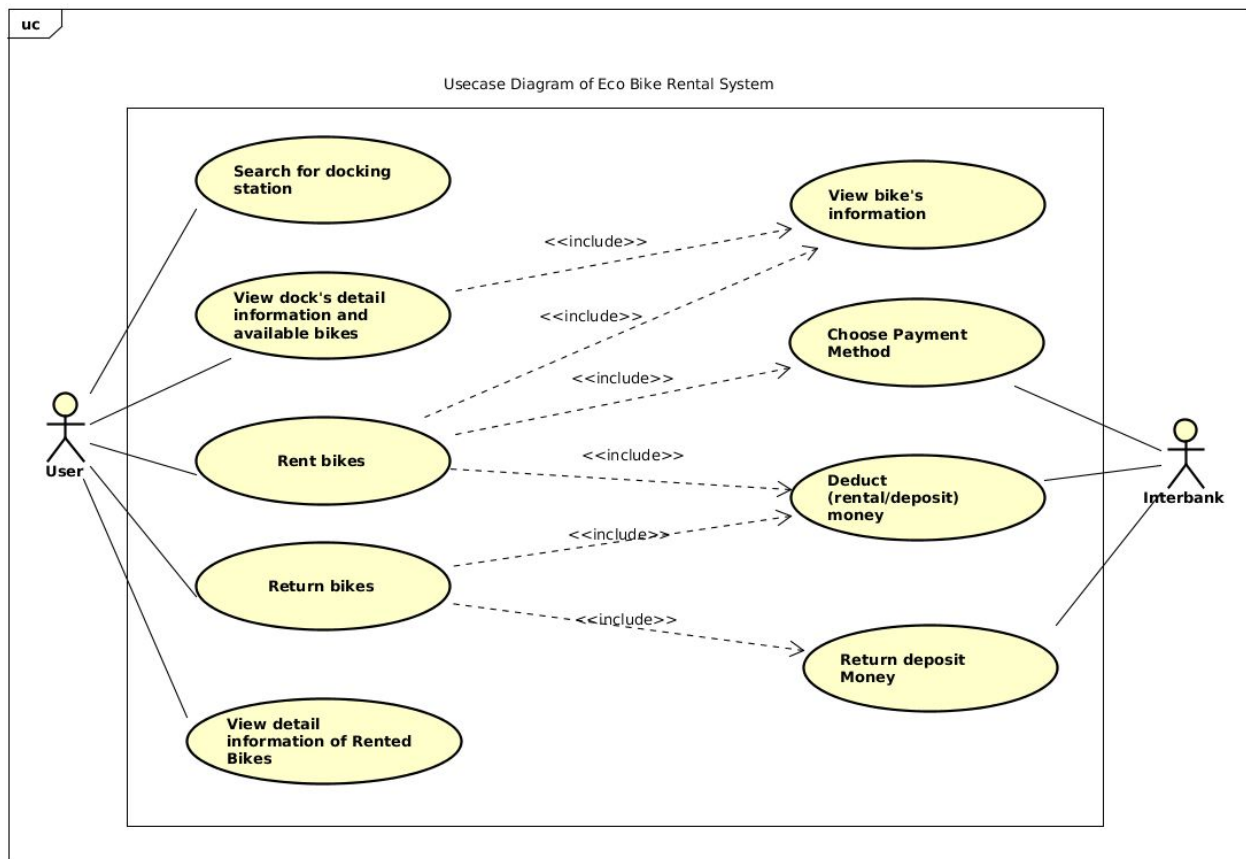
- [Reaves SPMP] "Software Project Management Plan Jacksonville State University of Computing and Information Sciences Web Accessible Alumni Database." Jacksonville State University, 2003.

## 2 Overall Description

### 2.1 Actors

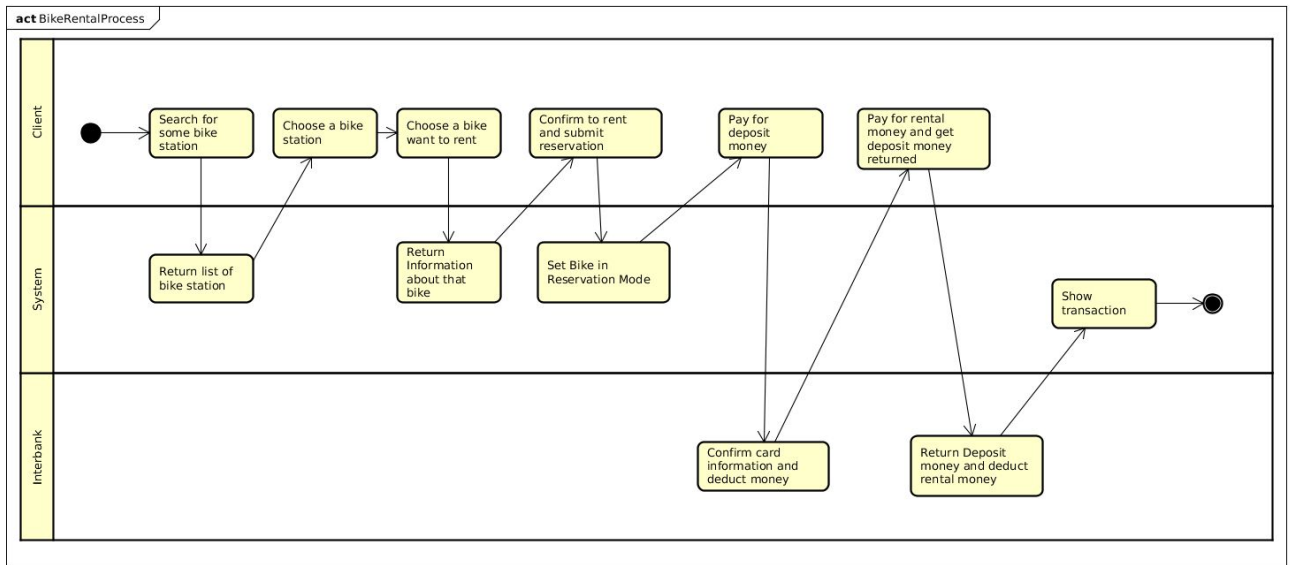
There are 2 main actors in the system: Customer and Interbank.

### 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 “Rent Bike”

#### Use Case “Rent Bike”

##### 1. Use case code

UC001

##### 2. Brief Description

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

##### 3. Actors

Customer

##### 4. Preconditions

None

##### 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

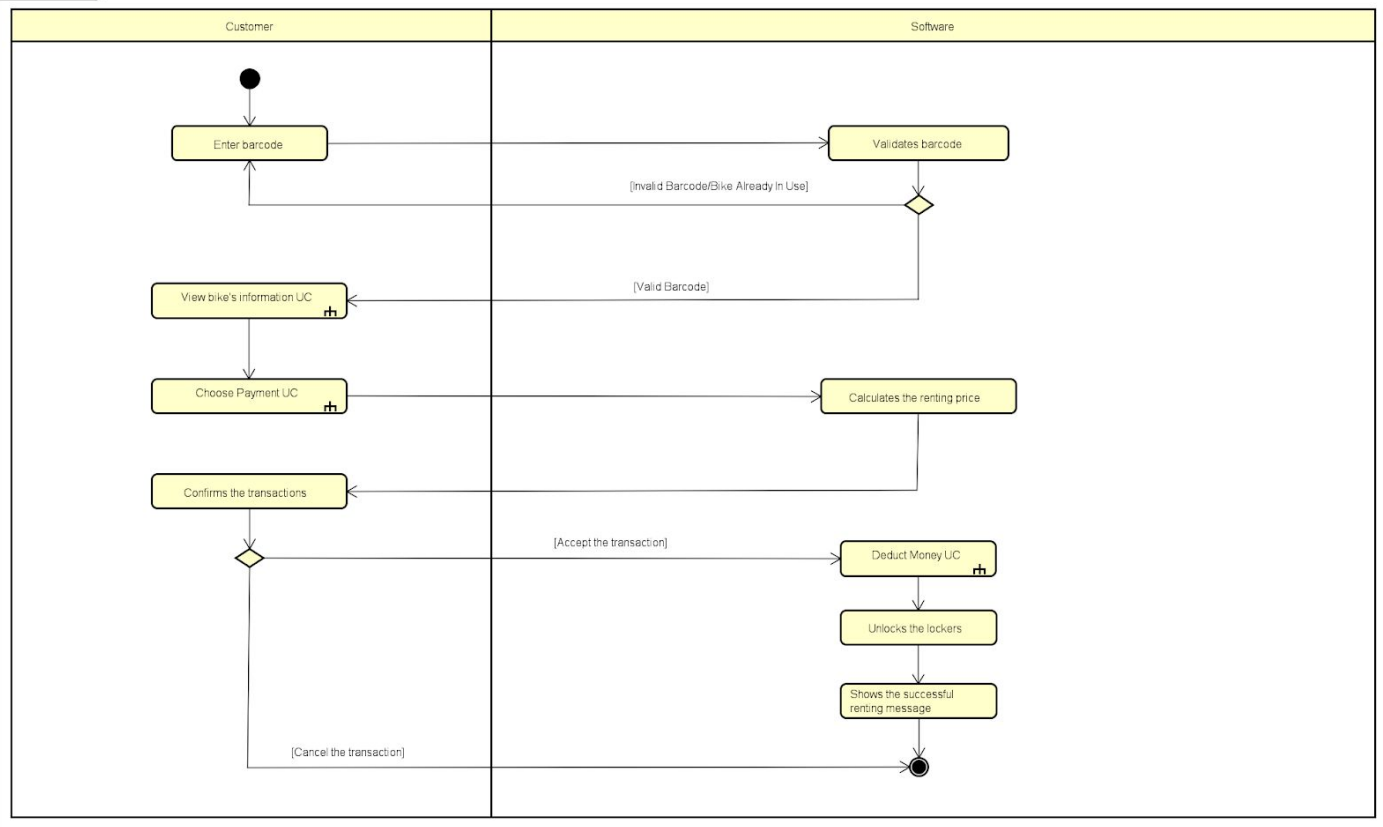
Step 10. Shows the successful renting message

## 6. Alternative flows

*Table N-Alternative flows of events for UC Rent Bike*

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

## 7. Activity diagram



## 8. Input data

Table A-Input data of Barcode

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

## 9. Output data

Table B-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<br>- Positive integer<br>- Right alignment | 430,000          |
| 4 | Barcode of the rented bike |   |  | 4fy7tvi7         |

## 10. Postconditions

### 3.2 Use case specification for “Choose Payment Method”

#### Use Case “Choose Payment Method”

##### 1. Use case code

UC002

##### 2. Brief Description

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

##### 3. Actors

Customer

Interbank

##### 4. Preconditions

Customer had logged in

##### 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

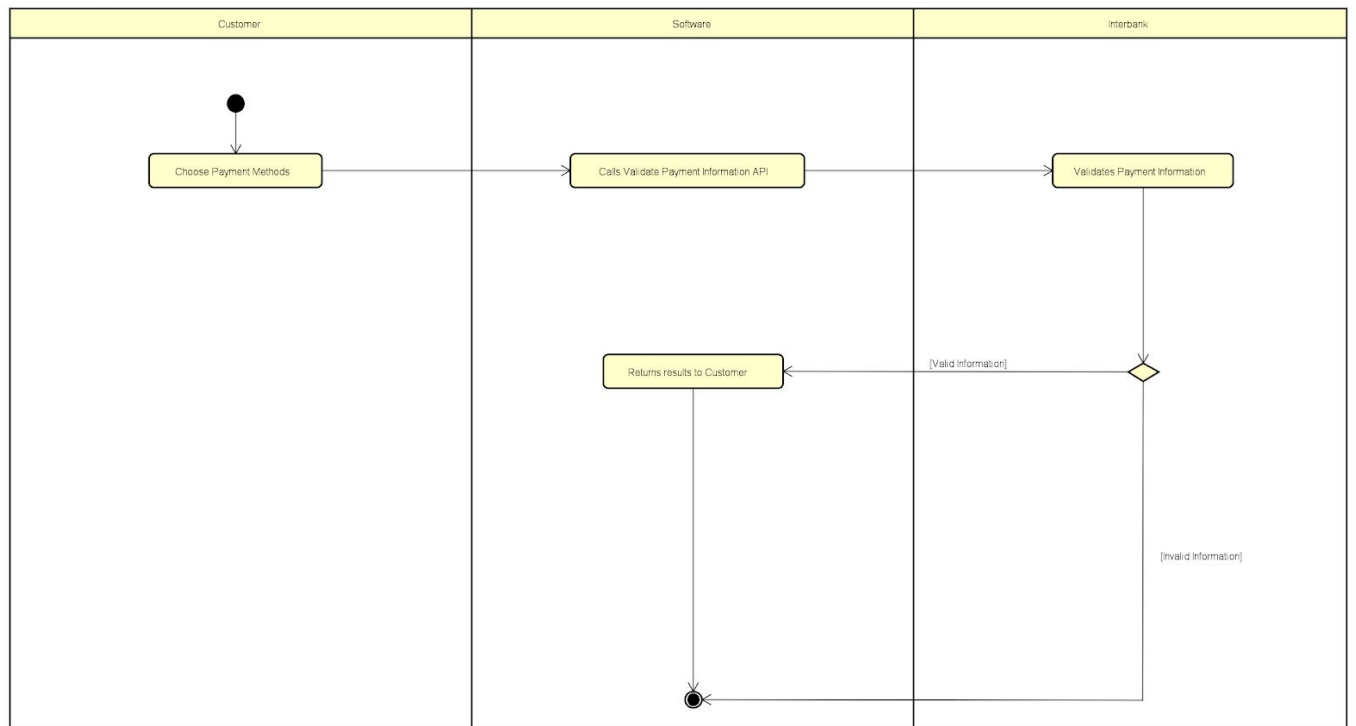
Step 4. Software returns result to customer

## 6. Alternative flows

Table N-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"><li>Sends the failure message to Software</li><li>Notify that wrong payment method to customer</li></ul> | The use case ends |

## 7. Activity diagrams



## 8. Input data

*Table A-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<br>2345 6789 |
| 3. | Cardholder Name  |                    | Yes       |                 | Nguyen Xuan<br>Hoang   |
| 4. | Card expire Date |                    | Yes       | Time in future  | 12/22                  |

## 9. Output data

## 10. Postconditions

None

### 3.3 Use case specification for "Deduct money"

## Use Case "Deduct money"

### 1. Use case code

UC003

### 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. Actors

Customer

### 4. Preconditions

None

### 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

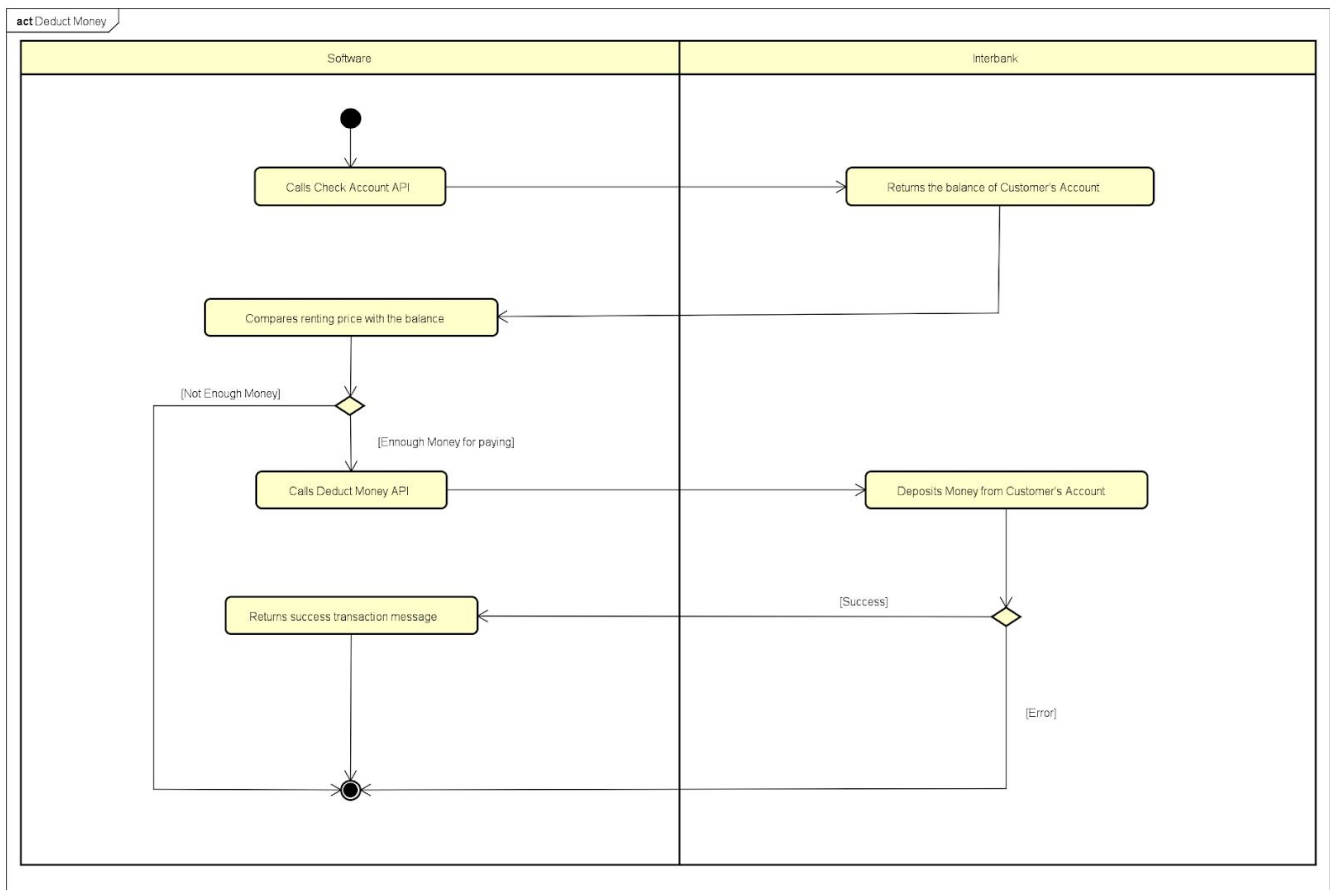
Step 6. The software returns status of transactions

## 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                 | ▪ Sends the failure message to software | The use case ends |

## 7. Activity diagrams



## 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       |                        |         |

## 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 deducted money of    | - 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<br>Nguyen Trung<br>Nghia |

## 10. Postconditions

None

### 3.4 Use case specification for "Return Bikes"

#### Use Case "Return Bikes"

##### 1. Use case code

UC004

##### 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. Actors

Customer

Interbank

##### 4. Preconditions

Customers had already rented bikes in EcoBikeRental System

## 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.

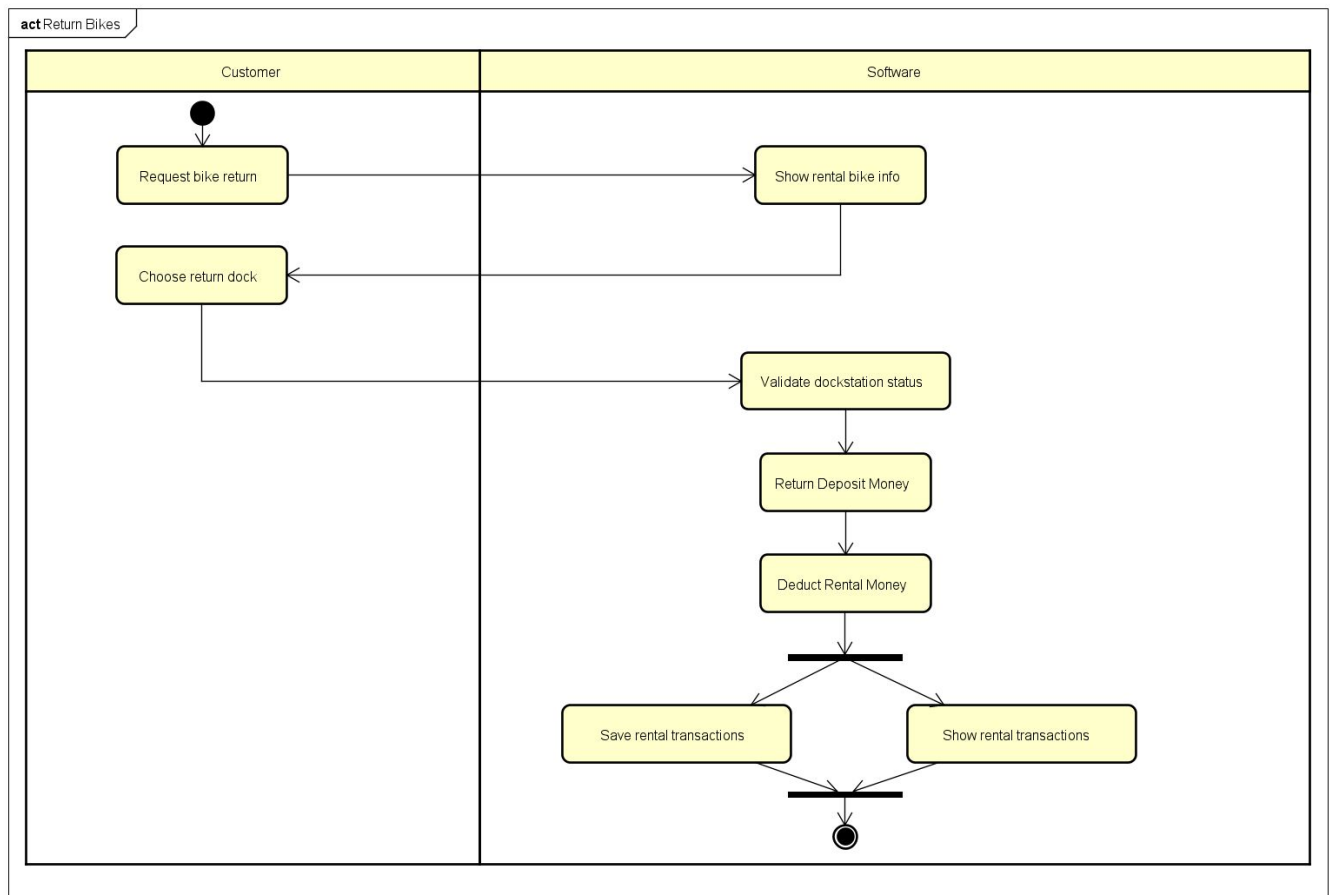
Step 7. The software shows and saves rental transactions

## 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 |

## 7. Activity diagrams



## 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 |

## 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<br>Have positive or negative sign<br>Right alignment | - 246,000           |
| 5 | Deposit Amount          | Total money for depositing the bike                                   | Comma for thousands separator<br>Have positive or negative sign<br>Right alignment | + 200,000           |
| 6 | Currency                |   |  | VND                 |
| 8 | Subtotal                | Total money the customer have to pay, include returned deposit amount | Comma for thousands separator<br>Have positive or negative sign<br>Right alignment | - 46,000            |
| 7 | Transaction status      | The status of the transaction   | Success - Failed   | Success             |

## 10. Postconditions

None

### 3.5 Use case specification for “Return Deposit Money”

#### Use Case “Return Deposit Money”

##### 1. Use case code

UC005

##### 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. Actors**

#### **3.1 Customer**

#### **3.2 Interbank**

### **4. Preconditions**

Customer had already deposit money for rented bike in EcoBikeRental System

### **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

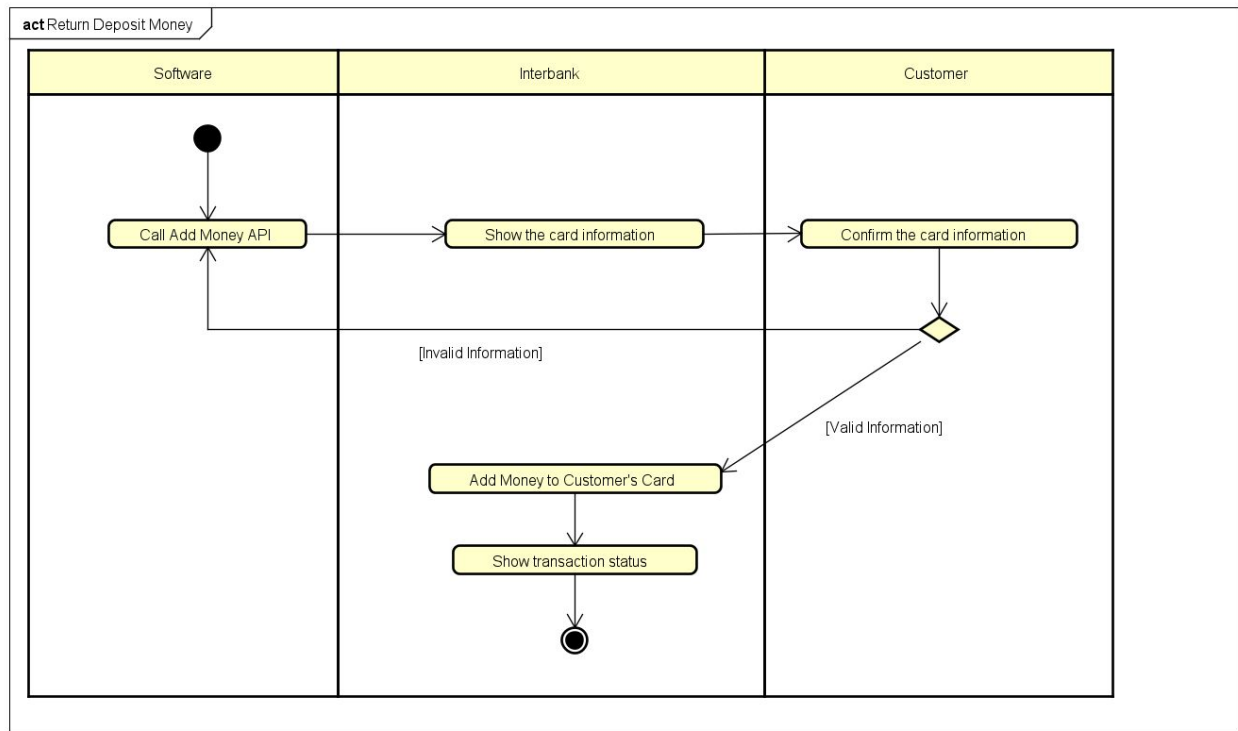
Step 5. The Software displays the success transaction.

### **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 |

### **7. Activity diagrams**



## 8. Input data

## 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<br>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<br>Chu Viet Dung                               |
| 4 | Return Deposit Amount    | Total deposit money to be returned  | Comma for thousands separator<br>Positive integer<br>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 |

## 10. Postconditions

None

### 3.6 Use case specification for “View Bikes Information”

#### Use Case “View Bikes Information”

##### 1. Use case code

UC006

##### 2. Brief Description

UC “View 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. Actors

##### 3.1 Customer

## 4. Preconditions

Successfully login

## 5. Basic Flow of Events

Step 1: Customer use the app to see information of bikes in rental

Step 2: Software check the user's rental details and collect the bike information in database

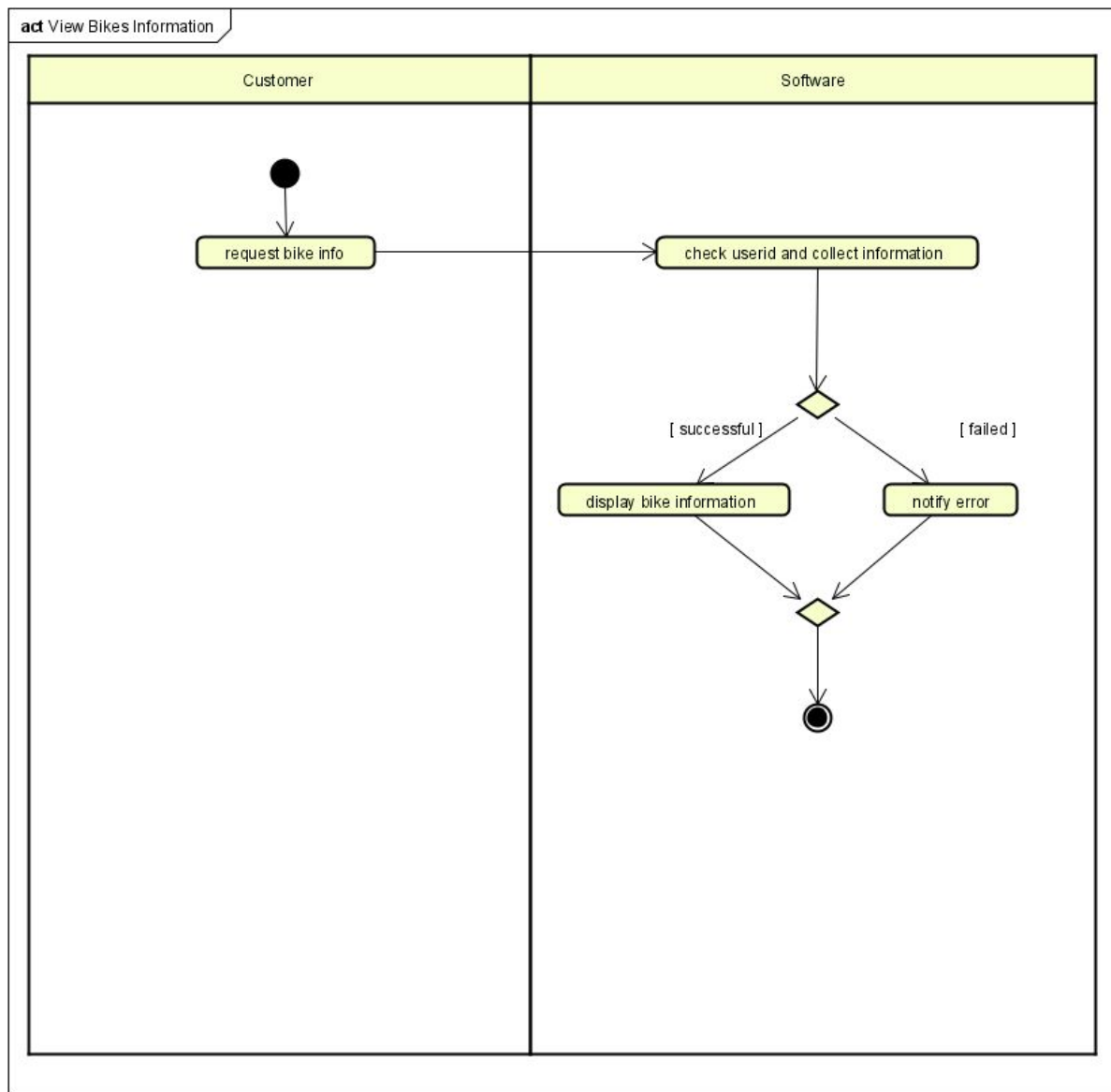
Step 3: Software display the view of bike information currently rented by the user

## 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 | ▪ The software notify for the error to the user | The use case ends |

## 7. Activity diagrams



## 8. Input data

Table A- Input data for View Bike Information

| No | Data fields | Description             | Mandatory | Valid condition | Example |
|----|-------------|-------------------------|-----------|-----------------|---------|
| 1. | Userid      | Unique id for each user | Yes       |                 | 11037   |

## 9. Output data

Table B- Output data for View Bike Information

| No | Data fields                         | Description  | Display format | Example |
|----|-------------------------------------|--------------|----------------|---------|
| 1  | Bike type                           | Type         | Code           | 1/2/3   |
| 2  | Renting time                        | Renting time | Time           | 4m20s   |
| 3  | Current amount to be paid           | Money        | Currency       | 69\$    |
| 4  | Current amount of battery (e-bikes) | Battery      | Percentage     | 69%     |

## 10. Postconditions

None

### 3.7 Use case specification for “View Bikes In Station”

## Use Case “View Bikes In Station”

### 1. Use case code

UC007

### 2. Brief Description

UC “View Bike in Station” allows the user to see all the 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. Actors

Customer

### 4. Preconditions

Successfully login

### 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

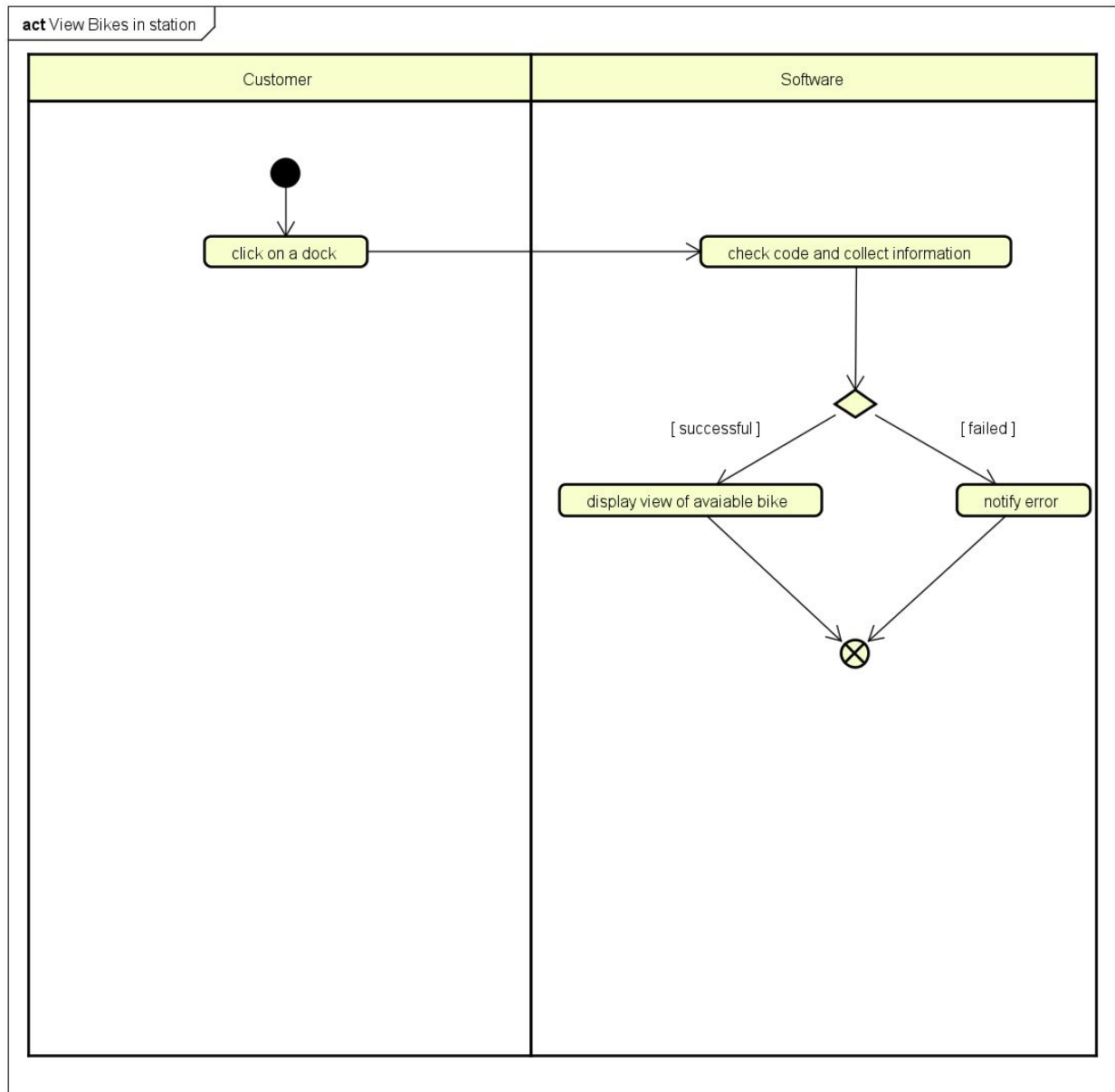
Step 3: Software display the view of available bike in a list/table for customer to choose

## 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"><li>▪ The software notify for the error to the user</li></ul> | The use case ends |

## 7. Activity diagrams



## 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 |

## 9. Output data



*Table B- Output data for View Bike In Station*

| No | Data fields                         | Description    | Display format  | Example |
|----|-------------------------------------|----------------|---|---------|
| 1  | Current amount of battery (e-bikes) | Battery        | <ul style="list-style-type: none"><li>Percentage</li></ul>                  | 58%     |
| 2  | Maximum time to use                 | Time remaining | <ul style="list-style-type: none"><li>Numbers and words</li></ul>           | 4h36m   |
| 3  | Bike's code                         | Code           | <ul style="list-style-type: none"><li>No space, words and numbers</li></ul> | TN234   |

**A**

**10. Postconditions**

Being able to choose renting option

## **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