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

**Software Requirement Specifications**

**EcoBikeRental Application**

***Subject: ITSS Software Development***

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

**Group number: 15**

**Group participants**

|  |  |
| --- | --- |
| **Name** | **Student ID** |
| **Vu Hong Quang** | **20205192** |
| **Vu Xuan Quy** | **20205193** |
| **Bui Van Thanh** | **20200585** |
| **Nguyen Hai Son** | **20205168** |

**TABLE OF CONTENTS**

[1. Introduction 3](#_Toc2123473032)

[1.1 Objective 3](#_Toc756038725)

[1.2 Scope 3](#_Toc1416475280)

[1.3 Glossary 3](#_Toc1243152862)

[2. Overall description 4](#_Toc284017403)

[2.1 Actors 4](#_Toc737864721)

[2.2 Use case diagram 5](#_Toc149517663)

[2.3 Business processes 5](#_Toc1505188388)

[3. Detailed requirement 6](#_Toc341148208)

[3.1 Usecase specifications for “Select dock” 7](#_Toc298829142)

[3.2 Usecase specifications for “Choose credit card” 8](#_Toc2117798337)

[3.3 Usecase specifications for “Check bike information”” 10](#_Toc1717750565)

[3.4 Usecase specifications for “Rent a bike” 12](#_Toc1425971082)

[3.6 Usecase specifications for “Return bike” 15](#_Toc1026237125)

[3.7 Usecase specifications for “Pay for rental fee” 17](#_Toc592446699)

[3.8 Usecase specifications for “Check bike information” 18](#_Toc310823967)

[4. Supplementary specification 20](#_Toc1529984593)

[4.1 Functionality 20](#_Toc1085511961)

[4.2 Usability 20](#_Toc410434842)

[4.3 Reliability 20](#_Toc510636973)

[4.4 Performance 21](#_Toc945993102)

[4.5 Supportability 21](#_Toc1668535974)

[4.6 Database design 21](#_Toc5363211)

[4.7 Other requirements 21](#_Toc630354078)

## **1. Introduction**

### **1.1 Objective**

The objective of this document is to provide an extensive overview of the EcoBikeRental system. It aims to elucidate the system's goals, functionalities, interfaces, operational scope, and responses to external influences. The intended audience for this document comprises both stakeholders and developers involved in the system.

### **1.2 Scope**

The EcoBikeRental system is designed specifically for EcoPark, featuring an hourly bike rental service with automated docking stations. Its primary purpose is to facilitate a seamless experience for customers seeking bike rentals, as well as to streamline administrative tasks for the system's supervisors.

When a customer accesses the EcoBikeRental application, it presents a map showing nearby bike stations based on the user's current location. The app allows users to select or search for a specific bike station, revealing relevant information such as distance and estimated walking time to that particular dock. Simultaneously, the application displays a list of available bikes at the selected dock, enabling customers to explore detailed information and current states of each bike before making a rental decision.

To initiate a bike rental, the customer scans the bike's barcode using the application, which unlocks the bike. The app then presents further details about the selected bike and calculates the rental deposit. The transaction is completed using credit cards linked to an interbank system. Throughout the rental period, the bike's status is continually updated to provide customers with real-time information.

When returning the bike, the customer securely places it in an empty dock and locks it. EcoBikeRental automatically computes the total charges based on the rental duration and generates a rental transaction record, which is sent to the customer via email.

The interbank system acts as a financial intermediary, facilitating payment processes for EcoBikeRental. It handles the rental deposit before the bike is rented and manages rental fees once the customer completes the rental. If the rental period is shorter than a predefined duration, the interbank system returns the customer's deposit, with this process being executed by EcoBikeRental.

On the administrator's side, there is the capability to manage the bike inventory within the system. This includes adding new bikes to the system by providing relevant bike information for validation and creating a new bike profile for tracking. Administrators can access lists of bikes and view detailed information about each bike. Moreover, they can update bike details or remove bikes from the system if they are no longer in use.

In summary, the EcoBikeRental system serves as an efficient solution for EcoPark, offering customers convenient bike rentals through automated docking stations while empowering administrators with tools to manage the bike inventory effectively.

### **1.3 Glossary**

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

|  |  |
| --- | --- |
| **Term** | **Definition** |
| Administrator | The person who uses EcoBikeRental application system for the purposes of monitoring list of bicycles in the system |
| Admin | Administrator |
| Bicycle | The transportation means to be rent in this application system |
| Bike | Bicycle |
| Card number | The ID number of the credit card |
| Cardholder name | The name of the owner of the credit card |
| Credit card | A card connected to the interbank, used for performing transaction |
| Customer | The person who uses EcoBikeRental 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, 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 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 | 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 EcoBikeRental 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 EcoBikeRental system. The activities of the customers with the application are adding, viewing and updating bike information, and deleting bike from the system.

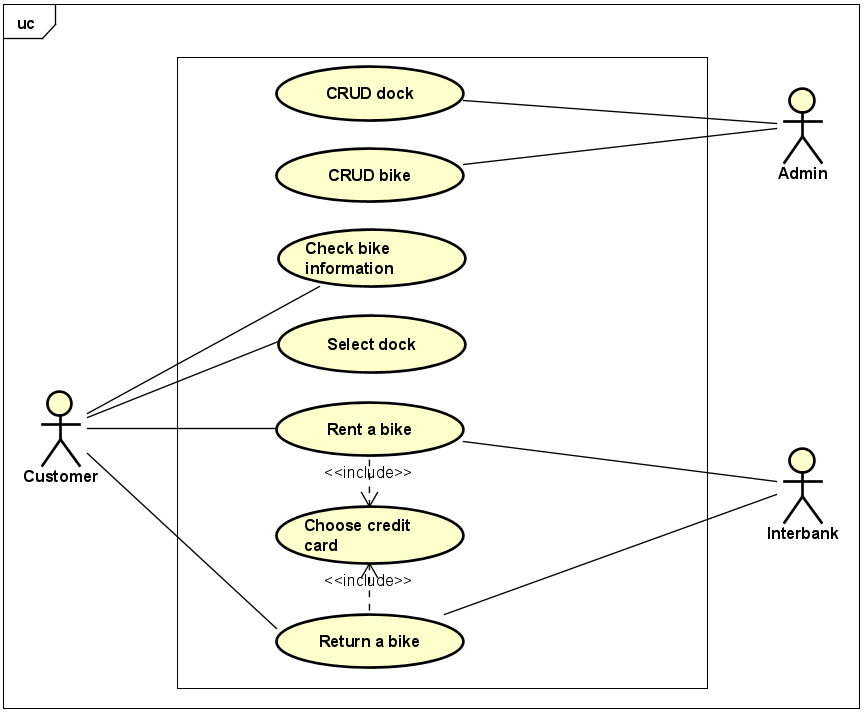
*2.1.3 Interbank*

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

### **2.2 Use case diagram**

The below diagrams illustrate the overall use case of the actors on the EcoBikeRental system

**Figure: Use case diagram of EcoBikeRental system**



### **2.3 Business processes**

The bellow diagram shows the business process for the EcoBikeRental system

Diagram

Description automatically generated

*Figure 2: Business process diagram for EcoBikeRental system*

## **3. Detailed requirement**

### **3.1 Usecase specifications for “Select dock”**

**3.1.1 Usecase Code: UC301**

**3.1.2 Brief description**

This usecase allows the customer to select a dock to view dock information and all docking point

**3.1.3 Actors**

Customer

**3.1.4 Precondition**

Customer has download the application successfully

**3.1.5 Basic flows of event**

**Step 1**: The customer opens the Bike app

**Step 2:** The system displays a list of docks and each dock's availability information

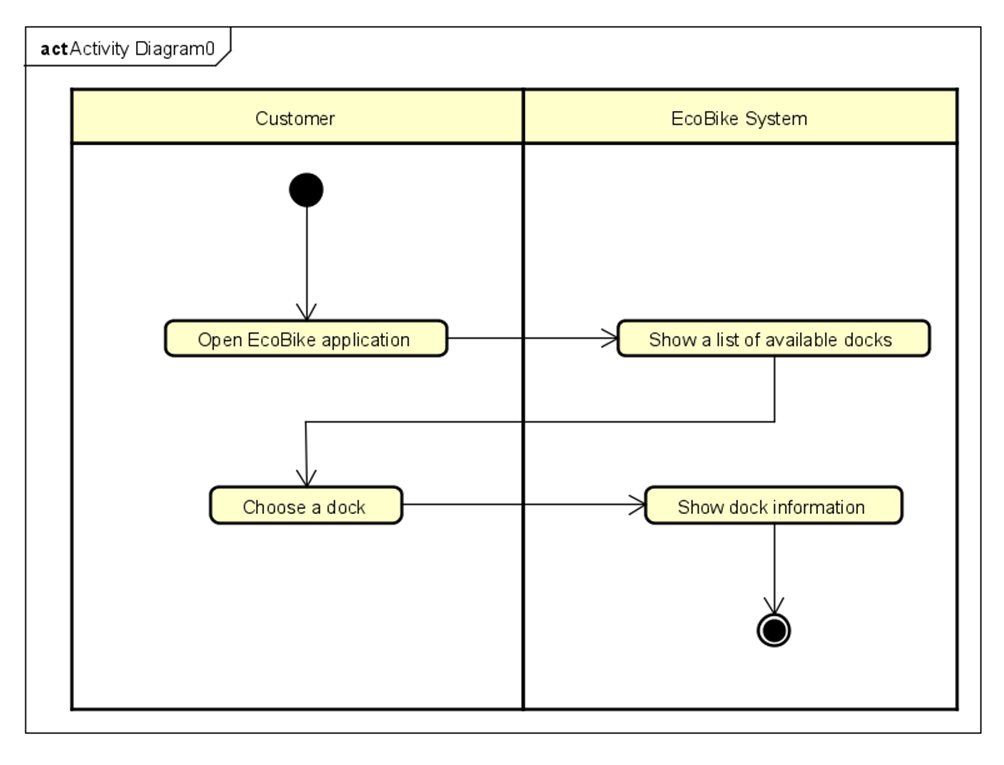
**Step 3:** The customer selects a dock from the list

**Step 4:** The system show dock detailed information, each docking point may have a bike or empty

**3.1.6 Alternative flows**

None

**3.1.7 Activity diagram**



**3.1.8 Priority:**

High

**3.1.9 Frequency of Use:**

High

**3.1.10 Notes and Issues:**

Dock availability needs to be updated in real time in case another customer returns a bike while the user is selecting a dock

**3.1.11 Postcondition:**

None

### **3.2 Usecase specifications for “Choose credit card”**

**3.2.1 Usecase Code: UC101**

**3.2.2 Brief description**

This usecase describes interaction between Customer and EcoBikeRental application when customer wants to select the credit card that they would do their payments

**3.2.3 Actors**

- Customer

**3.2.4 Precondition**

None

**3.2.5 Basic flows of event**

**Step 1**: Customer chooses a payment method for the transaction (bank or momo)

**Step 2:** Customer chooses pay by credit card

**Step 3:** EcoBikeRental displays credit card form

**Step 4:** Customer provides card information (serial number and bank’s name)

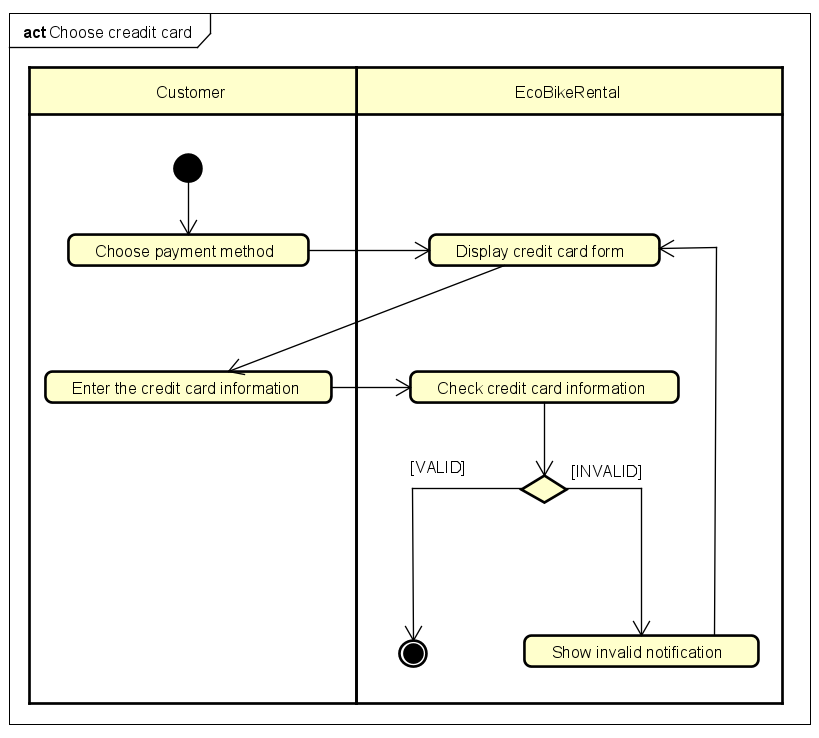
**Step 5:** EcoBikeRental checks the input information

**3.2.6 Alternative flows**

**Table: Alternative flow of “Choose credit card” use case**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Location** | **Conditions** | **Actions** | **Resume location** |
| 1 | Step 5 | Customer enters an invalid credit card | EcoBikeRental notifies that invalid credit card and automatically navigates to credit card form | At step 3 |

**3.2.7 Activity diagram**



**3.2.8 Input data**

**Table: Input data of card information**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Data fields** | **Description** | **Mandatory** | **Valid condition** | **Example** |
| 1 | Card holder name |  | Yes |  | VU HONG QUANG |
| 2 | Card number |  | Yes |  | 1234 5678 5431 1231 |
| 3 | Issuing bank |  | Yes |  | VietinBank |
| 4 | Security code |  | Yes |  | 123456 |

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

**3.3.1 Usecase Code:** **UC-210**

**3.3.2 Brief description**

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

**3.3.3 Actors**

* Customer

**3.3.4 Preconditions**

* Customer has entered the application successfully

**3.3.5 Basic flow of events**

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

**Step 2:** EcoBikeRental displays list of bikes available in dock

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

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

**3.3.6 Alternative flows of events**

None

**3.3.7 Activity diagram**

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

Diagram

Description automatically generated

**3.3.8 Input data**

None

**3.3.9 Output data**

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Data field** | **Description** | **Display format** | **Example** |
| 1 | Type | Type of the bike | - String | Normal bike |
| 2 | Image |  | - Image |  |
| 3 | Rental price | The price to rent the bike | - Integer | 20,000 |
| 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 | Type | Type of the bike | - String | Normal bike |
| 2 | License plate | The string of the bike’s license plate | - String | JP-12345 |
| 3 | Image |  | - Image |  |
| 4 | Rental price | The price to rent the bike | - Integer | 20,000 |
| 5 | Deposit | The deposit to rent the bike | - Integer | 400,000 |
| 6 | Time unit | The default time unit used to calculate rental fee | - String  - Abbreviated to 3 letters | min, hrs |
| 7 | Currency unit | The default currency unit for calculating rental fee | - String  - 3 letters  - All caps |  |
| 8 | Status | Status of the bike | - String | Rented  Free |
| 9 | Current dock | The dock where the bike is in | - String  - Can be empty when the bike is being rent | - Ecopark 12 Dock |
| 10 | Current battery | Current status of the battery; only for the Ebike type | - Percentage  - Integer | 60 |
| 11 | Time limit | Maximum time the bike can be rented; only for the Ebike type | - Minutes  - Integer | 180 |

**3.3.10 Postconditions**

None

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

**3.4.1 Use case code: UC240**

**3.4.2 Brief description**

This use case describes the interaction between Customer and EcoBikeRental when customer wants to rent a bike from the system

**3.4.3 Actor**

* Customer

**3.4.4 Preconditions**

* Customer logged in the system successfully

**3.4.5 Basic Flow of Event**

**Step 1:** Customer enters the barcode of the bike customer want to rent

**Step 2:** EcoBikeRental displays bike information

**Step 3:** Customer chooses to rent

**Step 4**: EcoBikeRental calls “Choose credit card” use case

**Step 5:** EcoBikeRental displays the transaction information

**Step 6:** Customer provide information about type of tracker and content

**Step 7**: Customer confirms the transaction

**Step 8:** EcoBikeRental checks the balance

**Step 9:** EcoBikeRental checks information that customer provided

**Step 10:** EcoBikeRental sends request to deducts money from customer’s credit card

**Step 11:** Interbank deducts money from customer’s credit card

**Step 12:** EcoBikeRental saves the transaction

**Step 13:** EcoBikeRental notifies the customer the transaction is successful

**Step 14:** EcoBikeRental start counting the rent bike time

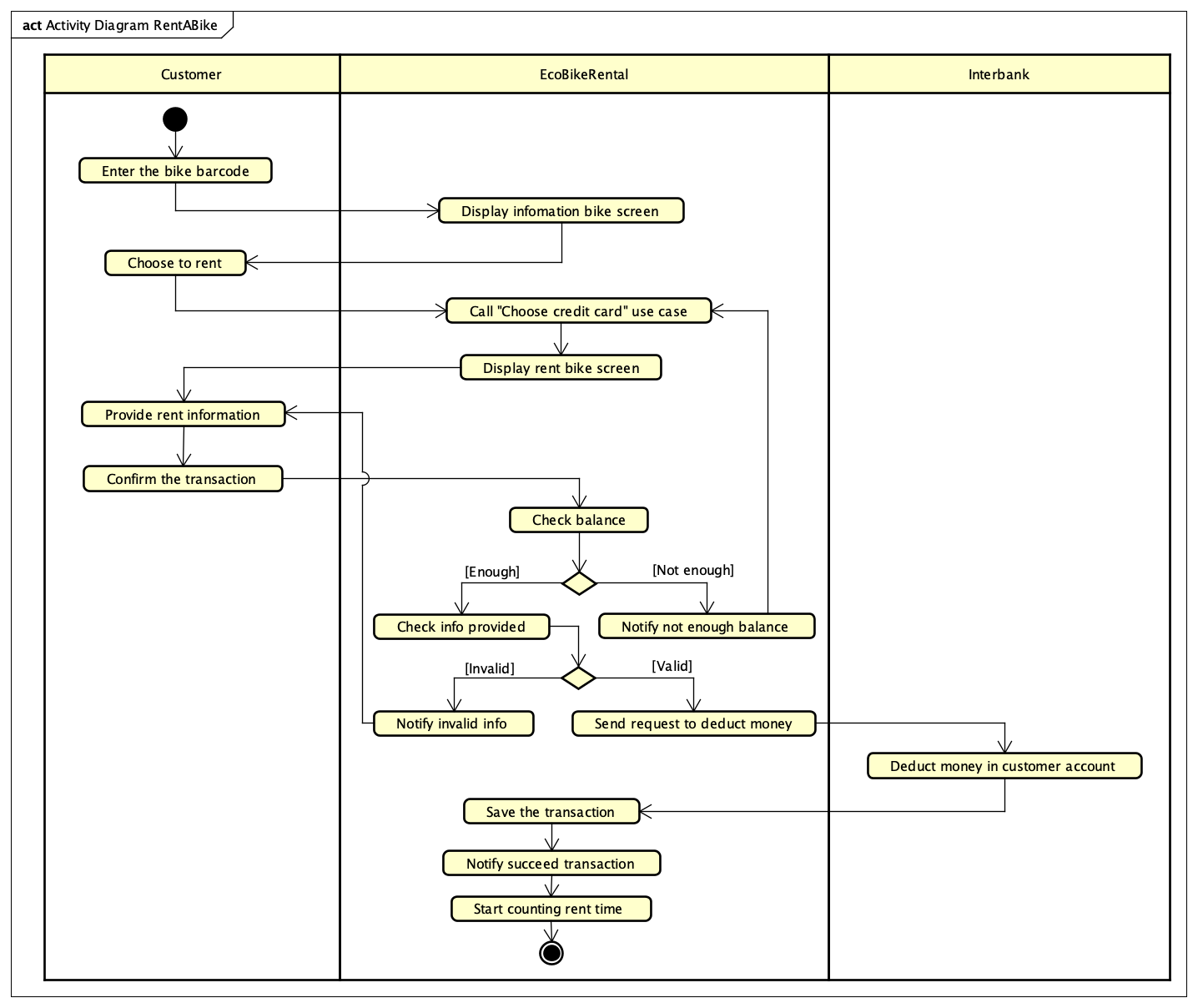
**3.4.6 Alternative flows of events**

**Table: Alternative flow of “Rent a bike” use case**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Location** | **Conditions** | **Actions** | **Resume location** |
| 1 | Any time before step 8 | Customer quits the renting process | EcoBikeRental return home page | Use case ends |
| 2 | At step 8 | The balance is not enough | EcoBikeRental notifies the customer that the balance is not enough | At step 4 |
| 3 | At step 9 | The information was provided is invalid | EcoBikeRental notifies the customer that the information was provided is invalid | At step 6 |

**3.4.7 Activity diagram**

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



**3.4.8 Input data**

**Table: Input data of enter the barcode**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Data fields** | **Description** | **Display format** | **Example** |
| 1 | Barcode | Barcode of the bike customer  chooses | **-** Integer | 12312321 |

**Table: Input data of provide rent information**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Data fields** | **Description** | **Display format** | **Example** |
| 1 | Type Tracker | Type of tracker use for the rent bike | **-** String | Minute |
| 2 | Transaction content | Content of the transaction write by customer | - String | Pay for rent bike |

**3.4.9 Output data**

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Data field** | **Description** | **Display format** | **Example** |
| 1 | Type | Type of the bike | - String | Normal bike |
| 2 | License plate | The string of the bike’s license plate | - String | JP-12345 |
| 3 | Image |  | - Image |  |
| 4 | Barcode | Barcode of the bike customer chooses | **-** Integer | 12312321 |
| 5 | Rental price | The price to rent the bike | - Integer | 20,000 |
| 6 | Deposit | The deposit to rent the bike | - Integer | 400,000 |
| 7 | Status | Status of the bike | - String | Rented  Free |
| 8 | Current dock | The dock where the bike is in | - String  - Can be empty when the bike is being rent | - Ecopark 12 Dock |
| 9 | Battery | Status of the battery; only for the Ebike type | - Percentage  - Integer  - Can be empty when the bike is not Ebike | 60 |

**Table: Output data of transaction information**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Data fields** | **Description** | **Display format** | **Example** |
| 1 | Card holder name | Name of customer on credit card | - String  - Uppercase | VU HONG QUANG |
| 2 | Cvv code | Id of the credit card | - Int | 123 |
| 3 | Balance | The balance of credit card | - Long | 1,000,000 |
| 4 | Card code | Code of the credit card | - Int | 1 |
| 5 | Transaction\_id | Id of the transaction | - Int | 345345345534 |
| 6 | CreatedAt | The time make the transaction | - Date | 2023-08-20T10:00:02 |
| 7 | Amount | Amount of transaction | - Long | 70,000 |
| 8 | Transaction content | Content of transaction | - String | Pay to rent bike |

**3.4.10 Post condition**

EcoBikeRental starts to count the rental time, customer returns to home page

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

**3.6.1 Usecase code: UC-250**

**3.6.2 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

**3.6.3 Actors**

- Customer

**3.6.4 Preconditions**

- Customer is renting at least one bike

**3.6.5 Basic flow of events**

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

**Step 2:** EcoBikeRental displays the list of docks to choose.

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

**Step 4:** EcoBikeRental calculates the rental fee.

**Step 5:** EcoBikeRental checks the balance to pay or not.

**Step 6:** Display the transaction information (rental fee, rental bike, rental time, start dock, returning dock, …)

Step 7: The customer confirms the transaction

**Step 8:** The software asks the bank to return the deposit (call “add money API”)

**Step 9:** EcoBikeRental calls “PAY FOR BIKE RENTAL” usecase

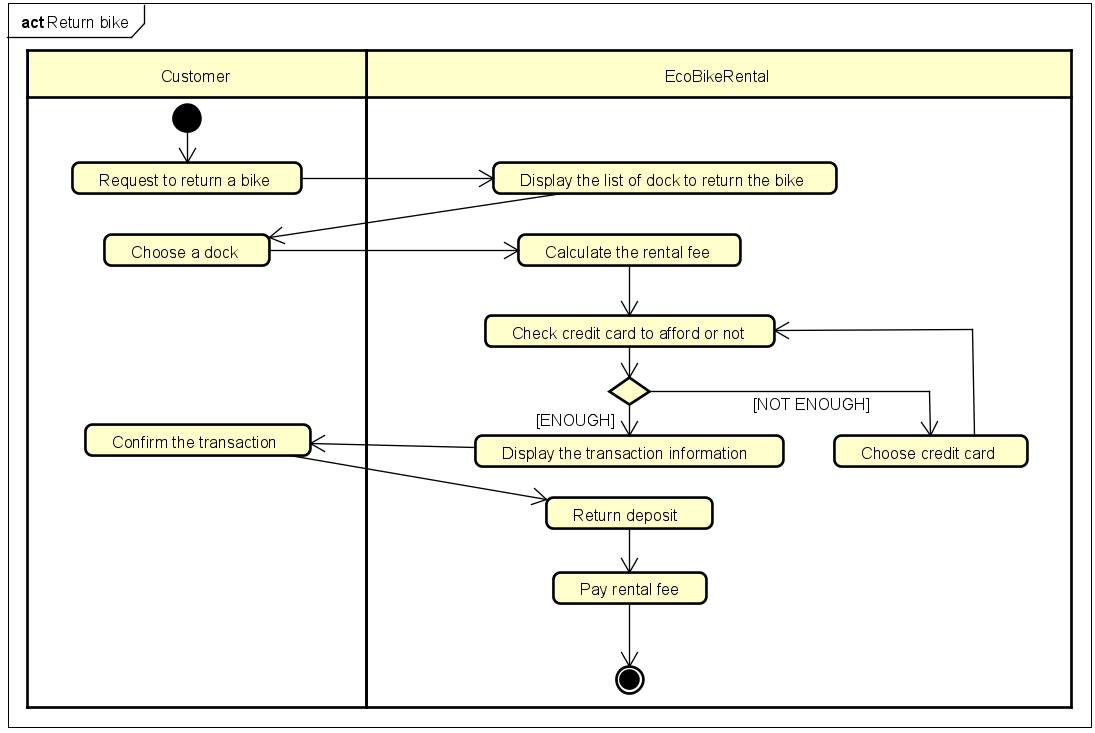
**3.6.6 Alternative flow**

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Location** | **Conditions** | **Actions** | **Resume location** |
| 1 | At step 6 | The amount is not enough to pay the rent | EcoBikeRental display a notification to annouce that the amount is not enough |  |
| 2 | At step 6 | The amount is not enough to pay the rent | Customer chooses another payment method | Use case “Choose credit card” |

**3.6.7 Activity diagram**

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



**3.6.8 Input data**

None

**3.6.9 Output data**

None

**3.6.10 Postconditions**

The transaction must show a notification

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

**3.7.1 Usecase code: UC-330**

**3.7.2 Brief description**

This use case describes the interaction between the EcoBikeRental 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 EcoBikeRental system to the interbank

**3.7.5 Basic flow of events**

**Step 1:** The EcoBikeRental asks the bank to pay the rental fee

**Step 2:** The EcoBikeRental saves the transaction info

**Step 3:** The EcoBikeRental displays the successful transaction notification

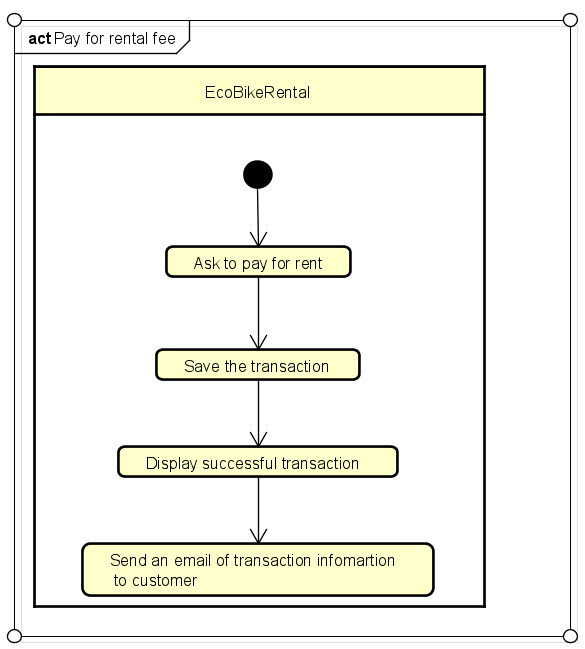
**Step 4:** The EcoBikeRental sends an email of transaction info to the customer

**3.7.6 Alternative flow**

None

**3.7.7 Activity diagram**

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



**3.7.8 Input data**

None

**3.7.9 Output data**

None

**3.7.10 Postconditions**

None

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

**3.8.1 Usecase code: UC-350**

**3.8.2 Brief description**

This use case describes interaction between the EcoBikeRental with the customer when the customer wants to check information about bike.

**3.8.3 Actors**

Customers

**3.8.4 Preconditions**

EcoBikeRental has some bike active in system.

**3.8.5 Basic flow of events**

Step 1: Customers click one bike to check information.

Step 2: EcoBikeRental display information about that bike.

Step 3: Customers click button ‘View Rent Info’ to check rent information

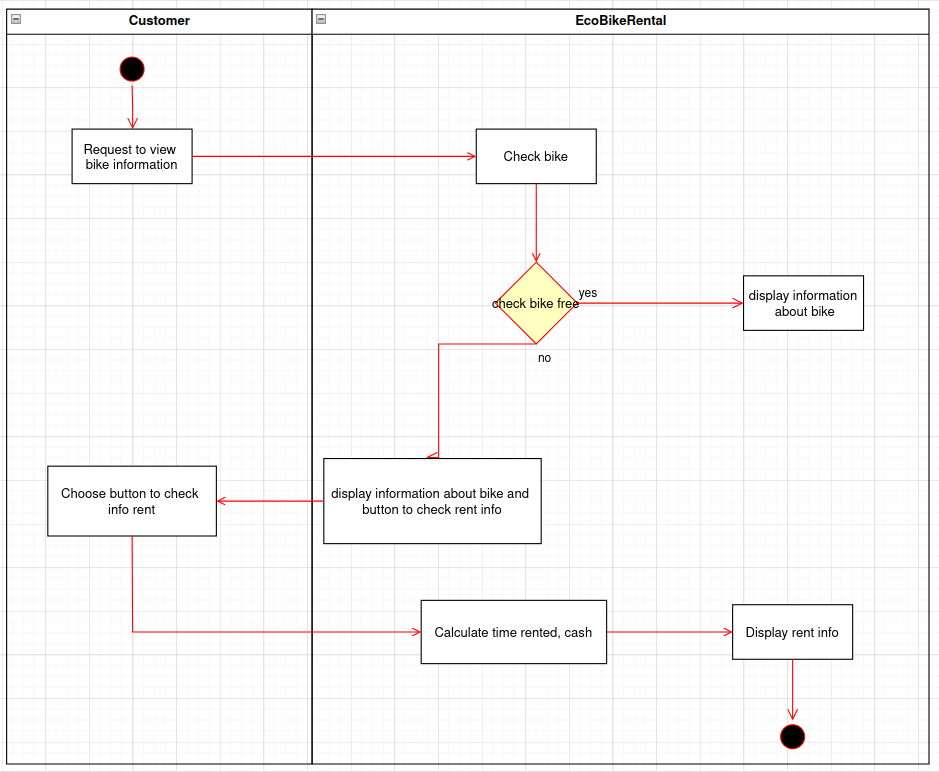
Step 4: EcoBikeRental display rent information of that bike.

**3.8.6 Alternative flow**

None

**3.8.7 Activity diagram**

**Figure: Activity diagram of “Check Information Bike” usecase**



**3.8.8 Input data**

**None**

**3.8.9 Output data**

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Data fields** | **Description** | **Display format** | **Example** |
| 1. | type | Type of bike | String | Standard bike |
| 2. | status | Status of bike | String | rented |
| 3. | battery | Battery of bike | Long | 100 |
| 4. | License plate | License plate of bike | String | 15-HP 6888 |
| 5. | deposit | Deposit of bike | Long | 50000 |
| 6. | barcode | Barcode of bike | String | 89d0660c-8c89-437e |
| 7. | description | Description of bike | String | Bike normal |

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Data fields** | **Description** | **Display format** | **Example** |
| 1. | Start time | Start time | Date | 2023-08-20 03:20:32 |
| 2. | cash | Cash | Long | 50000 |
| 3. | status | Status of tracker | String | stop |
| 4. | Time rented | Time rented | Long | 67 |

**3.8.10 Postconditions**

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**

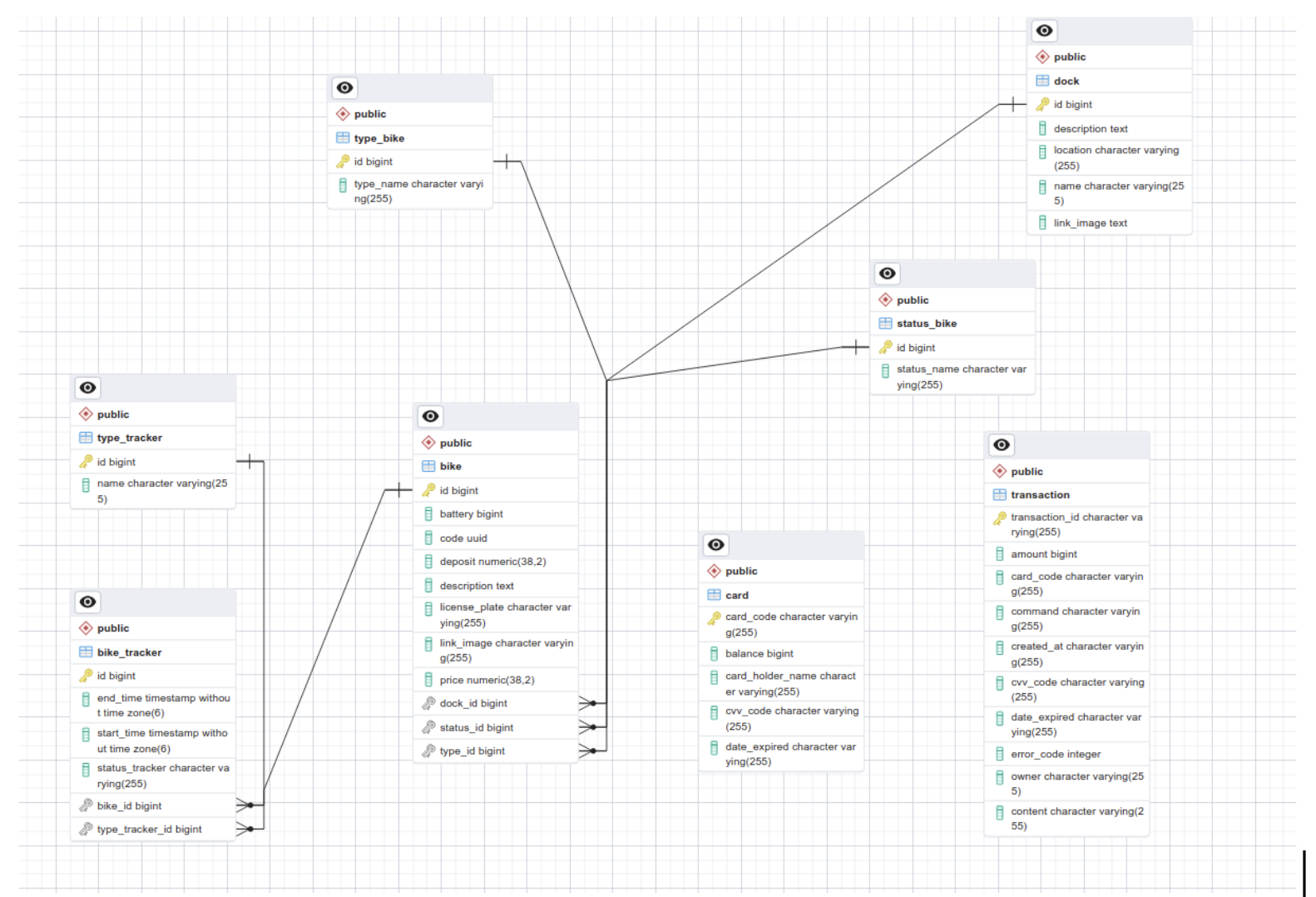
- The response time for the system is 01 second at normal time and 02 seconds during a peak load

- 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: Implementation of EcoBikeRental system’s database**

### **4.7 Other requirements**

- The system is platform-independent

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