

**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 11**

**Group members**

| <b>Name</b>            | <b>Student ID</b> |
|------------------------|-------------------|
| <b>Pham Tuan Long</b>  | <b>20205161</b>   |
| <b>Dang Trong Luat</b> | <b>20205187</b>   |
| <b>Dang Hoang Long</b> | <b>20200357</b>   |
| <b>Duong Duy Manh</b>  | <b>20205162</b>   |

## **1. Introduction**

The EcoBikeRental System is a pioneering web and mobile application set to redefine the realm of hourly bicycle rentals and returns within the Ecopark township. With a laser-focused approach on user needs, this meticulously crafted platform presents an effortlessly seamless and efficient solution for renting and returning bicycles at strategically located docking stations. Through the harmonious integration of rental management, payment processing, and supplementary features, the EcoBikeRental System emerges as an agent of transformation, empowering users to effortlessly partake in urban cycling.

Rooted in innovation and user-centric design, the EcoBikeRental System sets the stage for a revolutionary approach to urban mobility. Its core mission is to transcend conventional bike rental frameworks, offering an all-encompassing solution that converges convenience, mobility, and affordability. The system revolves around docking stations as pivotal interaction points, seamlessly connecting users with a world of biking possibilities.

The true essence of the EcoBikeRental System lies in its ability to seamlessly merge rental management, payment processing, and additional functionalities. This comprehensive synergy streamlines the rental and return processes into a fluid narrative, enabling the system to efficiently validate rental requests, compute charges, process payments, and promptly update bike availability. This orchestration, guided by the system's integrated components, underscores its transformative potential.

Ultimately, the EcoBikeRental System empowers users to engage in urban cycling with unprecedented ease. Through user-friendly web and mobile interfaces, users effortlessly access an array of available bicycles, each poised to facilitate real-world exploration with a digital touch. This user-centric design ethos aligns with broader sustainability goals, making the system a driving force in the evolution of eco-friendly urban transportation within the Ecopark township.

### **1.1. Scope**

The software's scope includes the following features:

- Docking Stations and Bike Information Display
- Bike Rental and Return Process

- Credit Card Payment Processing
- Additional Features (e.g., 24-hour pass, rental time pause)

## **2. Overall Requirements**

### **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 activities of the customers with the application are adding, viewing and updating bike information, and deleting bike from the system.

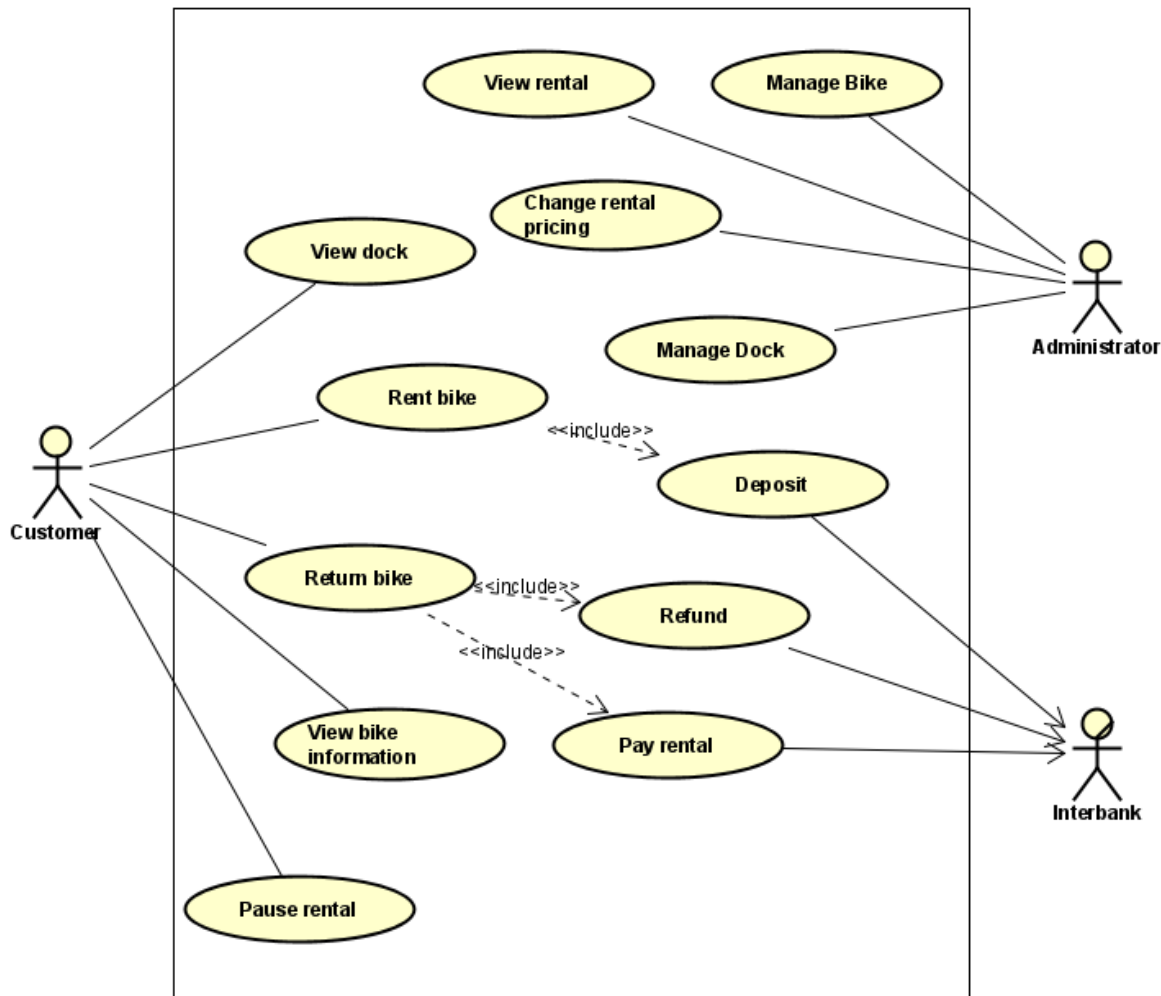
#### *2.1.3 Interbank*

The interbank is the actor who interacts with the system for the purpose of performing transactions issued by the customers. The activities 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**



### 3. Usecase Specification

#### 3.1 Administrator

##### 3.1.1: Brief Description

This use case describes the interaction between administrator and EcoBike application to manage bike, dock and pricing

##### 3.1.2: Actor

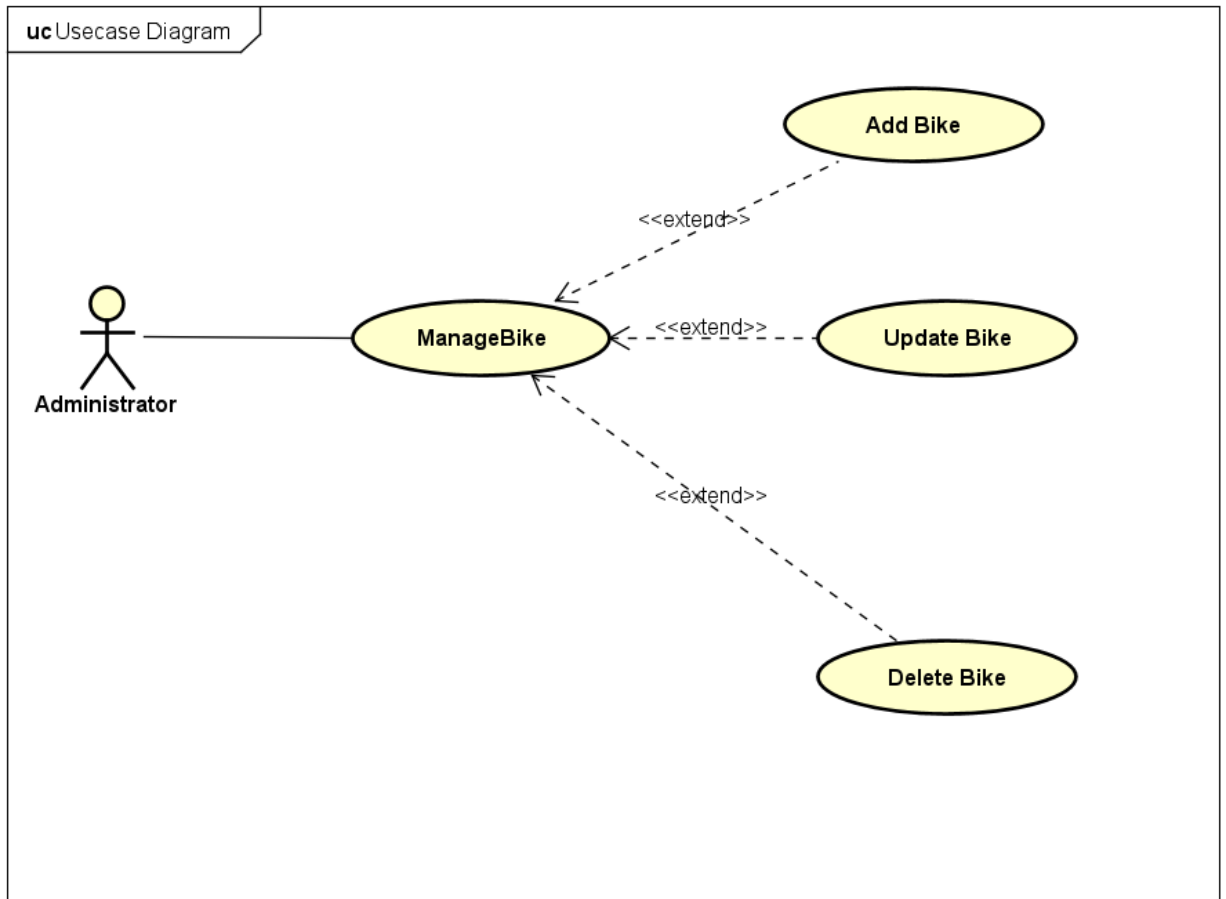
**Administrator**

##### 3.1.3 Preconditions

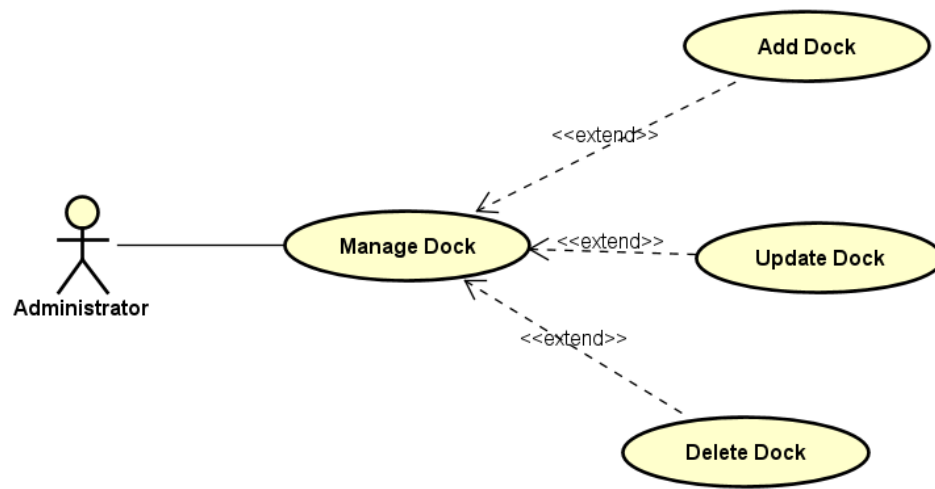
- User successfully logged in to the system as an Admin

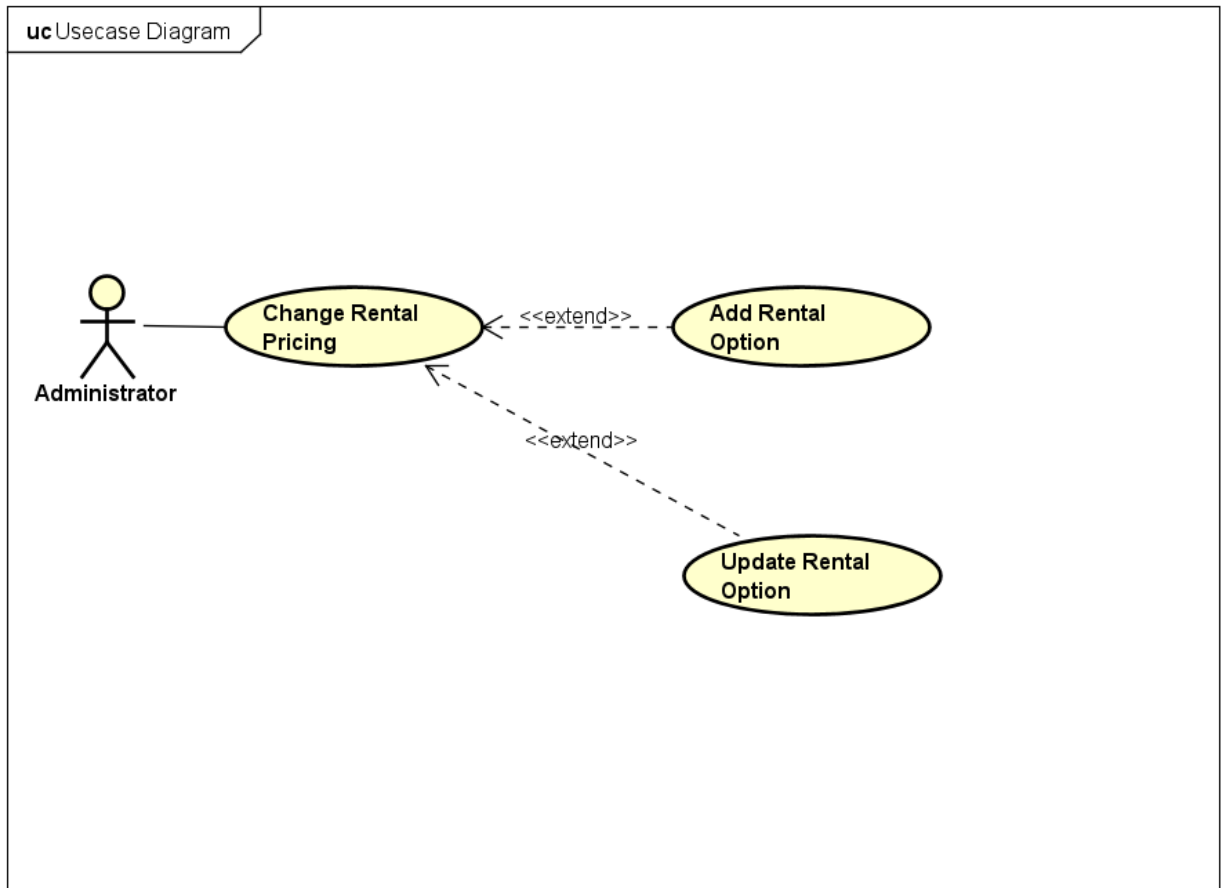
### 3.1.4: Usecase Diagram

- Usecase Diagram:



uc Usecase Diagram





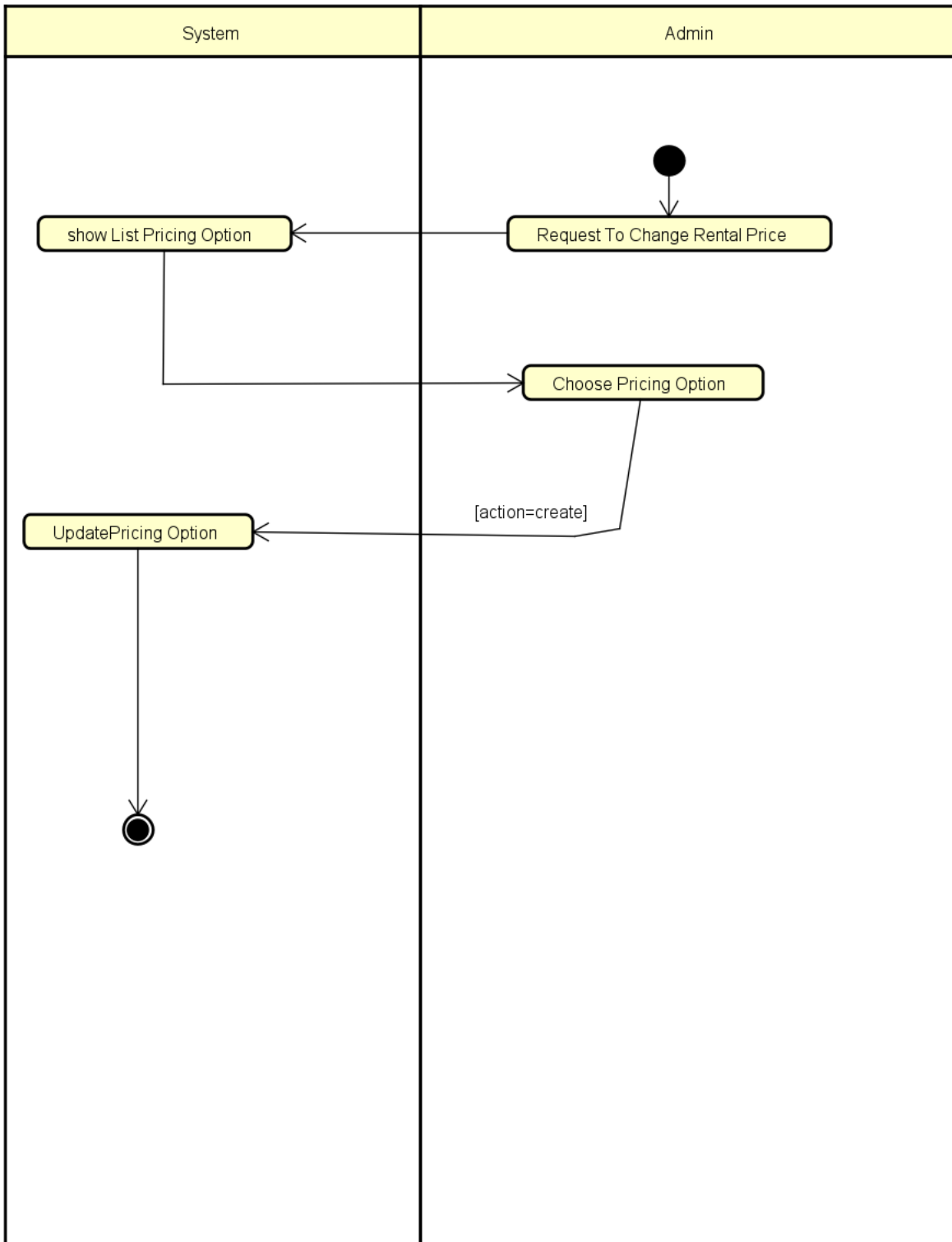
### 3.1.5 Basic Flow of Event:

- **Administrator Manage Bike:**
  - Administrator logs into the admin panel.
  - Administrator selects the "Manage Bikes" option.
  - System displays a list of existing bikes.
  - Administrator can choose to view, add, edit, or delete bikes.
- **Administrator Manage Dock:**
  - Administrator logs into the admin panel.
  - Administrator selects the "Manage Docks" option.
  - System displays a list of existing docks.
  - Administrator can choose to view, add, edit, or delete docks.
- **Administrator Manage Price:**
  - Administrator logs into the admin panel.
  - Administrator selects the "Manage Pricings" option.
  - System displays a list of existing pricings.
  - Administrator can choose to view, add, edit, or delete pricings.

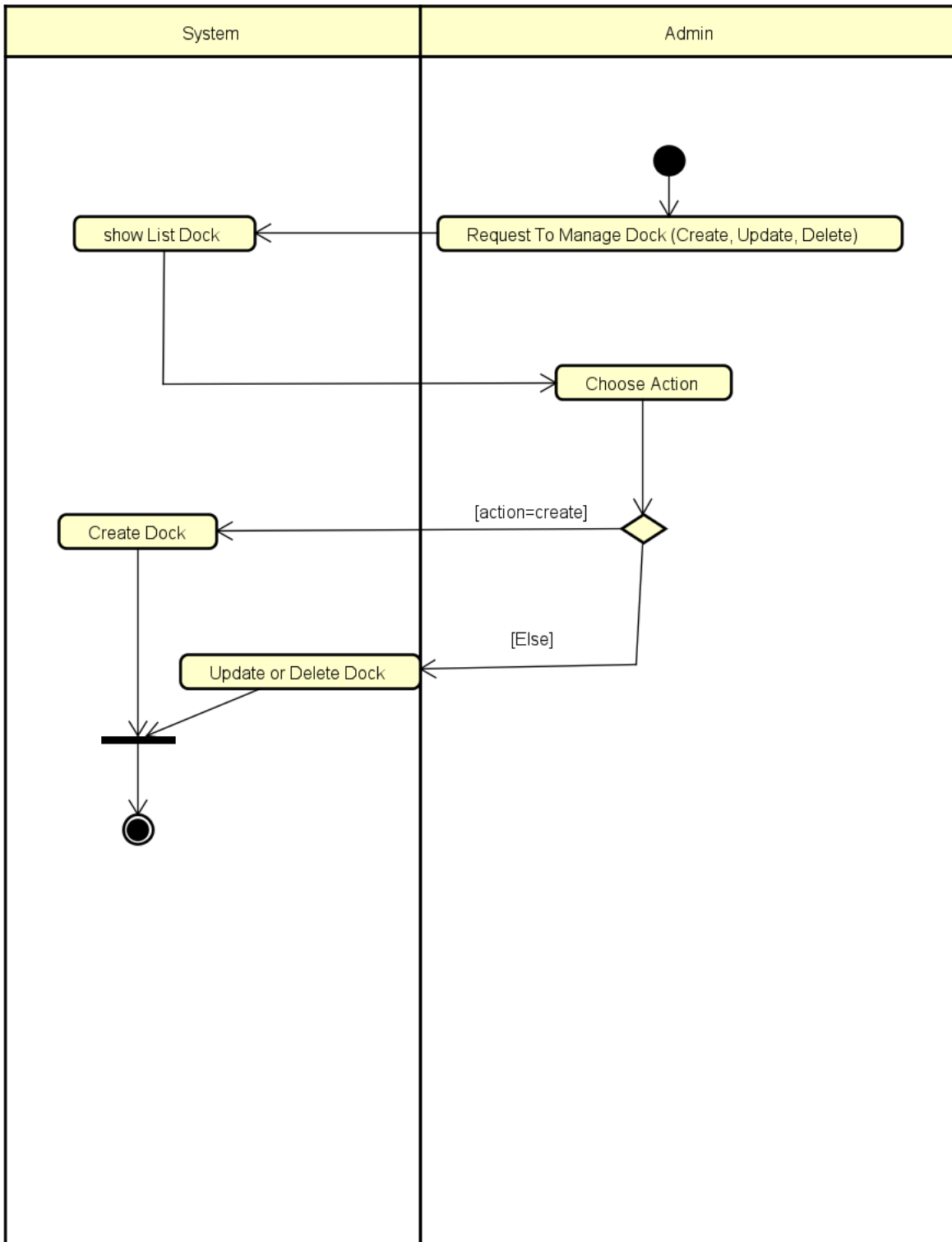
### **3.1.6 Activity Diagram:**

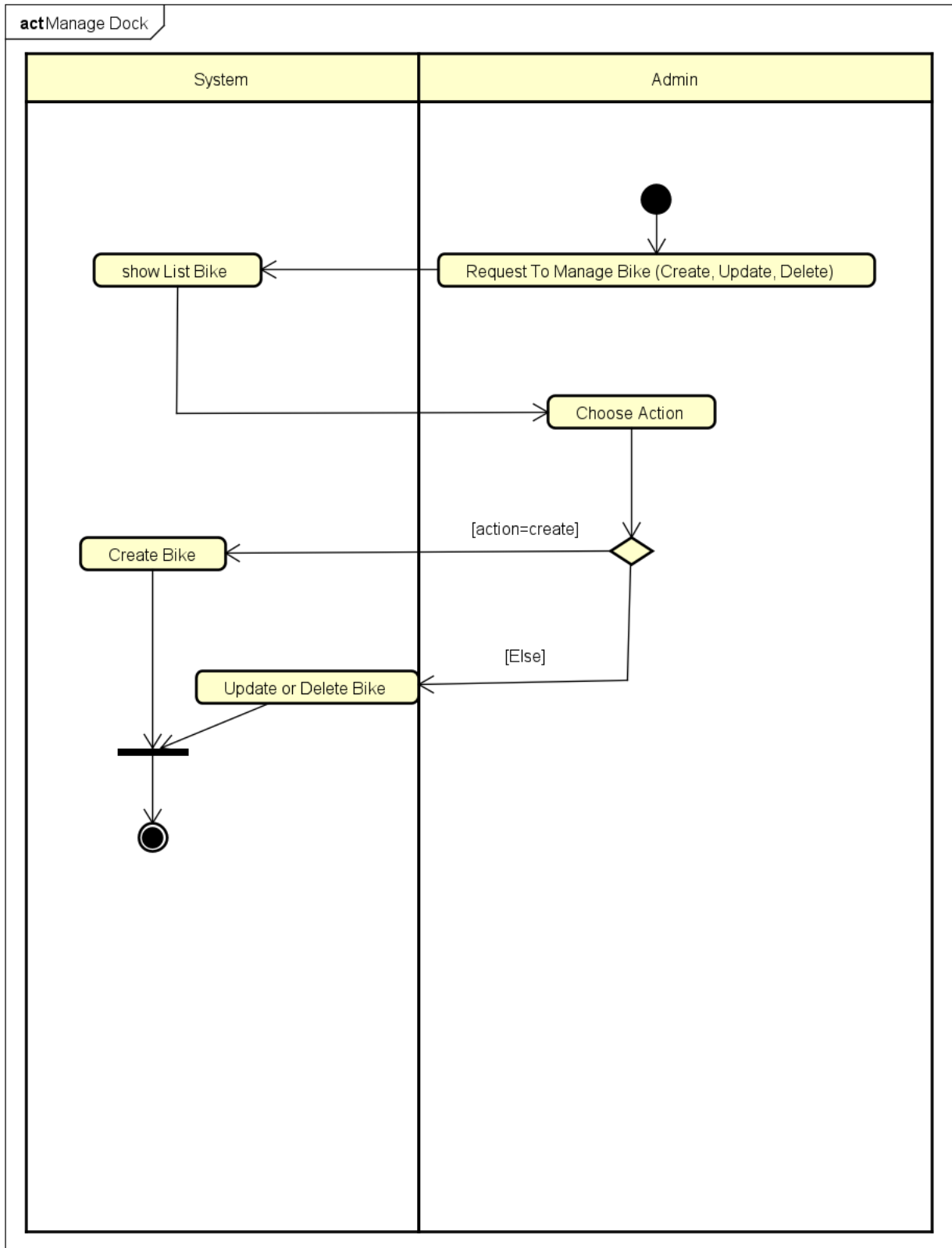


act Change Rental Price



actManage Bike





### 3.2 Rent A Bike

### 3.2.1 Usecase Code: UC-002

### 3.2.2 Brief Description:

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

### 3.2.3 Actor

- User

### 3.2.4 Preconditions

- User successfully logged in to the system

### 3.2.5 Basic Flow of Event

1. User scans the barcode on the lock using EcoBike application
2. EcoBike application shows bike information
3. The user chooses to deposit
4. EcoBike application calls "DEPOSIT" use case
5. The lock is automatically opened, allowing user to use the bike
6. EcoBike application starts counting the rental time

### 3.2.6 Alternative flows of events

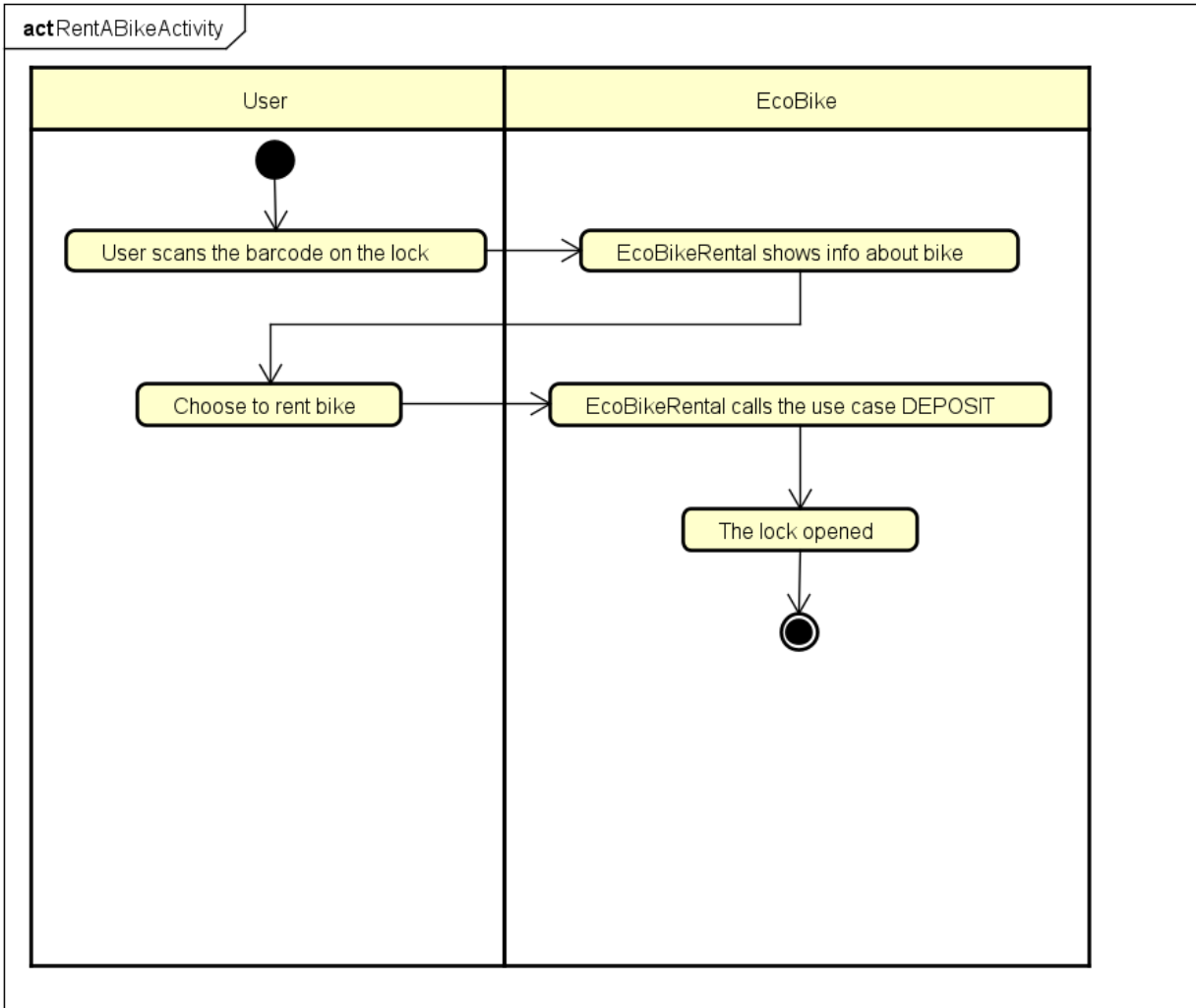
Table: Alternative flow of Rent A Bike use case

| No | Location               | Conditions                     | Actions                       | Resume location   |
|----|------------------------|--------------------------------|-------------------------------|-------------------|
| 1  | Any time before step 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 |
|--|--|--|--|---------------|

### 3.2.7 Activity Diagram

Figure: Activity diagram of “Rent a bike” usecase



### 3.2.8 Input Data

None

### 3.2.9 Output Data

Table: Output data of displaying bike information

| No | Data fields                | Description                 | Display format   | Example        |
|----|----------------------------|-----------------------------|--|----------------|
| 1  | Name                       |                             | - Right alignment  | DUONG DUY MANH |
| 2  | Bike type                  |                             | - Right alignment  | Electric Bike  |
| 3  | License type               |                             | - XX-XX111.11<br>- Right alignment                       | 73-D4738.31    |
| 4  | Bike image                 |                             |  |                |
| 5  | Bike status                |                             | - Right alignment  | Free           |
| 6  | Current battery percentage | Current battery of the bike | - Positive integer<br>- Follow by %<br>- Right alignment | 50%            |

### 3.2.10 Post condition

- User either successfully rents a bike or not

## 3.3 Deposit

### 3.3.1 Usecase code: UC-003

### 3.3.2 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

### 3.3.3 Actors

- Customer

### 3.3.4 Preconditions

None

### 3.3.5 Basic flow of events

1. User chooses a payment method for the transaction
2. User provides card information and transaction content
3. EcoBike app checks the input
4. EcoBike app calculates the deposit
5. User confirms the transaction

6. EcoBike app sends request to deducts money from user's credit card
7. Interbank deducts money from user's credit card
8. EcoBike app saves the transaction information
9. EcoBike app sends transaction information to the user

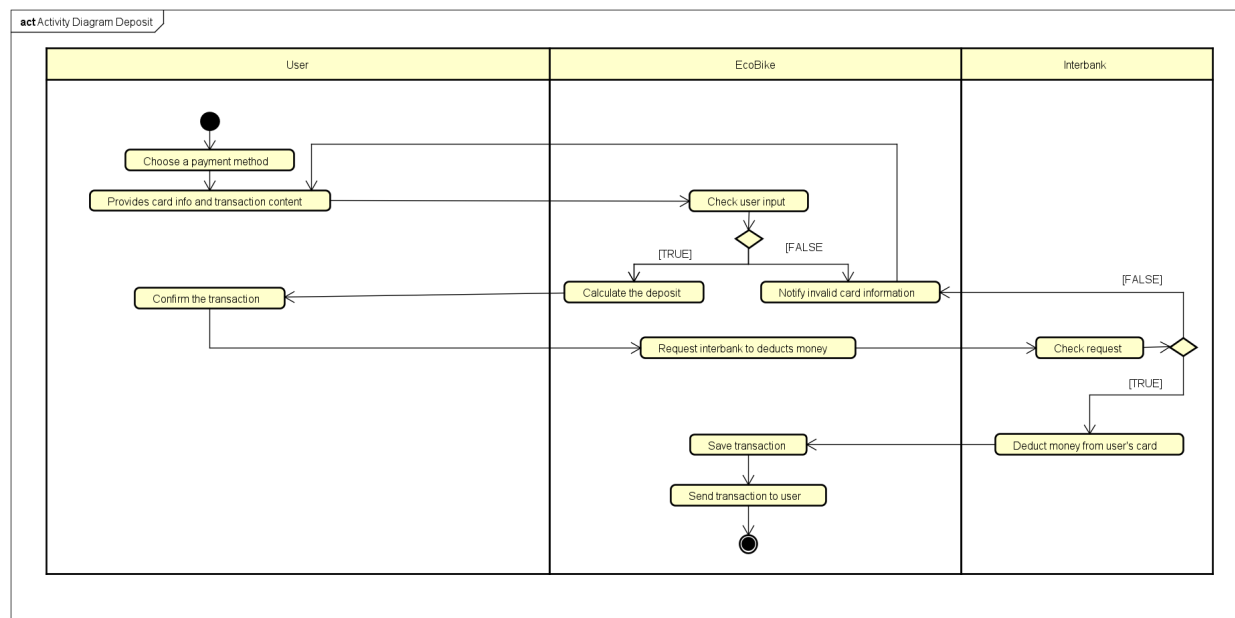
### 3.3.6 Alternative flow of events

| 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                             | At step 2                       |

|  |  |  |                       |  |
|--|--|--|-----------------------|--|
|  |  |  | balance is not enough |  |
|--|--|--|-----------------------|--|

### 3.3.7 Activity diagram

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



### 3.3.8 Input Data

**Table: Input data of card information**

| No | Data Fields      | Description | Mandatory | Valid Condition                                 | Example                |
|----|------------------|-------------|-----------|---|------------------------|
| 1  | Card holder name |             | Yes       |   | DUONG<br>DUY MANH      |
| 2  | Card number      |             | Yes       |   | 6362 3512<br>9685 7436 |
| 3  | Expiration date  |             | Yes       | Consist of month and last 2 digits of year only | 02/15                  |
| 4  | Security code    |             | Yes       |   | 356                    |

### 3.3.9 Output Data



**Table: Output data of transaction information**

| No | Data fields | Description | Display format    | Example        |
|----|-------------|-------------|-------------------|----------------|
| 1  | Payer       |             | - Right alignment | DUONG DUY MANH |
| 2  | Payee       |             | - Right alignment | Vietcombank    |
| 3  | Amount      |             | - Comma separator | 345,000        |

### **3.3.8 Post Conditions**

User successfully deposits the bike

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

### **3.4.1 Usecase code: UC-005**

### **3.4.2 Brief description**

The usecase how customer can interact with the Ecobike system to manage to return a bike after a rental.

### **3.4.3 Actors**

Customer

### **3.4.4 Preconditions**

Customer currently on a rental

### **3.4.5 Basic flow of events**

1. The customer requests to return the bike
2. The system displays list of docks
3. The customer chooses a dock to return bike
4. EcoBike calls “Pay rental” usecase

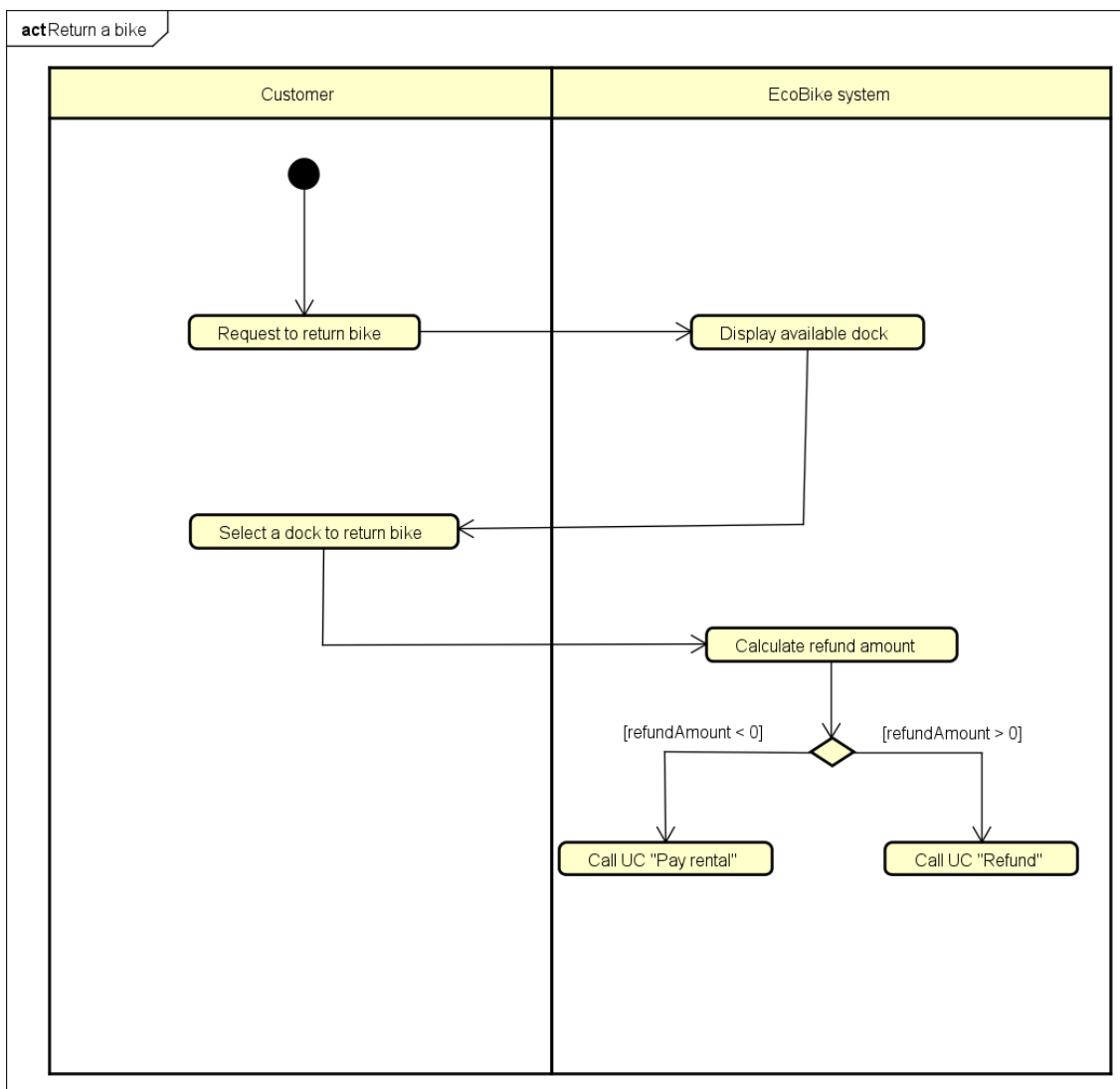
### **3.4.6 Alternative flow**

Table: Alternative flow of “Return a bike” usecase

| No. | Location  | Conditions                               | Actions                     | Resume location |
|-----|-----------|--|-----------------------------|-----------------|
| 1   | At step 2 | The customer not return the bike anymore | Return to the rental screen |                 |
| 2   | At step 3 | The dock                                 | Notify the                  | Step 2          |

|  |  |   |          |  |
|--|--|---|----------|--|
|  |  | chosen is not available or fully filled | customer |  |
|--|--|---|----------|--|

### 3.4.7 Activity diagram



### 3.4.8 Input data

### 3.4.9 Output data

### 3.4.10 Postcondition

### 3.5 Usecase specifications for “Pay rental”

#### 3.5.1 Usecase code: UC-006

#### 3.5.2 Brief description

The usecase how customer can perform payment on the system to end a rental.

#### 3.5.3 Actors

Customer

#### 3.5.4 Preconditions

Customer currently on a rental

#### 3.5.5 Basic flow of events

1. The customer requests to pay the rental
2. The system calculate fees for the rental
3. The system display the total fees for the rental
4. The customer input the card information
5. The system request the interbank to process the transaction
6. The interbank process the transaction and notify the system
7. The system saves the payment transaction
8. The system display the invoice and notify success payment
9. The system sends an emails of invoice to the customer

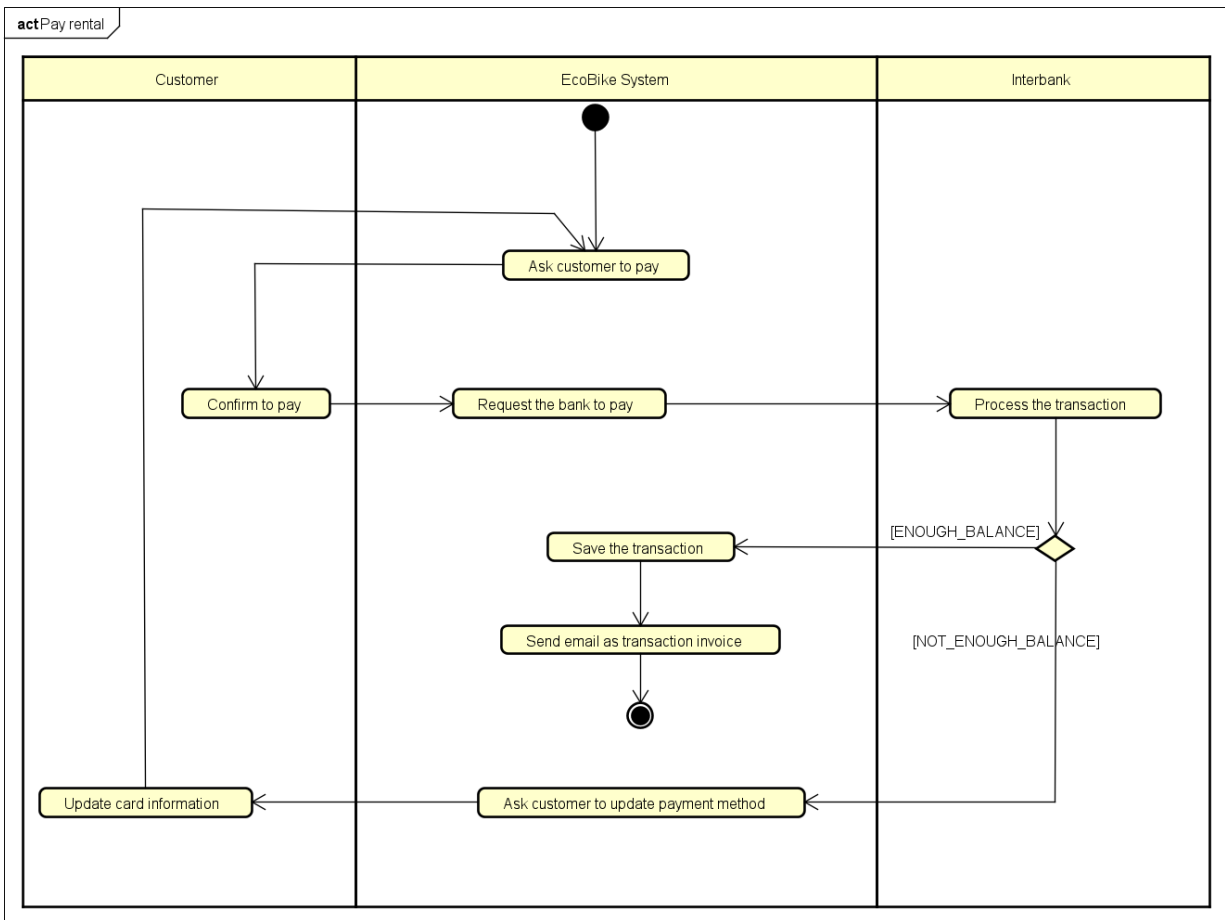
#### 3.5.6 Alternative flow

Table: Alternative flow of “Return a bike” usecase

| No. | Location  | Conditions                                    | Actions                                     | Resume location |
|-----|-----------|---|---|-----------------|
| 1   | At step 3 | The refund amount greater than the total fee  | Call UC “Refund”                            |                 |
| 2   | At step 5 | The card information is not valid             | Notify the customer                         | Step 4          |
| 3   | At step 7 | The interbank notify the system that the card | Redirect back to the card information input | Step 4          |

|  |  |                           |  |  |
|--|--|---------------------------|--|--|
|  |  | balance is not sufficient |  |  |
|--|--|---------------------------|--|--|

### 3.3.7 Activity diagram



### 3.3.8 Input data

| No | Data fields      | Description | Mandator y | Valid condition | Example                |
|----|------------------|-------------|------------|-----------------|------------------------|
| 1  | Card holder name |             | Yes        |                 | Pham Tuan Long         |
| 2  | Card number      |             | Yes        |                 | 0123 4567<br>8941 2354 |

|   |                 |  |     |   |         |
|---|-----------------|--|-----|---|---------|
| 3 | Expiration date |  | Yes | MM/YYYY, with month positive and less or equals to 12, year greater than or equal to current year | 12/2024 |
| 4 | Security code   |  | Yes |   |         |

### 3.3.9 Output data

| No | Data fields          | Description                    | Display format | Example        |
|----|----------------------|--------------------------------|----------------|----------------|
| 1  | Customer's name      | Customer information           |                | Pham Tuan Long |
| 2  | Bike's name          | Bike information               |                | Twin Bike      |
| 3  | Bike's license plate | Bike information               |                | ABC12345       |
| 4  | Time rented          | Time of the rental             | Time           | 1:25:03        |
| 5  | Total fee            | Rental fee minus refund amount | Number         | 100000         |

### 3.5.10 Postcondition

New transaction is stored.

## 3.6 Usecase specifications for “Refund”

### 3.6.1 Usecase code: UC-007

### 3.6.2 Brief description

The usecase how the system can process a refund for the customer

### 3.6.3 Actors

System

### 3.6.4 Preconditions

Customer currently on a rental

### 3.6.5 Basic flow of events

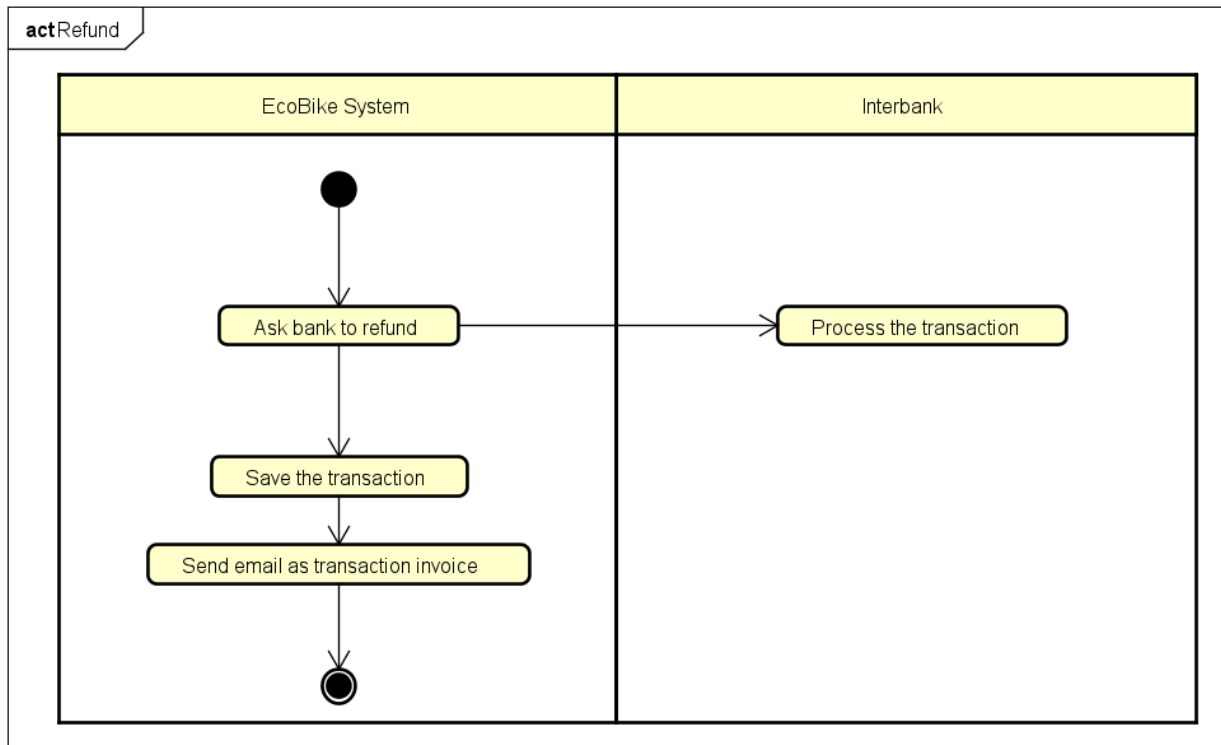
1. The system get the customer's card saved
2. The system ask bank to refund to the customer
3. The interbank process the transaction
4. The system save the transaction
5. Send email as transaction invoice

### 3.6.6 Alternative flow

Table: Alternative flow of "Return a bike" usecase

| No. | Location  | Conditions                             | Actions  | Resume location |
|-----|-----------|--|--|-----------------|
| 1   | At step 2 | The customer's card saved is not valid | Ask the customer to input the card information again |                 |

### 3.6.7 Activity diagram



### 3.6.8 Input data

### 3.6.9 Output data

| No | Data fields          | Description          | Display format | Example        |
|----|----------------------|----------------------|----------------|----------------|
| 1  | Customer's name      | Customer information |                | Pham Tuan Long |
| 2  | Bike's name          | Bike information     |                | Twin Bike      |
| 3  | Bike's license plate | Bike information     |                | ABC12345       |
| 4  | Time rented          | Time of the rental   | Time           | 1:25:03        |
| 5  | Total amount         | Refunded amount      | Number         | 100000         |

### **3.6.10 Postcondition**

New transaction is stored.

## **4. Constraints and Assumptions**

- The payment method is limited to credit cards.
- Each credit card can be used to rent only one bike at a time.
- The change in battery life for e-bikes while rented or returned is not accounted for.

## **5.. Supplementary specification**

### **5.1 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

### **5.2 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 error so that they can know how to fix it.

### **5.3 Reliability**

- The service must be able to run 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.

#### 5.4 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 05 second

#### 5.5 Supportability

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

### 6. Glossary

| 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   |