HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

School of Information and communications technology

Software Requirement Specification

Version 1.0

**Eco Bike Rental Software**

Subject: ITSS Software Development

**Group 05**

Nguyen Thai An – 20176677

Bui Tu Hoang – 20176761

Vu Minh Hoang – 20176765

Nguyen Manh Khang - 20176792

*Hanoi,* *October, 2020*

Table of contents

Table of contents 1

1 Introduction 2

1.1 Objective 2

1.2 Scope 2

1.3 Glossary 2

1.4 References 2

2 Overall Description 3

2.1 Actors 3

2.2 Use case diagrams 3

2.3 Business processes 3

3 Detailed Requirements 4

3.1 Use case specification for “Create EBR account” 5

3.2 Use case specification for “Setup payment method” 6

3.3 Use case specification for "View dock's detailed information” 8

3.4 Use case specification for “View bike's detailed information” 10

3.5 Use case specification for “Rent a bike” 12

3.6 Use case specification for “Pay deposit” 16

3.7 Use case specification for “Return a bike” 19

3.8 Use case specification for “Refund deposit after deducting rental fee” 22

4 Supplementary specification 25

4.1 Functionality 25

4.2 Usability 25

4.3 Reliability 25

4.4 Performance 25

4.5 Supportability 25

4.6 Other requirements 25

# Introduction

## Objective

This document provides the detailed description for Eco Bike Rental software and their functions in application.

This document is mainly focus on related software developers.

## Scope

Eco Bike Rental software’s aim, as its name, is to provide a bike renting service to customers.

The software’s goal includes creating account. At time of creating account, user is required to add payment method. After logging in, user will be provided with the ability to search for docks and see their information in details. At each dock, user can have knowledge of bikes parked at the dock. Bike renting and returning are key functions of this system.

Interbank acts as a bridge between users and the main system for executing payment transactions. Interbank will take place in the validating, adding money and deducting money processes.

## Glossary

*<Listing and explaining the terms appearing in the software’s profession and this document. Any assumption of the reader’s prior knowledge or experience on the subject is ill advised>*

## References

*<Listing the referenced material used in this document, including the one related to the project>*

# Overall Description

## Actors

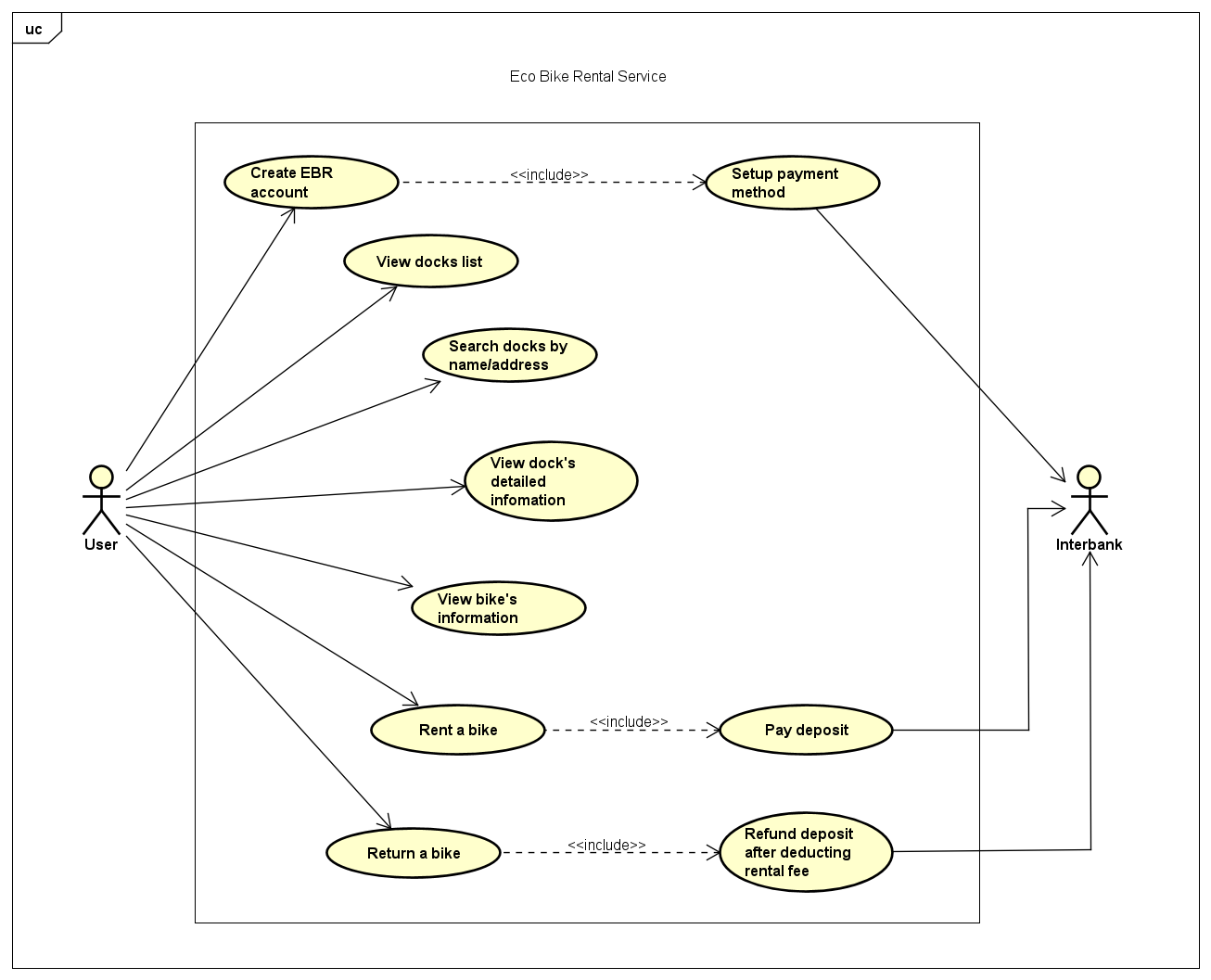
### User

Anyone who wants to use the eco bike rental service through this software.

### Interbank

The interbank system, used for making transaction when renting and returning a bike within the service.

## Use case diagrams



## Business processes

# Detailed Requirements

## Use case specification for “Create EBR account”

**Use Case “Create EBR Account”**

1. **Use case code**

UC001

1. **Brief Description**

This use case describes the interactions between user and EBR software when user wishes to create a new EBR account.

1. **Actors**
   1. **User**
2. **Preconditions**
3. **Basic Flow of Events**

Step 1. The user requests to start making a new EBR account

Step 2. The EBR software displays the account information form

Step 3. The user fills and submits the account information form

Step 4. The EBR software calls UC “Setup Payment Method”

Step 5. The EBR software creates a new account

Step 6. The EBR software displays the successfully created account notification.

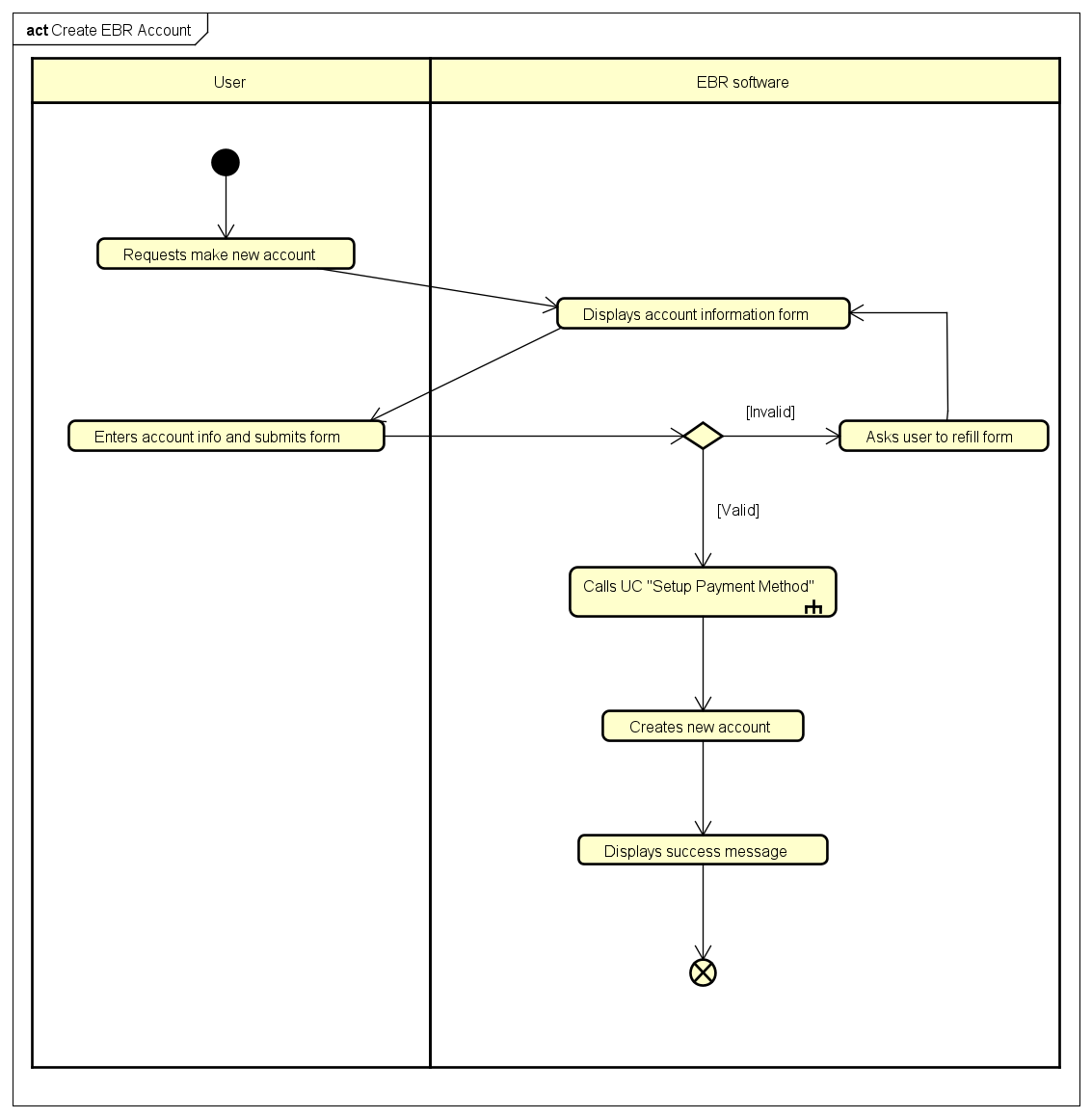
.

1. **Alternative flows**

*Table 1 - Alternative flow of events for UC “Name of the Use Case”*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Location | Condition | Action | Resume location |
| 1. | At step 4 | If has mandatory field(s) left blank | \_ The EBR software asks the user to refill the account information form | Resume at step 2 |
| 2. | At step 4 | If any fields is against its “valid condition” | \_ The EBR software notifies the user the invalid field(s) and asks the user to make changes | Resume at step 2 |

1. **Activity diagrams**



1. **Input data**

*Table 2 - Input data of account information form*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Data fields | Description | Mandatory | Valid condition | Example |
| 1. | Phone Number | Will be use as login id | Yes | 10 digits | 0987654321 |
| 2. | User Name |  | Yes |  | Nguyen Thai An |
| 3. | Password |  | Yes | At least 6 characters | 123456 |
| 4. | Email |  | No |  | thaian229@gmail.com |

1. **Output data**
2. **Postconditions**

## Use case specification for “Setup payment method”

**Use Case “Setup Payment Method”**

1. **Use case code**

UC002

1. **Brief Description**

This use case describes the interactions between the user, the interbank and the EBR software when the user wishes to setup a new payment method.

1. **Actors**
   1. **User**
   2. **Interbank**
2. **Preconditions**
3. **Basic Flow of Events**

Step 1. The EBR software displays credit card information form

Step 2. The user enters card info and submits

Step 3. The EBR software forwards info and asks the Interbank to link the card

Step 4. The interbank create the payment method

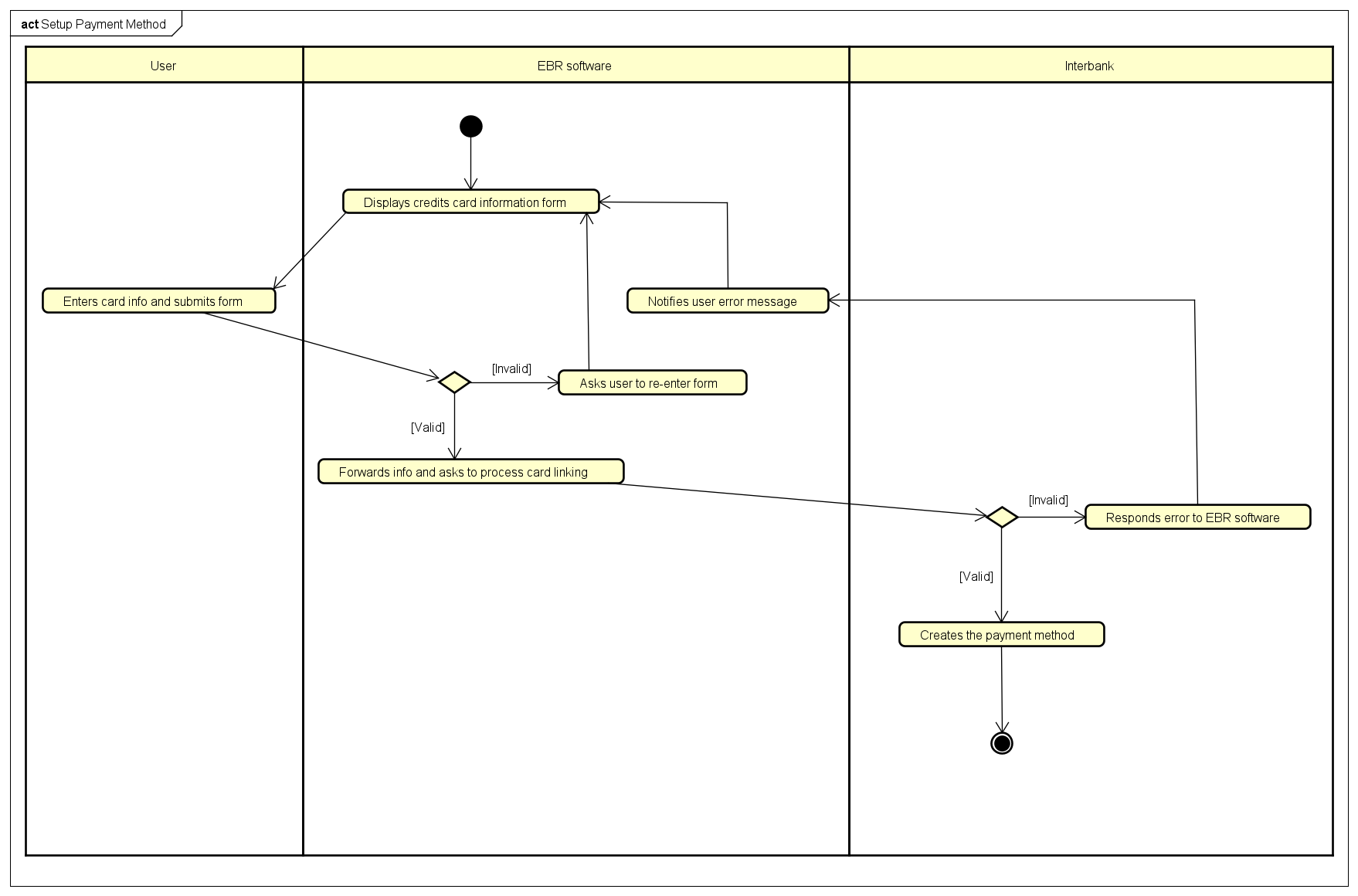
.

1. **Alternative flows**

*Table 1 - Alternative flow of events for UC “Name of the Use Case”*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Location | Condition | Action | Resume location |
| 1. | At step 3 | If has blank mandatory field(s) | \_ The EBR software asks the user to refill the form | Resume at step 1 |
| 2. | At step 4 | If card info is invalid | \_ The interbank informs the EBR software  \_ The EBR software notifies the user with error message | Resume at step 1 |

1. **Activity diagrams**

****

1. **Input data**

*Table 2 - Input data of credit card information form*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Data fields | Description | Mandatory | Valid condition | Example |
| 1. | Card Number |  | Yes | 16 digits | 1234 4321 2134 3214 |
| 2. | Card Type | Choose from a list | Yes |  | Visa |
| 3. | Expire Date |  | Yes | Form MM/YY | 08/25 |
| 4. | Security Code |  | Yes | Digits only | 012 |
| 5. | Billing address |  | Yes |  | No 7, alley 66, Dinh Dong lane, Bach Mai |

1. **Output data**
2. **Postconditions**

## Use case specification for “View dock’s detailed information”

**Use Case “View Dock’s Detailed Information”**

1. **Use case code**

UC005

1. **Brief Description**

This use case describes the interactions between user and EBR software when user wishes to view the detailed information of chosen dock.

1. **Actors**
   1. **User**
2. **Preconditions**

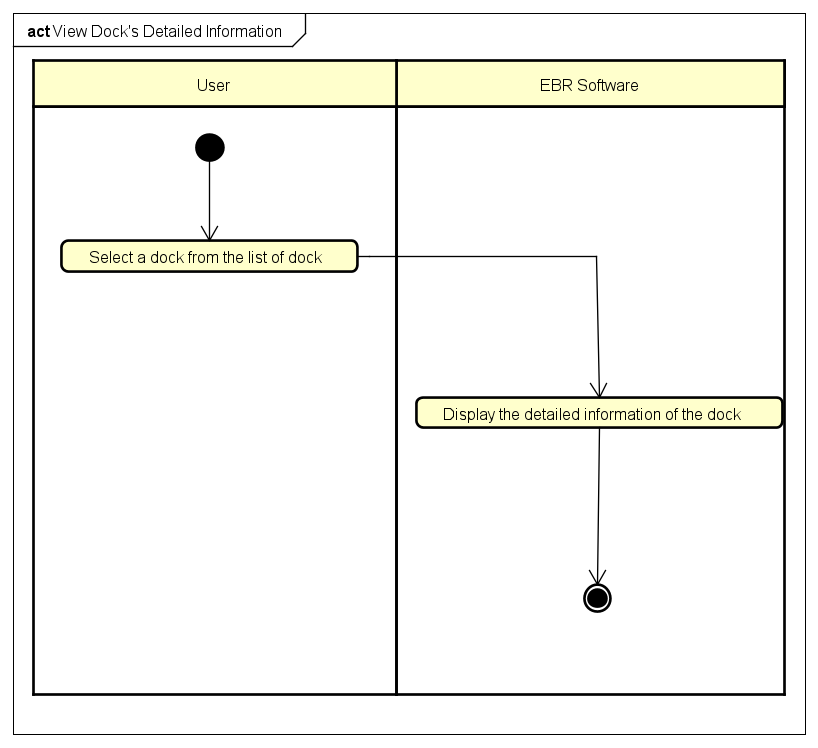
User must log in

1. **Basic Flow of Events**

Step 1. The user choose a dock from the list of docks

Step 2. The software return the information of the dock

1. **Alternative flows**
2. **Activity diagrams**

****

1. **Input data**
2. **Output data**

*Table 1 - Output data of view dock’s detailed information*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Data fields | Description | Display format | Example |
| 1. | Name | Name of the chosen dock |  | Dock No.01 |
| 2. | Address | The address of this dock |  | 12 Inner Road |
| 3. | Dock Area | The area of this dock | * Positive number * Right alignment | 60m2 |
| 4. | Number of Available Bikes | Number of available bikes in this dock | * Positive integer * Right alignment | 20 |
| 5. | Bike | Available bike in this dock |  | Standard Bike 01 |
| 6. | Number of empty slot | The number of empty docking point | * Positive integer * Right alignment | 10 |
| 7. | Distance | The distance from user’s location this dock | * Positive number * Right alignment | 100m |
| 8. | Walking Time | The calculated walking time from user’s location to this dock | Positive number | 2 minutes |

1. **Postconditions**

## Use case specification for “View bike’s detailed information”

**Use Case “View Bike’s Detailed Information”**

1. **Use case code**

UC006

1. **Brief Description**

This use case describes the interactions between user and EBR software when user wishes to view the detailed information of chosen bike.

1. **Actors**
   1. **User**
2. **Preconditions**

User must log in

1. **Basic Flow of Events**

Step 1. The user selects a bike from the list of bikes

Step 2. The software checks the information of the selected bike

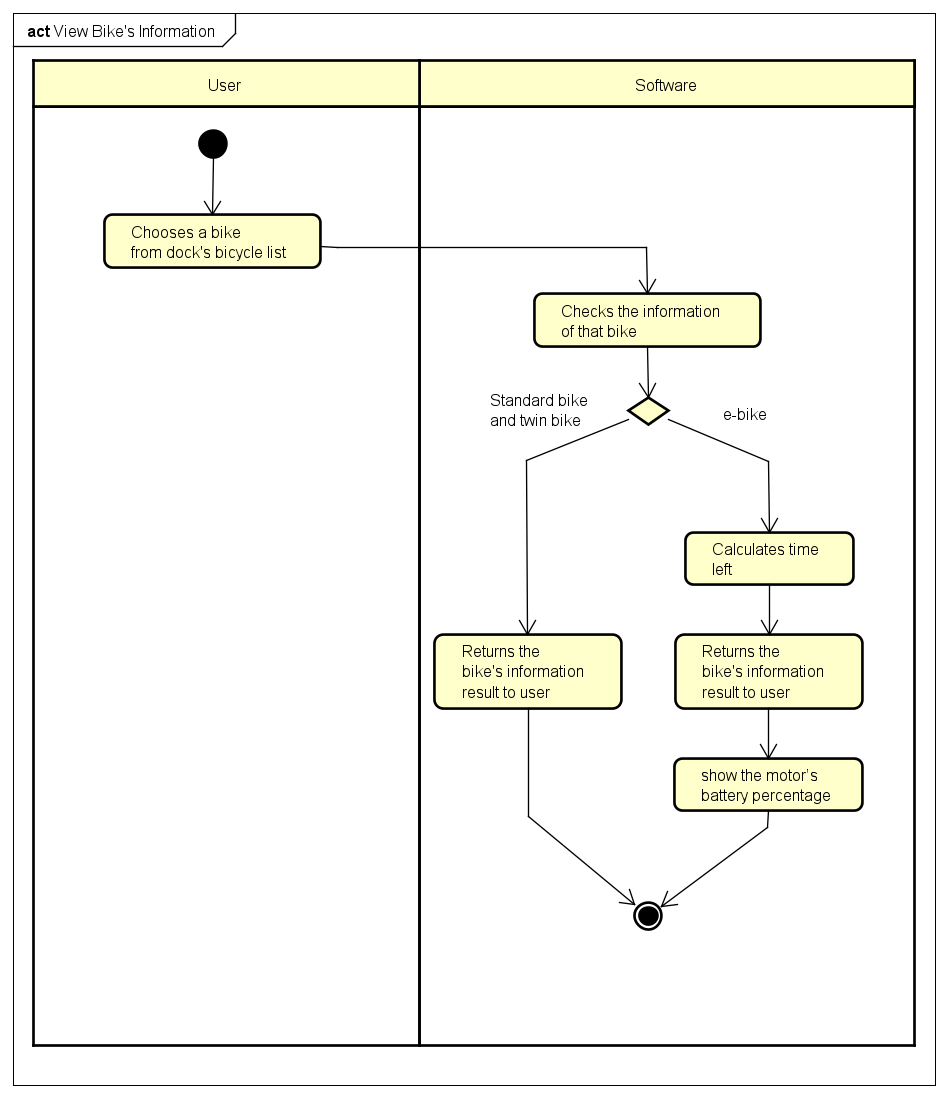
Step 3. The software returns the information of the bike

1. **Alternative flows**

*Table 1 - Alternative flow of events for UC “Name of the Use Case”*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Location | Condition | Action | Resume location |
| 1 | At Step 2 | The selected bike is an e-bike | The software returns the percentage of the electric motor’s battery and calculate the time left | Resume at Step 3 |

1. **Activity diagrams**

****

1. **Input data**
2. **Output data**

*Table 2 - Output data of view bike’s detailed information*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Data fields | Description | Display format | Example |
| 1. | Name | Name of the selected bike |  | E-bike 01 |
| 2. | Type | Type of this bicycle |  | E-bike |
| 3. | Saddle | Number of saddle of this bike | * Positive integer * Right alignment | 1 |
| 4. | Pedals | Number of pair of pedals | * Positive integer * Right alignment | 1 |
| 6. | Rear seat | Number of rear seat | * Positive integer * Right alignment | 1 |
| 7. | Battery | The electric motor’s battery percentage | * Positive number with percentage symbol * Right alignment | 60% |
| 8. | Time left | How much time is left | In minute | 180 minutes |

1. **Postconditions**

## Use case specification for “Rent a bike”

**Use Case “Rent A Bike”**

1. **Use case code**

UC008

1. **Brief Description**

This use case describes the interactions between user and EBR software when user wishes to rent a bike

1. **Actors**
   1. User
2. **Preconditions**

User logged in

1. **Basic Flow of Events**

Step 1. The user selects a bike

Step 2. The user enters corresponding barcode

Step 3. The EBR Software checks if barcode is valid

Step 4. The EBR Software checks if bike is availabe

Step 5. User chooses card for payment

Step 6. The EBR Software checks if payment card used for another session

Step 7. The EBR Software converts from barcode to rental code

Step 8. The EBR Software calls use case "Pay deposit"

Step 9. The EBR Software creates new renting session

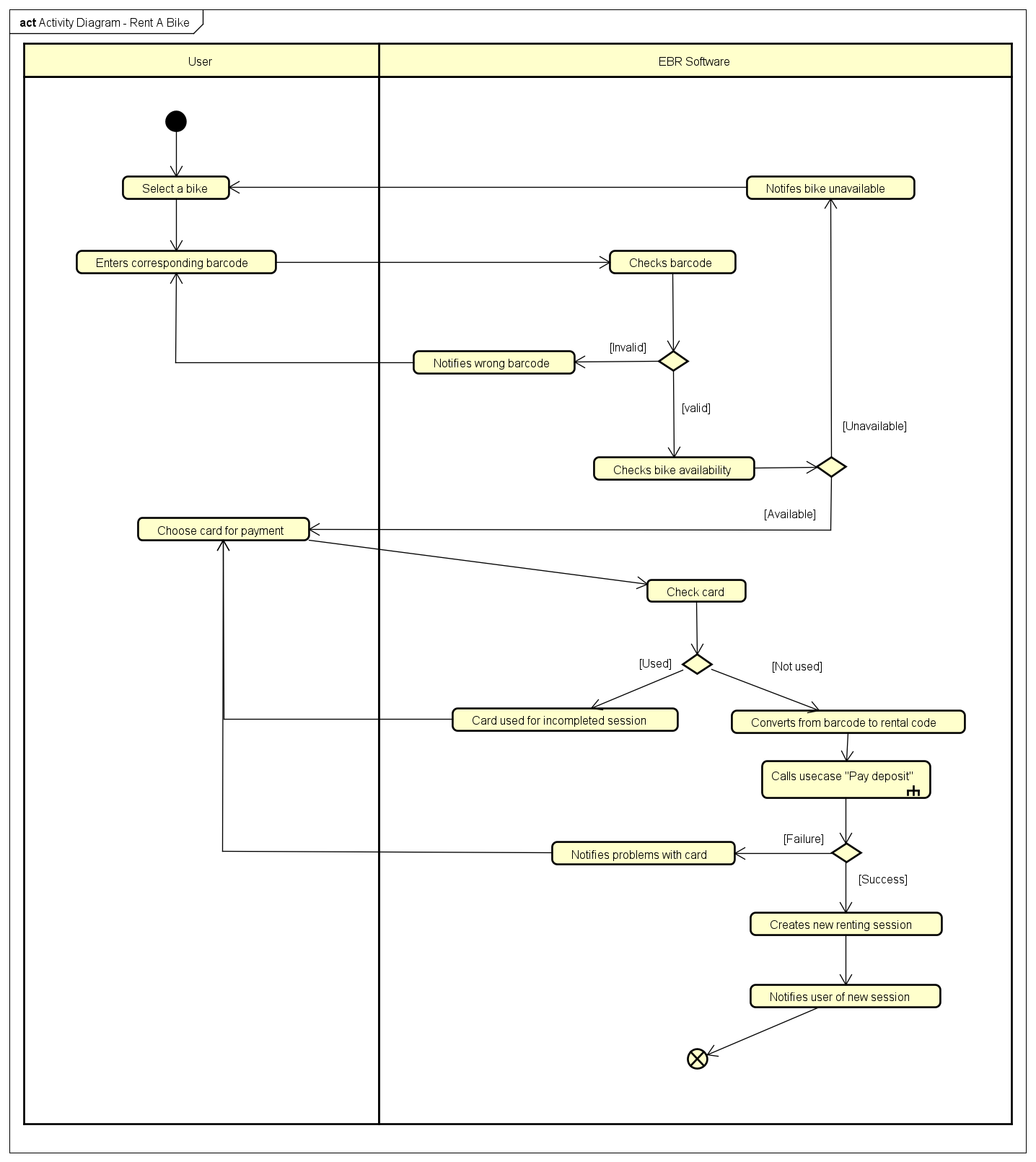
Step 10. The EBR Software notifies user of new session

1. **Alternative flows**

*Table 1 - Alternative flow of events for UC “Rent A Bike”*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Location | Condition | Action | Resume location |
| 1 | At Step 4 | Wrong barcode | Notifies wrong barcode | Step 2 |
| 2 | At Step 5 | Not available | Notifies rented bike | Step 1 |
| 3 | At Step 7 | Used | Let user choose a different card | Step 5 |
| 4 | At Step 9 | Problem with credit card | Let user choose a different card | Step 5 |

1. **Activity diagrams**

****

1. **Input data**

*Table 2 - Input data of bike renting*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Data fields | Description | Mandatory | Valid condition | Example |
| 1 | Barcode | Bike’s barcode | Yes | String | 1231abc212 |

1. **Output data**

*Table 3 - Output data of bike renting*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Data fields | Description | Display format | Example |
| 1 | Session code |  | String | Code123123 |
| 1 | Start time |  | hh:mm DD/MM/YYYY | 16:30 29/09/2020 |
| 2 | Deposit | Money deposited for the bike | • Comma for thousands separator  • Positive integer  • Right alignment | 123,000 |

1. **Postconditions**

## Use case specification for “Pay deposit”

**Use Case “Pay Deposit”**

1. **Use case code**

UC009

1. **Brief Description**

This use case describes the interactions between user and EBR software and Interbank when EBR Software wishes to make transaction with interbank

1. **Actors**
   1. User
   2. Interbank
2. **Preconditions**

User logged in

1. **Basic Flow of Events**

Step 1. The EBR Software displays the payment screen

Step 2. The user confirms transaction

Step 3. The EBR Software asks interbank to process transaction

Step 4. The Interbank processes transaction

Step 5. The Interbank begins transaction

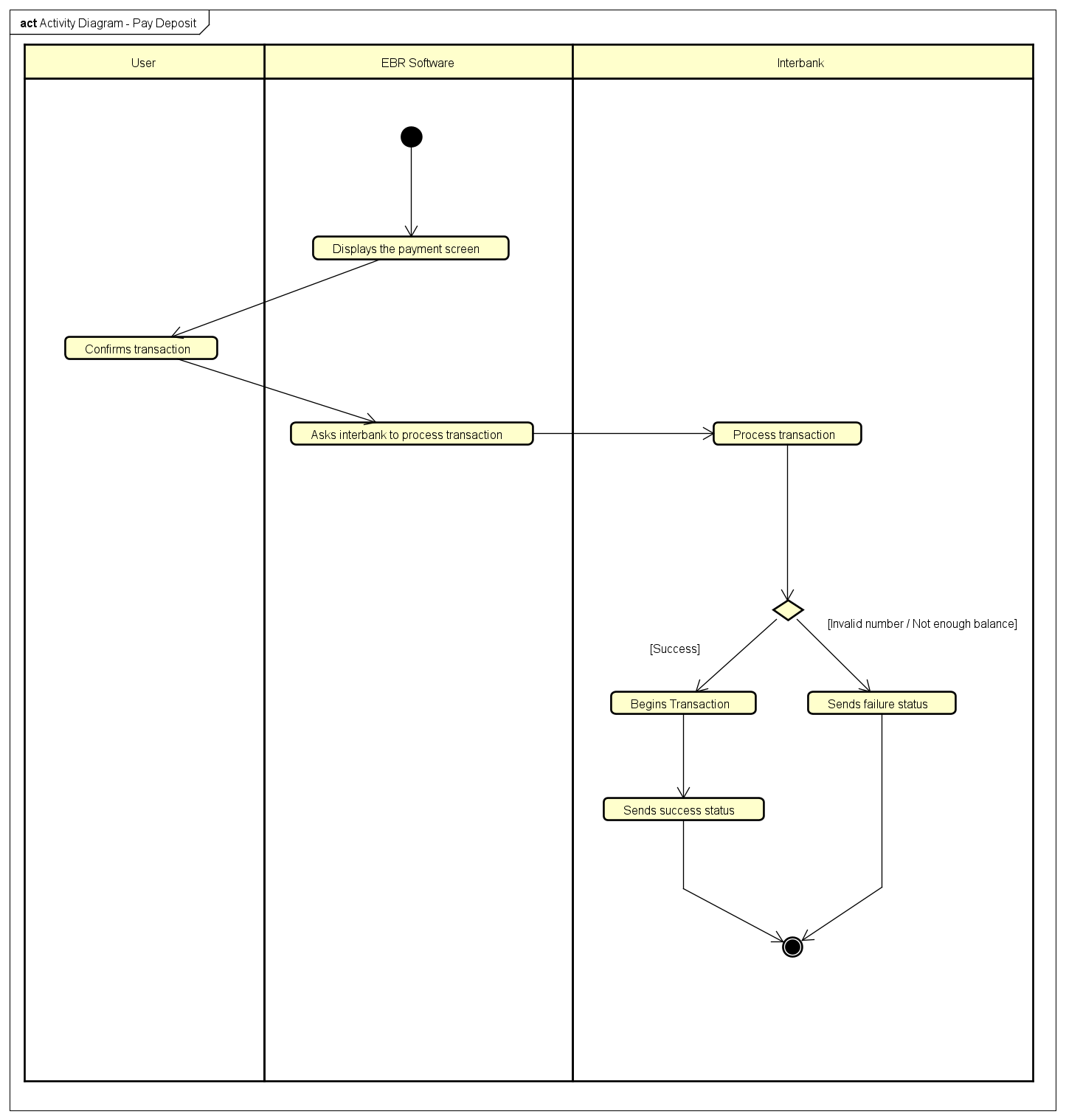
Step 6. Sends success status

1. **Alternative flows**

*Table 1 - Alternative flow of events for UC “Pay Deposit”*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Location | Condition | Action | Resume location |
| 1 | At Step 5 | Invalid card number | Notifies invalid card number | End of use case |
| 2 | At Step 5 | Not enough balance | Notifies not enough balance | End of use case |

1. **Activity diagrams**

****

1. **Input data**
2. **Output data**

*Table 2 - Output data of interbank transaction*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Data fields | Description | Display format | Example |
| 1 | Success state |  | Boolean | True |
| 2 | Error message |  | String | Invalid card number |

1. **Postconditions**

## Use case specification for “Return a bike”

**Use Case “Return a bike”**

1. **Use case code**

UC0010

1. **Brief Description**

This use case describes the interactions between User and EBR software when User wishes to return a bike.

1. **Actors**
   1. User
2. **Preconditions**

User rented a bike

1. **Basic Flow of Events**

Step 1. The User requests to return a bike

Step 2. The User enters the barcode

Step 3. EBR software checks the barcode

Step 4. EBR software converts barcode to rental code

Step 5. EBR software closes the user’s renting session

Step 6. EBR software Summarizes session information, calculates rental fee, calculates excess money and notifies to User

Step 7. User confirms information

Step 8. EBR software calls Use case “Refund deposit after deducting rental fee”

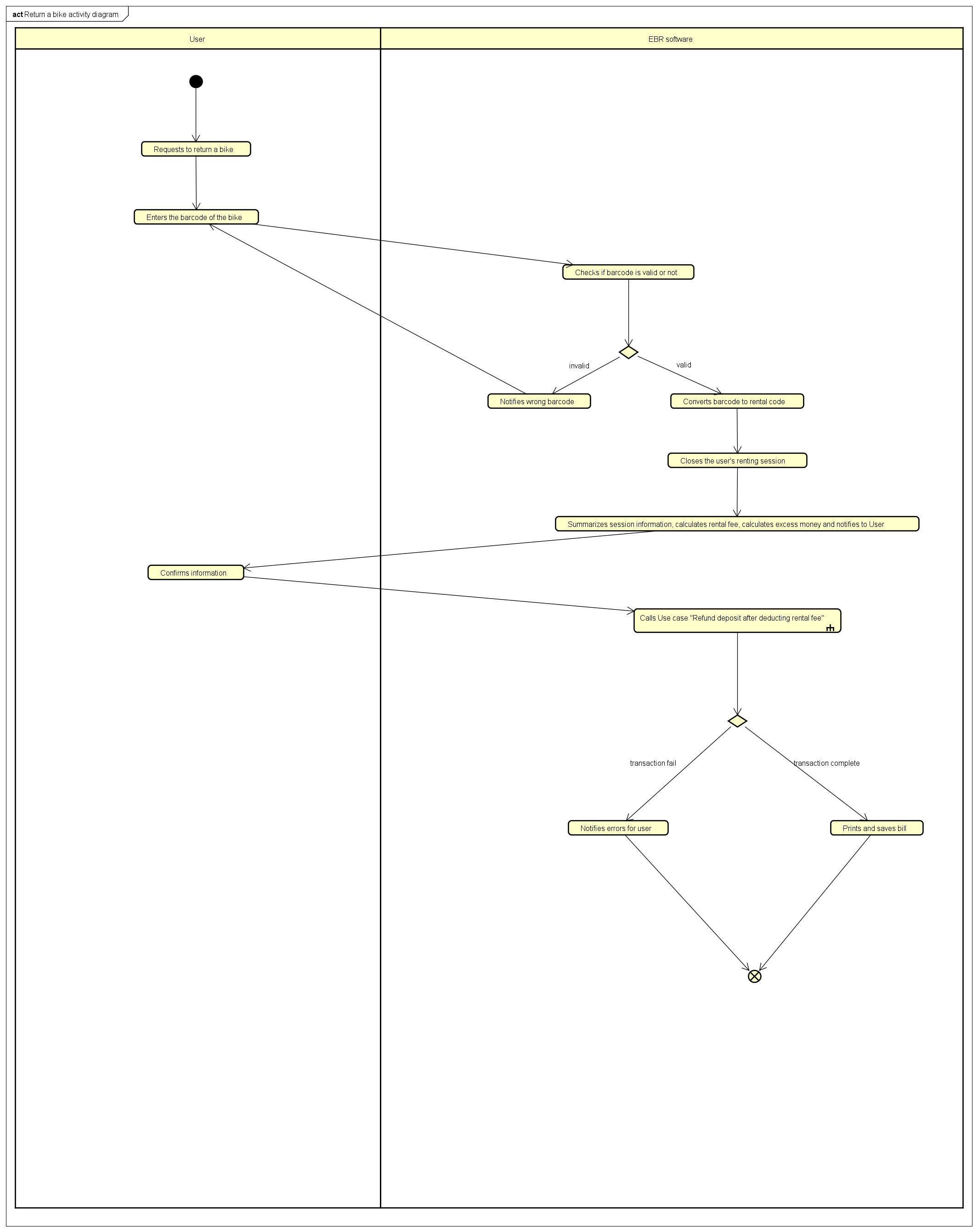
Step 9. EBR software saves and prints bill for user

1. **Alternative flows**

*Table 1 - Alternative flow of events for UC “Name of the Use Case”*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Location | Condition | Action | Resume location |
| 1 | At step 3 | Invalid barcode | Notifies wrong barcode | Step 2 |
| 2 | At step 8 | Transaction fail | Notifies transaction fail | End use case |

1. **Activity diagrams**

****

1. **Input data**

*Table 2 - Input data of …*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Data fields | Description | Mandatory | Valid condition | Example |
| 1 | Barcode | Bike’s barcode | Yes | String | 1231abc212 |

1. **Output data**

*Table 3 - Output data of …*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Data fields | Description | Display format | Example |
| 1 | End time | Ending time of the User’s renting session | hh : mm  DD/MM/YY | 12:20  10/10/2020 |
| 2 | Usage time |  | . A number of minutes usage | 120 |
| 3 | Renting fee |  | . Comma for thousands separator  . Positive integer  . Right alignment | 123,000 |
| 4 | Refund deposit | deposit after deducting rental fee | . Comma for thousands separator  . Positive integer  . Right alignment | 100,000 |

1. **Postconditions**

## Use case specification for “Refund deposit after deducting rental fee”

**Use Case “Refund deposit after deducting rental fee”**

1. **Use case code**

UC0011

1. **Brief Description**

This use case describes the interactions between EBR software and Interbank when EBR software wishes to refund deposit to User.

1. **Actors**
   1. Interbank
2. **Preconditions**
3. **Basic Flow of Events**

Step 1. EBR software asks for transaction

Step 2. Interbank validates payment method

Step 3. Proceeds transaction

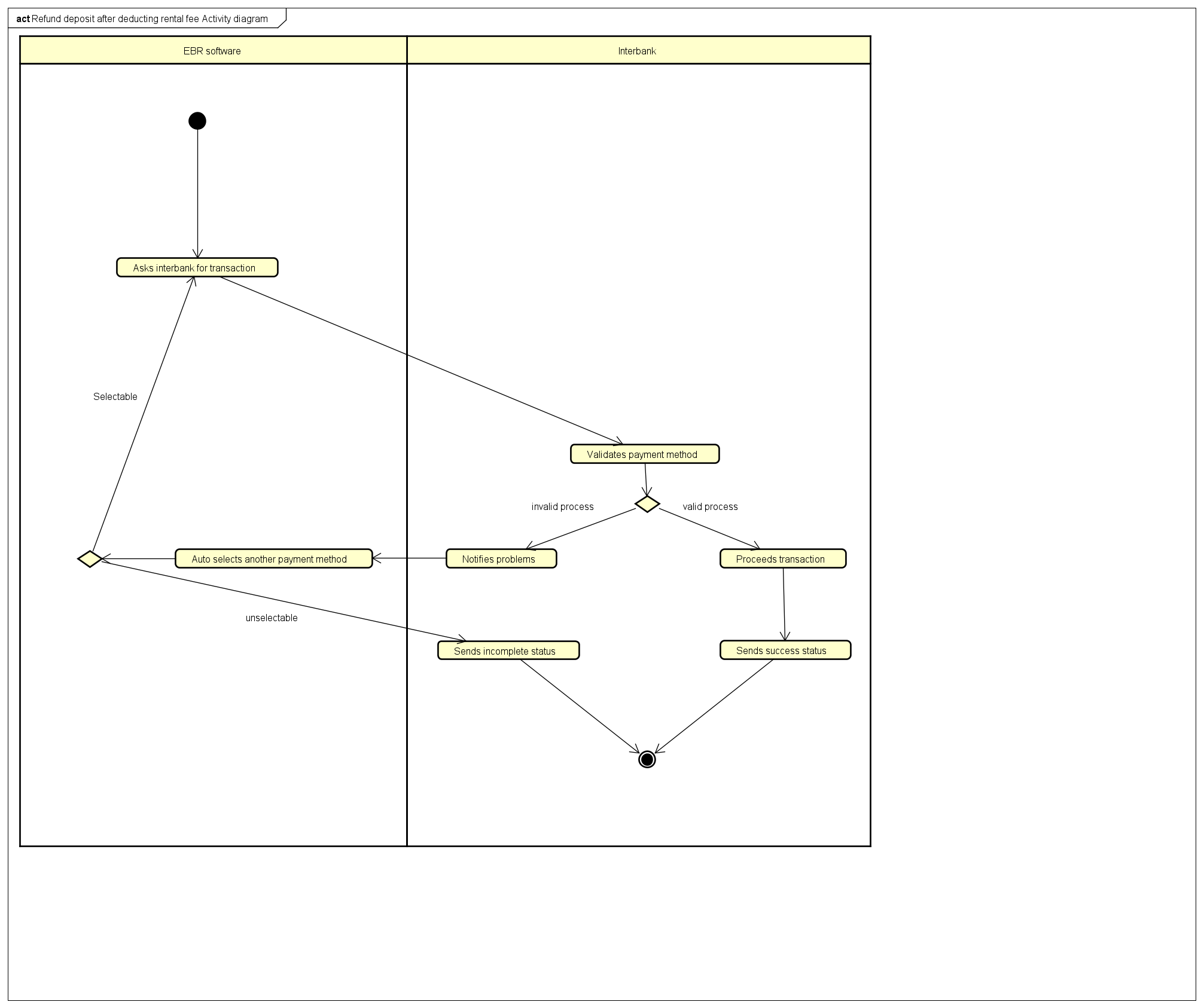
Step 4. Sends success result

1. **Alternative flows**

*Table 1 - Alternative flow of events for UC “Name of the Use Case”*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Location | Condition | Action | Resume location |
| 1 | At step 2 | Invalid process | . Notifies problems  . Auto selects another method | . If selectable, resume at step 1  . Else, send incomplete status then end the use case |

1. **Activity diagrams**

****

1. **Input data**
2. **Output data**

*Table 3 - Output data of …*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Data fields | Description | Display format | Example |
| 1 | Success status |  | . Success status as a message | . Your transaction done |
| 2 | Incomplete status |  | . Incomplete status and reason | . Your transaction was fail due to ... |

1. **Postconditions**

# Supplementary specification

## Functionality

## Usability

The software must be intuitive which allows novice users without any training to use.

## Reliability

This system is a 24/7 live service.

The system must be able to operate in an average of 200 hours without failure.

Must be repaired within 2 hours after any failure.

## Performance

The system shall serve 200 concurrent users without noticeable performance lost.

The system’s respond time shall be least than 1 second in average and least than 2 seconds under peak load.

## Supportability

## Other requirements