**Hanoi University of Science and Technology  
School of Information and Communication Technology**

**Software Requirement Specifications**

**Version 1.0**

**EcoBike Application**

***Subject: ITSS Software Development***

**Supervisor: Asst. Prof. Nguyen Thi Thu Trang**

**Group number: 6**

**Group participants**

|  |  |
| --- | --- |
| **Name** | **Student ID** |
| **Nguyen Thi Minh Chau** | **20184238** |
| **Tran Le Hai Duong** | **20184248** |
| **Nguyen Thanh Long** | **20184287** |

**Table of Contents**

[**1. Introduction** 3](#_Toc85357296)

[**1.1 Objective** 3](#_Toc85357297)

[**1.2 Scope** 3](#_Toc85357298)

[**1.3 Glossary** 4](#_Toc85357299)

[**2. Overall description** 4](#_Toc85357300)

[**2.1 Actors** 4](#_Toc85357301)

[**2.2 Usecase diagram** 5](#_Toc85357302)

[**2.3 Business processes** 5](#_Toc85357303)

[**3. Detailed requirement** 7](#_Toc85357304)

[**3.1 Usecase specifications for “CRUD bike”** 7](#_Toc85357305)

[**3.2 Usecase specifications for “Check bike information”** 15](#_Toc85357306)

[**3.3 Usecase specifications for “Rent a bike”** 18](#_Toc85357307)

[**3.4 Usecase specifications for “Deposit”** 20](#_Toc85357308)

[**3.5 Usecase specifications for “Return a bike”** 23](#_Toc85357309)

[**3.6** **Usecase specifications for “Return deposit”** 25](#_Toc85357310)

[**3.7 Usecase specifications for “Pay for bike rental”** 27](#_Toc85357311)

[**3.8 Usecase specifications for “Update payment method”** 29](#_Toc85357312)

[**4. Supplementary specification** 31](#_Toc85357313)

[**4.1 Functionality** 31](#_Toc85357314)

[**4.2 Usability** 31](#_Toc85357315)

[**4.3 Reliability** 31](#_Toc85357316)

[**4.4 Performance** 31](#_Toc85357317)

[**4.5 Supportability** 31](#_Toc85357318)

[**4.6 Database design** 31](#_Toc85357319)

[**4.7 Other requirements** 32](#_Toc85357320)

## **1. Introduction**

### **1.1 Objective**

The purpose of this document is to present a detailed description of the EcoBike 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 EcoBike system is designed for EcoPark, which has an hourly bike rental service with docking stations for customers to rent or return bikes automatically. This system provides convenience for customers to finding a dock of bike, signing up for renting a bike, paying renting fees and returning a bike. EcoBike also assists administrators in monitoring bikes in the system, which includes adding, viewing, updating and removing bike to/from the system.

In details, when the customer starts EcoBike, the application displays the map of the region around the current location of him. The customers can then select or search for a bike station to view its information, including the distance and estimated walking time to the selected dock. At the same time, EcoBike provides a list of available bikes at the dock at the meantime. Customer can also select a bike in order to get its information and current states before deciding to rent it.

To rent a bike, customer uses the application to scan the barcode on the lock and open the bike. Details about the selected bike is shown, and if he wants to rent it, the application calculates the deposit, and transactions is performed by using credit cards connected to an interbank. When being in rent, the state of the bike is always updated so that customers can have a better look on his current renting bike.

To return a bike, customer firmly pushes the bike to an empty dock point and closes the lock. EcoBike then calculates the total charges corresponding to renting time. At the same time, the system saves the rental transaction and sends an email of transactions to customers.

The interbank, which will be in connection with EcoBike system, will receive payment request from EcoBike after the customer has confirmed the rental. It will help the customer to pay deposit before renting a bike, and finish the fees after he is done with renting the bike. Also, if the customer rents a bike for less than a predefined time period, he will have his deposit returned. This return will also be issued by EcoBike and be performed by the interbank.

For the administrator side, he can manipulate the list of bikes in system for rental at any time. To add a new bike, he provides information for the application so that EcoBike can validates it and create a new bike profile for tracking its states. Administrator can choose to view lists of bikes, as well as detail information of these bike. Also, while being in the monitor screen, he can choose to update bike information or delete it from the list if it is not being used in the future anymore.

### **1.3 Glossary**

***Table 1:*** Terms used in the document

|  |  |
| --- | --- |
| **Term** | **Definition** |
| Administrator | The person who uses EcoBike application system for the purposes of monitoring list of bicycles in the system |
| Admin | as “administrator” |
| Bicycle | The transportation mean to be rent in this application system |
| Bike | as “bicycle” |
| Card number | The ID number of the credit card, printed on the credit card |
| Cardholder name | The name of the owner of the credit card, printed on the credit card |
| Credit card | A card connected to the interbank, used for performing transaction |
| Customer | The person who uses EcoBike application system for the purposes of renting bike |
| Database | Collection of all information monitored by this system |
| Deposit | An amount of money customer has to pay at first in order to rent a bike |
| Dock | A place where bicycles are put |
| Interbank | The organization in charges of performing payment and return deposit transactions in the system |
| Payment | An amount of money customer has to pay to rent a bike, including deposit and rental fee |
| Rent a bike | The action of using a bike in a period of time, with paying deposit and rental fee |
| Rental fee | An amount of money customer has to pay, outside of the deposit, which depends on the rental time |
| Rental time | The time period when the bike is being rented |
| Return a bike | The action of stopping using a bike after having rented |
| Software Requirement Specification | A document that completely describes all of the functions of a proposed system and the constraints under which it must operate. For example, this document. |
| Station | as “dock” |
| Transaction | The action of paying for bike deposit, bike rental or returning deposit |
| User | Customer or Administrator |

## **2. Overall description**

### **2.1 Actors**

*2.1.1 Customer*

The customer is the actor who interacts with the system for the purpose of renting a bike. The activities of the customers with EcoBike system includes checking and selecting dock, viewing bike information, renting and returning bike, performing payment.

*2.1.2 Administrator*

The administrator is the actor who interacts with the system for the purpose of monitoring bicycles in the EcoBike system. The activites of the customers with the application are adding, viewing and updating bike information, and deleting bike from the system.

*2.1.3 Interbank*

The interbank is the actor who interacts with the system for the purpose of performing transactions issued by the customers. The activities includes paying deposit, rental fees and returning deposit

### **2.2 Usecase diagram**

The below diagrams illustrate the overall usecase of the actors on the EcoBike system

**Figure: Usecase diagram of EcoBike system**

Diagram

Description automatically generated

### **2.3 Business processes**

The bellow diagram shows the business process for the EcoBike system

Diagram

Description automatically generated

Figure : Business process diagram for EcoBike system

## **3. Detailed requirement**

### **Usecase specifications for “CRUD bike”**

* + 1. **Usecase Code:** **UC-110**
    2. **Brief description**

This usecase describe interaction between Administrators and EcoBike application when he wants to create, read information, update or delete the bike from the system

* + 1. **Actors**
* Administrator
  + 1. **Preconditions**
* Administrator logs in the system successfully
* Administrator has entered the administrator page
  + 1. **Basic flow of events**

1. ***Creating new bike***

**Step 1:** Admin chooses to create new bike

**Step 2:** EcoBike system displays form for entering new bike information

**Step 3:** Admin enters new bike information

**Step 4:** EcoBike validates newly entered information

**Step 5:** Admin approves to create new bike

**Step 6:** EcoBike creates new bike instance in the system and displays new bike detail information

1. ***Reading bike information***

**Step 1:** Admin select to list information of the bike

**Step 2:** EcoBike displays all bikes currently in the system

**Step 3:** Administrator chooses one bike to read its detail information

**Step 4:** EcoBike displays board contains detail information of the bike and options for managing the bike

1. ***Update bike information***

**Step 1:** EcoBike calls “READING BIKE INFORMATION” usecase

**Step 2:** Administrator selects to update bike information

**Step 3:** EcoBike shows form for updating bike information

**Step 4:** Admin enters new information for the bike

**Step 5:** EcoBike validates newly entered information

**Step 6:** Admin approves to update bike information

**Step 7:** EcoBike updates bike information with the entered inputs and display bike new detail information

1. ***Deleting bike***

**Step 1:** EcoBike calls “READING BIKE INFORMATION” usecase

**Step 2:** Administrator selects to delete a bike

**Step 3:** EcoBike asks for confirmation

**Step 4:** Admin approves to delete the bike

**Step 5:** EcoBike clears the bike's profile from the system

* + 1. **Alternative flows of events**

**Table: Alternative flow of “creating new bike” usecase**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Location** | **Conditions** | **Actions** | **Resume Location** |
| 1 | Anytime before step 6 | Admin quits the creating bike process | - EcoBike asks for confirmation to quit the process |  |
| 1a |  | Admin accepts to quit the creating bike process | - EcoBike returns to the main page |  |
| 1b |  | Admin refuses to quit the creating bike process | - EcoBike re-display the form for entering new bike’s information | 2 |
| 2 | Step 4 | Admin enters input with incompatible type | - EcoBike highlights invalid fields and requires admin to provide information that must have required type | 2 |
| 3 | Step 4 | Admin enters duplicated information for unique fields | - EcoBike highlights invalid fields and requires admin to provide unique information | 2 |
| 4 | Step 4 | Admin does not enter mandatory fields | - EcoBike highlights missing fields and requires admin to provide valid information | 2 |
| 5 | Step 6 | EcoBike fails to create new instance in the system due to internal errors | - EcoBike notifies errors (with error code) and requires admin to input again | 2 |

**Table: Alternative flow of “reading bike information” usecase**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Location** | **Conditions** | **Actions** | **Resume Location** |
| 1 | Step 2 | Admin search for a specific bike | - Admin selects to search for a specific bike by name, ID or license plate  - Admin enters information to search  - EcoBike fetches matched data and display results |  |

**Table: Alternative flow of “updating bike information” usecase**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Location** | **Conditions** | **Actions** | **Resume Location** |
| 1 | Anytime before step 6 | Admin quits the update bike process | - EcoBike asks for confirmation to quit the process |  |
| 1a |  | Admin accepts to quit the update bike process | - EcoBike returns to the main page |  |
| 1b |  | Admin refuses to quit the update bike process | - EcoBike re-display the form for updating bike’s information | 2 |
| 2 | Step 5 | Admin enters input with incompatible type | - EcoBike highlights invalid fields and requires admin to provide information that must have required type | 3 |
| 3 | Step 5 | Admin enters duplicated information for unique fields | - EcoBike highlights invalid fields and requires admin to provide unique information | 3 |
| 4 | Step 5 | Admin does not enter mandatory fields | - EcoBike highlights missing fields and requires admin to provide valid information | 3 |
| 5 | Step 7 | EcoBike fails to update new instance in the system due to internal errors | - EcoBike notifies errors (with error code) and requires admin to try again | 3 |

**Table: Alternative flow of “deleting bike” usecase**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Location** | **Conditions** | **Actions** | **Resume Location** |
| 1 | Step 2 | Admin choose to delete multiple bike at once | - EcoBike display checkboxes for selecting bikes to delete  - Admin chooses bikes to delete by checking/unchecking boxes | 3 |
| 2 | Step 4 | Admin refuses to delete the bike(s) | - EcoBike re-display the page for selecting bike to delete | 1 |

* + 1. **Activity diagrams**

**Figure: Activity diagrams for “creating new bike” usecase**

Diagram

Description automatically generated

**Figure: Activity diagrams for “reading bike information” usecase**

Diagram

Description automatically generated

**Figure: Activity diagram of “updating bike information” usecase**

Diagram

Description automatically generated

**Figure: Activity diagram of “deleting bike” usecase**

Diagram

Description automatically generated

* + 1. **Input data**

**Table: Input data for “creating new bike” and “updating bike” usecase**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Data field** | **Description** | **Mandatory** | **Valid conditions** | **Example** |
| 1 | Bike name | Name of the bike | Yes | String | Thong Nhat Bike |
| 2 | Type | Type of the bike | Yes | String | Normal bike, E-bike |
| 3 | License plate code | The string of the bike’s license plate | Yes | - String  - Unique | BY-18-12345 |
| 4 | Image of the bike |  | No |  |  |
| 5 | Bike barcode | The barcode assigned to the bike to lock/unlock | Yes | - 12 digits  - Unique | 863127556142 |
| 6 | Bike rental price | The price to rent the bike | Yes | - Float  - Dot as decimal separator  - 2 digits after decimal separator | 12.00 |
| 7 | Time unit | The default time unit used to calculate rental fee | Yes | - Abbreviated to 3 letters  - Selected from list | mins, hrs |
| 8 | Currency unit | The default currency unit for calculating rental fee | Yes | Selected from list | VND, USD, EUR |

**Table: Input data for “reading bike information” usecase, “search for specific bike” alternative flow**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Data field** | **Description** | **Mandatory** | **Valid conditions** | **Example** |
| 1 | Search type | Type for searching with the provided keywords | No | Selected from list | ID, Name, License plate code |
| 2 | Search information | Information used to search for specific bike | Yes | - String  - No more than 32 characters | Thong Nhat Bike |

* + 1. **Output data**

**Table: Output data for displaying bike list**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Data field** | **Description** | **Display format** | **Example** |
| 1 | Bike name | Name of the bike | - String  - All caps | THONG NHAT BIKE |
| 2 | Image of the bike |  | Image |  |
| 3 | Current status | Current status of the bike, showing if it is being rented | - String | Rented  Free |
| 4 | Current dock | The dock where the bike is in | - String  - Can be empty when the bike is being rent | - My Dinh 12 Dock |
| 5 | Current battery | Current status of the battery; only for the Ebike type | - Percentage  - Integer | 60% |

**Table: Output data for displaying bike information**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Data field** | **Description** | **Display format** | **Example** |
| 1 | Bike name | Name of the bike | - String  - All caps | THONG NHAT BIKE |
| 2 | Bike type | Type of the bike | String | Normal bike |
| 3 | License plate code | The string of the bike’s license plate | - String  - All caps | BY-18-12345 |
| 4 | Bike barcode | The barcode assigned to the bike to lock/unlock | - String  - 12 digits | 863127556142 |
| 5 | Image of the bike |  | Image |  |
| 6 | Rental price | The price to rent the bike | - Float  - Dot as decimal separator  - 2 digits after decimal separator | 12.00 |
| 7 | Time unit | The default time unit used to calculate rental fee | - Abbreviated to 3 letters | mins, hrs |
| 8 | Currency unit | The default currency unit for calculating rental fee | - String  - 3 letters  - All caps | USD |
| 10 | Current status | Current status of the bike, showing if it is being rented | - String | Rented  Free |
| 11 | Current dock | The dock where the bike is in | - String  - Can be empty when the bike is being rent | - My Dinh 12 Dock |
| 12 | Current renter | Name of the renter | - String  - Can be empty when the bike is free | Nguyen Thu Thao |
| 13 | Current battery | Current status of the battery; only for the Ebike type | - Percentage  - Integer | 60% |
| 14 | Time rented | Total time being rented from creation | - Minutes  - Integer | 12 minutes |

**3.1.10 Post conditions**

None

### **Usecase specifications for “Check bike information”**

* + 1. **Usecase Code:** **UC-210**
    2. **3.2.2 Brief description**

This usecase describe interaction between Customer and EcoBike application when he wants to view information of bikes in the dock

* + 1. **Actors**
* Customer
  + 1. **Preconditions**
* Customer has entered the application successfully
  + 1. **Basic flow of events**

**Step 1:** Customer chooses a dock to display information

**Step 2:** EcoBike display list of bikes available in dock

**Step 3:** Customer choose a specific bike to view information

**Step 4:** EcoBike displays detail information about the bike and options for using the bikes

* + 1. **Alternative flows of events**

**Table: Alternative flow of “View bike” usecase**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Location** | **Conditions** | **Actions** | **Resume Location** |
| 1 | Step 2 | Customer searches for a specific bike | - Customer selects to search for a specific bike by name, ID or license plate  - Customer enters information to search |  |
| 1a |  | No bike matches the given search criteria | - EcoBike show errors notification | 3 |
| 1b |  | There is a bike matches given search criteria | - EcoBike fetches matched data and display results | 3 |

* + 1. **Activity diagram**

**Figure: Activity diagram of “View bike information” usecase**

**Diagram

Description automatically generated**

* + 1. **Input data**

None

* + 1. **Output data**

**Table: Output data for displaying bike list**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Data field** | **Description** | **Display format** | **Example** |
| 1 | Bike name | Name of the bike | - String  - All caps | THONG NHAT BIKE |
| 2 | Image of the bike |  | Image |  |
| 3 | Current dock | The dock where the bike is in | - String  - Can be empty when the bike is being rent | - My Dinh 12 Dock |
| 4 | Current battery | Current status of the battery; only for the Ebike type | - Percentage  - Integer | 60% |

**Table: Output data for displaying bike information**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Data field** | **Description** | **Display format** | **Example** |
| 1 | Bike name | Name of the bike | - String  - All caps | THONG NHAT BIKE |
| 2 | Bike type | Type of the bike | String | Normal bike |
| 3 | License plate code | The string of the bike’s license plate | - String  - All caps | BY-18-12345 |
| 4 | Image of the bike |  | Image |  |
| 5 | Rental price | The price to rent the bike | - Float  - Dot as decimal separator  - 2 digits after decimal separator | 12.00 |
| 6 | Time unit | The default time unit used to calculate rental fee | - Abbreviated to 3 letters | mins, hrs |
| 7 | Currency unit | The default currency unit for calculating rental fee | - String  - 3 letters  - All caps | USD |
| 8 | Current status | Current status of the bike, showing if it is being rented | - String | Rented  Free |
| 9 | Current dock | The dock where the bike is in | - String  - Can be empty when the bike is being rent | - My Dinh 12 Dock |
| 10 | Current battery | Current status of the battery; only for the Ebike type | - Percentage  - Integer | 60% |

**3.2.10 Postconditions**

None

### **Usecase specifications for “Rent a bike”**

* + 1. **Use case code: UC240**
    2. **Brief description**

This use case describes the interaction between user and EcoBike application when user wants to rent a bike from the system

* + 1. **Actor**
* User
  + 1. **Preconditions**
* User successfully logged in to the system
  + 1. **Basic Flow of Event**

**Step 1:** User scans the barcode on the lock using EcoBike app

**Step 2:** EcoBike app shows bike information, including:

**Step 3:** The user chooses to deposit

**Step 4:** EcoBike app calls "DEPOSIT" use case

**Step 5:** The lock is automatically opened, allowing user to use the bike

**Step 6:** EcoBike app starts counting the rental time

* + 1. **Alternative flows of events**

**Table: Alternative flow of Rent A Bike use case**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Location** | **Conditions** | **Actions** | **Resume location** |
| 1 | Any time before step 3 | User quits the renting process | EcoBike asks for confirmation |  |
| 1a |  | User accepts to quit | EcoBike displays main page |  |
| 1b |  | User refuses to quit | EcoBike shows current page | At step where the event happens |

**3.3.7 Activity diagram**

**Figure: Activity diagram of “Rent a bike” usecase**

Diagram

Description automatically generated

* + 1. **Input data**

None

**3.3.9 Output data**

**Table: Output data of displaying bike information**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Data fields** | **Description** | **Display format** | **Example** |
| 1. | Name |  | - Right alignment | NGUYEN THANH LONG |
| 2. | Bike type |  | - Right alignment | Electric bike |
| 3. | License plate |  | - XX-XX111.11  - Right alignment | 29-D1371.68 |
| 4. | Bike image |  |  |  |
| 5. | Bike status |  | - Right alignment | free |
| 6. | Current battery percentage | Current battery of the bike | - Positive integer  - Follow by %  - Right alignment | 50% |

* + 1. **Post condition**
* User either successfully rents a bike or not

### **Usecase specifications for “Deposit”**

**3.4.1 Usecase code:** **UC-310**

**3.4.2 Brief description**

This use case describes the interaction between user, interbank and EcoBike application when user proceeds to deposit when renting a bike from the system

**3.4.3 Actors**

- Customer

**3.4.4 Preconditions**

None

**3.4.5 Basic flow of events**

**Step 1:** User chooses a payment method for the transaction

**Step 2:** User provides card information and transaction content

**Step 3:** EcoBike app checks the input

**Step 4:** EcoBike app calculates the deposit

**Step 5:** User confirms the transaction

**Step 6:** EcoBike app sends request to deducts money from user’s credit card

**Step 7:** Interbank deducts money from user’s credit card

**Step 8:** EcoBike app saves the transaction information

**Step 9:** EcoBike app sends transaction information to the user

**3.4.5 Alternative flow of events**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Location** | **Conditions** | **Actions** | **Resume location** |
| 1 | Any time before step 4 | User quits the depositing process | EcoBike app asks for confirmation |  |
| 1a |  | User accepts to quit | EcoBike app displays renting page |  |
| 1b |  | User refuses to quit | EcoBike app shows current page | At step where the event happens |
| 2 | At step 3 | User lefts mandatory field blank | EcoBike app requires user to fill all the mandatory field | At step 2 |
| 3 | At step 3 | User’s inputs is in wrong format | EcoBike app requires user to refill the fields | At step 2 |
| 4 | At step 7 | The card information is invalid | EcoBike app notifies that the card information is invalid | At step 2 |
| 5 | At step 7 | The balance is not enough | EcoBike app notifies that the balance is not enough | At step 2 |

**3.4.6 Activity diagram**

**Figure: Activity diagram of “Deposit” usecase**

**Diagram

Description automatically generated**

**3.4.8 Input data**

**Table: Input data of card information**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Data fields** | **Description** | **Mandatory** | **Valid condition** | **Example** |
| 1 | Card holder name |  | Yes |  | NGUYEN THANH LONG |
| 2 | Card number |  | Yes |  | 1231 4561 7891 1111 |
| 3 | Expiration date |  | Yes | Consist of month and last 2 digits of year only | 01/23 |
| 4 | Security code |  | Yes |  | 123 |

**3.4.9 Output data**

**Table: Output data of transaction information**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Data fields** | **Description** | **Display format** | **Example** |
| 1 | Payer |  | - Right alignment | NGUYEN THANH LONG |
| 2 | Payee |  | - Right alignment | VietinBank |
| 3 | Amount |  | - Comma separator | 700,000 |

* + 1. **Post conditions**

User successfully deposits the bike

### **Usecase specifications for “Return a bike”**

**3.5.1 Usecase code: UC-250**

**3.5.2 Brief description**

This use case describes the interaction between the Eco-Bike software with the customer and the Interbank when the customer desires to return the bike they rented

**3.5.3 Actors**

- Customer

**3.5.4 Preconditions**

- Customer has successfully launch the app

**3.5.5 Basic flow of events**

**Step 1:** The customer requests to return the bike

**Step 2:** The software displays list of docks to choose

**Step 3:** The customer chooses a dock to return bike

**Step 4:** EcoBike calls “RETURN DEPOSIT” usecase

**Step 5:** EcoBIke calls “PAY FOR BIKE RENTAL” uscase

**3.5.6 Alternative flow**

**Table: Alternative flow of “Return a bike” usecase**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Location** | **Conditions** | **Actions** | **Resume location** |
| 1 | At step 3 | User quits the renting process | EcoBike asks for confirmation |  |
| 1a |  | User accepts to quit | EcoBike ends usecase |  |
| 1b |  | User refuses to quit | EcoBike continue the usecase main flow | At step where the event happens |
| 2 | At step 1 | If the customer hasn’t rent any bike | EcoBike notifies errors and ends usecase |  |

**3.5.7 Activity diagram**

**Figure: Activity diagram of “Return a bike” usecase**

Diagram

Description automatically generated

**3.5.8 Input data**

None

**3.5.9 Output data**

None

**3.5.10 Postconditions**

None

### **Usecase specifications for “Return deposit”**

**3.6.1 Usecase code: UC-320**

**3.6.2 Brief description**

This use case describes the interaction between the Eco-Bike software and the Interbank when software returns the deposit

**3.6.3 Actors**

- Interbank

**3.6.4 Preconditions**

- The user confirms to return bike

- There is internet connection between EcoBike system and the interbank

**3.6.5 Basic flow of events**

**Step 1:** The software asks the bank to return the deposit

**Step 2:** The interbank processes the transaction

**3.6.6 Alternative flow**

None

**3.6.7 Activity diagram**

**Figure: Activity diagram of “Return deposit” usecase**

Diagram

Description automatically generated

**3.6.8 Input data**

None

**3.6.9 Output data**

None

**3.6.10 Postconditions**

None

### **3.7 Usecase specifications for “Pay for bike rental”**

**3.7.1 Usecase code: UC-330**

**3.7.2 Brief description**

This use case describes the interaction between the Eco-Bike software with the customer and the Interbank when the customer pays for the rental bike

**3.7.3 Actors**

- Interbank

**3.7.4 Preconditions**

- There is internet connection from EcoBike system to the interbank

- The “RETURN DEPOSIT” usecase returns successfully

**3.7.5 Basic flow of events**

**Step 1:** The software calculates the rental fee

**Step 2:** The software asks the bank to pay the fee

**Step 3:** The interbank processes the transaction

**Step 4:** The software saves the transaction info

**Step 5:** The software displays the successful transaction notification

**Step 6:** The software sends an emails of transaction info to the customer

**3.7.6 Alternative flow**

**Table: Alternative flow of “Pay for bike rental”**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Location** | **Conditions** | **Actions** | **Resume location** |
| 1 | At step 1 | Customer doesn’t have to pay | EcoBike notifies that the customer doesn’t have to pay the fee and the event ends | Step 6 |
| 2 | At step 3 | The balance is not enough | The software notifies that the balance is not enough and calls the sub use case “Update Payment Method” | Step 2 |

**3.7.7 Activity diagram**

**Figure: Activity diagram of “Pay for bike rental” usecase**

Diagram

Description automatically generated

**3.7.8 Input data**

None

**3.7.9 Output data**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Data fields** | **Description** | **Display format** | **Example** |
| 1 | Receiver’s name |  |  | DO MINH HIEU |
| 2 | Sender’s name |  |  | Eco-Bike System |
| 3 | Issuing bank |  |  | VietinBank |
| 4 | Total | Total amount of money corresponding the rental time interval | - Comma for thousands separator  - Positive integer  - Right alignment | 50,000 |
| 5 | Currency |  |  | VND |
| 6 | Transaction description |  |  | Pay for rent a bike |

**3.7.10 Postconditions**

None

### **3.8 Usecase specifications for “Update payment method”**

**3.8.1 Usecase code: UC-331**

**3.8.2 Brief description**

This use case describes the interaction between the Eco-Bike software and the customer when the software asks the customer to update the payment method

**3.8.3 Actors**

- Customer

**3.8.4 Preconditions**

- EcoBike recognizes an error occurs when customer tries to pay for bike rental

**3.8.5 Basic flow of events**

**Step 1:** The software displays the screen of payment method

**Step 2:** The customer updates the transaction information accordingly

**Step 3:** The software checks the validity of newly entered information

**3.8.6 Alternative flow**

**Table: Alternative flow of “Update payment method”**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Location** | **Conditions** | **Actions** | **Resume location** |
| 1 | At step 3 | There is invalid information | EcoBike notifies about the errors fields and asks user to input again | Step 1 |
| 2 | At step 3 | The mandatory fields are blank | EcoBike notities about the mandatory fields and requires users to input them | Step 1 |

**3.8.7 Activity diagram**

**Figure: Activity diagram of “Update payment method” usecase**

Diagram

Description automatically generated

**3.8.7 Input data**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Data fields** | **Description** | **Mandatory** | **Valid condition** | **Example** |
| 1. | Card holder name |  | Yes |  | NGUYEN THANH LONG |
| 2. | Card number |  | Yes |  | 1231 4561 7891 1111 |
| 3. | Expiration date |  | Yes | Consist of month and last 2 digits of year only | 01/23 |
| 4. | Security code |  | Yes |  | 123 |
| 5 | Issuing bank |  | Yes |  | Viettinbank |

**3.8.8 Output data**

None

## **4. Supplementary specification**

### **4.1 Functionality**

- For every action users take when use the application, the program will print on the console about what users are doing and it happens in which classes.

- When there is an error, a message must be display and there is a difference between system‘s error, database‘s error and user’s error.

- General displaying format:

* For integer, comma for thousands separator
* For number, right alignment
* Font: Arial 14, black
* White background

### **4.2 Usability**

- The system allows novice users to use without any training

- The system must be able to use 24/7, serves 100 users at the same time without noticeable loss of performance

- There need to be a detailed guide for user’s error so that they can know how to fix it.

### **4.3 Reliability**

- The system must be able to operates in an average of 200 hours per week without failure. It also must be repaired within 02 hours after any typical failure

- The system should run smoothly, automatically and trustworthily.

### **4.4 Performance**

- The response time for the system is 01 second at normal time and 02 seconds during a peak load

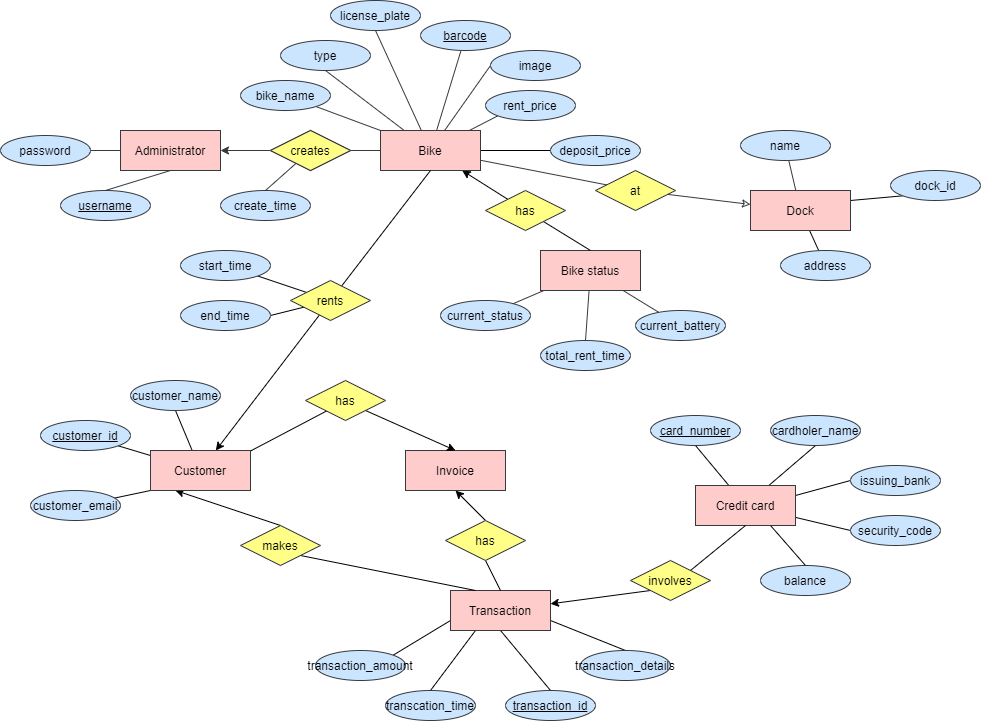
- The response time for performing any transaction must not exceed 01 second

### **4.5 Supportability**

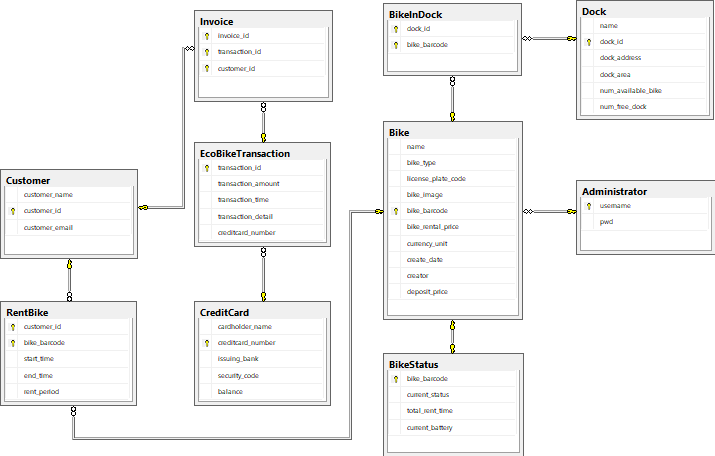
- Any error or failure should be quickly noticed and repaired.

### **4.6 Database design**

**Figure: ER-Diagram for EcoBike system’s database**



**Figure: Implementation of EcoBike system’s database**



### **4.7 Other requirements**

- The system is platform-independent

- All information about user’s credit card must be secured