**HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY**

**SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY**

**A picture containing logo

Description automatically generated**

**SOFTWARE REQUIREMENT SPECIFICATION**

**Project: Design and construct Ecobike Rental** **System**

**Group 10**

|  |  |
| --- | --- |
| Vũ Quang Trường | 20194198 |
| Nguyễn Huy Hoàng | 20194058 |
| Nguyễn Phúc Tân | 20194163 |

|  |  |
| --- | --- |
| **Instructor:** | Nguyễn Thị Thu Trang |
| **Subject:** | Software design and construction |
|  |  |
| **Hanoi, 12/2022** | |

Contents

[**A. Introduction** 3](#_Toc122901565)

[**I. Purpose** 3](#_Toc122901566)

[**II. Application overview** 3](#_Toc122901567)

[**III. Intended audience** 3](#_Toc122901568)

[**IV. Abbreviations** 3](#_Toc122901569)

[**V. Reference** 3](#_Toc122901570)

[**B. High level requirements** 4](#_Toc122901571)

[**I. Use case diagram** 4](#_Toc122901572)

[**II. Activity diagram** 5](#_Toc122901573)

[**C. Use case specialization** 12](#_Toc122901574)

[**I. Use case “Check info rented bike”** 12](#_Toc122901575)

[**II. Use case “Pause renting”** 15](#_Toc122901576)

[**III. Use case “Resume renting”** 16](#_Toc122901577)

[**IV. Use case “Check info parking lot”** 18](#_Toc122901578)

[**V. Use case “Check info bike in lot”** 20](#_Toc122901579)

[**VI. Use case “Rent bike”** 23](#_Toc122901580)

[**VII. Use case “Return bike”** 26](#_Toc122901581)

[**D. Other requirements** 28](#_Toc122901582)

# **A. Introduction**

## **I. Purpose**

Ecopark provides a bike rental service with various parking lots in the area. Customers can rent and return bike at the parking lot. The Ecobike Rental System is created to simulate this process.

This document provides the detailed descriptions for the Ecobike Rental System as well as the functional and non-functional requirements of the system.

## **II. Application overview**

The application is connected to a database which stores the information abount parking lots, bikes, users and rental activities. Users can use this application to rent a bike at any parking lot. During the rental time, they can pause as well as resume the rental process. They can also return the rented bike at any parking lot. The application is also connected to a bank interface to process the rental payment.

## **III. Intended audience**

This document is designed for:

* The system adminstrator
* The users.

## **IV. Abbreviations**

## **V. Reference**

[1] Systems Analysisand Design - Gary B. Shelly, Harry J. Rosenblatt,

Shelly Cashman Series, 2012

[2] Giáo trình phân tích và thiết kế hệ thống thông tin - Trần Đình Quế

[3] Slide bài giảng phân tích và thiết kế hệ thống - Nguyễn Hữu Đức

# **B. High level requirements**

## **I. Use case diagram**

The software consists of two actors: User and Interbank. The role of the Interbank is to handle transactions in the software.

Users can use the software to rent a bike if it is available for renting. During the time of renting, they can choose to manage the rented bike, which are pausing renting, resuming renting, checking its info and return the bike. Users can also check info of any parking lot and any of its bikes.

Diagram

Description automatically generated

**Figure 1:** UC Diagram level 1

Diagram

Description automatically generated

**Figure 2:** UC Diagram level 2 for UC “Manage rented bike”

## **II. Activity diagram**

Diagram

Description automatically generated

**Figure 3:** Activity diagram for use case ‘Rent bike’

Diagram

Description automatically generated

**Figure 4:** Activity diagram for use case ‘Pause renting’

Diagram

Description automatically generated

**Figure 5:** Activity diagram for use case ‘Resume renting’

Diagram

Description automatically generated

**Figure 6:** Activity diagram for use case ‘Check info rented bike’

Diagram

Description automatically generated

**Figure 7:** Activity diagram for use case ‘Return bike’

Diagram

Description automatically generated

**Figure 8:** Activity diagram for use case ‘Check info parking lot’

Diagram

Description automatically generated

**Figure 9:** Activity diagram for use case ‘Check info bike in lot’

# **C. Use case specialization**

## **I. Use case “Check info rented bike”**

**1. Use case code**

UC01

**2. Brief description**

UC “Check info rented bike” describes the interaction between customers

and Ecobike system when the customer wishes to check information of rented bike

**3. Actors**

3.1. User

**4. Preconditions**

There is an active network connection to the Internet

**5.** **Basic Flow of Events**

Step 1. The customer views rented bike

Step 2. The customer requests to check information of rented bike

Step 3. The Ecobike system checks the information of rented bike

Step 4. The Ecobike system displays the information of rented bike

**6. Alternative flows**

**Table X. Alternative flow of events for UC “Check info rented bike”**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Location** | **Condition** | **Action** | **Resume location** |
| 1 | At step 1 | If the customer is not renting any bike | ● The Ecobike system show notification | At step 1 |

**7.** **Input data**

**8.** **Output data**

**Table X. Output data of displaying information of rented bike**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Data fields** | **Description** | **Display format** | **Example** |
| 1 | ID | ID of bike |  | B0001 |
| 2 | Type | Type of bike |  | Normal bike |
| 3 | Name |  |  |  |
| 4 | Country | Country of manufacture |  |  |
| 5 | Color | Color of bike |  | White |
| 6 | Date of manufacture |  |  | 01/01/2022 |
| 7 | Price | Price of rented bike |  | 1000000 VND |
| 8 | Timeline | Display list start time, pause time, resume time in chronological order |  |  |
| 9 | Total rented time |  |  |  |
| 10 | Fee | Total fees of rented bike |  |  |
| 11 | Battery | Remaining battery of the bike |  | 95% |
| 12 | License plate number |  |  | MD1-20221 |

**9.** **Postconditions**

## **II. Use case “Pause renting”**

**1.** **Use case code**

UC02

**2.** **Brief description**

UC “Pause renting” describes the interaction between customers

and Ecobike system when the customer wishes to pause renting

**3.** **Actors**

3.1. User

**4.** **Preconditions**

There is an active network connection to the Internet

**5.** **Basic Flow of Events**

Step 1. The customer views rented bike

Step 2. The customer requests to pause renting

Step 3. The Ecobike system update the information of rented bike

Step 4. The Ecobike system displays success message

**6.** **Alternative flows**

**Table X. Alternative flow of events for UC “Pause renting”**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Location** | **Condition** | **Action** | **Resume location** |
| 1 | At step 1 | If the customer is not renting any bike | ● The Ecobike system show notification “NO RENTED BIKE” | At step 1 |
| 2 | At step 3 | If system fails to update | ● The Ecobike system show notification “FAIL” | At step 2 |

**7.** **Input data**

**8.** **Output data**

**9.** **Postconditions**

The logs have been updated accordingly.

## **III. Use case “Resume renting”**

**1.** **Use case code**

UC03

**2.** **Brief description**

UC “Resume renting” describes the interaction between customers

and Ecobike system when the customer wishes to resume renting.

**3.** **Actors**

3.1. User

**4.** **Preconditions**

There is an active network connection to the Internet and customer is stopping bike rental

**5.** **Basic Flow of Events**

Step 1. The customer views rented bike

Step 2. The customer requests to resume renting

Step 3. The Ecobike system update the information of rented bike

Step 4. The Ecobike system displays success message

**6.** **Alternative flows**

**Table X. Alternative flow of events for UC “Resume renting”**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Location** | **Condition** | **Action** | **Resume location** |
| 1 | At step 1 | If the customer is not renting a car | ● The Ecobike system show notification “NO RENTED BIKE” | At step 1 |
| 2 | At step 3 | If system fails to update | ● The Ecobike system show notification “FAIL” | At step 2 |

**7.** **Input data**

**8.** **Output data**

**9.** **Postconditions**

The logs have been updated accordingly.

## **IV. Use case “Check info parking lot”**

**1. Use case code**

UC04

**2.** **Brief description**

UC “Check info parking lot” describes the interaction between customers

and Ecobike system when the customer wishes to check information of rented bike

**3.** **Actors**

3.1. User

**4.** **Preconditions**

There is an active network connection to the Internet

**5.** **Basic Flow of Events**

Step 1. The customer views the available parking lot.

Step 2. The customer clicks on a parking lot.

Step 3. The Ecobike system checks the information of the lot.

Step 4. The Ecobike system displays the information of the lot.

**6.** **Alternative flows**

**7.** **Input data**

**8.** **Output data**

**Table XI. Output data of displaying information of a parking lot**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Data fields** | **Description** | **Display format** | **Example** |
| 1 | ID | ID of the parking lot | PL\_\_ | PL01 |
| 2 | Location |  |  | 1st Giai Phong, Hanoi |
| 3 | Name |  |  | Giai Phong Parking Lot |
| 4 | Distance | Distance from the lot to the customer location |  | 3 km |
| 5 | Travelling time | Travelling time (on foot) from the lot to the customer location |  | 20 min |
| 3 | Total number of slots |  | Int | 50 |
| 4 | Number of empty slots |  | Int | 10 |
| 5 | List of bikes | List of bikes in the parking lot as well as their status (rented or not) |  |  |

**9.** **Postconditions**

## **V. Use case “Check info bike in lot”**

**1. Use case code**

UC05

**2.** **Brief description**

UC “Check info bike in lot” describes the interaction between customers

and Ecobike system when the customer wishes to check information of a bike in a particular parking lot.

**3.** **Actors**

3.1. User

**4.** **Preconditions**

There is an active network connection to the Internet

**5.** **Basic Flow of Events**

Step 1. The customer views the available parking lot.

Step 2. The customer clicks on a parking lot.

Step 3. The Ecobike system checks the information of the lot.

Step 4. The Ecobike system displays the information of the lot.

Step 5. The customer clicks on a bike in the bike list displayed.

Step 6. The Ecobike system checks the information of the selected bike.

Step 7. The Ecobike system displays the information of the bike.

**6.** **Alternative flows**

**7.** **Input data**

**8.** **Output data**

**Table XII. Output data of displaying information of a bike in a parking lot**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Data fields** | **Description** | **Display format** | **Example** |
| 1 | ID | ID of bike |  | B0001 |
| 2 | Type | Type of bike |  | Normal bike |
| 3 | Name |  |  |  |
| 4 | Country | Country of manufacture |  |  |
| 5 | Color | Color of bike |  | White |
| 6 | Date of manufacture |  |  | 01/01/2022 |
| 7 | Price | Price of rented bike |  | 1000000 VND |
| 8 | Battery | Remaining battery of the bike |  | 95% |
| 9 | Maximum usage time | Maximum usage time corresponding to the remaining battery |  | 2 hours |
| 10 | License plate number |  |  | MD1-20221 |
| 11 | Slot ID |  |  | S01 |
| 12 | Is rented | Boolean |  | False |

**9.**  **Postconditions**

## **VI. Use case “Rent bike”**

**1. Use case code**

UC06

**2. Brief description**

UC “Rent bike” allows customers to rent a bike.

**3. Actors**

3.1. User

**4. Preconditions**

There is an active network connection to the Internet.

**5. Basic Flow of Events**

Step 1. The customer selects a park.

Step 2. The Ecobike system displays the park information.

Step 3. The customer chooses to rent bike.

Step 4. The Ecobike system displays the input form.

Step 5. The customer enters and submits the ID of the bike they want to rent.

Step 6. The Ecobike system ensures that the customer want to rent that bike.

Step 7. The Interbank processes payment.

Step 8. The Ecobike system saves the transaction and unlock the bike.

**6. Alternative flows**

**Table VII. Alternative flow of events for UC “Rent bike”**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Location** | **Condition** | **Action** | **Resume location** |
| **1** | At step 5 | If the code submitted is invalid or non-existent | ● The Ecobike system show notification “BIKE CODE INVALID” or “BIKE CODE NON-EXISTENT” | At step 5 |
| **2** | At step 6 | If the customer click on “NO” | ● | At step 5 |
| **3** | At step 7 | If transaction fails | ● The Ecobike system show reason of failing transaction | At step 5 |

**7. Input data**

**Table VIII. Input data of rent bike**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Data fields** | **Description** | **Mandatory** | **Valid Condition** | **Example** |
| **1** | ID | ID of bike | Yes |  | B0001 |

**8. Output data**

**Table IX. Output data of rent bike**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Data fields** | **Description** | **Display format** | **Example** |
| **1** | Transaction fee | Transaction fee for renting deposit |  | 100,000 |

**9. Postconditions**

## **VII. Use case “Return bike”**

**1. Use case code**

UC07

**2. Brief description**

UC “Return bike” allows customers to return a rented bike.

**3. Actors**

**3.1.** User

**4. Preconditions**

There is an active network connection to the Internet and the customer is renting a bike.

**5. Basic Flow of Events**

Step 1. The customer chooses to return bike.

Step 2. The Ecobike system displays the list of available parks.

Step 3. The customer selects a park.

Step 4. The Ecobike system ensures that the customer want to return the rented bike to the selected park.

Step 5. The Interbank processes payment.

Step 6. The Ecobike system saves the transaction and lock the bike.

**6. Alternative flows**

**Table X. Alternative flow of events for UC “Return bike”**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Location** | **Condition** | **Action** | **Resume location** |
| **1** | At step 4 | If the customer click on “NO” | ● | At step 3 |
| **2** | At step 5 | If transaction fails | ● The Ecobike system show reason of failing transaction | At step 3 |

**7. Input data**

**8. Output data**

**Table XII. Output data of rent bike**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Data fields** | **Description** | **Display format** | **Example** |
| **1** | Transaction fee | Renting fee and returning deposit |  | 100,000 |

**9. Postconditions**

# **D. Other requirements**

# The Ecobike system has to satisfy the use cases that require fast response. The system needs to ensure the security of customer information such as travel schedules, credit cards,... In addition, the software works with transactions related to the economic interests of related parties, so it is necessary to ensure accuracy and reliability as well as consistency across the entire system.

The main properties that the system needs to ensure:

* **Easy to use:** The system, once created, must provide an easy-to-use user interface. Consistent operations, limiting user input steps where possible
* **Response Time:** The maximum response time of the system is 1 second at normal or 2 seconds at peak.
* **Availability:** The system must always satisfy the needs of customers 24/7
* **Simultaneity:** The software can serve 100 users at a time without any change in performance.
* **Fault tolerance:** The system can operate for 200 hours continuously without error. It can be back to normal within 2 hours after the fault
* **Cross-platform compatibility**
* **Security:** The system must ensure the safety of user data and personal transaction information
* **Testability:** The recommendation system must keep a history of resolved orders
* **Maintenance:** Due to the increasing and diverse customer needs, as well as the explosive development of new technologies, the system needs to be easy to maintain and upgrade.
* **Design constraints:**

 ◦ Right-aligned number, displayed thousands separated by dots.

 ◦ Left aligned letters.

 ◦ Font: Arial 16, black color.