**HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY**

**School of Information and Communication Technology**

Software Requirement Specification

**ECOBIKE RENTAL SOFTWARE**

**Group 18**

Nguyen Mau Tra – 20200624

Nguyen Kieu Trang – 20205174

Hoang Duc Trung – 20200640

Nguyen Khanh Trung - 20205133

*Hanoi,* *July, 2023*

Table of contents

Table of contents 1

1 Introduction 3

1.1 Objective 3

1.2 Scope 3

1.3 Glossary 3

1.4 References 3

2 Overall Description 4

2.1 Actors 4

2.2 Use case diagrams 4

2.3 Business processes 4

3 Detailed Requirements 5

3.1 Use case specification for “Setup payment method” 5

3.2 Use case specification for "View dock's detailed information” 7

3.3 Use case specification for “View bike's detailed information” 9

3.4 Use case specification for “Rent a bike” 12

3.5 Use case specification for “Pay deposit” 14

3.6 Use case specification for “Return a bike” 17

3.7 Use case specification for “Refund deposit after deducting rental fee” 20

4 Supplementary specification 22

4.1 Functionality 22

4.2 Usability 22

4.3 Reliability 22

4.4 Performance 22

4.5 Supportability 22

4.6 Other requirements 22

# 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

|  |  |
| --- | --- |
| Word | Meaning |
| deposit | to pay someone an amount of money when you make an agreement with that person to pay for or buy something, that either will be returned to you later, if the agreed arrangement is kept, or that forms part of the total payment |
| deduct | to take away an amount or part from a total |
| transaction | an occasion when someone buys or sells something, or when money is exchanged or the activity of buying or selling something |

## References

# Overall Description

## Actors

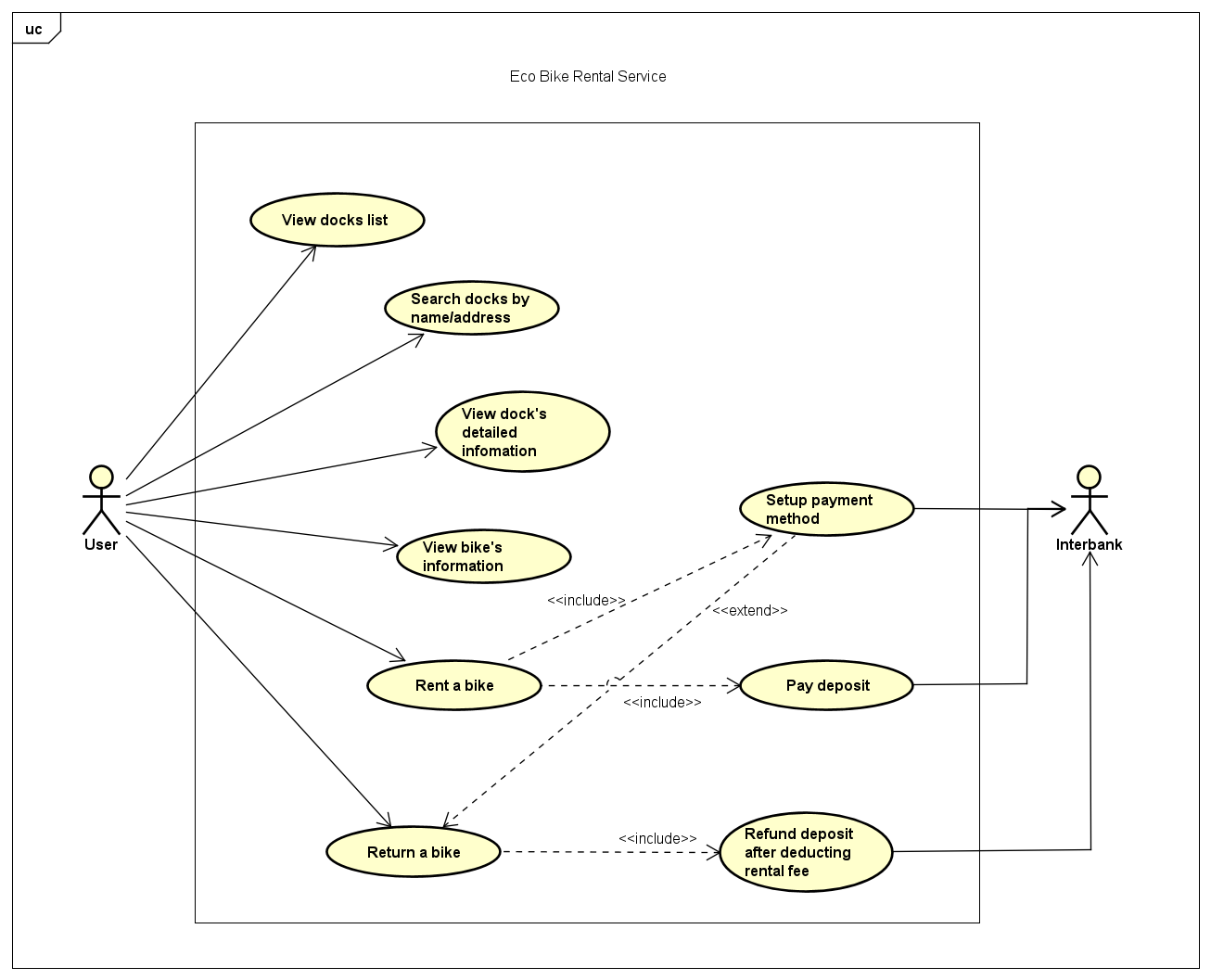
### User

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 “Setup payment method”

**Use Case “Setup Payment Method”**

1. **Use case code**

UC001

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. **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

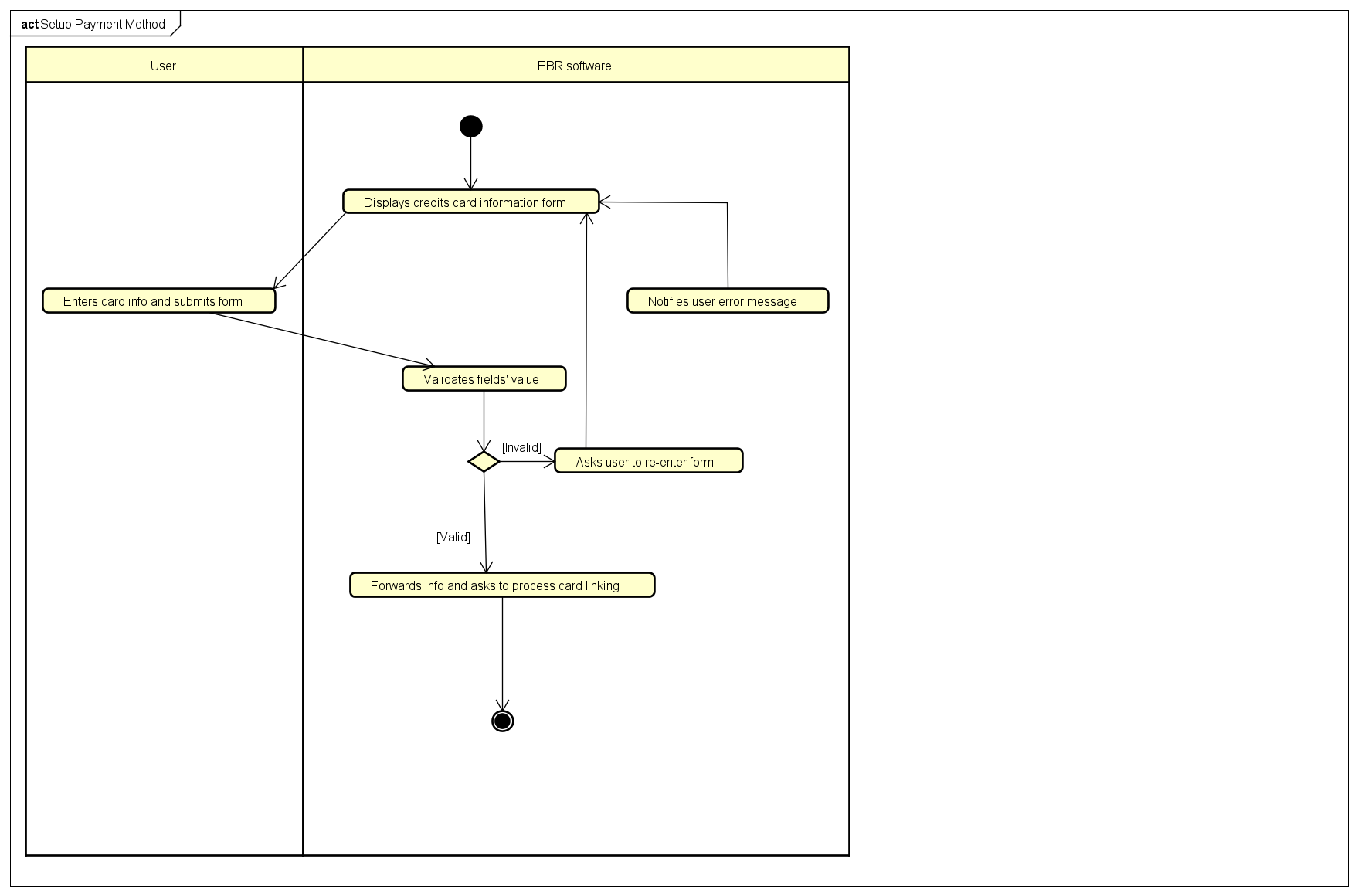
.

1. **Alternative flows**

*Table 1 - Alternative flow of events for UC “Setup Payment Method”*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Location | Condition | Action | Resume location |
| 1. | At step 2 | If has blank mandatory field(s) | \_ The EBR software asks the user to refill the form | Resume at step 1 |
| 2. | At step 2 | If has invalid field formats | \_ The EBR software asks user to fix field’s values | 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 |

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**

UC002

1. **Brief Description**

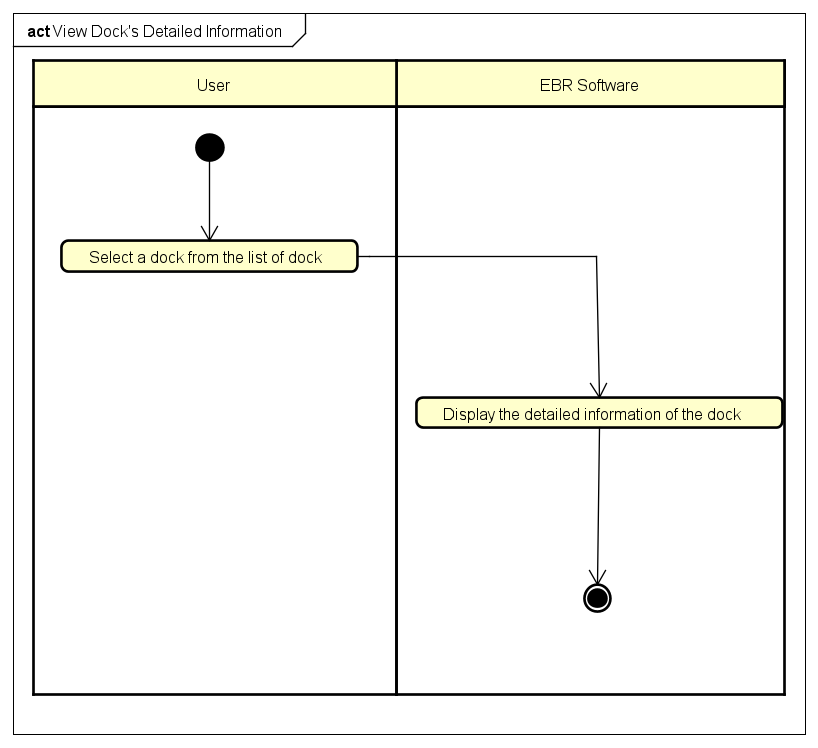
This use case describes the interactions between user and EBR software when user wishes to view the detailed information of chosen dock.

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

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

Step 2. The software returns the information of the dock

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

****

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

*Table 2 - 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 slots | 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**

UC003

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**
3. **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

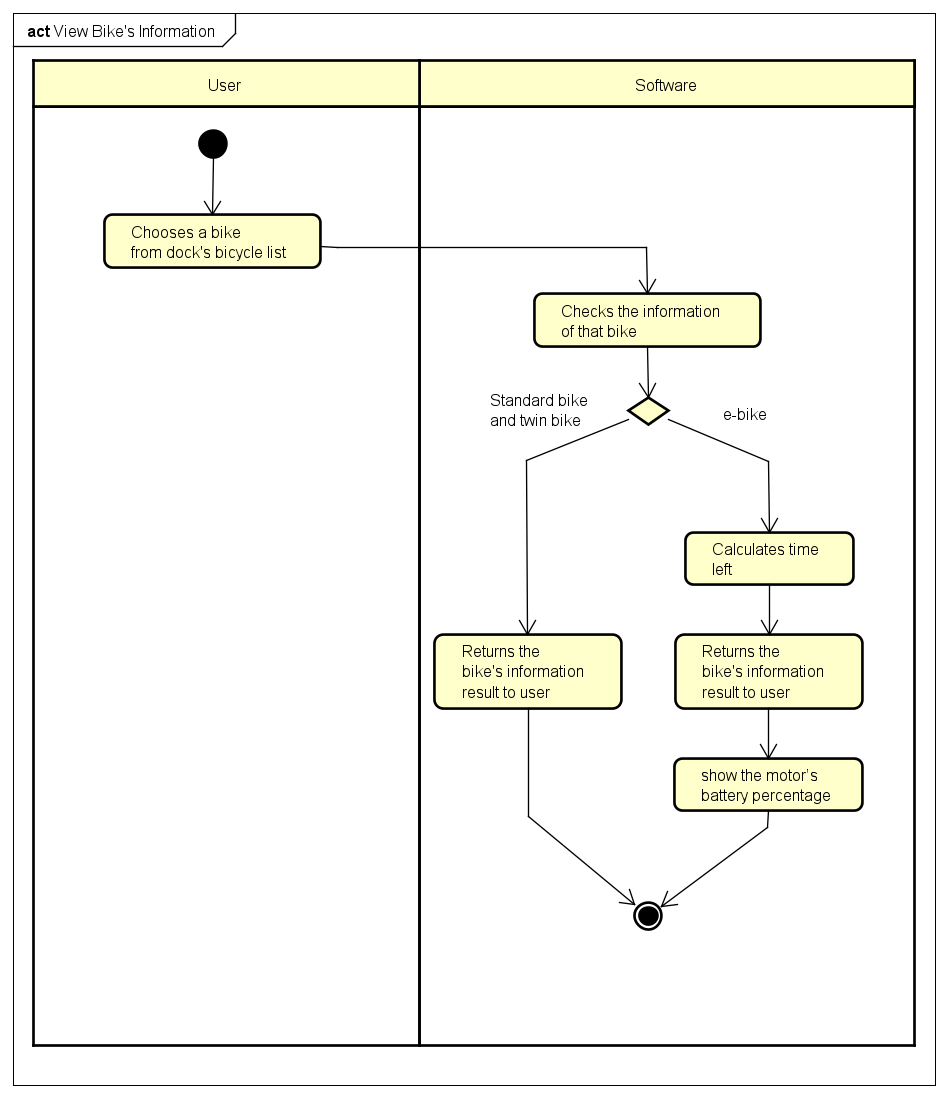
Step 3. The software returns the information of the bike

1. **Alternative flows**

*Table 3 - 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 4 - 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 saddles 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 seats | * 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**

UC004

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**
3. **Basic Flow of Events**

Step 1. The user requests to rent a bike

Step 2. EBR software calls use case “Setup payment method”

Step 3. EBR software calls use case “Pay deposit”

Step 4. The EBR Software creates new renting session

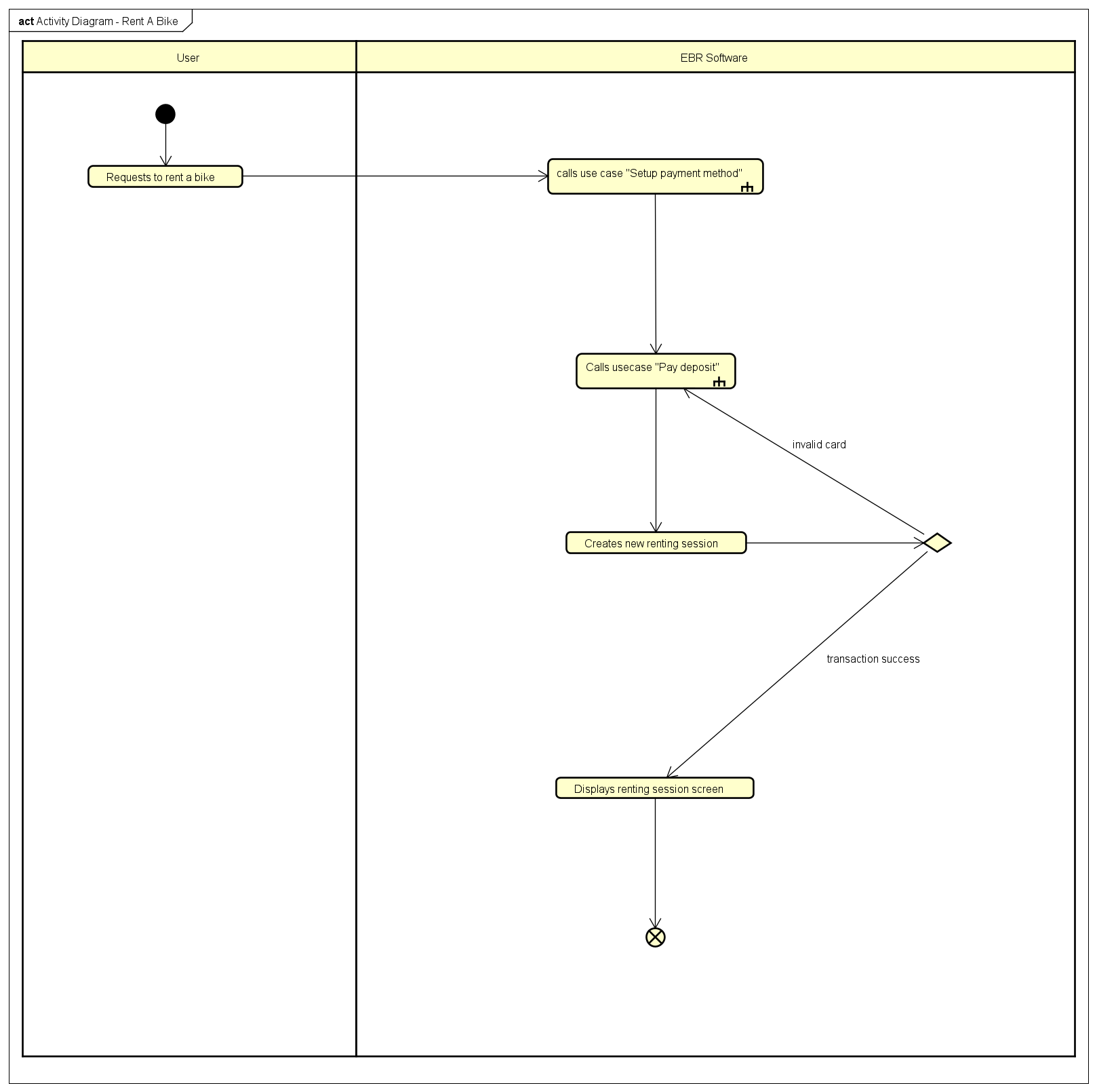
Step 5. The EBR Software displays renting session screen

1. **Alternative flows**

*Table 5 - Alternative flow of events for UC “Rent A Bike”*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Location | Condition | Action | Resume location |
| 1 | At Step 3 | Problem with credit card | Let user choose a different card | Step 2 |

1. **Activity diagrams**



1. **Input data**

*Table 6 - 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 7 - 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**

UC005

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**
3. **Basic Flow of Events**

Step 1. The EBR Software displays the payment confirmation screen

Step 2. The user confirms transaction

Step 3. The EBR Software asks interbank to process transaction

Step 4. The Interbank processes transaction

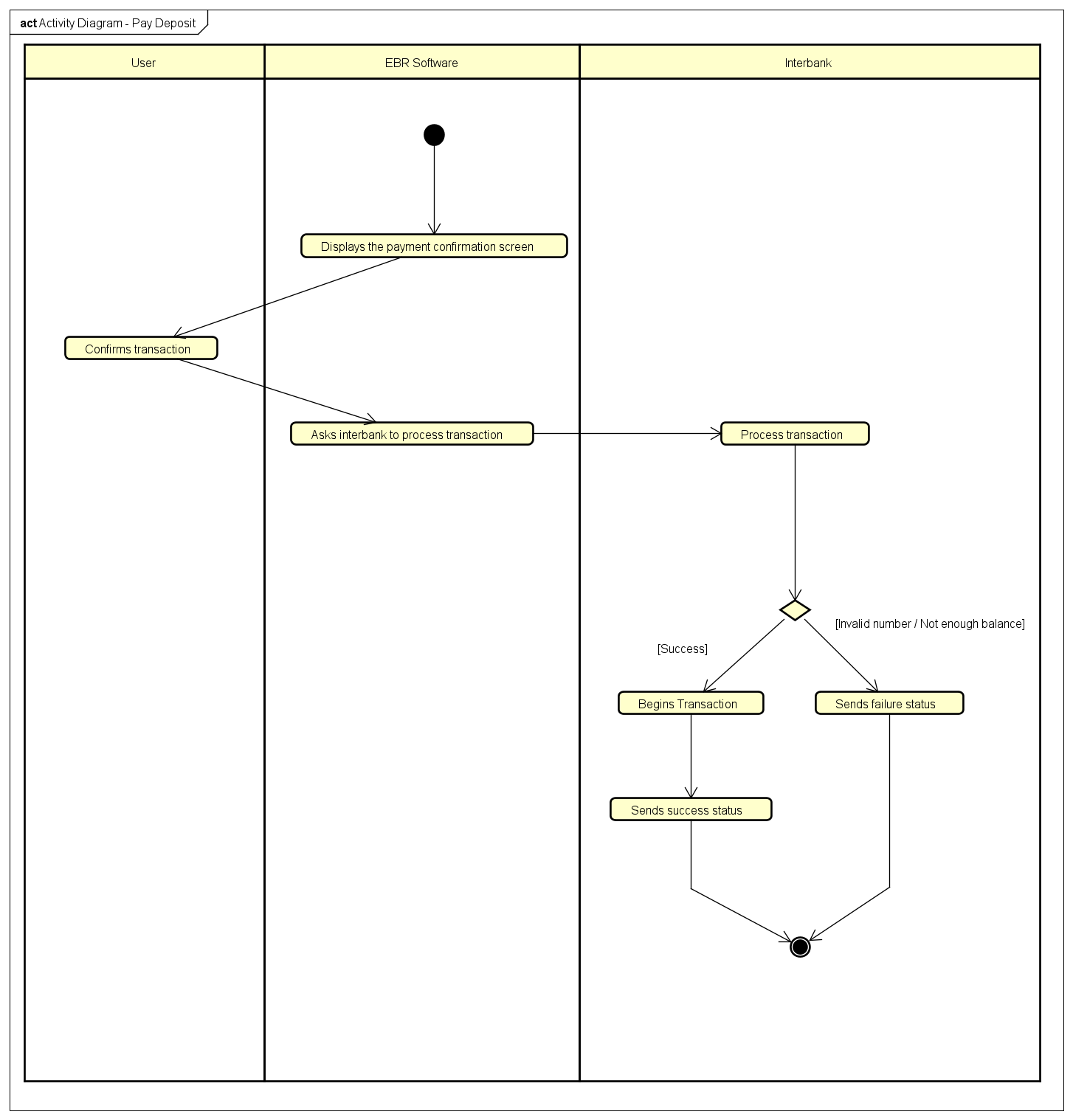
Step 5. The Interbank begins transaction

1. **Alternative flows**

*Table 8 - 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 9 - 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**

UC006

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. User requests to return a bike

Step 2. EBR software requests user to choose a dock to return bike

Step 3. EBR software Summarizes session information, calculates rental fee, calculates excess money and notifies to User

Step 4. User confirms information

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

Step 6. EBR software calls Use case “Refund deposit after deducting rental fee”

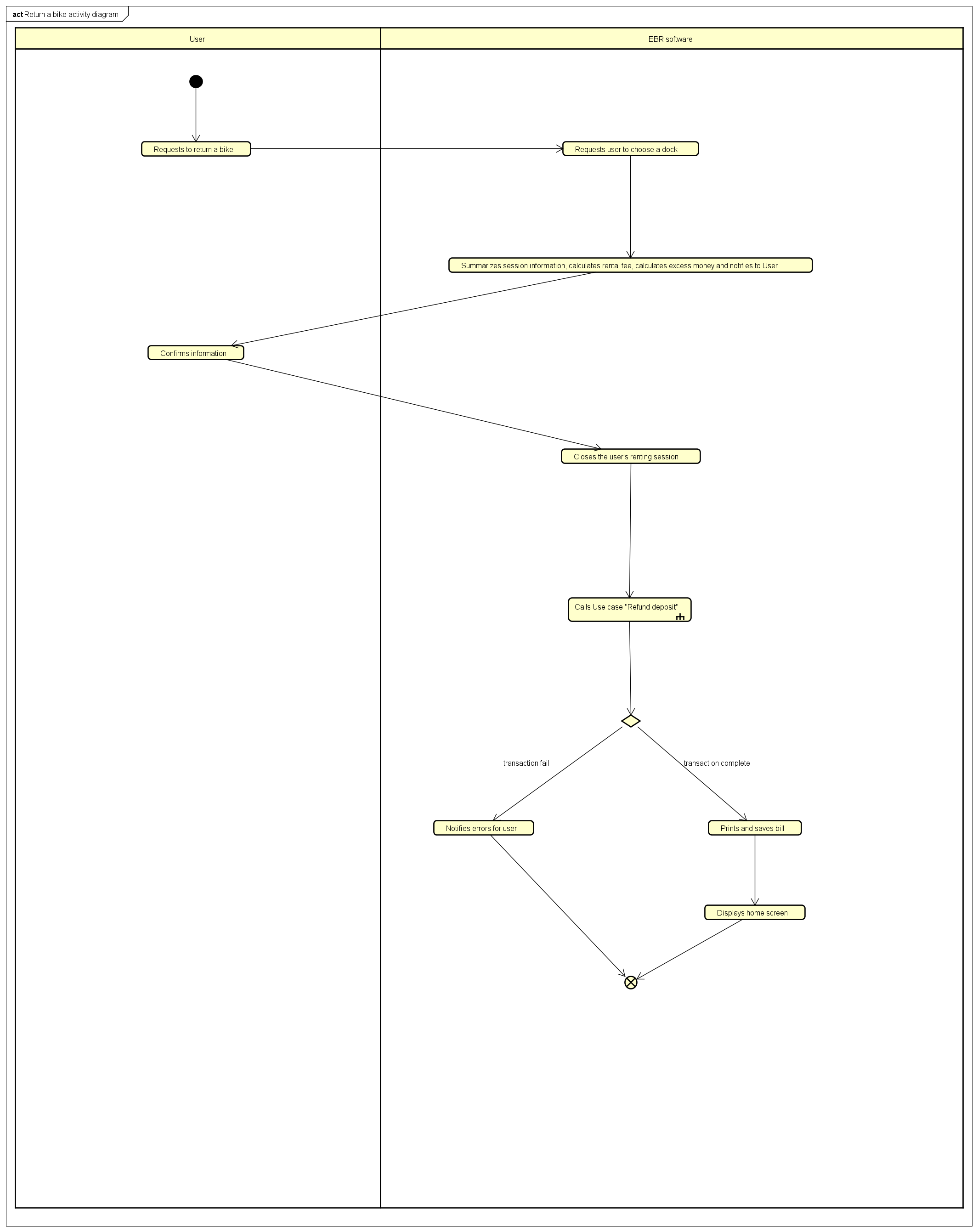
Step 7. EBR software returns home screen

1. **Alternative flows**

*Table 10 - Alternative flow of events for UC “Name of the Use Case”*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Location | Condition | Action | Resume location |
| 1 | At step 6 | Transaction fail | Notifies transaction fail | End use case |

1. **Activity diagrams**



1. **Input data**

*Table 11 - Input data of “Return a Bike”*

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

1. **Output data**

*Table 12 - Output data of “Return a Bike”*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 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”

**Use Case “Refund Deposit”**

1. **Use case code**

UC007

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 calls use case “Setup payment method"

Step 2. EBR software asks for transaction

Step 3. Interbank validates payment method

Step 4. Proceeds transaction

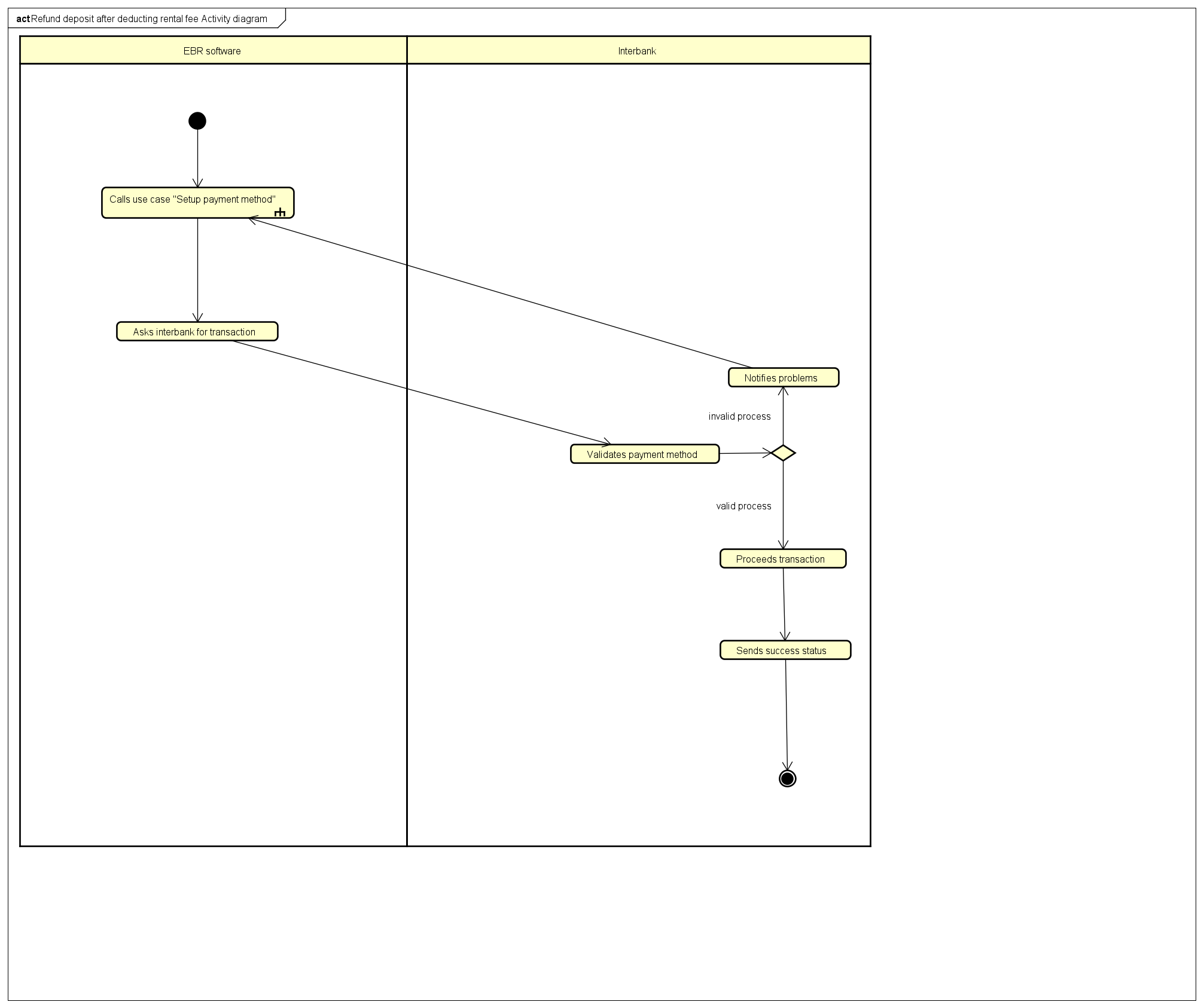
Step 5. Sends success result

1. **Alternative flows**

*Table 13 - Alternative flow of events for UC “Refund Deposit”*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Location | Condition | Action | Resume location |
| 1 | At step 3 | Invalid process | Notifies problems | Resume at step 1 |

1. **Activity diagrams**



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

*Table 14 - Output data of “Refund Deposit”*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 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 failed due to ... |

1. **Postconditions**

# Supplementary specification

## Functionality

User can cancel almost every current activity by backing to previous activity or to go to the base activity (Docks view)

## 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 100 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