
# Software Requirements Specification

**for**

# 2. Banking Portal System

**Version 0.2**

**Prepared by**

**Student Roll Number: B200059CS** **Student Name: Vaisakh Ramachandran**

**Student Roll Number: B200841CS** **Student Name: Rishit Kumar Chordia**

**Project Owner:** **Piyush Keswani**

**Course:** **CS4097D Object Oriented Systems Laboratory**

**Date:** **10/11/2022**

# 1 Introduction

## 1.1 Document Purpose

The project is to create a Banking Portal System which could be set up at a kiosk in a bank. This small desktop application will support bank account holders to perform basic banking activities. It will also facilitate the bank managers to supervise the bank operations and to provide customer service. The purpose is to showcase a mechanism to enable fast, contactless service for a user while maintaining a degree of security.

## 1.2 Product Scope

Users should be able to register through their already existing accounts or enabled to create a new account. They then have access to banking facilities such as transferring funds, checking account balance and viewing their bank statements.

The provision for bank managers to log in allows them to view the details of all existing accounts, classifying them based on various criteria as needed. They also provide services to users such as unblocking blocked accounts or changing their account type.

## 1.3 Definitions, Acronyms and Abbreviations

| Abbreviation | Full-form |
|---|---|
| ATM | Automated Teller Machine |
| BPS | Banking Portal System |
| NRI | Non-Resident Indian |

## 1.4 Document Conventions

Font style – Arial font size 12

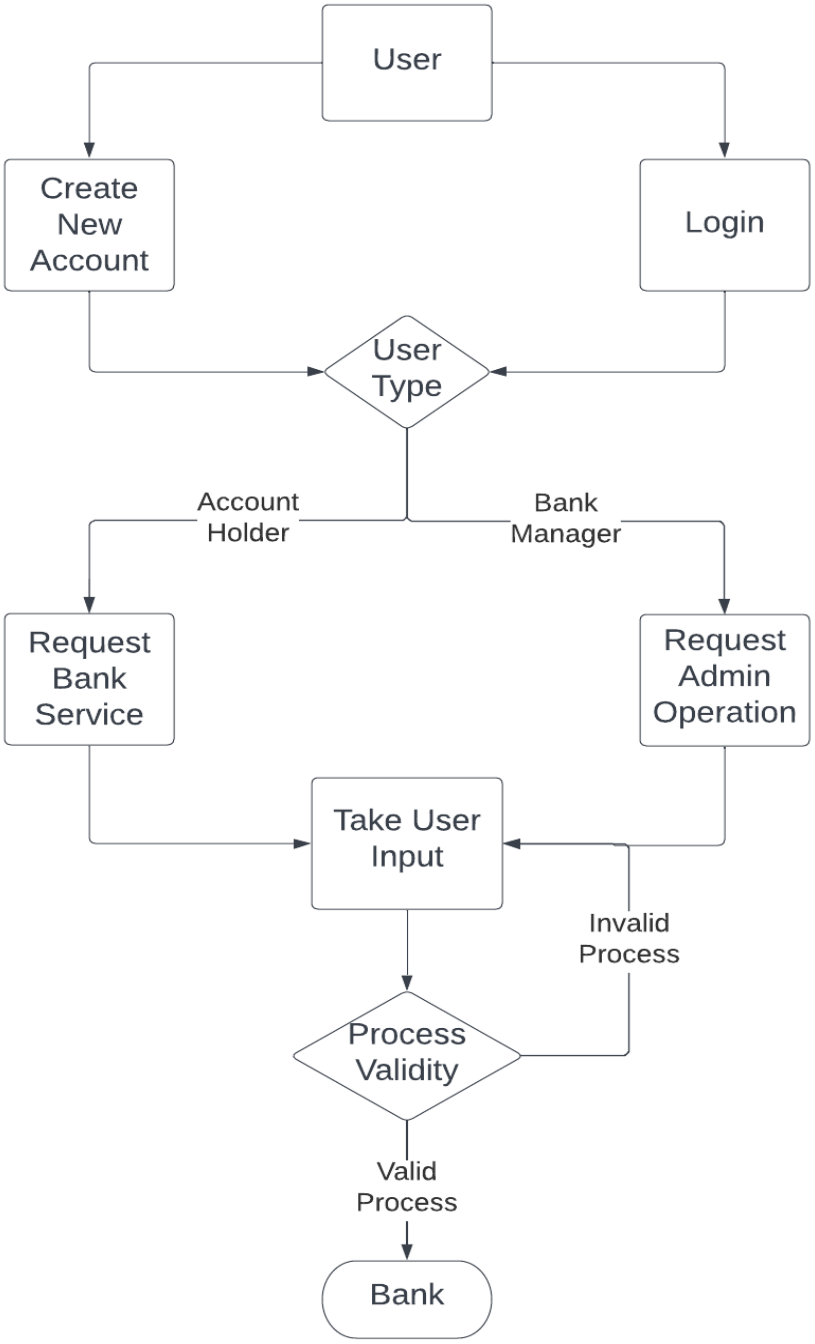
## 1.5 References and Acknowledgements

https://www.javatpoint.com/uml-use-case-diagram - Use Case Diagrams
https://www.youtube.com/watch?v=zid-MVo7M-E – Use Case conventions
https://lucid.app/ - Creating the Use Case diagram

# 2 Overall Description

## 2.1 Product Overview

The product is an upgrade on the already existing self service bank kiosks, the passbook printing kiosk and the ATM kiosk, with a few additional functionalities. It acts as a self-contained product, contacting and updating the bank databases of users and transactions as required. The following flow diagram illustrates the working of the BPS.



## 2.2 Product Functionality

List of the major functions provided by the system for a User:

- Login
- Create New Account

List of the major functions provided by the system for an Account Holder:

- Change Password
- Check Balance
- Print Statement
- Transfer Funds
- Logout

List of the major functions provided by the system for a Bank Manager:

- Change Account Type
- Unblock Account
- View Transactions
- View All Accounts
- View All Indian Accounts
- View All NRI Accounts
- View All Blocked Accounts
- Logout

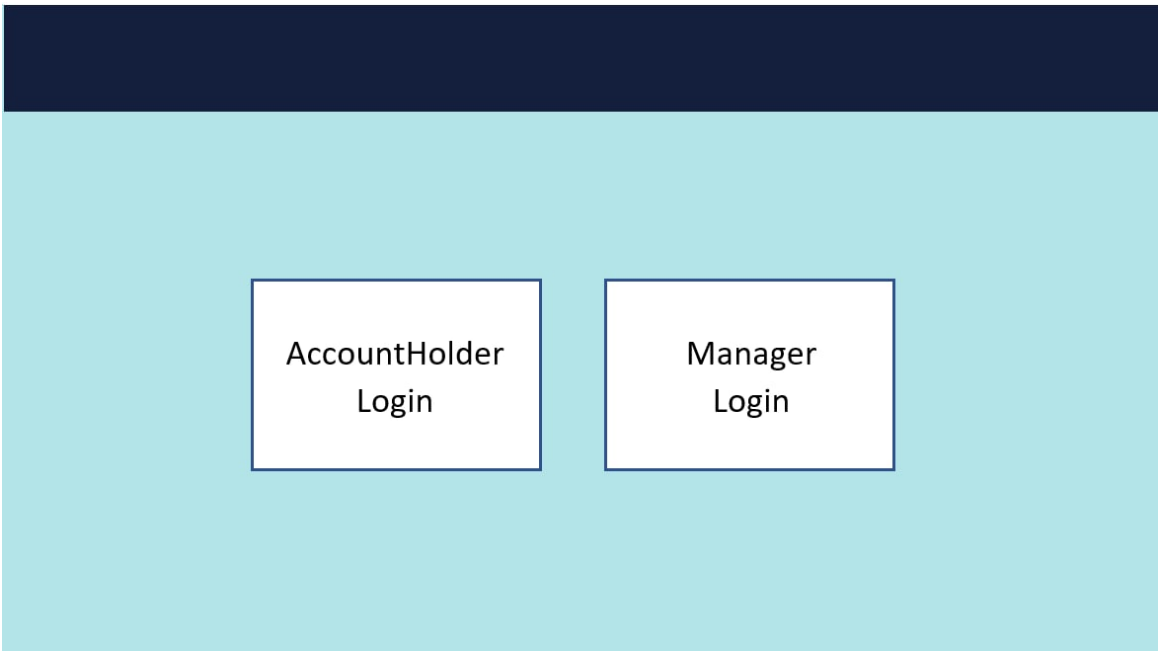
List of helper functions assisting the major functions:

- Check If Account Exists
- Check Transaction Validity
- Calculate Transaction Fees
- Block Account

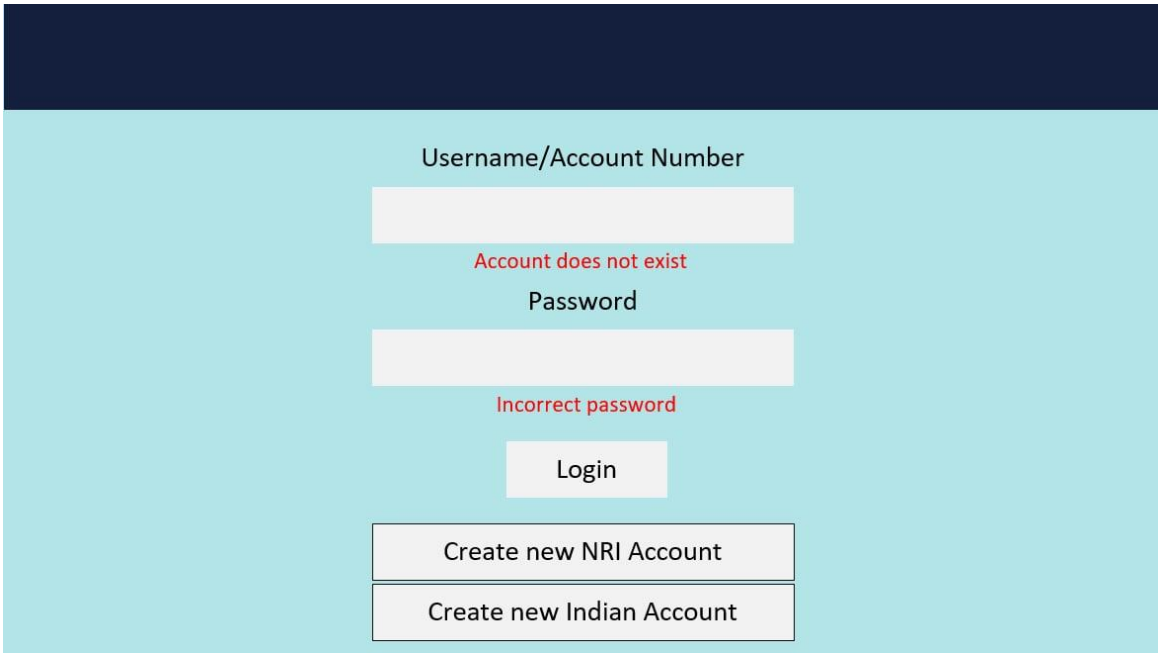
# 3 Specific Requirements

## 3.1 User Interfaces

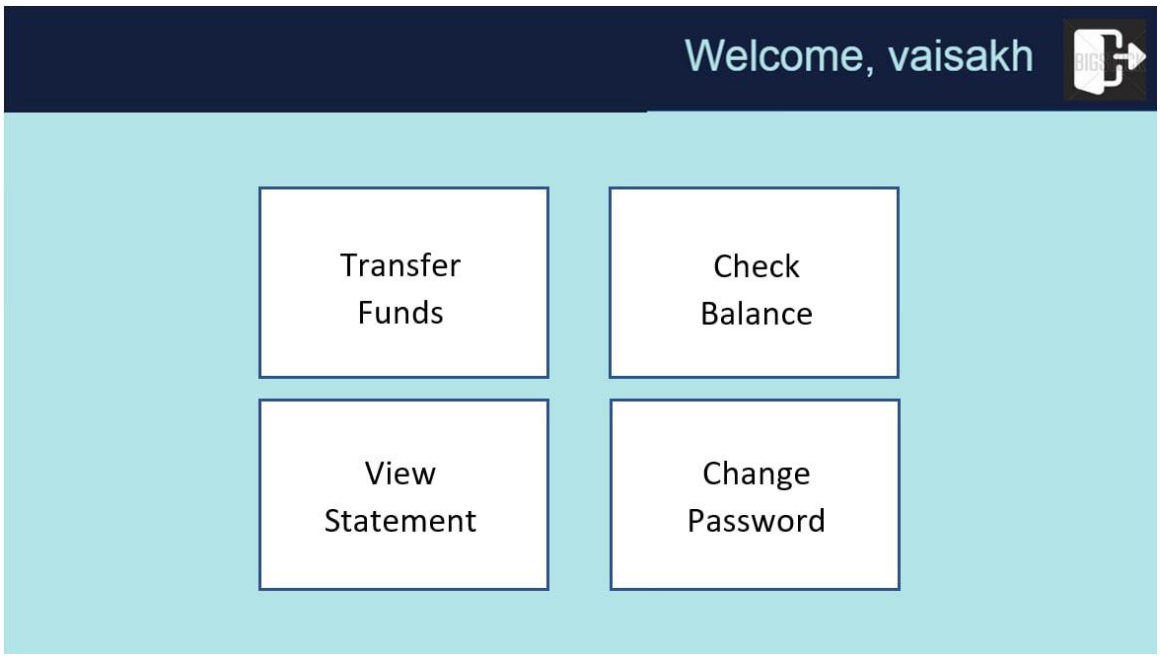
The following is the home screen of the kiosk:



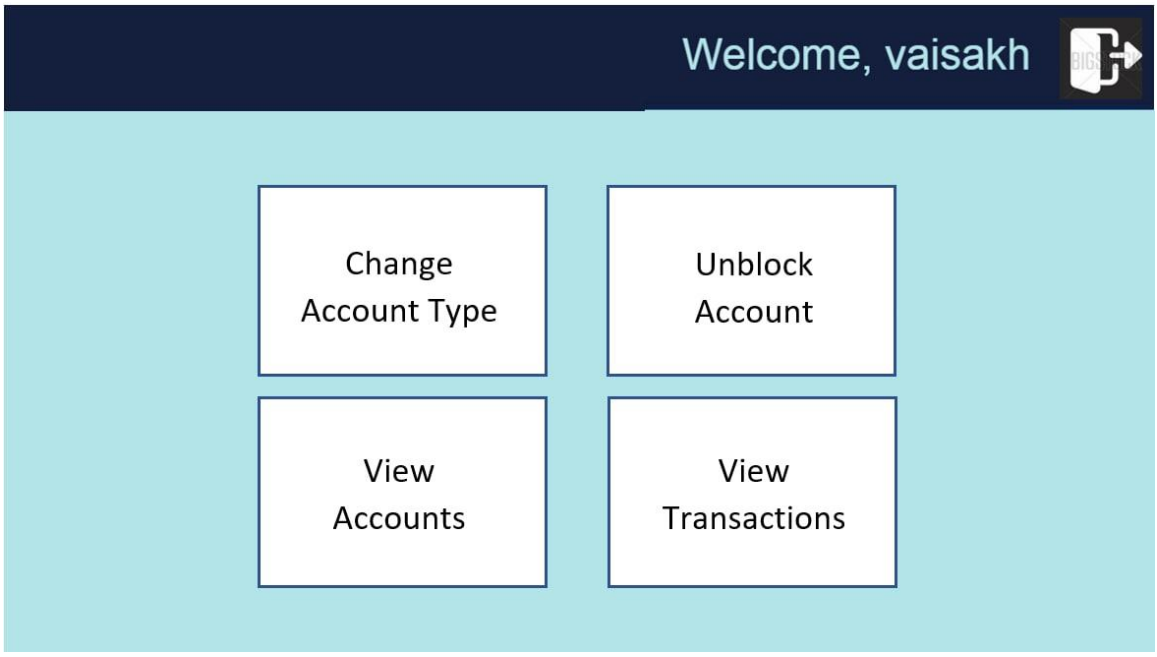

On clicking on AccountHolder Login, the User is led to the following screen where they are required to enter their credentials. The manager login is a similar screen without the create account options:



The Account Holder then has the following services available to them:




The Bank Manager is also given their set of services as shown below:




Each of these options for both types of users lead to individual functional interfaces that provide the desired service.

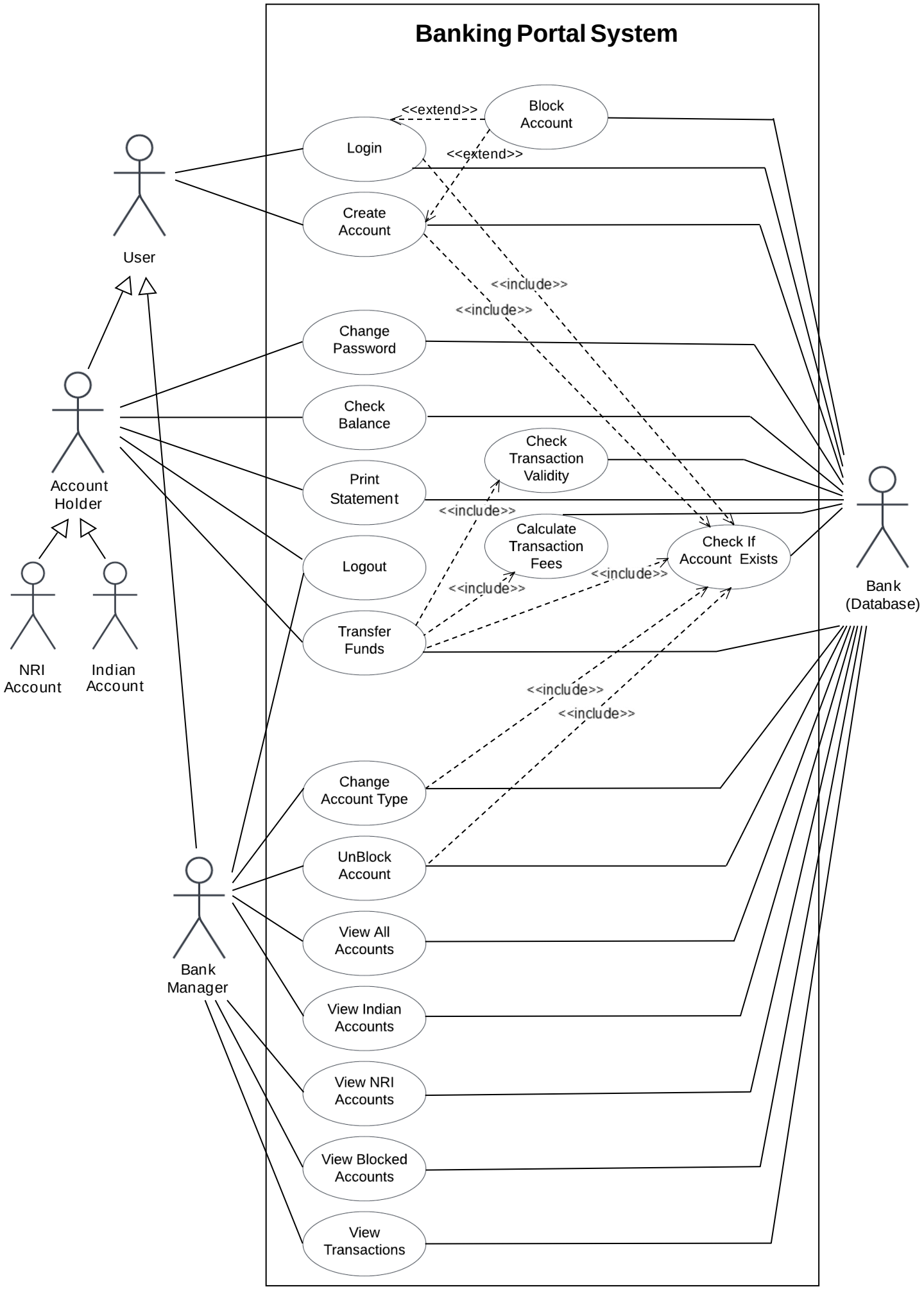
## 3.2 Functional Requirements

List of the major features provided/utilized by the system:

- Check if Account Exists: Checks if an account with given account number or username exists.
- Login: Allows user to log into their account.
- Create New Account: Allows a new user to create an account.
- Block Account: Blocks an account after five incorrect password attempts.
- Change Password: Allows an account holder to change their password.
- Check Balance: Displays the balance in the account.
- Print Statement: Displays the account statement.
- Transfer Funds: Allows account holder to send money to another account.
- Check Transaction Validity: Checks if the account maintains minimum balance after the transaction.
- Calculate Transaction Fees: Charges fee incurred during the transaction based on the user's account type.
- Logout: Logs the user out of their account.
- Change Account Type: Changes an NRI account to an Indian account.
- Unblock Account: Unblocks a blocked account.
- View Transactions: Displays all transactions through the bank.
- View Accounts: Displays all accounts or accounts falling under these criteria (NRI accounts, Indian accounts or blocked accounts).

## 3.3 Use Case Model

The following page contains a Use Case Model describing the functionality and interactions involved in the BPS.


Banking Portal System
Block Account
<<extend>>
Login
<<extend>>
Create Account
User
<<include>>
<<include>>
Change Password
Check Balance
Check Transaction Validity
Print Statement
<<include>>
Calculate Transaction Fees
Check If Account Exists
Logout
<<include>>
<<include>>
Transfer Funds
Account Holder
NRI Account
Indian Account
Bank (Database)
<<include>>
<<include>>
Change Account Type
UnBlock Account
View All Accounts
View Indian Accounts
View NRI Accounts
View Blocked Accounts
View Transactions
Bank Manager

# Design Document
## for
# Banking Portal System

Version 1.0

Prepared by Team 2:
(Based on SRS Version 0.2 prepared by Team 2)


| | | |
|---|---|---|
| **Rishit Kumar Chordia** | **B200841CS** | **rishit_b200841cs@nitc.ac.in** |
| **Vaisakh Ramachandran** | **B200059CS** | **vaisakh_b200059cs@nitc.ac.in** |


**Project Owner:** **Piyush Keswani**

**Course:** **CS4096 Object Oriented Systems Laboratory**

**Date:** **14/11/2022**

# Glossary

| | |
|---|---|
| SRS | Software Requirements Specification |
| BPS | Banking Portal System |
| 1NF | First Normal Form |
| 2NF | Second Normal Form |
| 3NF | Third Normal Form |

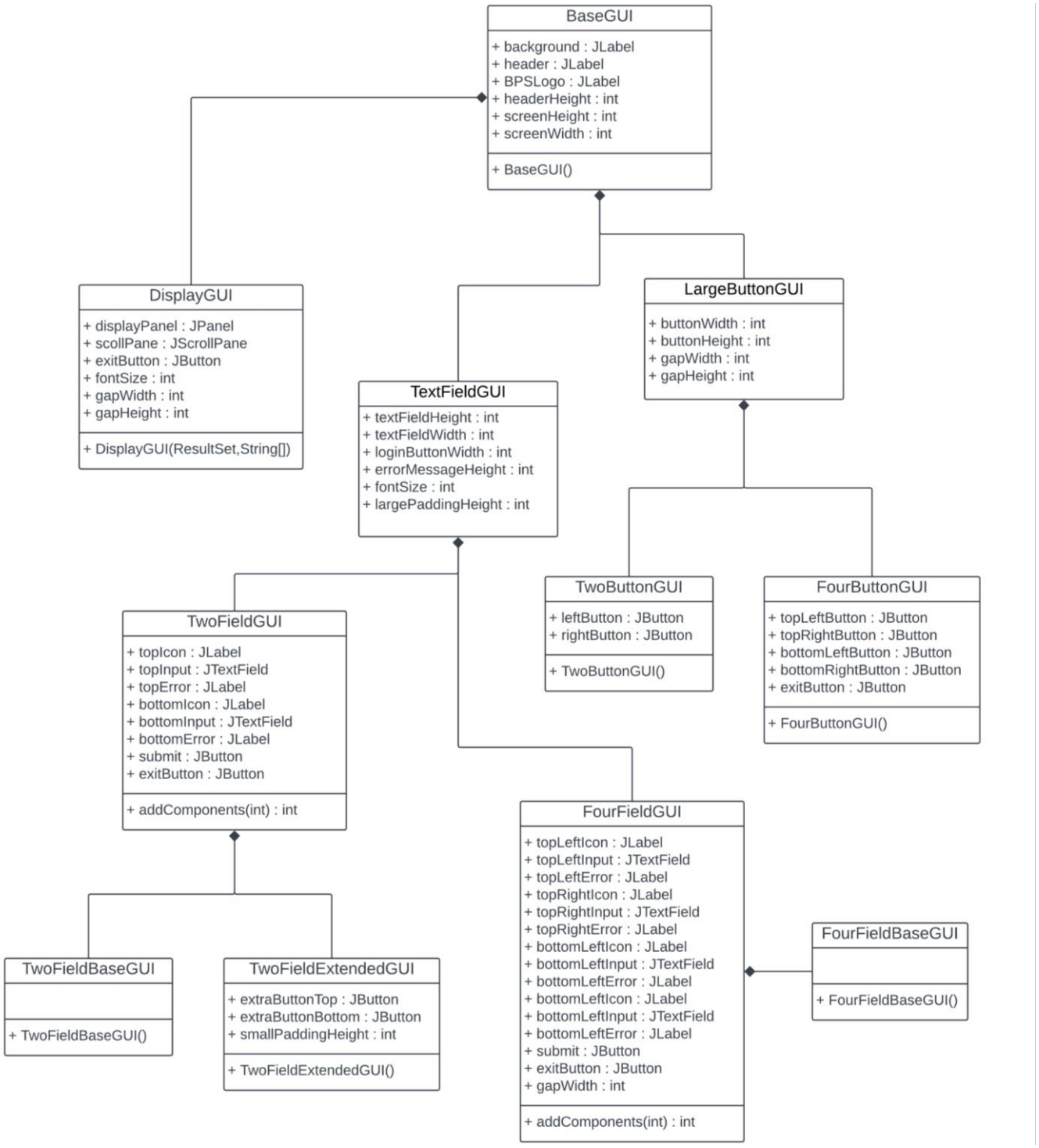
# Table of contents

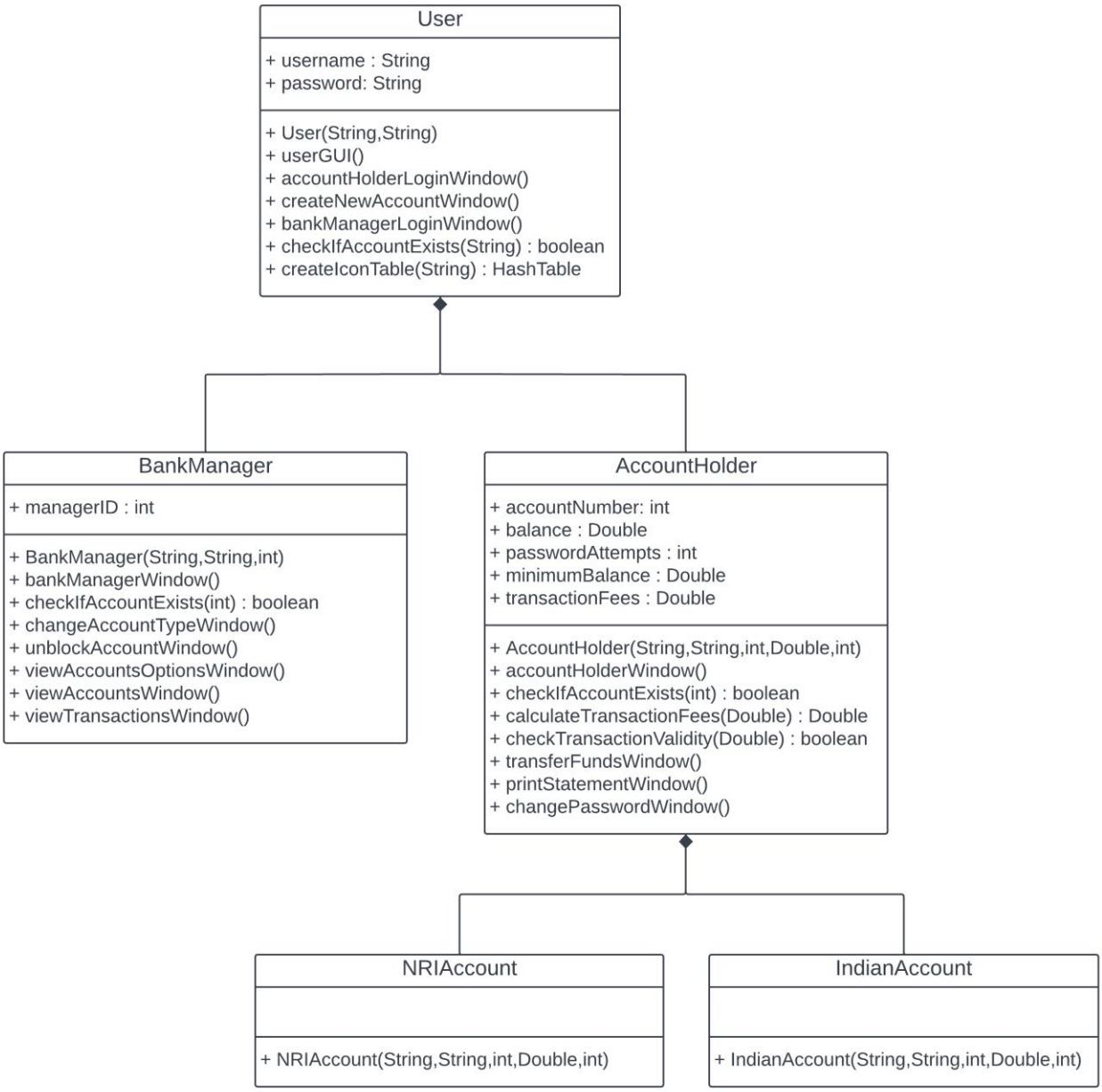
# 1. Detailed Design through UML diagrams

## 1.1 System model using Class Diagram

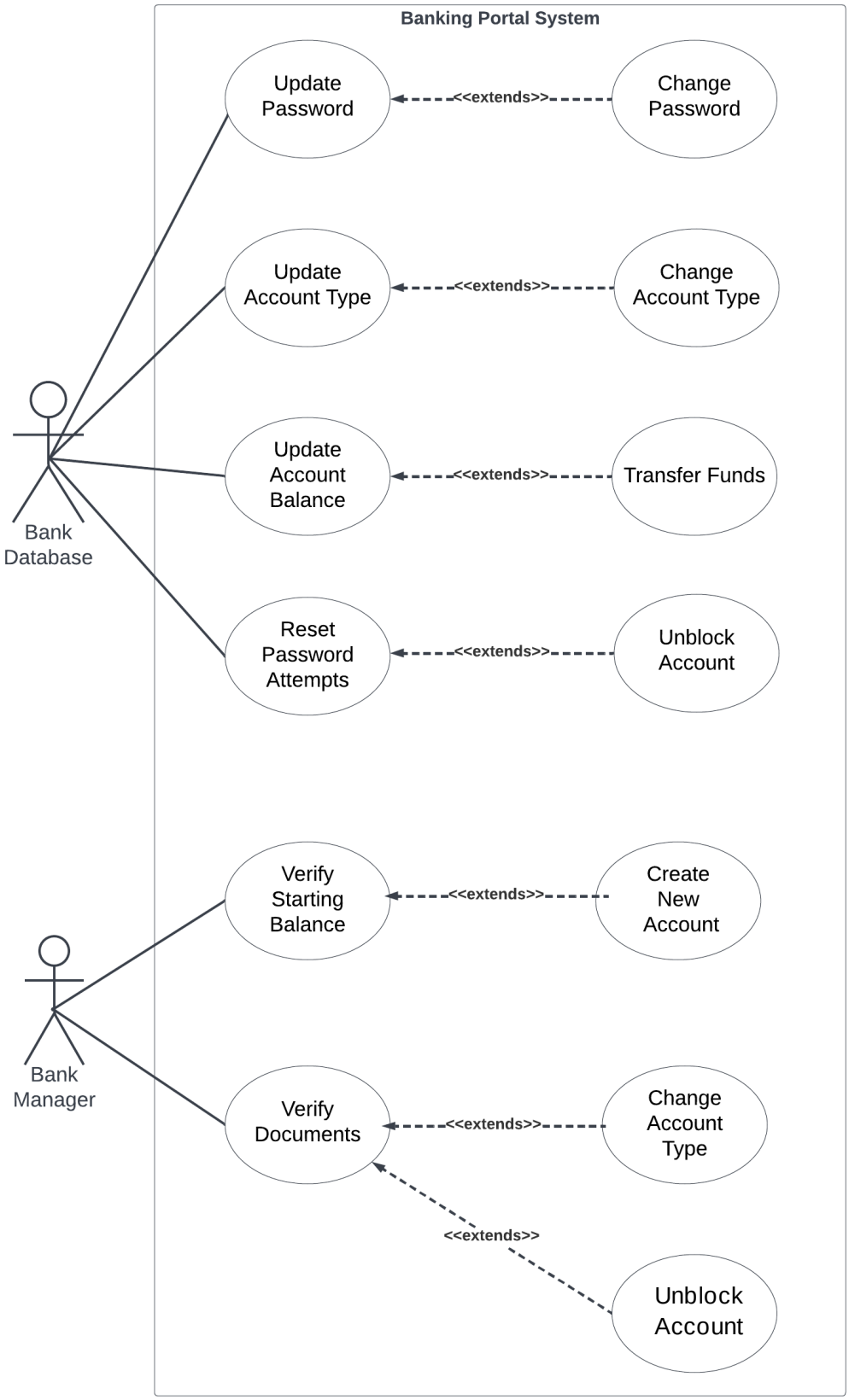
Class Diagram in the Unified Modelling Language is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods) and the relationships among classes.

### 1.1.1 Class Diagram

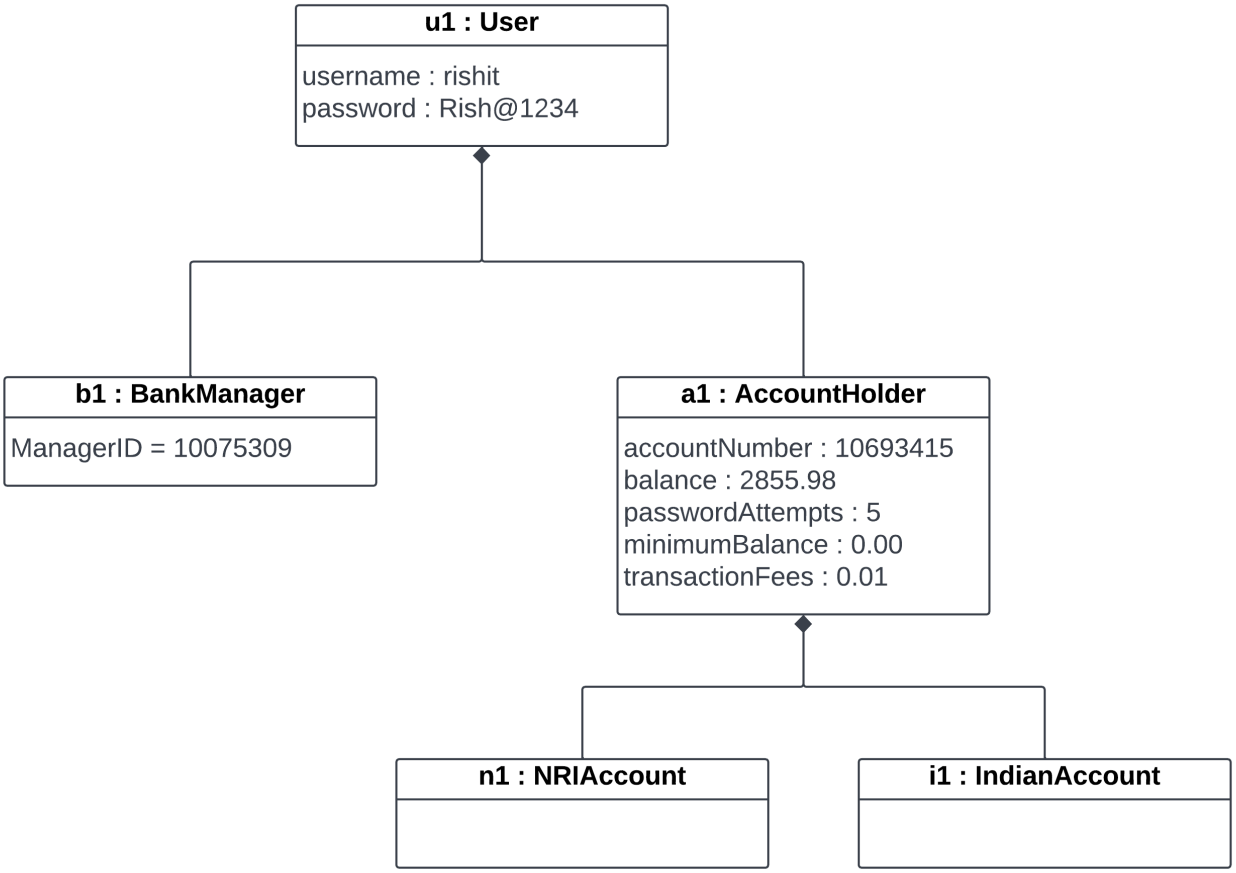



User
+ username : String
+ password: String
+ User(String,String)
+ userGUI()
+ accountHolderLoginWindow()
+ createNewAccountWindow()
+ bankManagerLoginWindow()
+ checkIfAccountExists(String) : boolean
+ createIconTable(String) : HashTable
BankManager
+ managerID : int
+ BankManager(String,String,int)
+ bankManagerWindow()
+ checkIfAccountExists(int) : boolean
+ changeAccountTypeWindow()
+ unblockAccountWindow()
+ viewAccountsOptionsWindow()
+ viewAccountsWindow()
+ viewTransactionsWindow()
AccountHolder
+ accountNumber: int
+ balance : Double
+ passwordAttempts : int
+ minimumBalance : Double
+ transactionFees : Double
+ AccountHolder(String,String,int,Double,int)
+ accountHolderWindow()
+ checkIfAccountExists(int) : boolean
+ calculateTransactionFees(Double) : Double
+ checkTransactionValidity(Double) : boolean
+ transferFundsWindow()
+ printStatementWindow()
+ changePasswordWindow()
NRIAccount
+ NRIAccount(String,String,int,Double,int)
IndianAccount
+ IndianAccount(String,String,int,Double,int)

# 1.2 Responsibilities - Usecase Diagram



## 1.3 Static snapshot of the system - Object Diagram



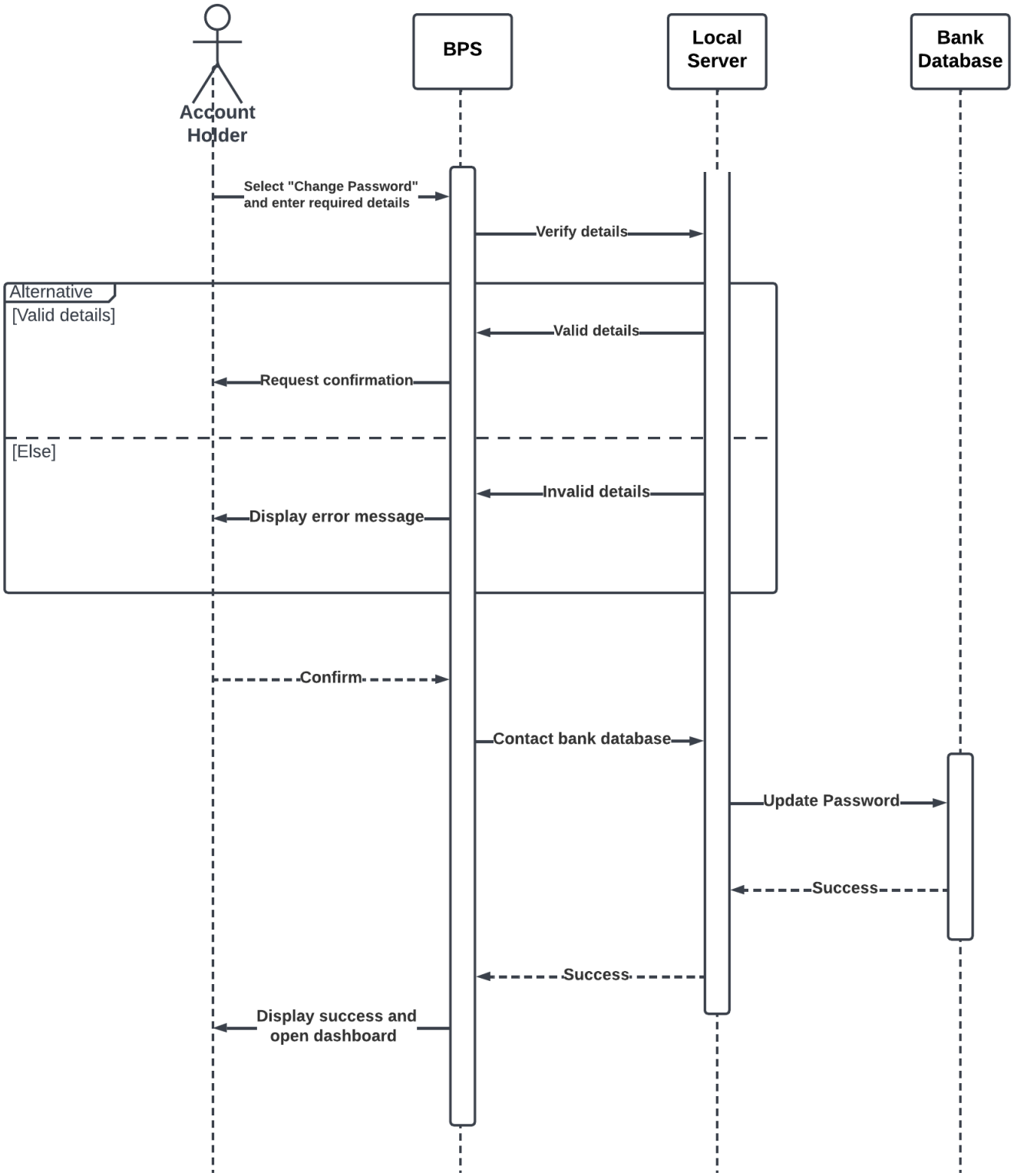

## 1.4 System Interactions through Sequence Diagrams

Sequence diagrams are interaction diagrams that show the sequence of messages exchanged by the set of objects performing a certain task. A sequence diagram shows, as parallel vertical lines (lifeline), different processes or objects that live simultaneously, and as horizontal arrows, the messages exchanged between them, in the order in which they occur.

### 1.4.1 Account Holder Functionalities

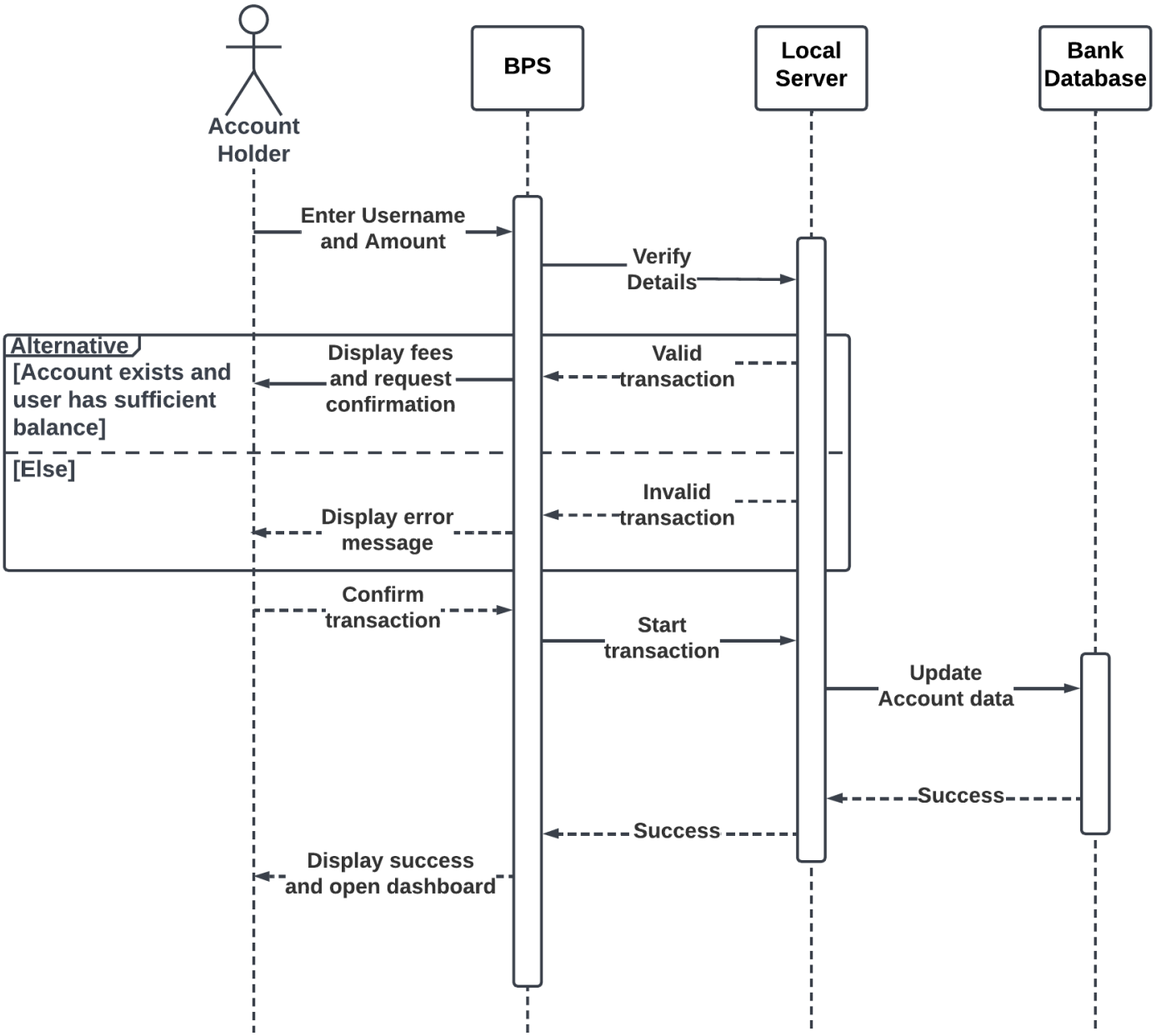
#### 1.4.1.1 Change Password

The following diagram represents the “Change Password” functionality.



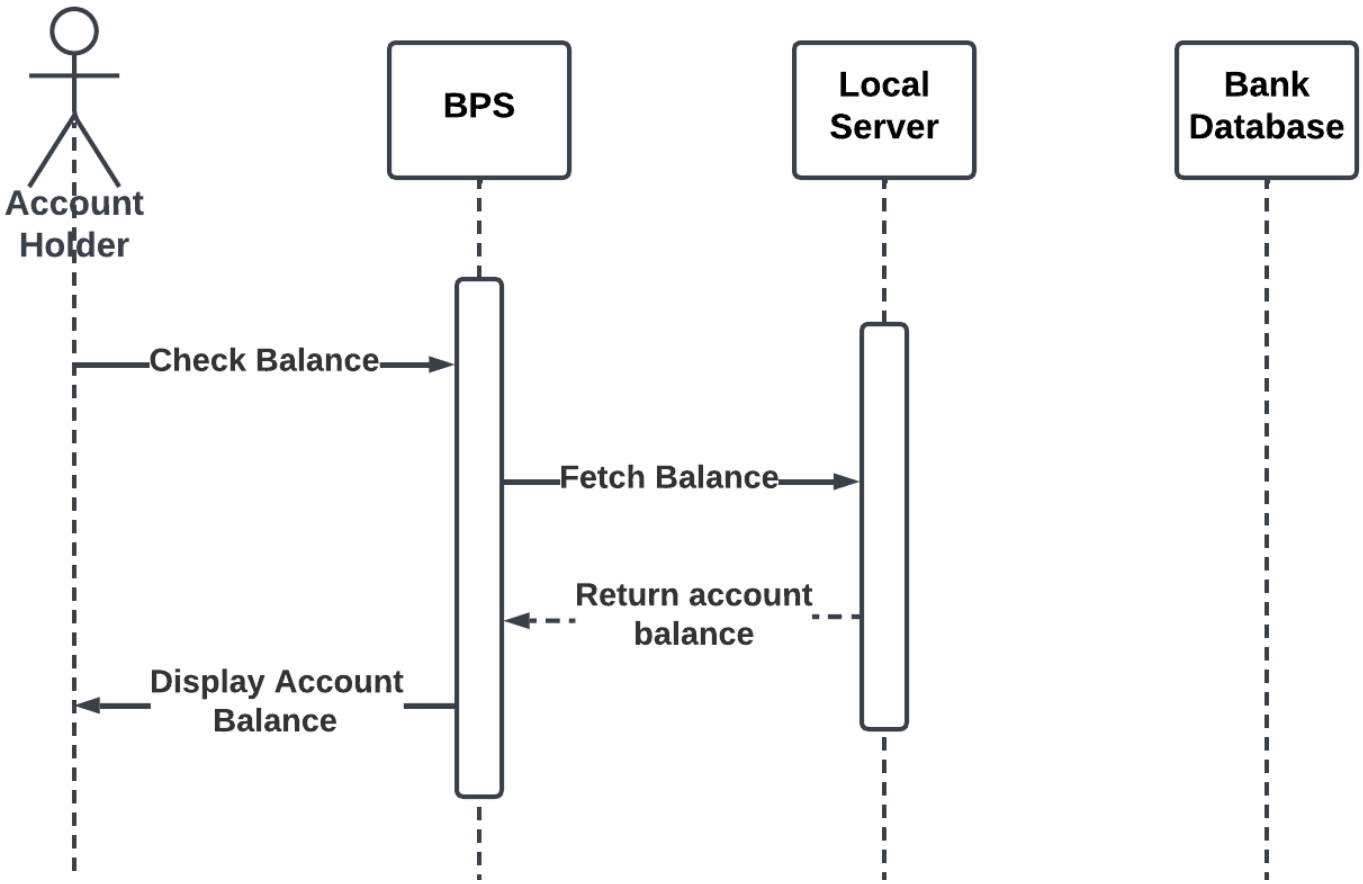
### 1.4.1.2 Transfer Funds

The following diagram represents the “Transfer Funds” functionality.



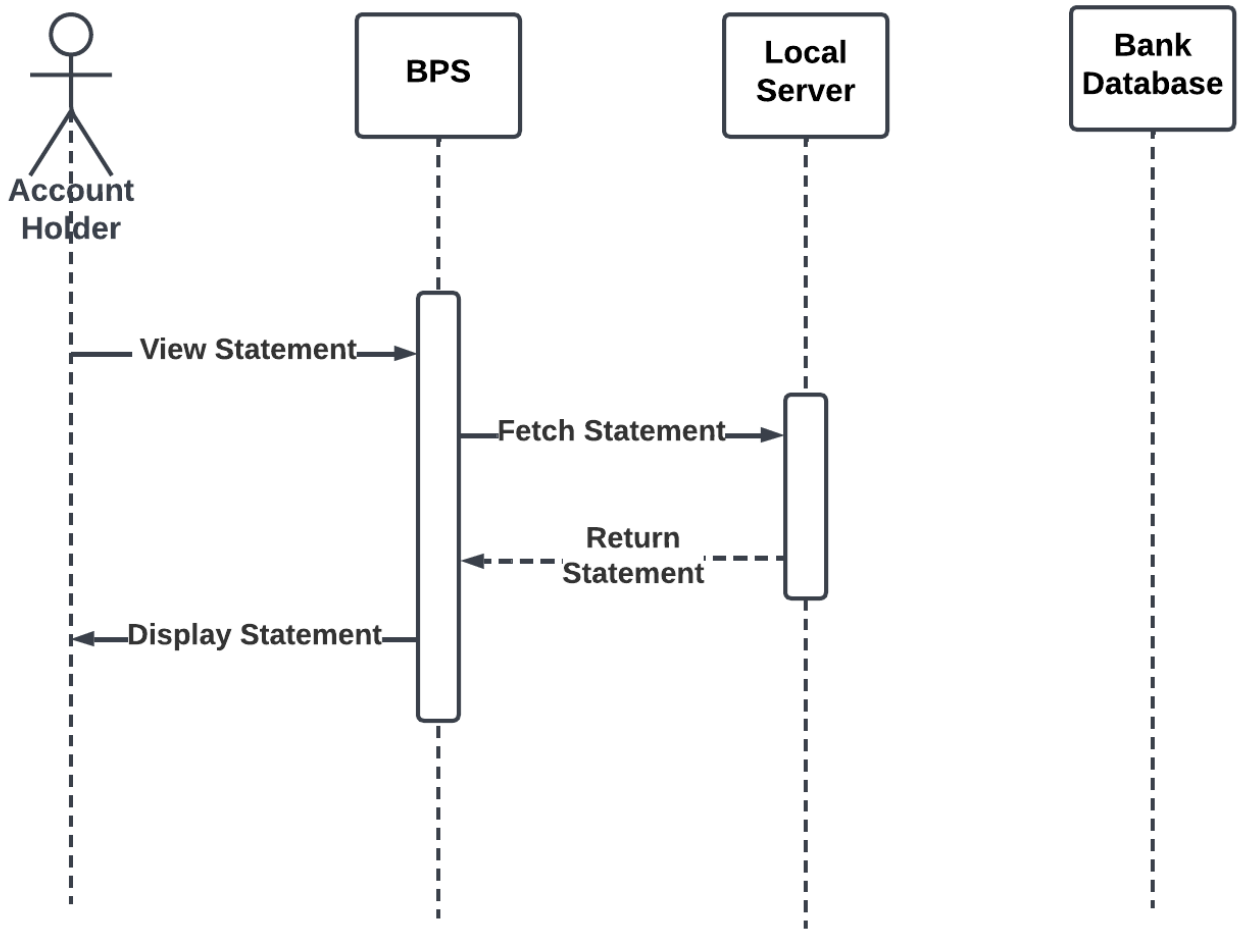
#### 1.4.1.3 Check Balance

The following diagram represents the “Check Balance” functionality.



#### 1.4.1.4 Print Statement

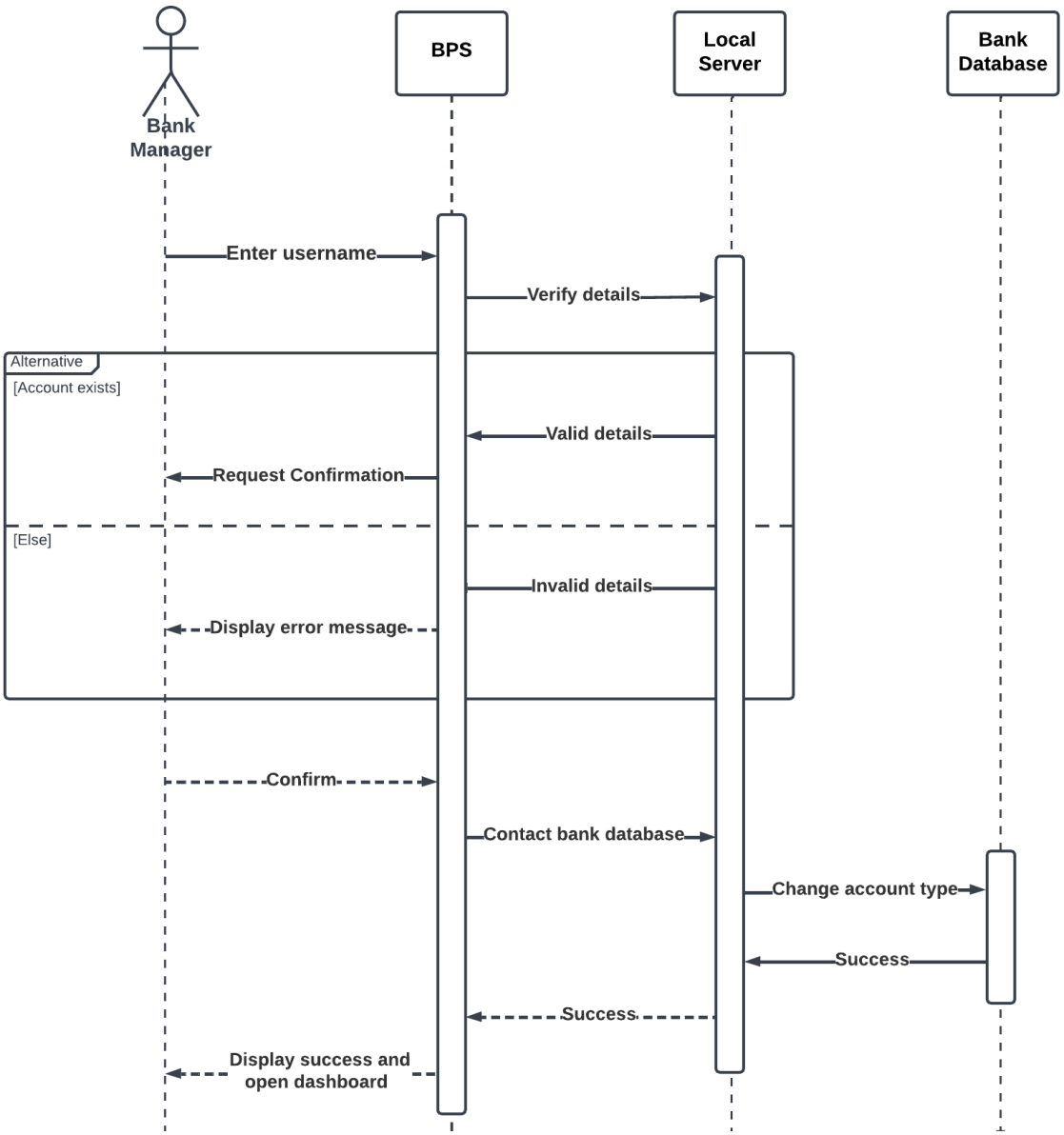
The following diagram represents the “Print Statement” functionality.



### 1.4.2 Bank Manager Functionalities

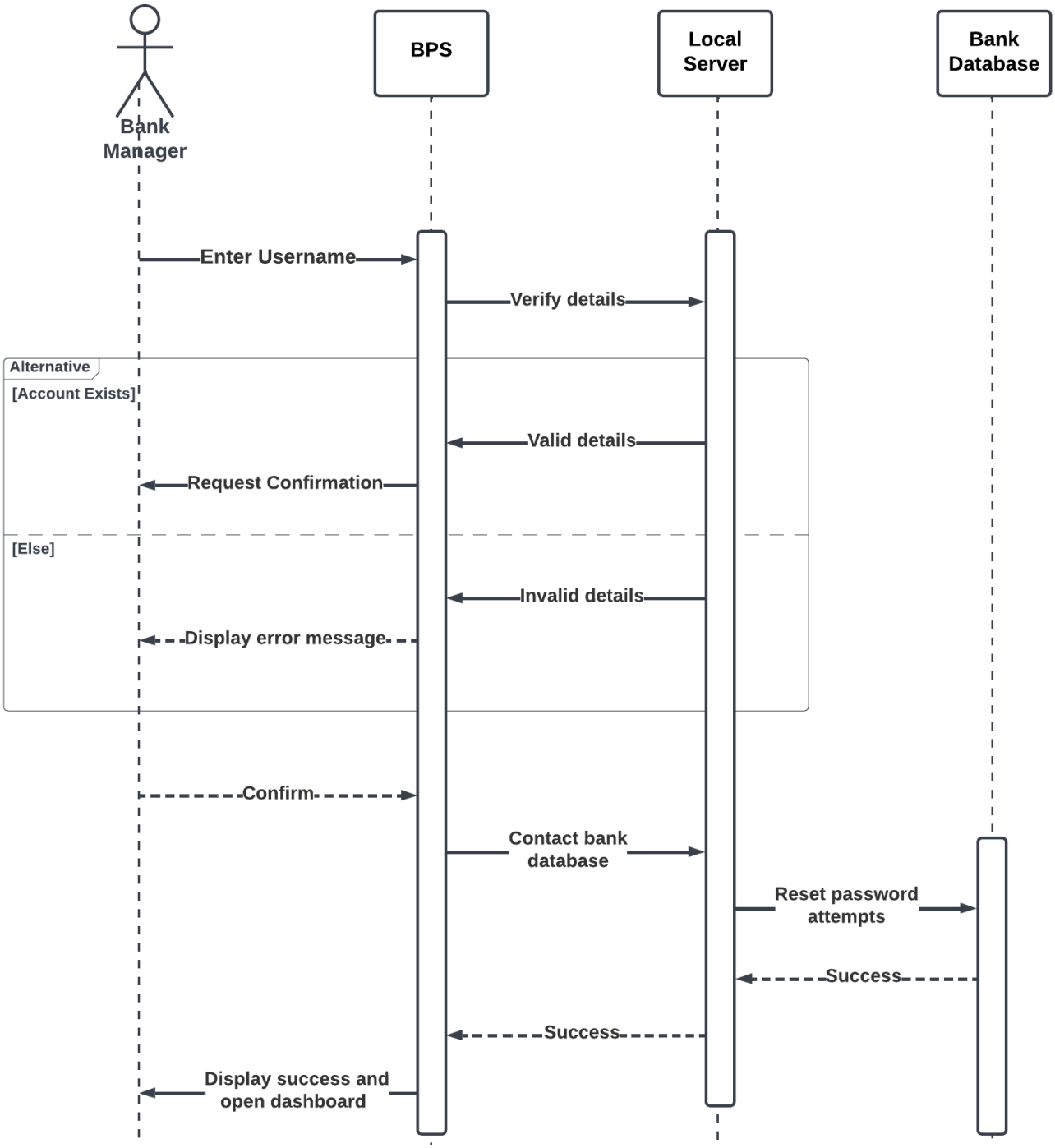
#### 1.4.2.1 Change Account Type

The following diagram represents the “Change Account Type” functionality.



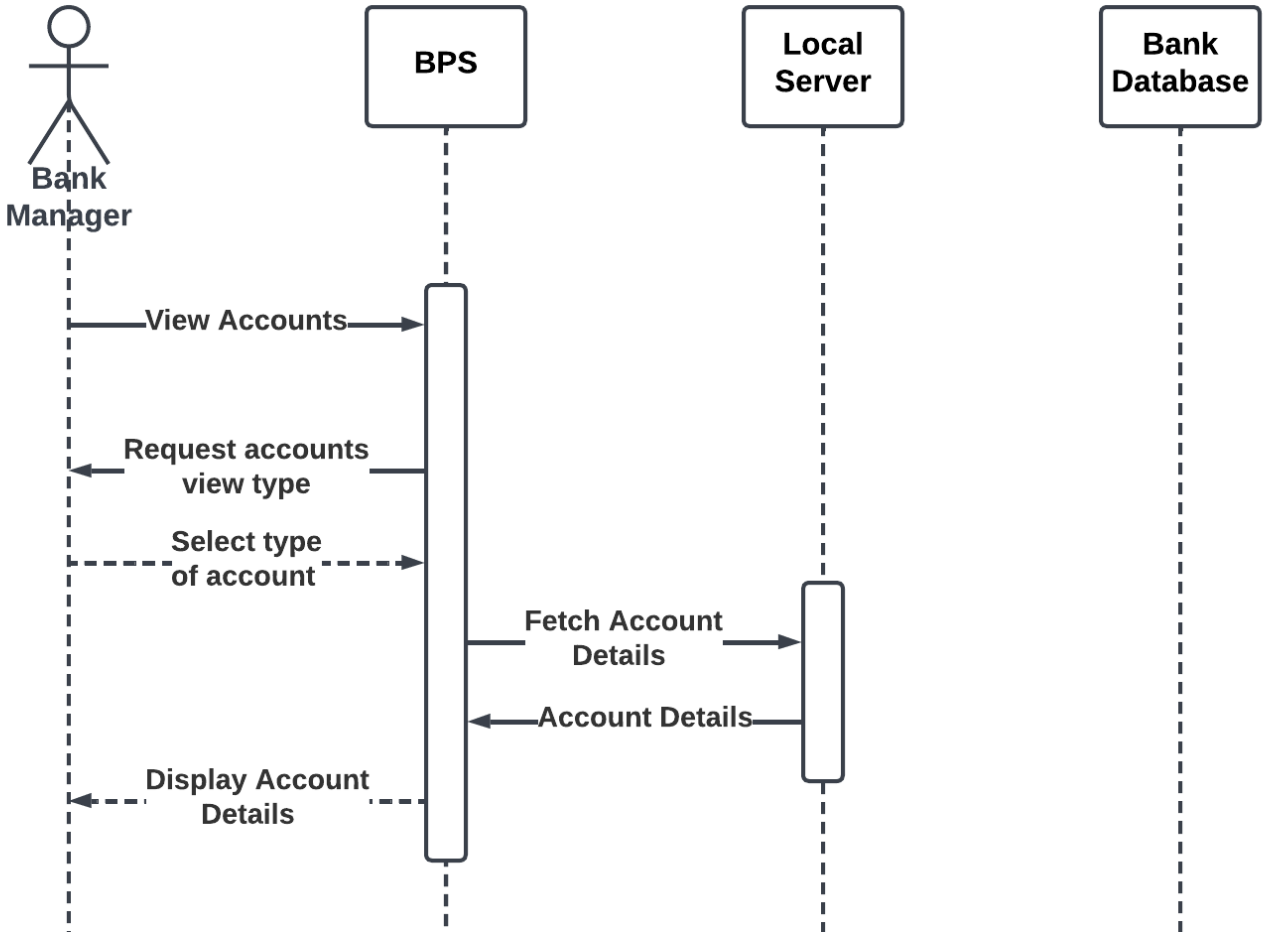
### 1.4.2.2 Unblock Account

The following diagram represents the “Unblock Account” functionality.



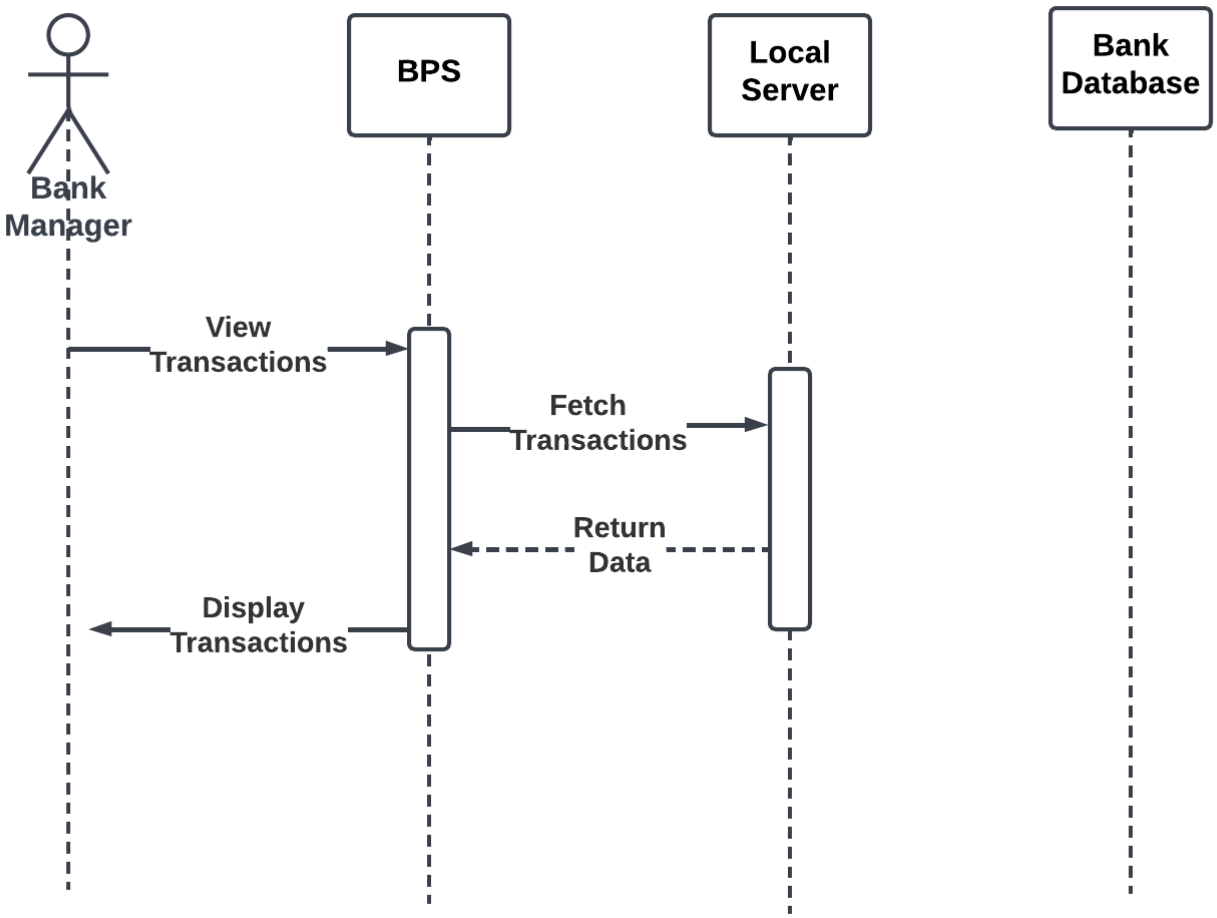
### 1.4.2.3 View Accounts

The following diagram represents the “View Accounts” functionality.



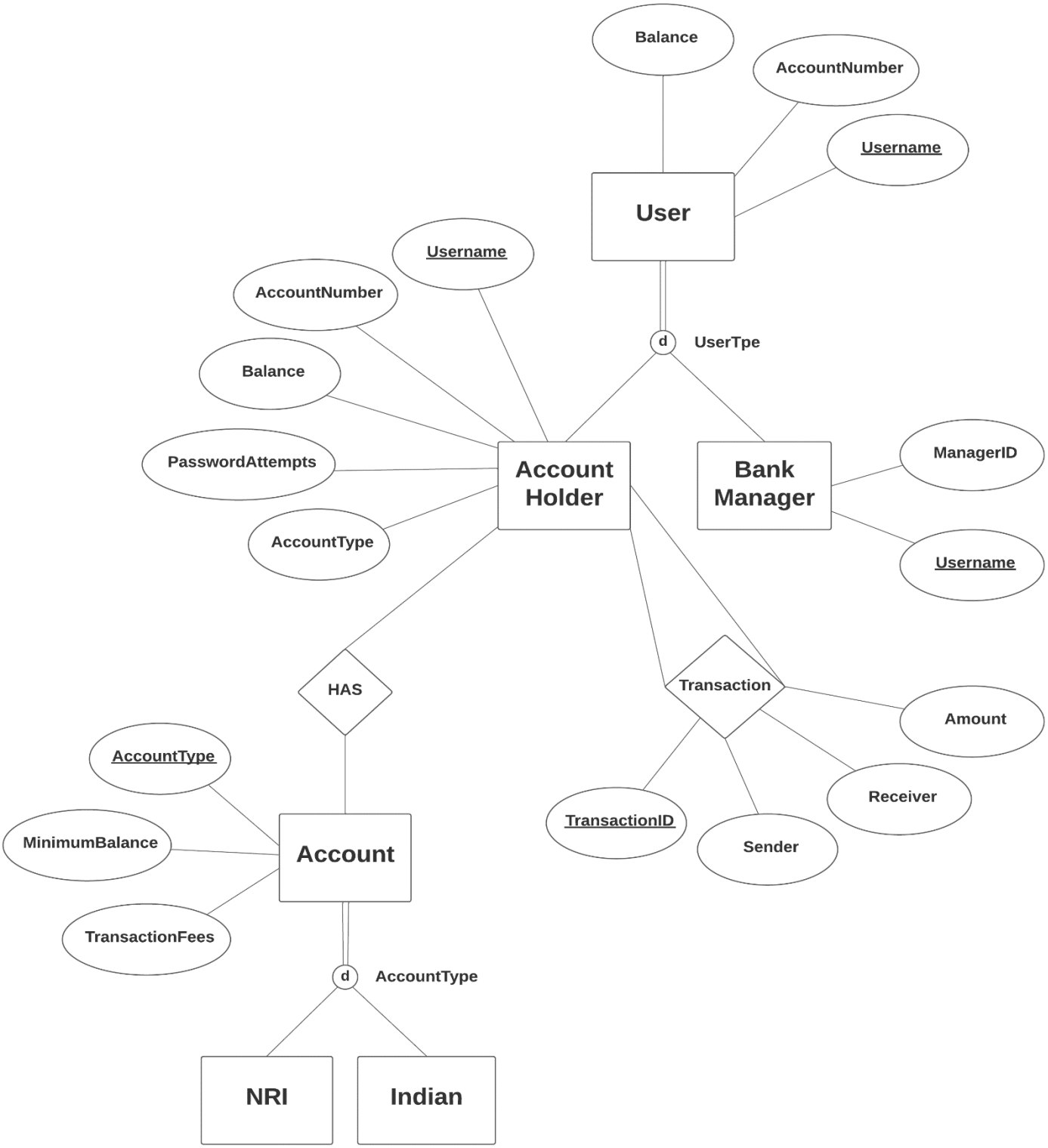
### 1.4.2.4 View Transactions

The following diagram represents the “View Transactions” functionality.



# 2. Database Design

## 2.1 ER Diagram

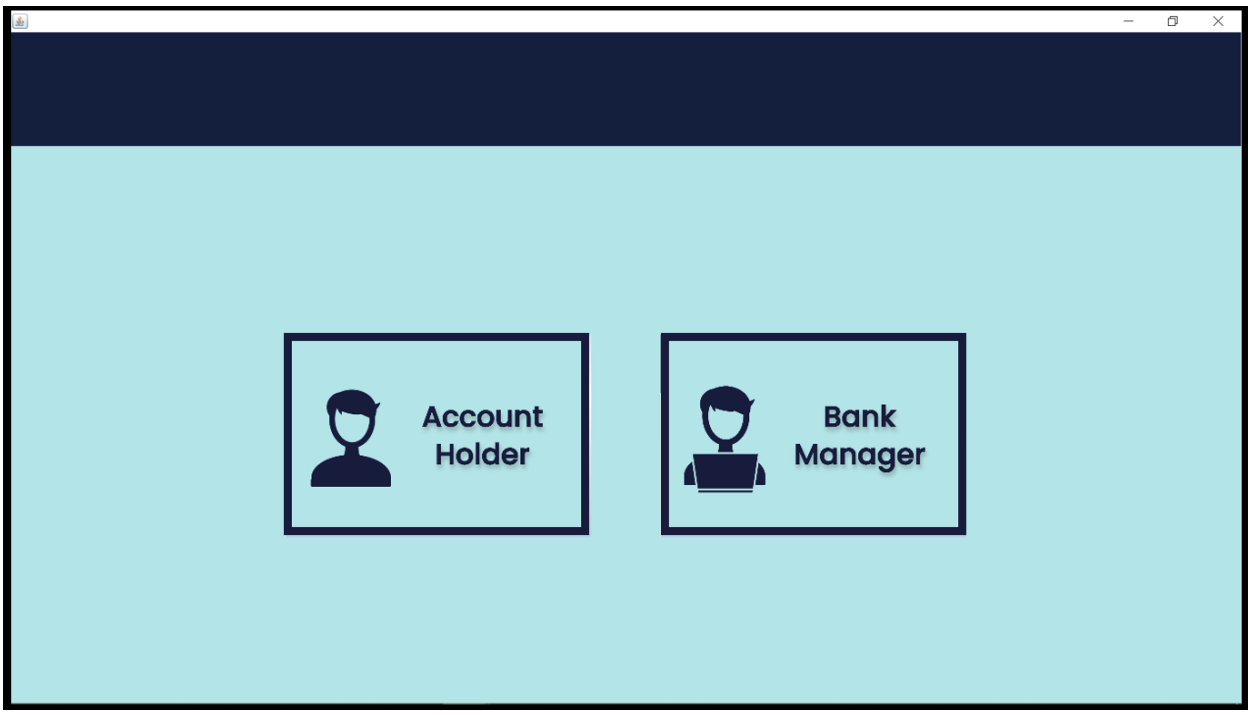



The proposed database has been designed to remove partial dependencies, making it 2NF normalized. In addition, all transitive dependencies have also been removed, thus making it an efficient 3NF normalized database.

## 3.2 User Interface Prototyping

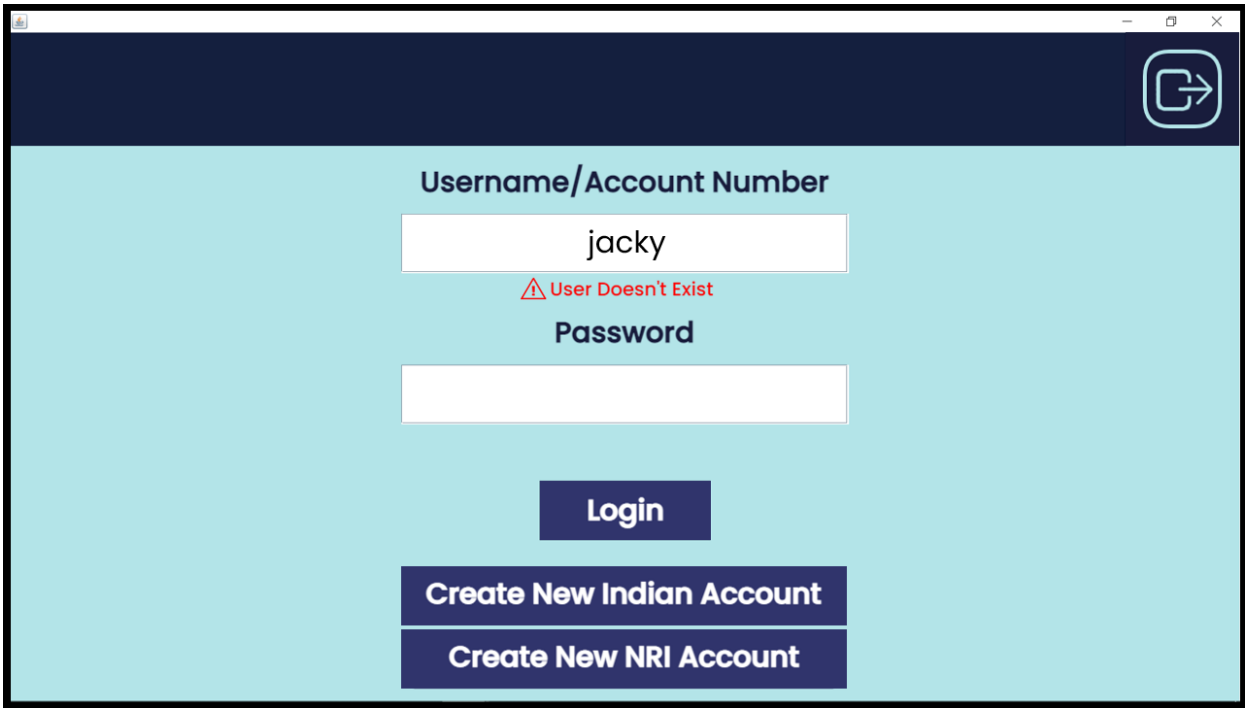
### 3.2.1 BPS Home Screen

Screen created by the TwoButtonGUI class.



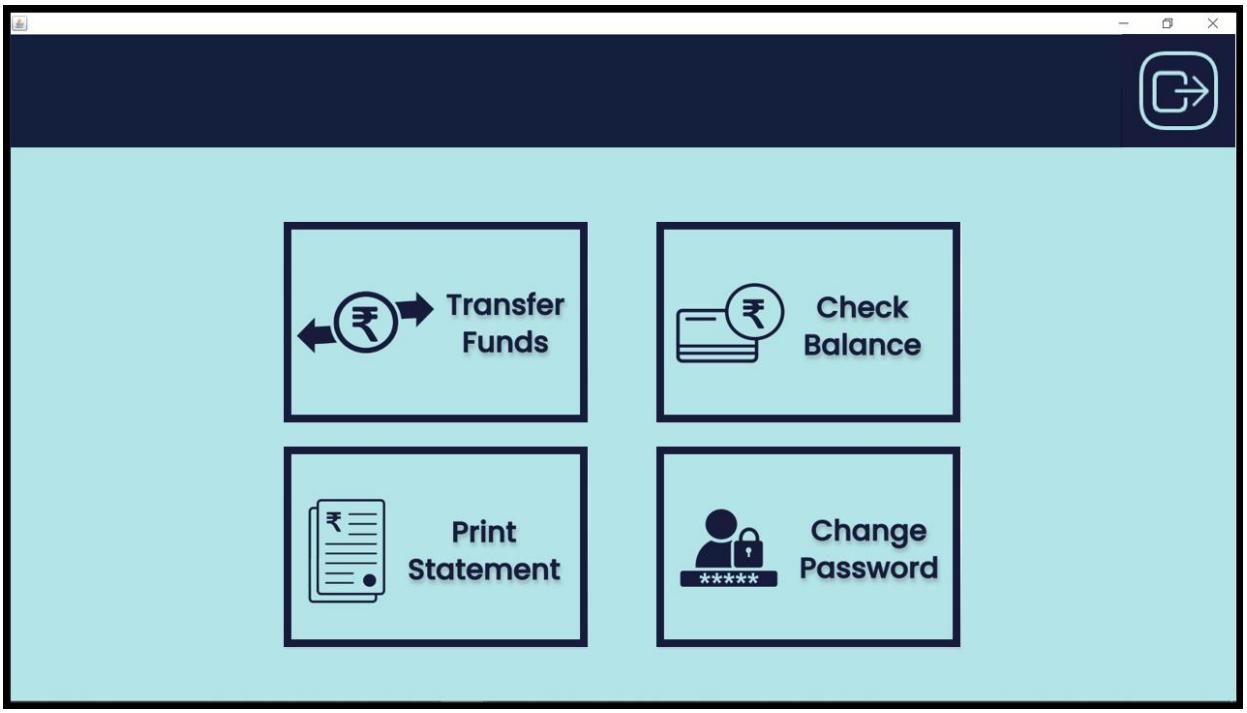

### 3.2.2 Account Holder Login Screen

Screen created by the TwoFieldExtendedGUI class.



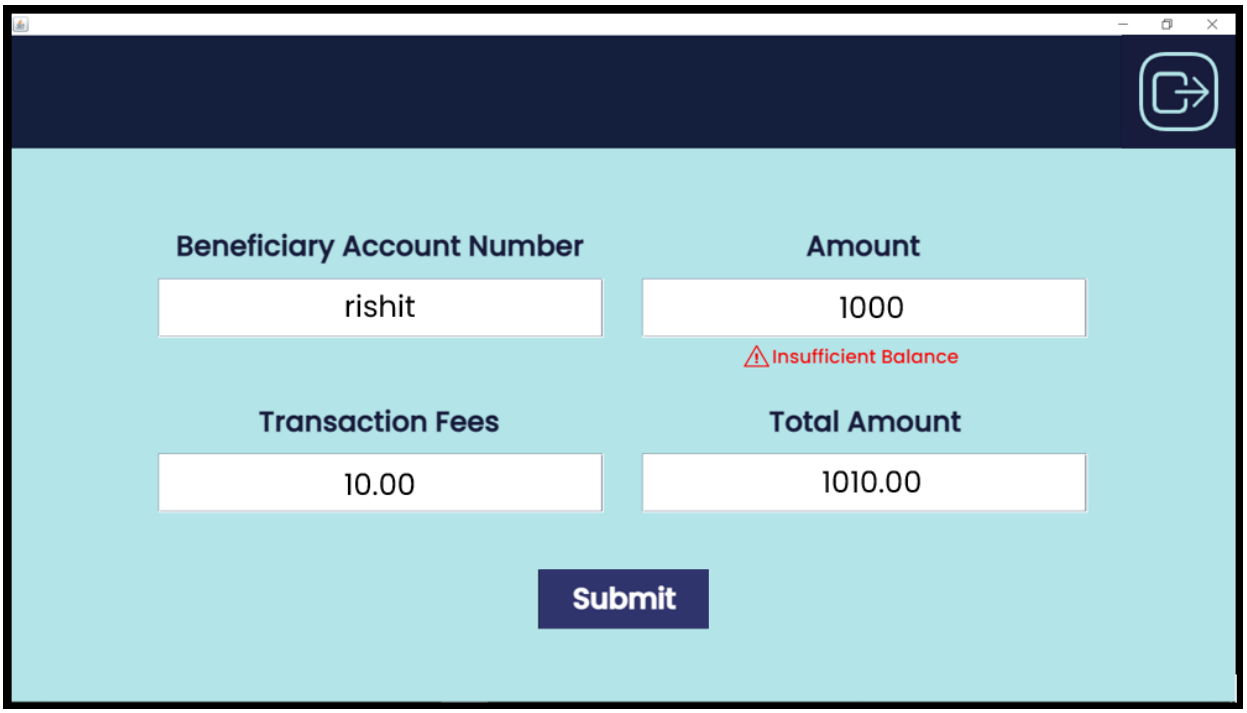
### 3.2.3 Account Holder Dashboard

Screen created by the FourButtonGUI class. Also used for Bank Manager Dashboard and View Accounts functionality.



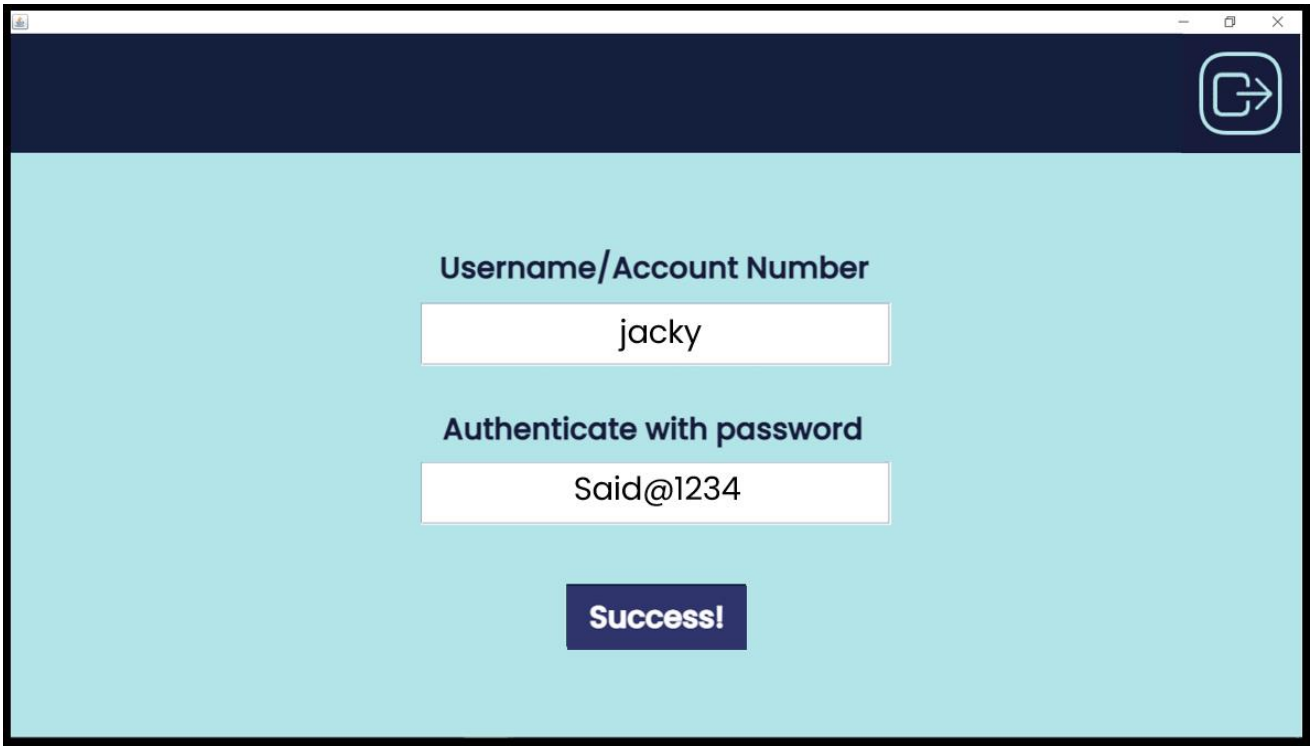

### 3.2.4 Transfer Funds Screen

Screen created by the FourFieldBaseGUI class. Also used for Change password and Create new account functionalities.



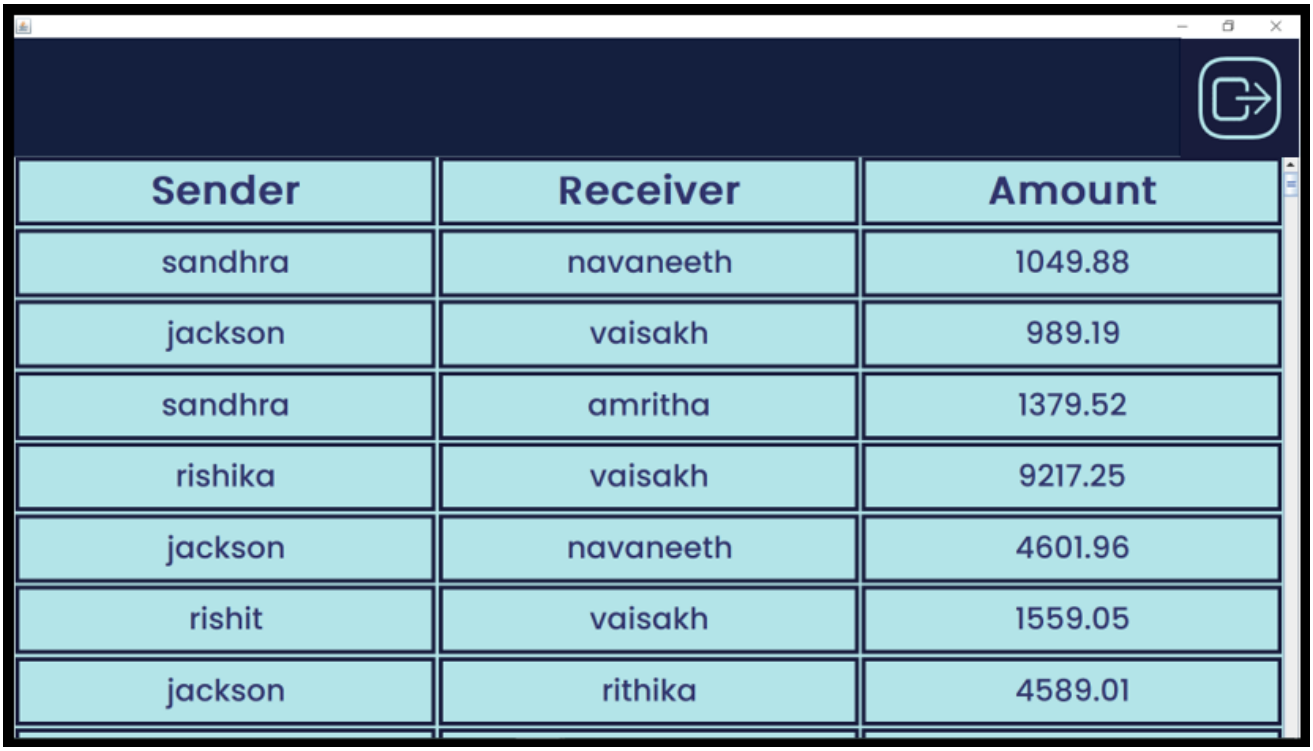
### 3.2.5 Change Account Type

Screen created by the TwoFieldBaseGUI class. Also used in Unblock account functionality.




### 3.2.6 View Transactions

Screen created by the DisplayGUI class. Also used in View Accounts and Print Statement functionalities.



| Sender | Receiver | Amount |
|---|---|---|
| sandhra | navaneeth | 1049.88 |
| jackson | vaisakh | 989.19 |
| sandhra | amritha | 1379.52 |
| rishika | vaisakh | 9217.25 |
| jackson | navaneeth | 4601.96 |
| rishit | vaisakh | 1559.05 |
| jackson | rithika | 4589.01 |

# 4. Test Cases

## 4.1 Test Case #1 (TC_LOGIN_01)

**Author:** Vaisakh Ramachandran

**Test Case Description:**

Test scenario: Verify AccountHolder login.
Test case: Entering an invalid set of login credentials and then a valid one.

**Pre-Conditions:**

Needs to be an Account registered with the bank.

**Test Steps:**

Select AccountHolder login.
Enter incorrect username.
Enter correct password.
Click login button.
Enter correct username.
Enter incorrect password.
Click login button.
Enter correct username.
Enter correct password.
Click login button.

**Test Data:**

Incorrect username: vaisakh1
Correct username: Vaisakh
Incorrect password: Vais@123
Correct password: Vais@1234

**Expected Result:**

Invalid username error message appears.
Incorrect password error message appears.
Successful login

**Post Condition:**

AccountHolder dashboard opens.

## 4.2 Test Case #2 (TC_LOGIN_02)

**Author:** Vaisakh Ramachandran

**Test Case Description:**

Test scenario: Verify AccountHolder login.
Test case: Entering an invalid set of login credentials.

**Pre-Conditions:**

Needs to be an Account registered with the bank.

**Test Steps:**

Select AccountHolder login.
Enter incorrect password five times.
Enter correct password.
Click login button.

**Test Data:**

Correct username: Vaisakh
Incorrect password: Vais@123
Correct password: Vais@1234

**Expected Result:**

“Account blocked, contact manager” error message appears.

**Post Condition:**

The account cannot be logged into.

## 4.3 Test Case #3 (TC_CHECKBAL_01)

**Author:** Vaisakh Ramachandran

**Test Case Description:**

Test scenario: View account balance.
Test case: Logging in and selecting “check balance”

**Pre-Conditions:**

Needs to be an Account registered with the bank.

**Test Steps:**

Select AccountHolder login.
Enter correct username and password.
Click login button.
Click “Check Balance”.

**Test Data:**

Correct username: Amritha
Correct password: Amri@1234

**Expected Result:**

Account balance is displayed for five seconds on the same box and then disappears to show “check balance” option on the same box.

**Post Condition:**

The balance is displayed.

## 4.4 Test Case #4 (TC_FUNDTRANSFER_01)

**Author:** Vaisakh Ramachandran

**Test Case Description:**

Test scenario: Transfer funds to another user.
Test case: Logging in and selecting “transfer funds” and entering details required.

**Pre-Conditions:**

Needs to be an Account registered with the bank.

**Test Steps:**

Select AccountHolder login.
Enter correct username and password.
Click login button.
Click “Transfer Funds”.
Enter incorrect beneficiary account number.
Enter amount to transfer.
Click submit.
Enter correct beneficiary account number.
Enter amount.
Click submit.

**Test Data:**

Correct username: Vaisakh
Correct password: Vais@1234
Incorrect Beneficiary Account Number: rishit123
Amount: 2000.00
Correct Beneficiary Account Number: Rishit

**Expected Result:**

"User doesn't exist" error message is displayed on clicking "Submit" the first time. "Insufficient funds" error message is displayed on clicking "Submit" the second time.

**Post Condition:**

Fund can not be transferred.

## 4.5 Test Case #5 (TC_FUNDTRANSFER_02)

**Author:** Vaisakh Ramachandran

**Test Case Description:**

Test scenario: Transfer funds.

Test case: Logging in, checking balance, transferring funds and checking balance again.

**Pre-Conditions:**

Needs to be an Account registered with the bank.

**Test Steps:**

Select AccountHolder login.
Enter correct username and password.
Click login button.
Click "Check Balance".
Click "Transfer Funds".
Enter beneficiary account number.
Enter amount to transfer.
Click 'Submit'.
Click 'Confirm'.
Click 'Check Balance'.

**Test Data:**

Correct username: Aswin

Correct password: Aswi@1234
Beneficiary Account Number: Rishit
Amount: 2000.00

**Expected Result:**

The account balance of the user has been reduced by the total amount transferred (inclusive of transaction fee).

**Post Condition:**

Funds transferred.

## 4.6 Test Case #6 (TC_PRINTSTATEMENT_01)

**Author:** Rishit Kumar Chordia

**Test Case Description:**

Test scenario: Print account statement.
Test case: Logging in and selecting “Print Statement”.

**Pre-Conditions:**

Needs to be an Account registered with the bank.

**Test Steps:**

Select AccountHolder login.
Enter correct username and password.
Click login button.
Click “Print Statement”.

**Test Data:**

Correct username: Jackson
Correct password: Jack@1234

**Expected Result:**

The user’s account statement pops up on the screen.

**Post Condition:**

Account statement printed.

## 4.7 Test Case #7 (TC_CHANGEACCPWD_01)

**Author:** Rishit Kumar Chordia

**Test Case Description:**

Test scenario: Change account password.
Test case: Logging in, selecting “Change Password” and entering required details.

**Pre-Conditions:**

Needs to be an Account registered with the bank.

**Test Steps:**

Select AccountHolder login.
Enter correct username and password.
Click login button.
Click “Change Password”.
Enter incorrect username.
Enter current password.
Enter new password.
Re-enter new password.
Click ‘Submit’.
Enter correct username.
Enter incorrect current password.
Enter new password.
Re-enter new password.
Click ’Submit’.
Enter correct username.
Enter correct current password.
Enter new password as current password.
Re-enter new password as current password.
Click ‘Submit’.
Enter correct username.
Enter correct current password.
Enter new password.
Re-enter new password incorrectly.
Click ‘Submit’.
Enter correct username.
Enter correct current password.
Enter weak new password.
Re-enter new weak password correctly.
Click ‘Submit’.
Enter correct username.
Enter correct current password.
Enter new password.

Re-enter new password correctly.
Click ‘Submit’.
Logout.
Login with same username.
Login with new password.

**Test Data:**

Correct username: Jackson
Current password: Jack@1234
Incorrect username: jacky
Incorrect password: jacky@1234
New weak password: abc
New password: Jackey@1234

**Expected Result:**

“Incorrect username” error message is shown upon clicking ‘Submit’ the first time. Subsequent clicks of ‘Submit’ button results in “Incorrect password” error message, an error message indicating the use of a weak password and “Passwords don’t match” error messages, before finally successfully changing the password. A login attempt with the new password will be successful.

**Post Condition:**

Account password successfully changed.

## 4.8 Test Case #8 (TC_VIEWTRANSACTIONS_01)

**Author:** Rishit Kumar Chordia

**Test Case Description:**

Test scenario: View transactions performed by the bank.
Test case: Logging in as manager and selecting “View Transactions”.

**Pre-Conditions:**

Needs to be an Account registered with the bank.

**Test Steps:**

Select BankManager login.
Enter incorrect username
Enter correct password.
Click login button.
Enter correct username.

Enter incorrect password.
Click login button.
Enter correct username.
Enter correct password.
Click login button.
Click “View Transactions”.

**Test Data:**

Incorrect username: Saidalavi12
Incorrect password: said@1234
Correct username: Saidalavi
Correct password: Said@1234

**Expected Result:**

“Incorrect username” error will be displayed on clicking login button first, then it displays an “incorrect password” error and finally successfully logs you in, after which clicking the “View Transactions” button displays all transactions.

**Post Condition:**

Transactions are displayed.

## 4.9 Test Case #9 (TC_VIEWACCOUNTS _01)

**Author:** Rishit Kumar Chordia

**Test Case Description:**

Test scenario: View accounts registered with the bank.
Test case: Logging in as manager and selecting “View Accounts”.

**Pre-Conditions:**

Needs to be an Account registered with the bank.

**Test Steps:**

Select BankManager login.
Enter username and password.
Click login button.
Click ‘View Accounts’.
Click ‘View All Accounts’.
Go to dashboard.
Click ‘View All NRI Accounts’.
Go to dashboard.

Click ‘View all Indian Accounts’.
Go to dashboard.
Click ‘View Blocked Accounts’.

**Test Data:**
Username: Saidalavi
Password: Said@1234

**Expected Result:**
All four types of account will be displayed.

**Post Condition:**
Accounts are displayed as per the option chosen.

## 4.10 Test Case #10 (TC_UNBLOCKACCOUNT _01)

**Author:** Rishit Kumar Chordia

**Test Case Description:**
Test scenario: Unblock the account of an account holder.
Test case: Logging in as manager and selecting “Unblock Account”.

**Pre-Conditions:**
Needs to be an Account registered with the bank.

**Test Steps:**
Select BankManager login.
Enter username and password.
Click login button.
Click “Unblock Account”.
Enter username of a blocked account incorrectly.
Enter password of manager.
Click “Confirm”.
Enter username of a blocked account correctly.
Enter password of manager incorrectly.
Click “Confirm”.
Enter username of a blocked account correctly.
Enter password of manager correctly.
Click “Confirm”.

**Test Data:**
Username of manager: Saidalavi

Password of manager: Said@1234
Blocked account username: Rishika

**Expected Result:**

An incorrect username error appears on clicking the confirm button the first time. Clicking it a second time results in an incorrect password error, after which it successfully works.

**Post Condition:**

The account is unblocked.

# References

Lucid Software. (2018, August 27). How to Make a UML Sequence Diagram [Video]. YouTube. https://www.youtube.com/watch?v=pCK6prSq8aw&ab_channel=LucidSoftware

Udacity. (2015, February 24). Object Diagram [Video]. YouTube. https://www.youtube.com/watch?v=3g-pWGssaZ8&ab_channel=Udacity

(2018). What is Object Diagram? Visual Paradigm. https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-object-diagram/