HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

School of Information and communications technology

# Software Requirement Specification

Version 1.0

# EcoBikeRental

# Subject: ISD.VN.20211

Hoàng Mai Đức Long - 20184136

Trần Đức Hiếu - 20184104

Nguyễn Viết Huy - 20184120

*Hanoi, 10/2021*

# 1 Introduction

## 1.1 Objective

The purpose of this document is to present a detailed description of the EcoBikeRental 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 software system, Ecobike Rental, will be a system for users to rent and return bikes automatically in the area of Ecopark township.

The software allows users to view information of all available docks and bikes; as well as to find a dock, rent a bike, return the bike and pay for rent. To pay for rent, users need to link the software to Interbank account or E-wallet.

## 1.3 Glossary

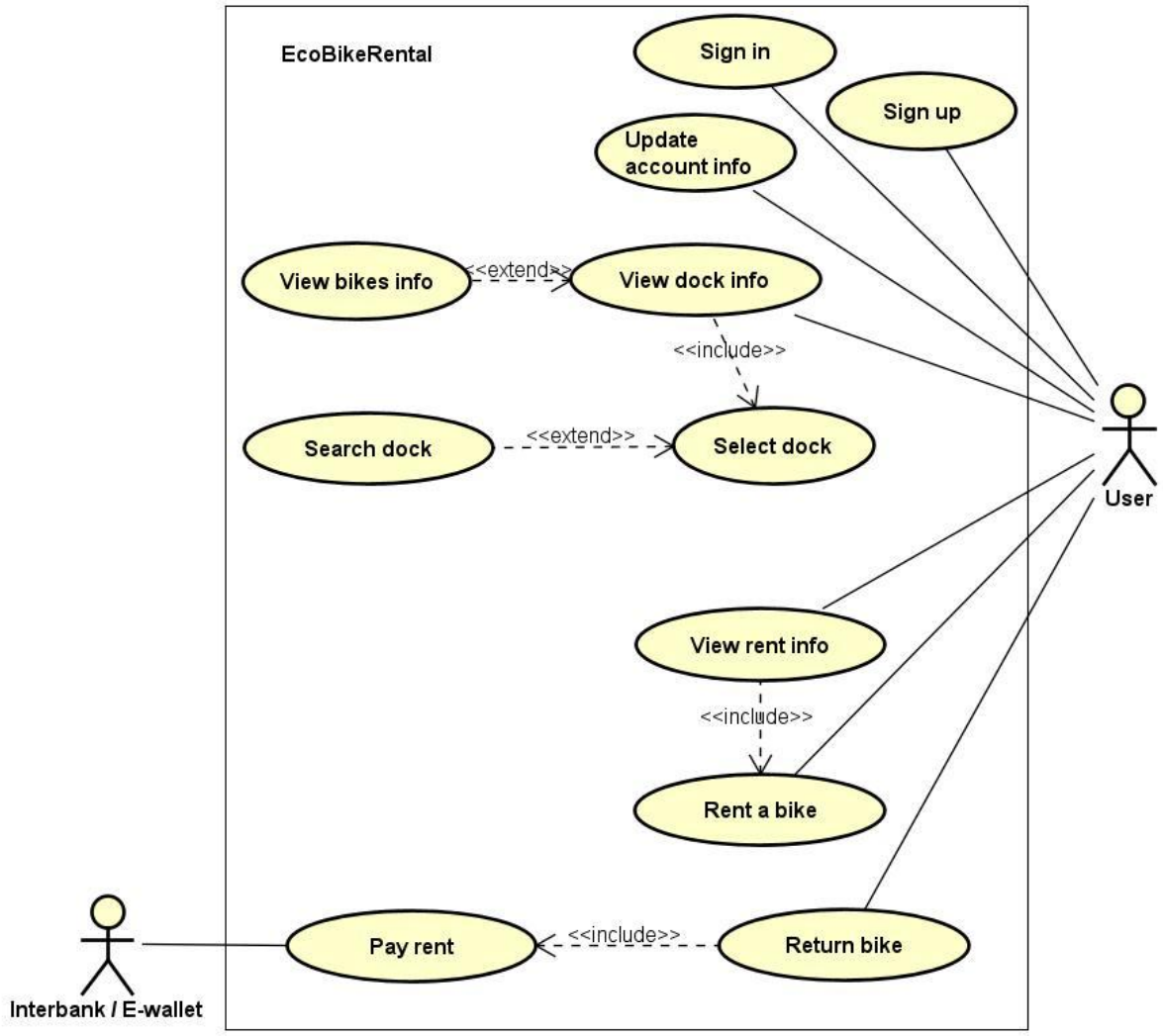
## 1.4 References

# 2 Overall Description

## 2.1 Actors

1) User
2) Interbank
3) The software (EcoBikeRental system)

## 2.2 Use case diagrams



# 3 Detailed Requirements

## 3.1 Use case specification for “View Dock Information”

### Use Case “View Dock Information”

#### 1. Use case code

UC001

#### 2. Brief Description

This use case describes the interaction between User and EcoBikeRental system when the User wishes to find and view the dock information.

#### 3. Actors

##### 3.1 User

#### 4. Preconditions

None

#### 5. Basic Flow of Events

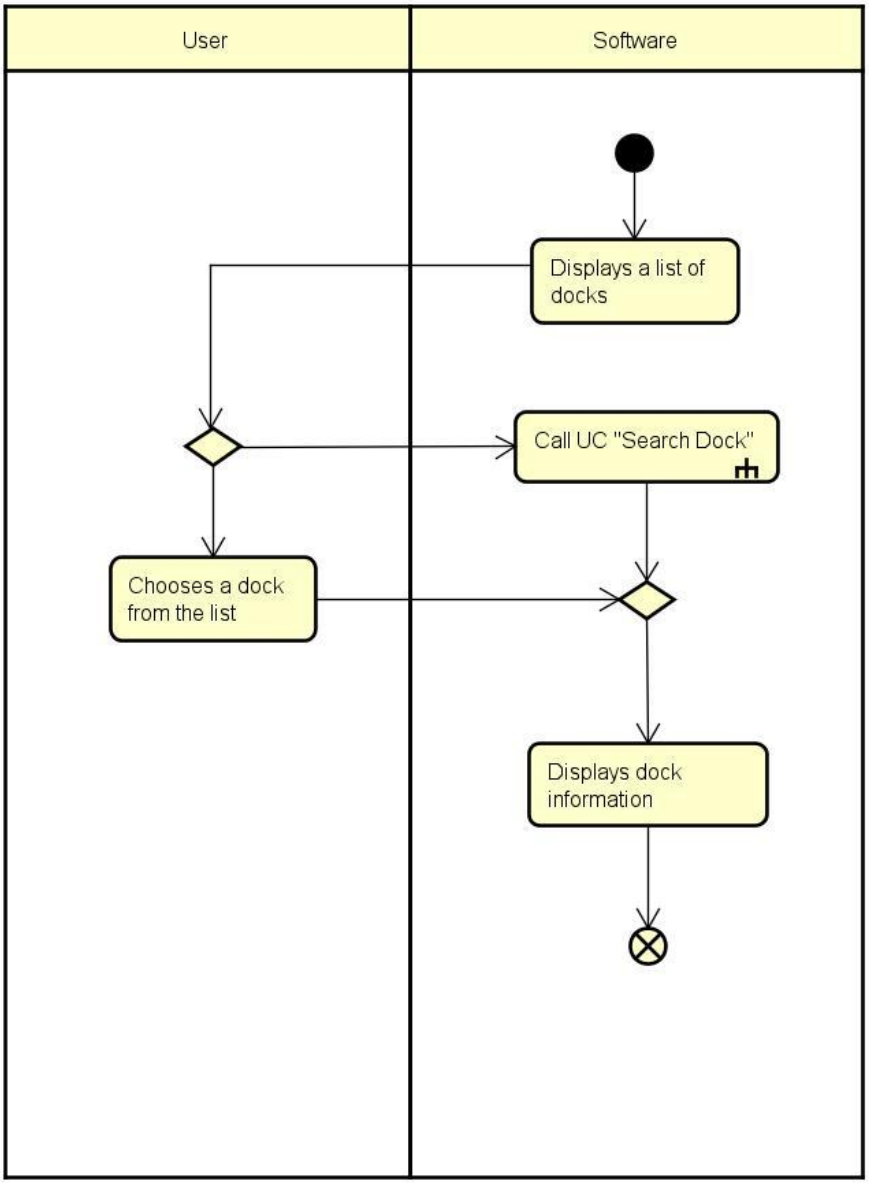
1. The Software displays a list of docks.
2. The User selects a dock from the list.
3. The Software displays the chosen dock’s information (see Table 3-2).

#### 6. Alternative flows

Table 3-1. Alternative flows of events for UC “View Dock Information”

| No | Location | Condition | Action | Resume location |
|---|---|---|---|---|
| 1. | At Step 2 | User searches for dock | Call use case “Search Dock” | Resumes at Step 3 |

## 7. Activity diagrams




## 8. Input data

None

## 9. Output data

Table 3-2. Dock display data

| No | Data fields | Description | Display format | Example |
|---|---|---|---|---|
| 1. | Name | | Text | |
| 2 | Address | | Text | |
| 3 | Area | | Area measurement displayed in square meters ($m^2$) | 1000 $m^2$ |
| 4 | Number of Bikes | | Integer | |
| 5 | Number of empty spots | | Integer | |
| 6 | Distance | Distance from user's location to the dock. | Distance measurement displayed in meter (m) | 500 m |
| 7 | Walk time | The amount of time for the user to walk to the dock. | Time measurement displayed in minute (min) | 10 min |

## 10. Postconditions

None

## 3.2 Use case specification for “Rent bike”

### Use Case “Rent bike”

1. **Use case code**

UC002

2. **Brief Description**

This use case describes the interactions between the EcoBikeRental system with the customer when the customer rent a bike.

3. **Actors**

3.1. Customer

3.2. EcoBikeRental system

4. **Preconditions**

None

5. **Basic Flow of Events**

1. Customer select the parking.

2. Customer enter the barcode of the rent bike.

3. EcoBikeRental system checks the barcode of the rent bike.

4. EcoBikeRental system calls API and converts barcode to barcode.

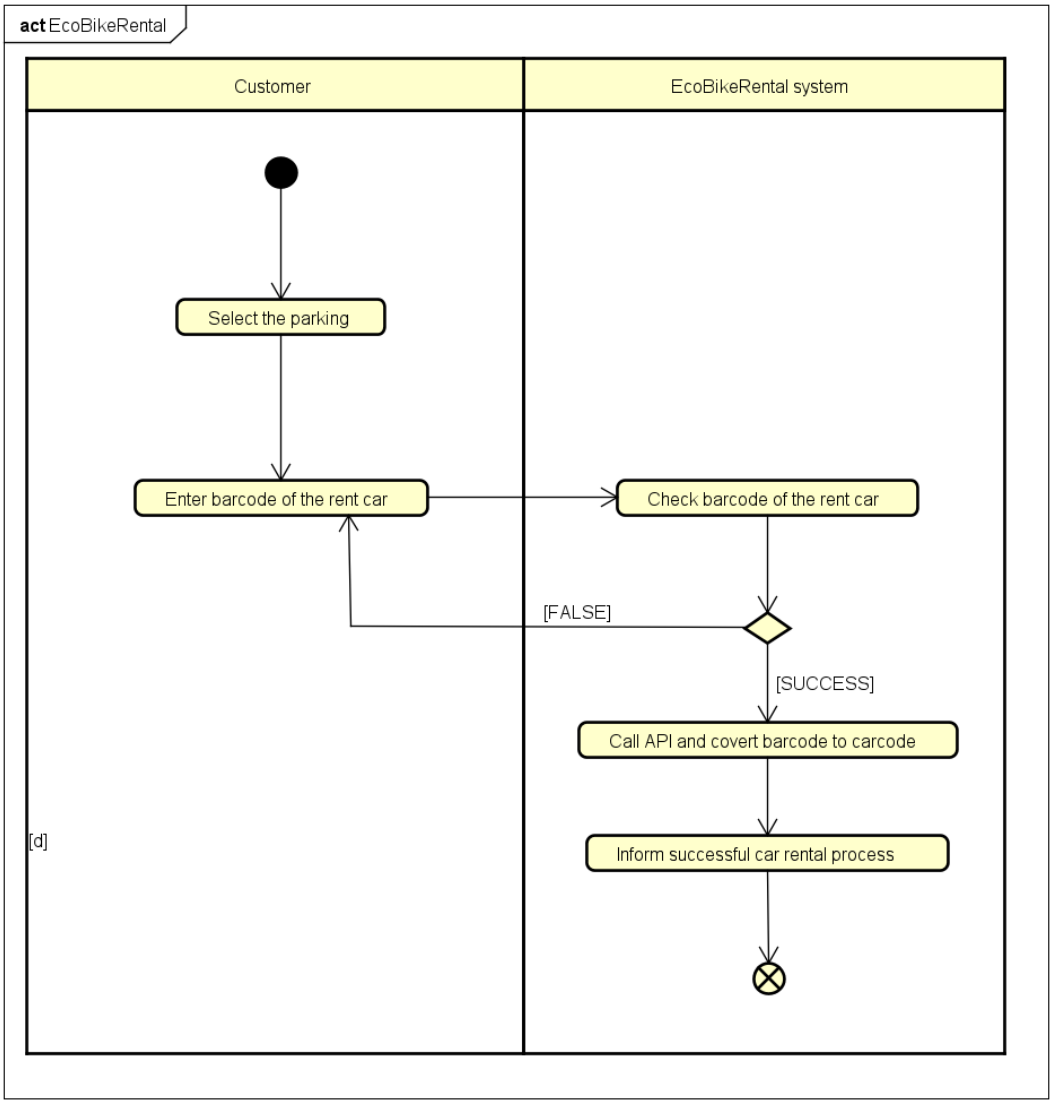
5. EcoBikeRental system inform successful bike rental process.

6. **Alternative flows**

| No | Location | Condition | Actions | Resume location |
|---|---|---|---|---|

| | | | | |
|---|---|---|---|---|
| 1 | Step 3 | If customer enter wrong barcode | EcoBikeRental system<br>notifles that the barcode does not exist | Step 2 |

## 7. Activity diagrams




## 8. Input data

Table A-Input data of rent bike

## 3.3 Use case specification for “Return bike”

### Use Case “Return a bike”

1. **Use case code**

   UC003

2. **Brief Description**

   This use case describes the interaction between customers and EcobikeRental when the customer wishes to return a car.

3. **Actors**

   3.1. Customer

   3.2. EcoBikeRental system

4. **Preconditions**

   None

5. **Basic Flow of Events**

   5.1 Customer chooses a feature which allows the customer to choose a dock.

   5.2 The application displays a list of available docks.

   5.3 Customers go to the parking lot.

   5.4 The customer chooses a location in the current dock.

   5.5 the user firmly pushes the bike into an empty docking point.

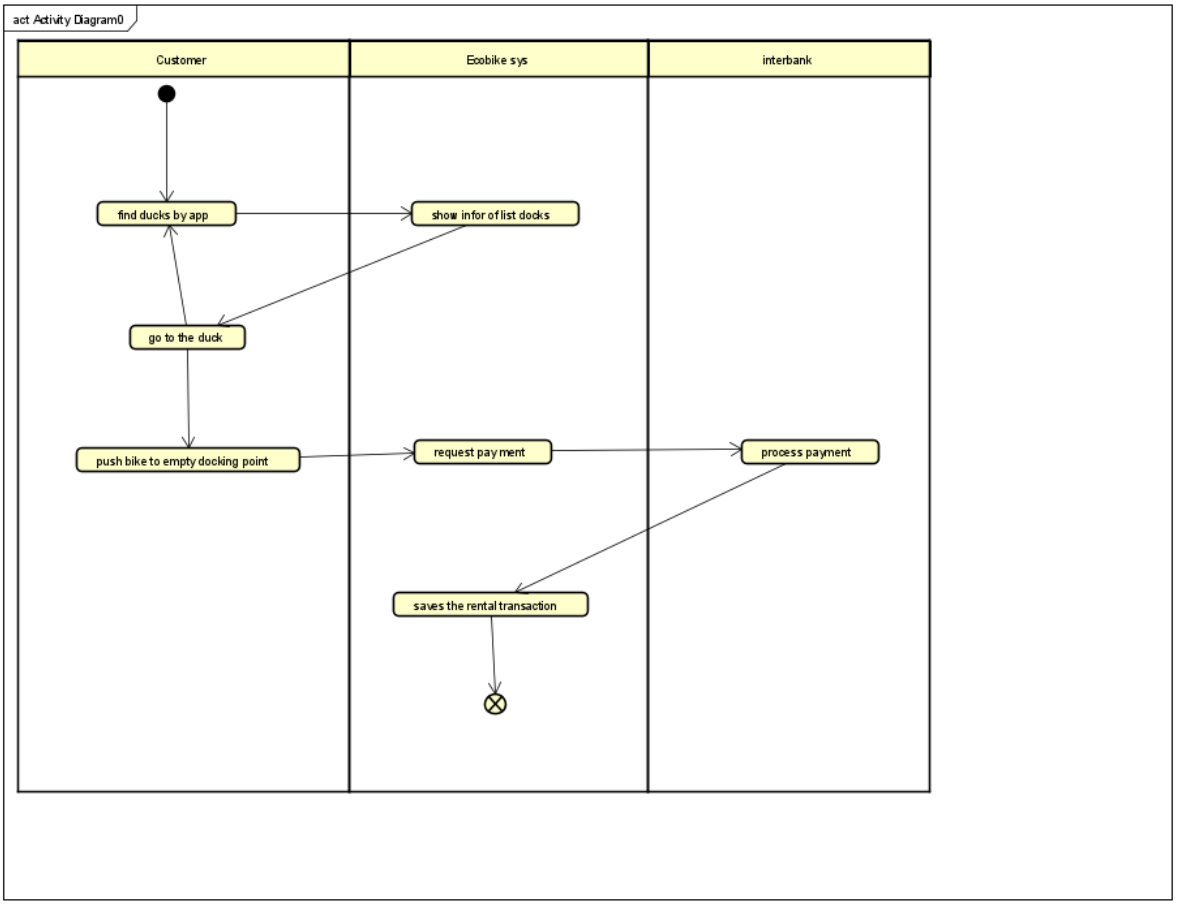
   5.6 System automatically pays  and returns the deposit.

   5.7 The system saves the rental transaction.

6. **Alternative flows**

| No | Location | Condition | Alternative flow | Resume location |
|---|---|---|---|---|
| | | | | |
| | At Step 5.4 | If the Parking is full | § Notice that the current parking is full | Resumes at Step 5.1 |
| | At Step 5.4 | If customer not use app to find ducks | § | Resumes at Step 5.3 |

## 7. Activity diagrams




## 9. Output data

Table B-Output data of return bike

| No | Data fields | Description | Display format | Example |
|---|---|---|---|---|
| **1.** | Carcode | | String | 123 |
| **2.** | Car name | | string | Honda |
| **3.** | Car ID | | string | 234 |
| **4.** | Time | | string | 8h |
| **5.** | Cost | | string | 5$/1h |
| **6.** | total | | | 40$ |

# 4 Supplementary specification

## 4.1 Functionality

1) Rent bike: The user enters the corresponding barcode of the bike that he or she wants to rent instead of scanning the barcode. Then the system will call an API to convert the barcode into a rental code

2) View bike info:

- When not renting bike: customers can see the detail information of the car and then choose the car that they want to rent.
- When renting a bike: customers can see more information about usage time and costs of renting a bike.

3) Update account info: Customer can update personal info (name, age, work address,.. ) in the system.

4) Return bike: For returning a bike, the application has a feature which allows the user to choose a dock and pay by use credit card

## 4.2 Usability

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

## 4.3 Reliability

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

## 4.4 Performance

The response time for the system is 1 second at normal and 2 seconds during a peak load if it is not explicitly stated

## 4.5 Supportability

Not specified

## 4.6 Other requirements

Not specified