HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

School of Information and communications technology

Software Requirement Specification

Version 1.1

Eco Bike Rental Subject: ISD.VN.20211

Group 4

Hoàng Mai Đức Long - 20184136 Trần Đức Hiếu - 20184104 Nguyễn Viết Huy - 20184120

*Hanoi, 10/2021*

**1 Introduction 3**

|  |  |  |  |
| --- | --- | --- | --- |
| **1.1** | | **Objective** | **3** |
| **1.2** | | **Scope** | **3** |
| **1.3** | | **Glossary** | **3** |
| **1.4** | | **References** | **3** |
| **2** | **Overall Description** | | **4** |
|  | **2.1 Actors** | | **4** |
|  | **2.2 Overall Use case Diagrams** | | **4** |
| **3** | **Detailed Requirements** | | **5** |
|  | **3.1 Use case specification for “View Dock Information”** | | **5** |
|  | 3.2 Use case specification for “Rent bike” | | 7 |
|  | **3.3 Use case specification for “Return bike”** | | **9** |
|  | 3.4 Use case specification for “Search Dock” | | 11 |
| **4** | **Supplementary specification** | | **14** |
|  | **4.1 Functionality** | | **14** |
|  | **4.2 Usability** | | **14** |
|  | **4.3 Reliability** | | **14** |
|  | **4.4 Performance** | | **14** |
|  | **4.5 Supportability** | | **14** |
|  | **4.6 Other requirements** | | **14** |

# Introduction

### 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.

### Scope

This software system, Ecobike Rental, will be a system for users to rent and return bikes automatically in the area of Ecopark township.

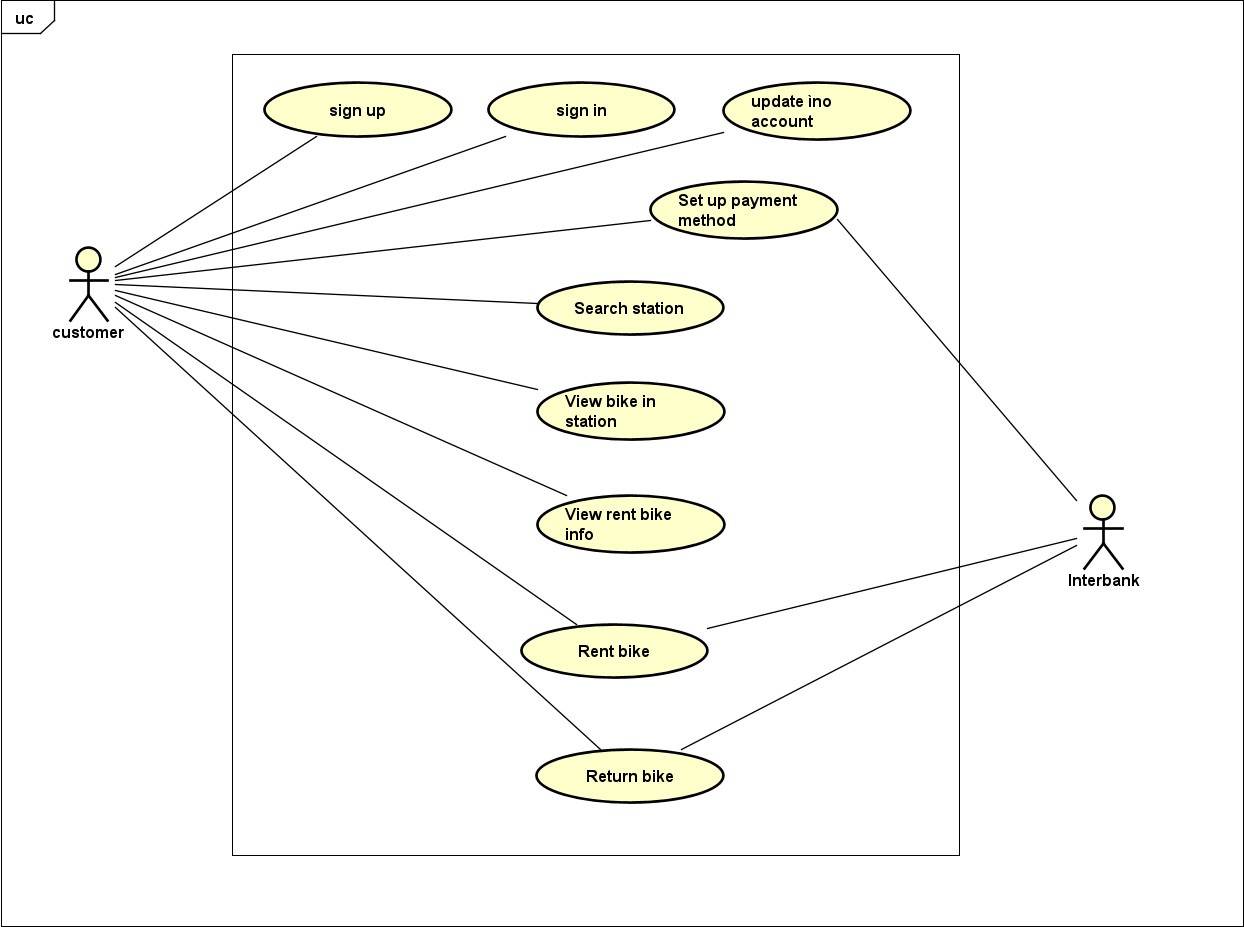
The software allows users to view information of all available docks and bikes; as well as to find a dock, rent a bike, return the bike and pay for rent. To pay for rent, users need to link the software to an Interbank account or E-wallet.

### Glossary

* 1. **References**

1. **Overall Description**
   1. **Actors**
      1. User
      2. Interbank
      3. The software (EcoBikeRental system)

### Overall Use case Diagrams



1. **Detailed Requirements**
   1. **Use case specification for “View Bike Information”**

#### Use Case “View Bike Information”

1. **Use case code**

UC001

#### Brief Description

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

#### Actors

* 1. **User**

1. **Preconditions**

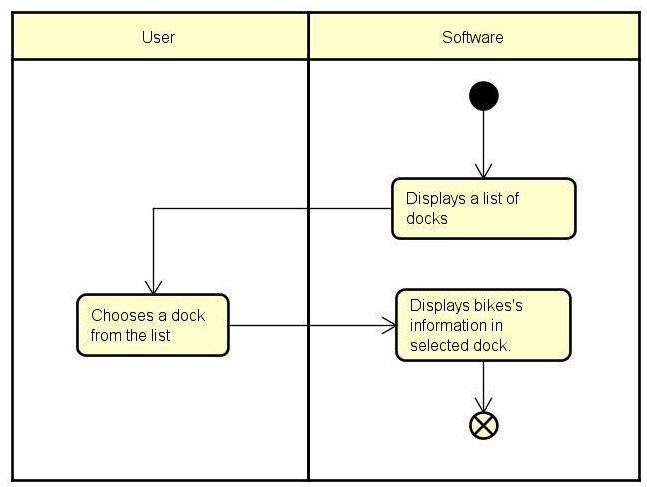
None

#### Basic Flow of Events

1. Software displays a list of docks.
2. User selects a dock from the list.
3. Software displays a list of bikes in the dock and their detailed information (see Table 3-1)

#### Alternative flows

1. **Activity diagrams**



1. **Input data**

None

#### Output data

**Table 3-1. Bike display data**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Data fields** | **Description** | **Display format** | **Example** |
| 1 | Type |  | String | Standard bike |
| 2 | Deposit |  | Number with currency | VND 400.000 |
| 3 | Battery | Remaining battery level, E-bike only | Percentage | 80% |
| 4 | Time | Estimated remaining operating time, E-bike only | Time, in hh:mm format | 04:32 |

1. **Postconditions**

None

* 1. **Use case specification for “Rent bike”**

#### Use Case “Rent bike”

1. **Use case code**

UC002

#### Brief Description

This use case describes the interactions between the user and the EcoBIkeRental system when the user wishes to rent a bike.

#### Actors

* 1. User
  2. EcoBikeRental system

#### Preconditions

None

#### Basic Flow of Events

1. The user selects the parking.
2. The user enters the barcode of the rented bike.
3. System checks the barcode of the rented bike.
4. System selects rent bike method ( rent bike normal && rent bike 24h)
5. The user selects payment method
6. System checks user's balance and deduce the deposit
7. System calls API and converts barcode to rental code.
8. System informs the success of the bike rental process.
9. System saves bike rental process

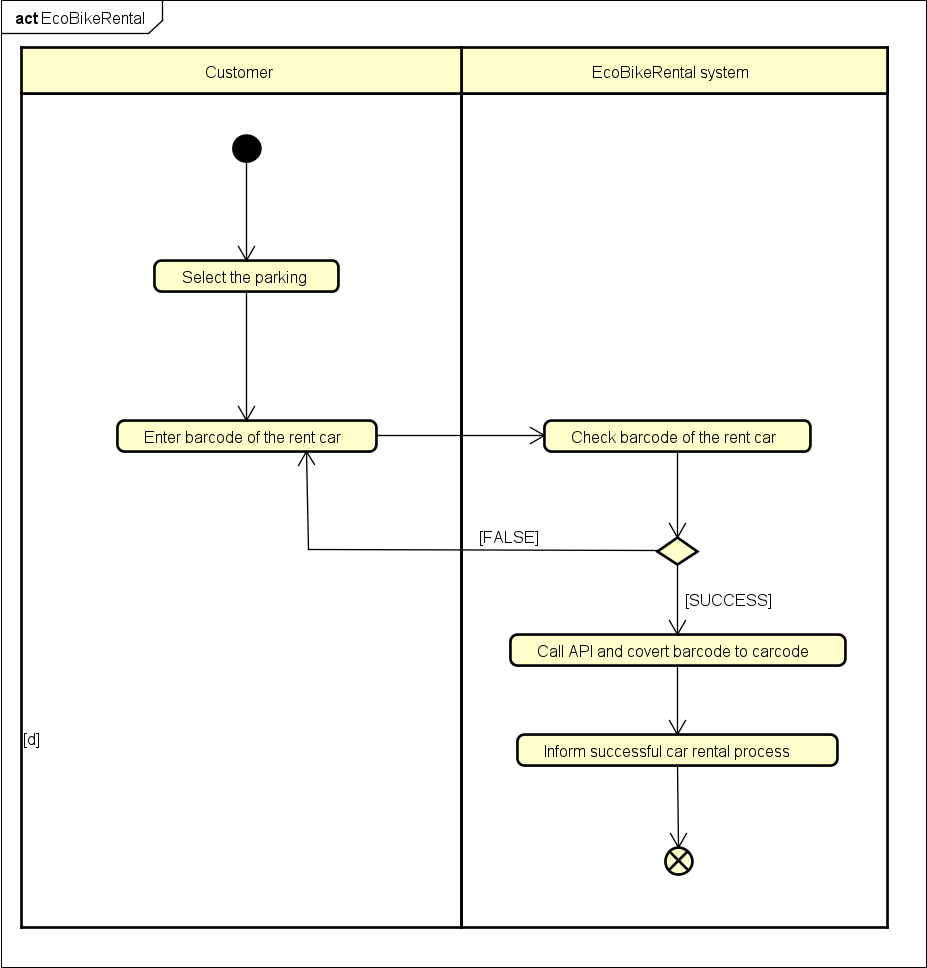
#### Alternative flows

**Table 3.2.1 – Alternative flows for UC “Rent Bike”**

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | Step 3 | If customer enter wrong barcode | EcoBikeRental system notifles that the barcode does not exist | Step 2 |
| 2 | Step 4 | If customer select rent bike 24h |  | Step 5 |
| 3 | Step 6 | If customer’s balance isn’t enough | EcoBikeRental system notifles that the balance is not enough | Step 5 |

1. **Activity diagrams**



1. **Input data**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Data fields** | **Description** | **Mandatory** | **Valid condition** | **Example** |
| **1.** | Barcode |  | Yes |  | 12345 |

1. **Output data**

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

1. **Postconditions**

None

* 1. **Use case specification for “Return bike”**

## Use Case “Return Bike”

1. **Use case code**

UC003

1. **Brief Description**

This use case describes the interactions between the EcoBike software with the customer and Interbank when the customer desires to return bike

1. **Actor** 3.1.Customer 3.2.Interbank
2. **Preconditions**

the customer is renting bike

1. **Basic flow events**
   1. Ecobike app shows the list station
   2. Customer chooses a station to return bike
   3. Customer selects feature return bike on app
   4. App shows a info invoice
   5. Customer confirm invoice
   6. App shows form payment
   7. Customer enter card infomation
   8. Ecobike app requests interbank process the transaction
   9. Ecobike app save the transaction
   10. Ecobike app save invoice
   11. Ecobike update dock info
2. **Alternative Flow**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Locatio n** | **Condition** | **Action** | **Resume Location** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 5.8 | The customer's bank card has a problem | Show error | End use case |

1. **Input data**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Data fields** | **Description** | **Mandator y** | **Valid conditi on** | **Example** |
| 1 | Bike type |  | yes |  | bicycle |
| 2 | Id bike |  | yes |  | Ab0x1234 |
| 3 | Rental time |  | yes |  | 8h |

1. **Output data**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Data fields** | **description** | **Display format** | **Example** |
| **1** | **Rental fee** |  | **number** | **63482,000** |
| **2** | **Refund money** |  | **number** | **782038,000** |
| **3** | **Id transaction** |  | **string** | **Xxx0x12345** |

1. **Postconditions**
   1. **Use case specification for “Search Dock”**

**Use Case “Search Dock”**

1. **Use case code**

UC004

1. **Brief Description**

This use case describes the interaction between the User and EcoBikeRental system when the User wishes to view dock’s information by searching for dock.

#### Actors

* 1. **User**

1. **Preconditions**

None

#### Basic Flow of Events

1. User enters the dock's name/address (see Table 3.4.2 for details).
2. Software queries for matching dock.
3. If found, Software displays the result dock’s information (see Table

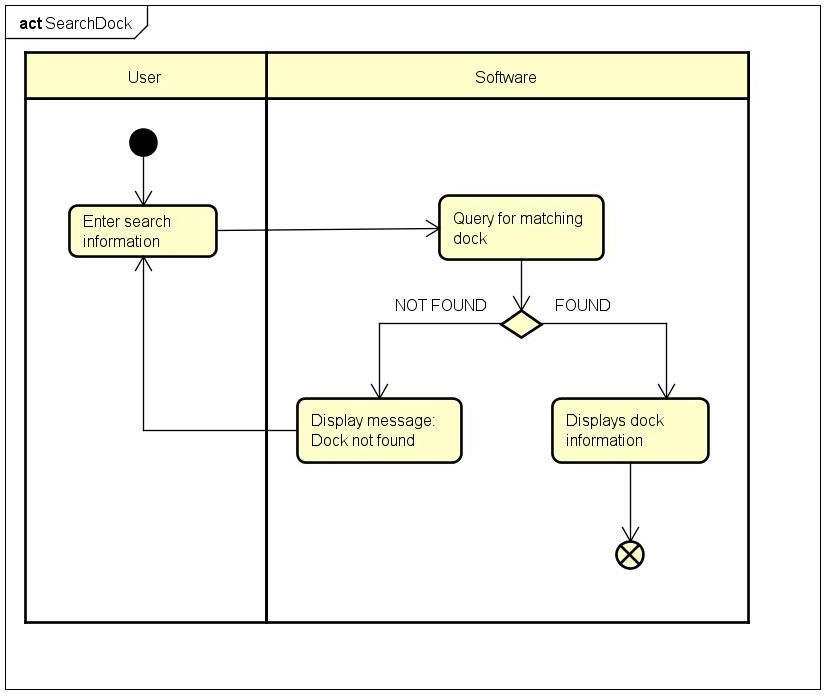
3.4.3 for details).

#### Alternative flows

**Table 3.4.1 - Alternative flows of events for UC “Search Dock”**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Location** | **Condition** | **Action** | **Resume location** |
| 1 | At Step 3 | No matching dock found. | Software displays error: “No matching dock found” | Resumes at Step 1 |

1. **Activity diagrams**



1. **Input data**

**Table 3.4.2 - Search input data**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Data fields** | **Description** | **Mandatory** | **Valid condition** | **Example** |
| 1 | Name | Name of the dock to search | No |  | Dock1 |
| 2 | Address | Address of the dock to search | No |  | 23, Tower1 |

1. **Output data**

**Table 3.4.3 - Dock display data**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **N**  **o** | **Data fields** | **Description** | **Display format** | **Example** |
| 1 | Name |  | String |  |
| 2 | Address |  | String |  |
| 3 | Area |  | Area measurement displayed in square meters (m2) | 1000 m2 |
| 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 |

1. **Postconditions**

None

# Supplementary specification

## Functionality

When an error occurs, the software must display the error to the user, including error’s information and guide for the user.

## Usability

All functions need to be designed so that users can use the software with no or little training. If the user makes a mistake, The system must notify the user. The notify must be detailed and easy to understand, so that the user can locate the mistake and redo.

## Reliability

The software is expected to:

* Operates 24/7.
* Serves 100 users at the same time without noticeable loss of performance.
* Operates in an average of 200 hours without failure.

## Performance

* Response time for any user interface interaction is 1 second at normal, 2 seconds at peak load.
* Response time for any transaction must not exceed 1 second.

## Supportability

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

## Other requirements

Not specified