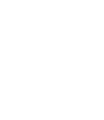


***Software Requirement and Design***

***Specifications***

***ATM***



***Version: [2.0]***

|  |  |
| --- | --- |
| *Course Code* | SE-2002 |
| *Instructor* | Syeda Rubab Jaffar |
| *Project Team* | Zayn Khan (21k-3828) |
| *Submission Date* | 5/11/23 |



Table of Contents

[1. Introduction 5](#_Toc133770696)

[1.1. Purpose of Document 5](#_Toc133770697)

[1.2. Intended Audience 5](#_Toc133770698)

[1.3 Definition of Terms, Acronyms and Abbreviations 5](#_Toc133770699)

[1.4 Document Convention 7](#_Toc133770700)

[2. Overall System Description 8](#_Toc133770701)

[2.1. Project Background 8](#_Toc133770702)

[2.2. Project Scope: 8](#_Toc133770703)

[2.3. Not In Scope 9](#_Toc133770704)

[2.4. Project Objectives 9](#_Toc133770705)

[2.5. Stakeholders 10](#_Toc133770706)

[2.6. Operating Environment 10](#_Toc133770707)

[2.7. Constraints 10](#_Toc133770708)

[2.8. Assumptions & Dependencies 11](#_Toc133770709)

[3. External Interface Requirements 12](#_Toc133770710)

[3.1. Hardware Interfaces 12](#_Toc133770711)

[3.2. Software Interfaces 13](#_Toc133770712)

[3.3. Communications Interfaces 14](#_Toc133770713)

[4. Functional Requirements 14](#_Toc133770714)

[4.1. Functional Hierarchy 14](#_Toc133770715)

[4.2. Use Cases 18](#_Toc133770716)

[4.2.1. Login 18](#_Toc133770717)

[4.2.2. Withdraw Cash 19](#_Toc133770718)

[*4.2.3.* BILL PAYMENT 21](#_Toc133770719)

[4.2.4. Deposit Cash 22](#_Toc133770720)

[4.2.5. Transfer Funds 23](#_Toc133770721)

[4.2.6. Balance Inquiry 25](#_Toc133770722)

[4.2.7. Change Pin 26](#_Toc133770723)

[4.2.8. Mini Statement 28](#_Toc133770724)

[*4.2.9. Check Balance* 29](#_Toc133770725)

[4.2.10. Verify account 31](#_Toc133770726)

[5. Non-functional Requirements 33](#_Toc133770727)

[5.1. Performance Requirements 33](#_Toc133770728)

[5.2. Safety Requirements 33](#_Toc133770729)

[5.3. Security Requirements 33](#_Toc133770730)

[5.4. User Documentation 34](#_Toc133770731)

[6. System Architecture 36](#_Toc133770732)

[6.1. System Level Architecture 36](#_Toc133770733)

[6.2. Software Level Architecture: 37](#_Toc133770734)

[7. Detailed System Design 38](#_Toc133770735)

[7.1.Class Diagram: 38](#_Toc133770736)

[7.1.1Class Diagram Description: 38](#_Toc133770737)

[7.2. Detailed GUI 40](#_Toc133770738)

[8. Application Design 57](#_Toc133770739)

[8.1 Collaboration Diagram 57](#_Toc133770740)

[8.2 Sequence Diagram 58](#_Toc133770741)

[8.2.1 Login 58](#_Toc133770742)

[8.2.2 Account Type 59](#_Toc133770743)

[8.2.3 CashWithdraw 60](#_Toc133770744)

[8.2.4 Cash Deposit 61](#_Toc133770745)

[8.2.5 Funds Transfer 62](#_Toc133770746)

[8.2.6 Pin change 63](#_Toc133770747)

[8.2.7 Electricity Bill paid 64](#_Toc133770748)

[8.3. State Transition Diagram 65](#_Toc133770749)

[8.4. Activity Diagram 66](#_Toc133770750)

[8.4.1 Login 66](#_Toc133770751)

[8.4.2 Account type 67](#_Toc133770752)

[8.4.3 Cash Withdraw 68](#_Toc133770753)

[8.4.4 Cash Deposit 69](#_Toc133770754)

[8.4.5 Funds Transfer 69](#_Toc133770755)

[8.4.6 Pin Change 70](#_Toc133770756)

[8.4.7 Electricity Bill Paid 71](#_Toc133770757)

[9.References 72](#_Toc133770758)

[9.1. Websites 72](#_Toc133770759)

[9.2 Books 72](#_Toc133770760)

[10. Appendices 72](#_Toc133770761)

[10.1. Appendix A: Glossary 72](#_Toc133770762)

# 1. Introduction

## 1.1. Purpose of Document

Automated Teller machines are a crucial part of people’s daily lives. It provides quick access to cash withdrawals and balance inquiries. Moreover, they make life easier by reducing human interaction, as the user is only interacting with the machine.

The SRS covers the whole system, all the functionality, requirements, constraints, etc. Every feature will be discussed in detail in the following sections. The SRS will be discussing the external interface and all the functional requirements along with functionality.

This SRS will create a framework for the development team and will help them clearly define what they need to develop. It will also lay groundwork not only for the development team but other teams such as quality assurance, operations, maintenance, etc.This will also act as a document for reference and/or proof.

## 1.2. Intended Audience

There is a specific audience that should read this SRS, the developer (who would build the whole system), the tester (who would test the system through test cases and pre-developed puts of that system), the system owner (who checks the system if its sound against the requirements that has been elicited). On the other hand, the marketing team or the advertising team will not read the whole SRS; they would skim the idea of the product, features, quality, and what’s new in this software and thus market the software such that it hits the target audience. Lastly, the designer would read the entire document line by line and would make sure that he doesn’t miss one single bit of detail since he must design accordingly.

The following document should be read in a simple way heading to heading in a linear flow.

## 1.3 Definition of Terms, Acronyms and Abbreviations

|  |  |
| --- | --- |
| ***Term*** | ***Description*** |
| **KID** | Key Information Document |
| **BA** | Business analyst |
| **OTP** | One Time Password |
| **PIN** | Personal Identification Number |
| **ATM** | Automated Teller Machine |
| **JDK** | Java Development Kit |
| **FTP** | File Transfer Protocol |
| **GSM** | Global System for Mobile communications |

## 1.4 Document Convention

Body Size:

Font: Arial

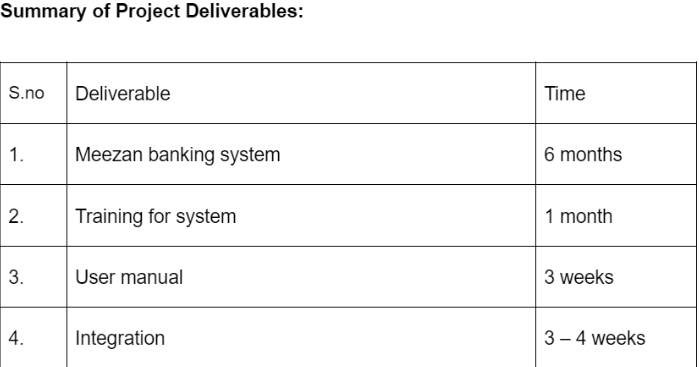
Size: 12

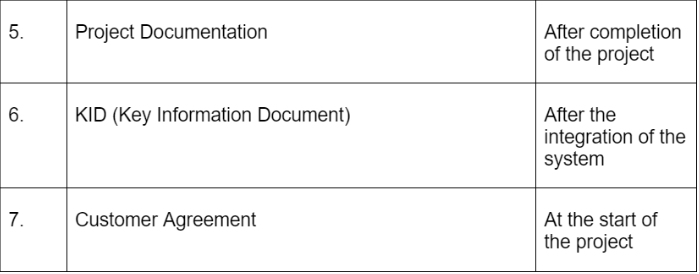
# 2. Overall System Description

## 2.1. Project Background

As we all know, when it comes to paying electricity bills, things get very disorganized. People are in long queues waiting for their turns, sometimes fights to break out, and sometimes people return without their payment being made as the time slot gets finished before their turn, and the whole process repeats. The idea of paying electricity bills through an ATM allows clients to come in at any time of day and pay their bills. This ease of access will lead to shorter queues for bill payment without much hassle and fuss. The clients will not miss the deadline for bill payments and will save money once they have paid for the overdue date. Additional to this, sometimes users don’t have access to internet and they want to transfer funds to their loved ones. The system will allow them to transfer funds without the use of internet.

## 2.2. Project Scope:





**Statement of Project’s success criteria:**

* 50% reduction in queue lengths for electricity via bank within the first 5 months.
* 25% cost reduction by lesser employees within 1 year.
* 30% of the current crop of clients (paying manually via banks) use our system within the first 3 months.
* At least 50% 4 stars and above reviews of our system (via feedback) within the first 6 months.

## 2.3. Not In Scope

**Project Exclusions:**

* UI for users (only UX).
* Free of cost future updates in the system as per hardware changes.
* Dealing with unauthorized actions, such as blocking another person using someone else’s card for cash withdrawal (will be done by the bank staff themselves if the client reports/requests for lost card/card blocking, etc.).

## 2.4. Project Objectives

The product is developed on the demand of a Meezan bank which wants a new ATM interface for their ATM outlets. It’s a replacement for the previous ATM interface. The bank wants to introduce new features to the ATMs like cash deposits and bill payments which could revolutionize the ATM market. Another reason to replace the old system is, earlier the average user would take about 4 to 7 minutes to withdraw cash, as the users were having difficulty using the previous interface. The increasing number of complaints from the users convinced the management of the bank to design a new and user-friendly interface.

Furthermore system increases the number of options users have to transfer funds to other accounts. They will have the option to transfer funds through bank, app, and atm. This solves the problem of users who sometimes don’t have access to internet and cannot transfer funds through web app and don’t want to wait in long queues at banks.

## 2.5. Stakeholders

1. **Atm User:** external.
2. **Project Manager:** internal.
3. **Project team members:** internal.
4. **Project sponsors:** external.
5. **Resource manager:** internal.
6. **Government agencies:** external.
7. **Bank Regulator:** external.
8. **Bank Manager:** external.
9. **Subject matter experts:** internal.
10. **Documentation Writers:** internal.
11. **Electricity department(bank):** external.
12. **Security Department(bank):** external
13. **Safety Department(bank):** external.
14. **CEO of Bank:** external.
15. **Development team:** internal.
16. **Maintenance team:** internal.
17. **Design team:** internal.

## 2.6. Operating Environment

The software will be used in all ATMs throughout the country. The ATMs will be in a vestibule or a room. It will have a card reader to scan the card, a keypad to enter the pin, a safe to store the money, a printer to print the receipt, a Cash dispenser, and a screen to prompt the users. To update customer transactions and save user information in the bank's database, the ATM system will work in tandem with the bank's system. The software will work on white-label ATMs and dial-up ATMs.

## 2.7. Constraints

* The system is coded in Java.
* Deadline for completion (4 months).
* The ATM must service at most one person at a time.
* Users can only use the cards of banks acceptable by the system.
* The number of invalid pin entries attempted must not exceed three. After three unsuccessful login attempts, the card is seized/blocked and needs to be replaced by the bank.
* Simultaneous access to an account through both the ATM and the bank is not supported.
* The minimum amount of money a user can withdraw is Rs 500/- and the maximum amount of money a user can withdraw in a session is Rs.15,000/- and the maximum amount he can withdraw in a day is different for different cards.

1. For silver, Rs.15000/-.
2. For gold, Rs.25000/-.
3. For platinum, Rs.50000/-.

* Before the transaction is carried out, a check is performed by the machine to ensure that a minimum amount of Rs 1000/- is left in the user’s account after the withdrawal. Withdrawal will be rejected if the amount left in the account is less than Rs 1000/-.
* The minimum amount of money a user can deposit is Rs 500/- and the maximum amount he can withdraw in a day is different for different cards.

1. For silver, Rs.10000/-.
2. For gold, Rs.15000/-.
3. For platinum, Rs.25000/-.

* The software requires a minimum memory of 20GB.
* The database used should be Oracle7.0.
* ATM has limited storage of currency notes.

## 2.8. Assumptions & Dependencies

 **Assumptions:**

* Every client knows how to read and understand Urdu and/or the English language/s.
* The ATM will be functional 24/7, only non-functional when the bank management refills it with cash.
* Every user using the ATM has a valid ATM card.
* Every ATM has the required hardware mentioned in the SRS.

**Dependencies:**

* Bank management fills ATMs with cash for them to be functional.
* The ATM should have an electricity supply 24/7 as it would not be operational/functional without it.
* Security in front of ATMs for safe transactions, otherwise no one will use it.

# 3. External Interface Requirements

## 3.1. Hardware Interfaces

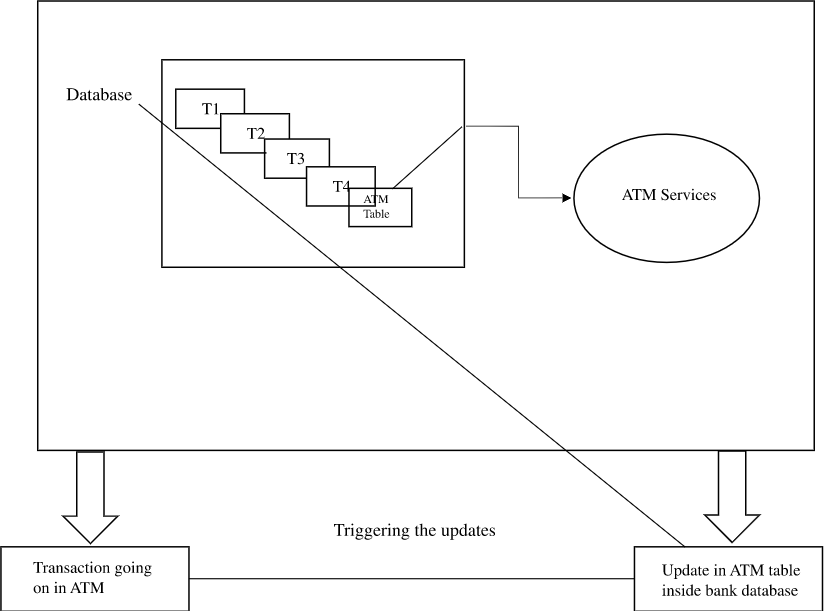
1. There must be an ATM card that should fulfill the requirement of the physical dimension.
   1. The width of an ATM card must be 85.47mm to 85.72mm.
   2. The height of the ATM card must be 53.92mm to 54.03mm.
   3. The thickness of the ATM card must be 0.76mm +-0.08mm.
2. The card reader should be a magnetic stripe reader.
3. The card reader shall have a Smart card option.
4. There shall be a 40-column dot matrix receipt printer.
5. The envelope depository shall be a maximum of 4.5" in width, 10" in length and

0.5" thickness.

1. Screen resolution of at least 800X600 is required for proper and complete viewing of screens. Higher resolution would not be a problem.
2. Two Serial ports, 1st for running communication headache and 2nd for backup. For source code configuration
3. Two Universal Serial Bus for data sharing.
4. The ATM Machine can count the currency notes.
5. The ATM Machine can have a touch screen for convenience.
6. The ATM Machine must connect to the bank’s network.
7. Numpad should be available for the user to enter the pin and perform actions.
8. The ATM should be 24/7 and have an internet connection to easily connect to the database.
9. A backup internet connection should be available if the main internet connection goes out of the connection.
10. The backup power supply should be connected to the system in case the main supply is cut off the machine should be working uninterruptedly.
11. The software is developed for Windows 32-bit or 64-bit.

## 3.2. Software Interfaces

1. The database used to keep records of user accounts shall be Oracle7.0.
2. The software is developed for windows 7 and onwards.
3. The software is developed using the JAVA programming language thus it requires a JDK driver to run.
4. The system should have a JDBC driver to have a connection with the database.
5. Java swings library is used for GUI purposes.



## 3.3. Communications Interfaces

The machine needs to communicate with the main branch for each session for various functions such as login verification, account access, etc. so the following are the various communication interface requirements that are needed to be fulfilled to run the software successfully: -

1. Protocols used for data transfer shall be File Transfer Protocol (FTP) 3 and HTTPS protocol.
2. The application can be run on any JAVA compiler such as IntelliJ, NetBeans, and Eclipse.
3. The JDK version should be JDK 17 or onwards for indexing purposes.
4. The ATM card activation will be acknowledged via email to the customer.
5. GSM is used for ATM message transmission.
6. Communication security is recommended strongly. Confidential information is guaranteed to be secure.
7. If the card or pin is invalid, an invalid message will take place.
8. We can choose the transaction method when the pin is correct.
9. Whenever a transaction is done it is contacted by the user through message and email.
10. End-to-end encryption should be done while entering the password to ensure the user account is safe.

# 4. Functional Requirements

## 4.1. Functional Hierarchy

**Functional requirement:**

* If no ATM card is in the ATM, the system should display an initial display. **FR1**

1. **Functional requirements:**

* If the ATM is running out of money, no card should be accepted. An error message is

displayed. **FR2.1**

* If the amount of cash is less than the requested amount, Display an error message. Return ATM card. **FR2.2**
* The ATM closes the session and waits for another user when done. **FR2.3**

1. **Functional requirements:**

* The ATM must check if the entered ATM card is valid. **FR3.1**
* It will be valid if,

1. the information on the card can be read and matched from the database.

2. it is not expired.

* Display an error message and return the ATM card if it is invalid.**FR3.2**

1. **Functional requirements:**

* If the ATM card is valid, the ATM should read the serial number and bank code and initiate the authorization dialog. **FR4**

1. **Functional requirements:**

* The serial number from the ATM card should be logged. **FR5**

1. **Functional requirements:**

* If the card is accepted print a message asking the user is requested to enter his password. **FR6.1**
* The ATM verifies the bank code and password with the bank computer. **FR6.2**
* Display the response received accordingly. **FR6.3**

1. **Functional requirements:**

* If a card was entered more than three times in a row at any ATM and the password

was wrong each time, the card is blocked inside the ATM. **FR7.1**

1. **Functional requirements:**

* If the card is accepted and the user selects the withdrawal option, print a message asking the user to input the amount. **FR8.1**
* The amount entered for withdrawal is compared with the amount in the customer’s account and the cash in the ATM. **FR8.2**
* Accept/Reject the request according to the response received and display the message. **FR8.3**
* If the request is rejected, ask the customer if he wants another transaction. **FR8.3.1**
* If yes, repeat the withdrawal process. **FR8.3.1.1**
* Return the card and display the welcome screen. **FR8.3.1.2**

1. **Functional requirements:**

* If the amount is accepted:
* If the amount is greater than the amount allowed in one session, reject the transaction. **FR9.1**
* Print an error message asking to input the amount less than Rs15000/-. **FR9.1.1**
* If the amount is greater than the amount allowed in one day, reject the transaction. **FR9.2**
* Print an error message asking to input the amount less than Rs15000/- if the user is a silver card holder. **FR9.2.1**
* If the amount is greater than the amount allowed in one day, reject the transaction. **FR9.3**
* Print an error message asking to input the amount less than Rs25000/-. if the user is a gold card holder. **FR9.3.1**
* If the amount is greater than the amount allowed in one day, reject the transaction. **FR9.4**
* Print an error message asking to input the amount less than Rs50000/-. if the user is a platinum card holder. **FR9.4.1**

1. **Functional requirements:**

* Accept the withdrawal amount and initiate the withdrawal sequence. **FR10.1**
* Print the message that the transaction was successful. **FR10.1.1**

1. **Functional requirements:**

* If the user selects the deposit option, print a message to ask to enter the amount user wants to deposit. **FR11.1**
* If the amount is greater than the amount allowed in one session, reject the deposit. **FR11.2**
* Print an error message asking to input the amount less than Rs10000/-. **FR11.2.1**

**(NOTE: Only deposits through ATMs have a limit because ATMs have limited storage, this rule will not apply to deposits through banks).**

* If the amount is greater than the amount allowed in one day, reject the deposit. **FR11.3**
* Print an error message asking to input the amount less than Rs10000/- if the user is a silver card holder. **FR11.3.1**
* If the amount is greater than the amount allowed in one session, reject the deposit. **FR11.4**
* Print an error message asking to input the amount less than Rs15000/- if the user is a gold card holder. **FR11.4.1**
* If the amount is greater than the amount allowed in one session, reject the deposit. **FR11.5**
* Print an error message asking to input the amount less than Rs25000/- if the user is a platinum card holder. **FR11.5.1**

1. **Functional requirements:**

* If the user selects the balance inquiry option, print the balance of the account. **FR12.1**

1. **Functional requirements:**

* If the user selects the loan information option, print the balance of the account. **FR13.1**

1. **Functional requirements:**

* If the user selects the help option, print the help manual. **FR14.1**

1. **Functional requirements:**

* If the user selects the PIN change, display a message asking the user to enter his current PIN for confirmation and display the message forgot PIN. **FR15.1**
* If the user enters the current PIN, match the PIN from the database. **FR15.2**
* If the PIN matches, display the message asking the user to enter the new PIN. **FR15.2.1**
* Else, display the message that the PIN does not match please renter. **FR15.2.2**
* If the user enters the wrong PIN 3 times in a row, reject the PIN change request. **FR15.3**
* Else, display that the PIN change request is successful. **FR15.4**
* If the user selects Forgot PIN, sent an OTP to the user’s number and display a message asking the user to enter the OTP. **FR15.5**
* Match the OTP, if matches, repeat **FR15.2.1** and **FR15.4**

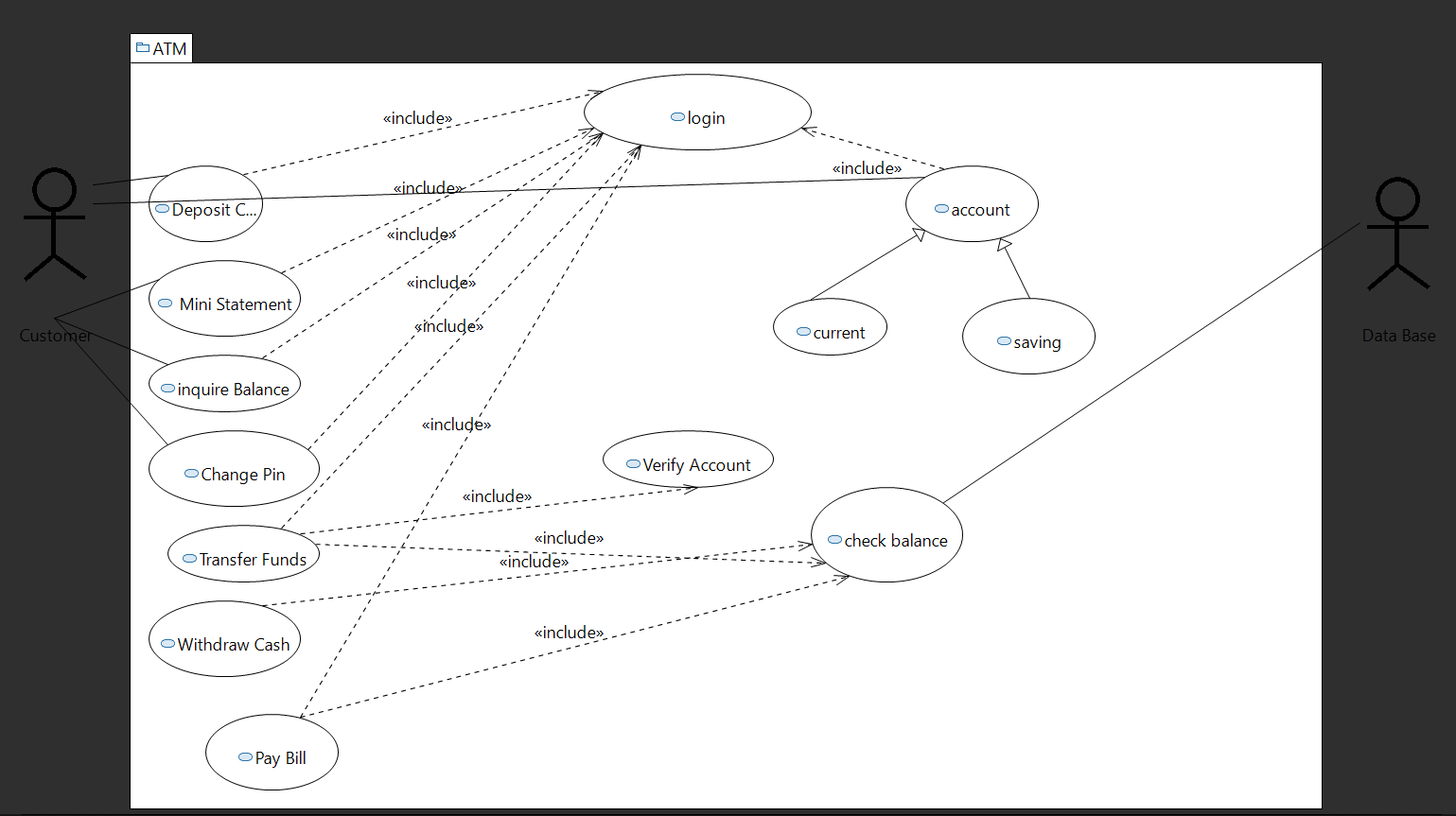
1. **Functional requirements:**

* If the user selects the bill payment option, display a message asking the user to enter the consumer number. **FR16.1**
* Retrieve the information about the bill from the database. **FR16.2**
* Deduct the amount from the account of the user. **FR16.3**
* Display a message, bill paid. **FR16.3.1**
* If the consumer number is not in the database, reject the consumer number. **FR16.4**
* Display a message, wrong input please renter. **FR16.4.1**
* Repeat **FR16.1, FR16.2, FR16.3** and **FR16.3.1.**

1. **Functional requirements:**

* If the user selects the mini statement, print the last 10 transactions from the user’s account. **FR17.1**

## 4.2. Use Cases



### 4.2.1. Login

|  |  |
| --- | --- |
| ***Use Case Description*** | |
| ***Use Case name:*** | Login to ATM |
| ***Use Case Description:*** This use case defines a user entering their ATM card for transactions or bill payments. The user’s ATM card and PIN code are validated. Once the user is verified, they are proceeded to the transaction screen/menu. If a user is not verified, they are prompted to re-enter their PIN code. 3 consecutive erroneous PIN code entries will result in the blocking of ATM accounts | |
| ***Primary actor:*** User | ***Other actors: ATM*** |
| ***Stakeholders: Management, user*** |  |
| ***Relationships***  ***Includes: NONE***  ***Extends: NONE*** | |
| ***Pre-conditions:*** The user must have an ATM card | |
|  | |
| ***Flow of Events:***  - The user enters their ATM card  - The ATM verifies the card  **-**The user is prompted to enter the PIN code  - The user enters the PIN code  **-**The ATM verifies the PIN code | |
| ***Alternative and exceptional flows:***  The user has entered an invalid card, the user is prompted to enter a valid card and the bank is informed about this activity.  The user enters an invalid PIN and the user is prompted to enter the correct PIN.  The user enters the wrong PIN code again. They are then prompted to re-enter the PIN  code. The user enters the wrong PIN code again. The ATM card is blocked and the bank is informed to investigate and resolve this issue. | |
| ***Post-conditions:*** The login activity is recorded and added to the user record. | |

### 4.2.2. Withdraw Cash

|  |  |
| --- | --- |
| ***Use Case Description*** | |
| ***Use Case name:*** | Cash withdrawal |
| ***Use Case Description:***  This use case defines the user as selecting the cash withdrawal option from the transaction menu/screen. After doing so the user is prompted to enter the amount they want to withdraw. The entered amount is verified whether it is a multiple of 500 and is less than the current balance. If it conforms with the criteria, the user gets the amount. | |
| ***Primary actor:*** User | ***Other actors: ATM*** |
| ***Stakeholders: Management, user*** |  |
| ***Relationships***  ***Includes: Login, check balance***  ***Extends: NONE*** | |
| ***Pre-conditions:*** The user must have an ATM card and select cash withdrawal | |
|  | |
| ***Flow of Events:***  -The user selects the cash withdrawal option  -The ATM asks for the amount  -The user enters the amount  -The amount is validated  **-**The amount is given to the user after the validation | |
| ***Alternative and exceptional flows:***   The user has entered an invalid amount, either it is not multiple of 500 or the amount is greater than or equal to the current balance. | |
| ***Post-conditions:***  The transaction is recorded to show it to the user at the end of the month bank statement. | |

### *4.2.3.* BILL PAYMENT

|  |  |
| --- | --- |
| ***Use Case Description*** | |
| ***Use Case name:*** | Pay Bill |
| ***Use Case Description:*** This use case defines a user entering their ATM card for bill payments. The user’s ATM card and PIN code are validated. Once the user is verified, they are proceeded to the transaction screen/menu. If a user is not verified, they are prompted to re-enter their PIN code. The user is prompted to enter the amount of bill they want to pay and the bill number. After entering the amount the user is asked to enter the money through a receiver sensor. The money is then validated and the user’s account balance is updated. | |
| ***Primary actor:*** User | ***Other actors: ATM*** |
| ***Stakeholders: Management, user, bill company*** |  |
| ***Relationships***  ***Includes: login, check balance***  ***Extends:*** | |
| ***Pre-conditions:*** The user must have an ATM card | |
|  | |
| ***Flow of Events:***  -The user enters their ATM card  -The ATM verifies the card  **-**The user is prompted to enter the PIN code  -The user enters the PIN code  **-**The ATM verifies the PIN code  -User selects Pay bill option from main menu.  -ATM asks user to enter amount of bill and bill number .  -User deposits that amount.  -Amount is deducted from user’s account. | |
| ***Alternative and exceptional flows:***  The user has entered an invalid card, the user is prompted to enter a valid card and the bank is informed about this activity.  The user enters an invalid PIN, and the user is prompted to enter the correct PIN.  The user enters the wrong PIN code again. They are then prompted to re-enter the PIN code. The user enters the wrong PIN code again. The ATM card is blocked and the bank is informed to investigate and resolve this issue.  The amount entered and the money inserted does not match. | |
| ***Post-conditions:*** The transaction is recorded to show it to the user at the end of the month bank statement. | |

### 4.2.4. Deposit Cash

|  |  |
| --- | --- |
| ***Use Case Description*** | |
| ***Use Case name:*** | Deposit cash |
| ***Use Case Description:***  This use case describes the event of depositing cash through an ATM. The user’s credentials are validated. Once the user is verified, The user is prompted to enter the amount of cash they want to deposit. After entering the amount the user is asked to enter the money through a receiver sensor. The money is then validated and the user’s account balance is updated. | |
| ***Primary actor:*** User | ***Other actors: ATM*** |
| ***Stakeholders: Management, user*** |  |
| ***Relationships***  ***Includes: Login***  ***Extends: None*** | |
| ***Pre-conditions:*** The user must have an ATM card  The user must be an account holder of Meezan bank. | |
|  | |
| ***Flow of Events:***  -The user enters their ATM card  -The system verifies the card  -The user is prompted to enter the PIN code  -The user enters the PIN code  **-**The system verifies the PIN code  -The user proceeds to the transaction screen/menu  **-**The user selects deposit cash from the menu **:** The system asks the user to enter the amount they want to deposit.  **-**The user enters the amount and inserts the amount through the receiver sensor.  The system validates the amount and adds the amount to the user's account balance. | |
| ***Alternative and exceptional flows:***  The user has entered an invalid card, the user is prompted to enter a valid card and the bank is informed about this activity.  The user enters an invalid PIN, and the user is prompted to enter the correct PIN.  The user enters the wrong PIN code again. They are then prompted to re-enter the PIN code. The user enters the wrong PIN code again. The ATM card is blocked and the bank is informed to investigate and resolve this issue.  The amount entered and the money inserted does not match. | |
| ***Post-conditions:***  The transaction is recorded to show it to the user at the end of the month bank statement. | |

### 4.2.5. Transfer Funds

|  |  |
| --- | --- |
| ***Use Case Description*** | |
| ***Use Case name:*** | Transfer funds |
| ***Use Case Description:***  This use case describes the event of depositing cash through an ATM. The user’s credentials are validated. Once the user is verified, The user is prompted to enter the amount of cash they want to transfer and the account number of person they want to trasnfer. After entering the amount the user is asked to enter the money through a receiver sensor. The money is then validated and the user’s and receivers account balance is updated. | |
| ***Primary actor:*** User | ***Other actors: ATM*** |
| ***Stakeholders: Management, user, reciever*** |  |
| ***Relationships***  ***Includes: Login, check balance, verify Account***  ***Extends: NONE*** | |
| ***Pre-conditions:*** The user must have an ATM card  The user must be an account holder of Meezan bank. | |
|  | |
| ***Flow of Events:***  -The user enters their ATM card  -The system verifies the card  -The user is prompted to enter the PIN code  -The user enters the PIN code  **-**The system verifies the PIN code  -The user proceeds to the transaction screen/menu  **-**The user selects trasnfer funds from the menu **:** The system asks the user to enter the amount they want to deposit and account number of reciever.  **-**The user enters the amount and inserts the amount through the receiver sensor.  -The system validates the amount and adds the amount to the user's account balance. | |
| ***Alternative and exceptional flows:***   The user has entered an invalid card, the user is prompted to enter a valid card and the bank is informed about this activity.  The user enters an invalid PIN, and the user is prompted to enter the correct PIN.  The user enters the wrong PIN code again. They are then prompted to re-enter the PIN code.  The user enters the wrong PIN code again. The ATM card is blocked and the bank is informed to investigate and resolve this issue.  The amount entered and the money inserted does not match.  The account number of receiver doesn’t exist | |
| ***Post-conditions:*** The transaction is recorded to show it to the user at the end of the month bank statement. | |

### 4.2.6. Balance Inquiry

|  |  |
| --- | --- |
| ***Use Case Description*** | |
| ***Use Case name:*** | View Balance |
| ***Use Case Description:***  This use case describes the event of viewing your current balance.. The user’s credentials are validated. Once the user is verified, he selects view balance option and his balance is shown on the screen. | |
| ***Primary actor:*** User | ***Other actors: ATM*** |
| ***Stakeholders: user*** |  |
| ***Relationships***  ***Includes: Login***  ***Extends: NONE*** | |
| ***Pre-conditions:*** The user must have an ATM card | |
|  | |
| ***Flow of Events:***  -The user enters their ATM card  -The system verifies the card  -The user is prompted to enter the PIN code  -The user enters the PIN code  **-**The system verifies the PIN code  -The user proceeds to the transaction screen/menu  **-**The user selects view balance from the menu.  -The menu is shown. | |
| ***Alternative and exceptional flows:***  -The user has entered an invalid card, the user is prompted to enter a valid card and the bank is informed about this activity.  -The user enters an invalid PIN, and the user is prompted to enter the correct PIN.  -The user enters the wrong PIN code again. They are then prompted to re-enter the PIN code. -The user enters the wrong PIN code again. The ATM card is blocked and the bank is informed to investigate and resolve this issue. | |
| ***Post-conditions:*** Balance is shown, then user exists. | |

### 4.2.7. Change Pin

|  |  |
| --- | --- |
| ***Use Case Description*** | |
| ***Use Case name:*** | Change Pin |
| ***Use Case Description:***  This use case describes the event of changing account pin through an ATM. The user’s credentials are validated. Once the user is verified, the user selects pin change menu. He is asked to enter current pin. Then he is asked to enter new pin, re enter the new pin. The new pin is then updated in the data base. | |
| ***Primary actor:*** User | ***Other actors: ATM*** |
| ***Stakeholders: Management, user*** |  |
| ***Relationships***  ***Includes: Login***  ***Extends: NONE*** | |
| ***Pre-conditions:*** The user must have an ATM card  The user must be an account holder of Meezan bank. | |
|  | |
| ***Flow of Events:***  -The user enters their ATM card  -The system verifies the card  -The user is prompted to enter the PIN code  -The user enters the PIN code  **-**The system verifies the PIN code  -The user proceeds to the transaction screen/menu  **-**The user selects pin change from menu.  -Enters current pin.  -Enters new pin.  -Re enters new pin.  -Pin is updated in database. | |
| ***Alternative and exceptional flows:***  The user has entered an invalid card, the user is prompted to enter a valid card and the bank is informed about this activity.  The user enters an invalid PIN, and the user is prompted to enter the correct PIN.  The user enters the wrong PIN code again. They are then prompted to re-enter the PIN code. The user enters the wrong PIN code again. The ATM card is blocked and the bank is informed to investigate and resolve this issue.  The current pin is not valid.  The new pin is not 4 digit long.  The re entered new pin does not match the previous entered new pin. | |
| ***Post-conditions:*** The pin is updated in database. | |

### 4.2.8. Mini Statement

|  |  |
| --- | --- |
| ***Use Case Description*** | |
| ***Use Case name:*** | Check mini statement |
| ***Use Case Description:***  This use case describes the event of viewing your last ten transactions.. The user’s credentials are validated. Once the user is verified, the user selects mini statement menu. He is shown record of his last ten transactions. | |
| ***Primary actor:*** User | ***Other actors: ATM*** |
| ***Stakeholders: Management, user*** |  |
| ***Relationships***  ***Includes: Login***  ***Extends: NONE*** | |
| ***Pre-conditions:*** The user must have an ATM card  The user must be an account holder of Meezan bank. | |
|  | |
| ***Flow of Events:***  -The user enters their ATM card  -The system verifies the card  -The user is prompted to enter the PIN code  -The user enters the PIN code  **-**The system verifies the PIN code  -The user proceeds to the transaction screen/menu  **-**The user selects mini stataement from menu.  -Last ten transactions are shown. | |
| ***Alternative and exceptional flows:***  The user has entered an invalid card, the user is prompted to enter a valid card and the bank is informed about this activity.  The user enters an invalid PIN, and the user is prompted to enter the correct PIN.  The user enters the wrong PIN code again. They are then prompted to re-enter the PIN code. The user enters the wrong PIN code again. The ATM card is blocked and the bank is informed to investigate and resolve this issue. | |
| ***Post-conditions:*** The user views statement and exists. | |

### *4.2.9. Check Balance*

|  |  |
| --- | --- |
| ***Use Case Description*** | |
| ***Use Case name:*** | ***Check Balance*** |
| ***Use Case Description:***  This use case defines the system checking the amount of balance present in user’s account. If the user selects withdraw, pay bill, transfer or withdraw option, system checks the amount present in user’s account. If it is greater than the amount input by the user, the function is performed by the system. | |
| ***Primary actor:*** data base | ***Other actors: user*** |
| ***Stakeholders: Management, user*** |  |
| ***Relationships***  ***Includes: Login***  ***Extends: NONE*** | |
| ***Pre-conditions:*** The user must have an ATM card and selects either cash withdrawal, pay bill, withdraw, or transfer funds. | |
|  | |
| ***Flow of Events:***  -The user selects one of the option from withdrawal, transfer, pay bill.  -User enters the amount.  -System makes a check of current balance in user’s account.  -Balance is greater than the input amount.  -Function is performed by the system. | |
| ***Alternative and exceptional flows:***   The current balance is less than the amount entered by the user. | |
| ***Post-conditions:***  The transaction is recorded to show it to the user at the end of the month bank statement. | |

### 4.2.10. Verify account

|  |  |
| --- | --- |
| ***Use Case Description*** | |
| ***Use Case name:*** | ***Verify account*** |
| ***Use Case Description:***  This use case defines the system making a check through data base to ensure that the account exists in the data base. System matches the account number entered by the user when transferring funds. If the match exists, user can proceed further with the process. | |
| ***Primary actor:*** system | ***Other actors: user*** |
| ***Stakeholders: Management, user*** |  |
| ***Relationships***  ***Includes: NONE***  ***Extends: NONE*** | |
| ***Pre-conditions:*** The user must have an ATM card and select fund transfer | |
|  | |
| ***Flow of Events:***  -The user selects transfer funds option.  -User enters the account number of reciever.  -System makes a check if the account number exists.  -Account number matches with the input.  -User enters amount to transfer.  -Amount is then transferred to the receiver. | |
| ***Alternative and exceptional flows:***  The account number does not exist in the data base.  User is given error message.  User is asked to input number again. | |
| ***Post-conditions:***  The transaction is recorded to show it to the user at the end of the month bank statement. | |

# 5. Non-functional Requirements

## 5.1. Performance Requirements

PER-1 – Card verification time must not exceed 0.8 seconds under normal server workload and 1 second under peak server load.

PER-2 - The PIN verification time must not exceed 0.3 seconds under normal server workload and 0.5 seconds under peak server workload.

PER-3 - Account balance transfer time must not exceed 3 seconds under normal server workload and 4 seconds under peak server workload.

PER-4 – Cash withdrawal transaction time must not exceed 4 seconds under normal server workload and 5 seconds under peak server workload.

PER-5 – Deposit transaction time after insertion of the deposit envelope must not exceed 5 seconds under normal server workload and 6 seconds under peak server workload.

The PIN verification time must not exceed 0.3 sec. under

normal server workload and 0.5 sec. under peak server workload.

The PIN verification time must not exceed 0.3 sec. under

normal server workload and 0.5 sec. under peak server workload.

The PIN verification time must not exceed 0.3 sec. under

normal server workload and 0.5 sec. under peak server workload.

The PIN verification time must not exceed 0.3 sec. under

normal server workload and 0.5 sec. under peak server workload.

The PIN verification time must not exceed 0.3 sec. under

normal server workload and 0.5 sec. under peak server workload.

The PIN verification time must not exceed 0.3 sec. under

normal server workload and 0.5 sec. under peak server workload.

The PIN verification time must not exceed 0.3 sec. under

normal server workload and 0.5 sec. under peak server workload.

The PIN verification time must not exceed 0.3 sec. under

normal server workload and 0.5 sec. under peak server workload.

The PIN verification time must not exceed 0.3 sec. under

normal server workload and 0.5 sec. under peak server workload.

The PIN verification time must not exceed 0.3 sec. under

normal server workload and 0.5 sec. under peak server workload.

The PIN verification time must not exceed 0.3 sec. under

normal server workload and 0.5 sec. under peak server workload.

PER-6 - Account balance display time must not exceed 2 seconds under normal server workload and 3 seconds under peak server workload.

## 5.2. Safety Requirements

SAF-1 - There all be a secured she vault with a combination locking system.

SAF-2 – the user can terminate system operations if he feels any security threat from outside.

SAF-3 - The product cabinet cover shall be manufactured using Fiberglass for security purposes.

SAF-4 – The cabin door must have an ATM card swipe slot which opens only when the user swipes his/her atm card.

## 5.3. Security Requirements

SEC-1 – The password shall be 4 digits long.

SEC-2 - The system shall have two levels of security i.e., ATM card and pin verification.

SEC-3 - The system shall lock a user’s account after four consecutive

Unsuccessful logon attempts within a period of five minutes

## 5.4. User Documentation

* The ATM interface will contain a help option that will let the user choose between:

1. online assistance, and a 24/7 helpline facility to resolve the day-to-day

operational issues at ATMs. The helpline should also be accessible through special phone terminals placed inside all ATM booths/cabins. Banks should ensure that these phones are in working order.

1. a video tutorial that will show the users how to use different functionalities of the ATM like cash deposit or withdrawal, etc.

* User Manual and Video Tutorial for the ATM will be sent to the bank.

***SDS***

# 6. System Architecture

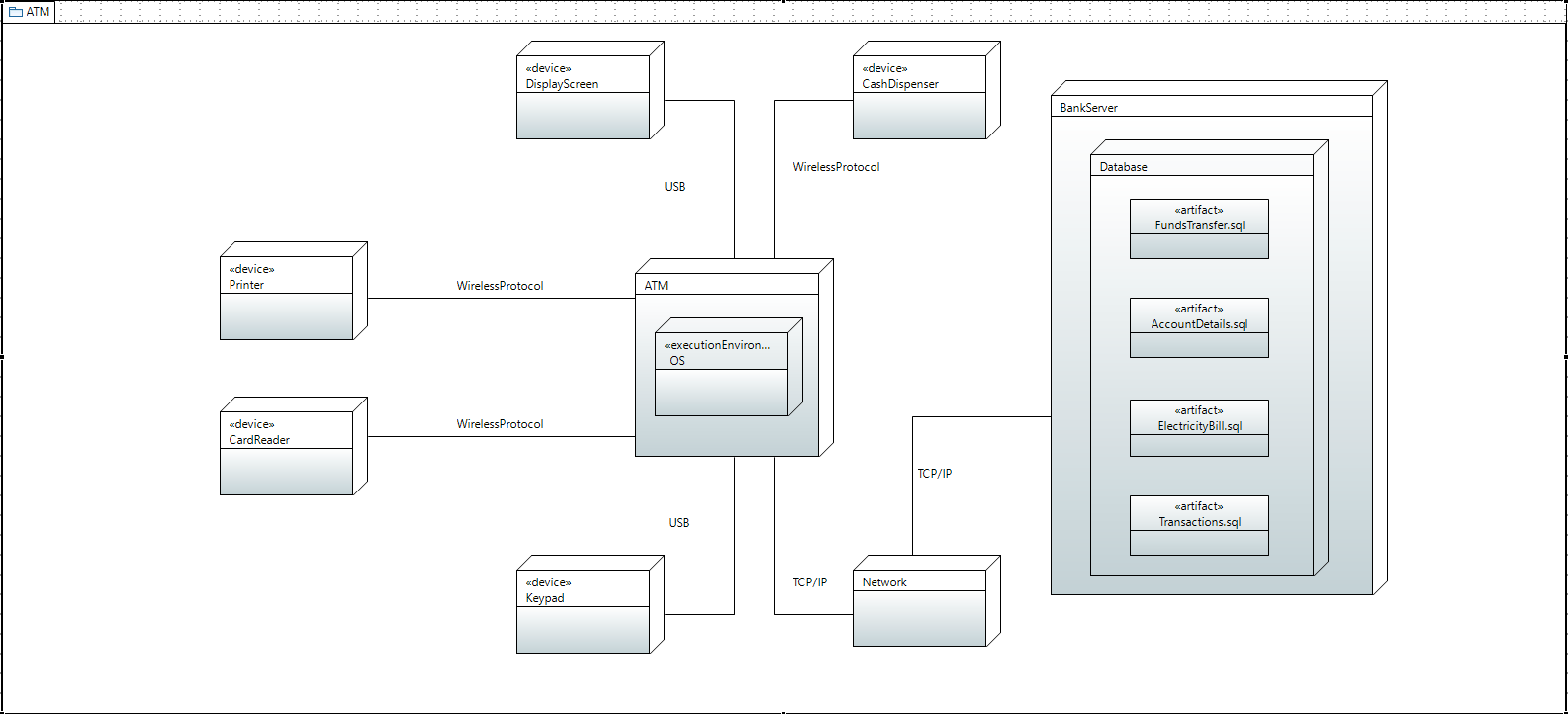
The Automated Teller Machine (ATM) system architecture can be described as a complex network of hardware and software components working together to provide banking services to customers.

At a high level, the ATM system architecture can be divided into four main components:

1. Customer Interface: This component is responsible for the interaction between the customer and the ATM machine. It includes the ATM keypad, screen, card reader, cash dispenser, and receipt printer. The customer interface is designed to be easy to use and secure, with features like encryption and PIN protection to ensure the safety of customer data.
2. ATM Controller: This component manages the customer interface and communicates with the rest of the ATM system. It is responsible for handling customer transactions, verifying customer identities, and maintaining the ATM's internal security measures. The ATM controller also communicates with the bank's central server to authorize transactions and update account balances.
3. Network Interface: This component connects the ATM to the bank's central server through a network infrastructure. It is responsible for transmitting customer data securely between the ATM and the bank's server, using encryption and other security measures to prevent unauthorized access or tampering.
4. Central Server: This component manages the overall ATM system, including transaction processing, database management, and security. The central server is responsible for authorizing transactions, updating customer account balances, and maintaining a secure record of all ATM transactions.

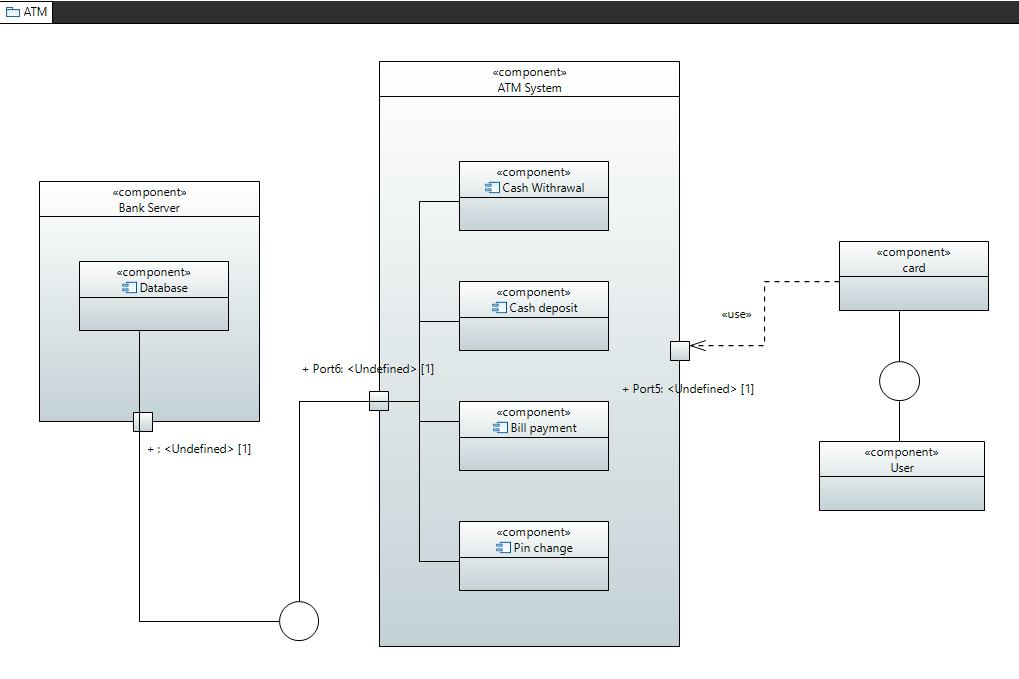
Overall, the ATM system architecture is designed to be reliable, secure, and easy to use for customers. By integrating hardware and software components and leveraging network infrastructure, the ATM system provides a fast and convenient way for customers to access their bank accounts and perform transactions 24/7.

## 6.1. System Level Architecture



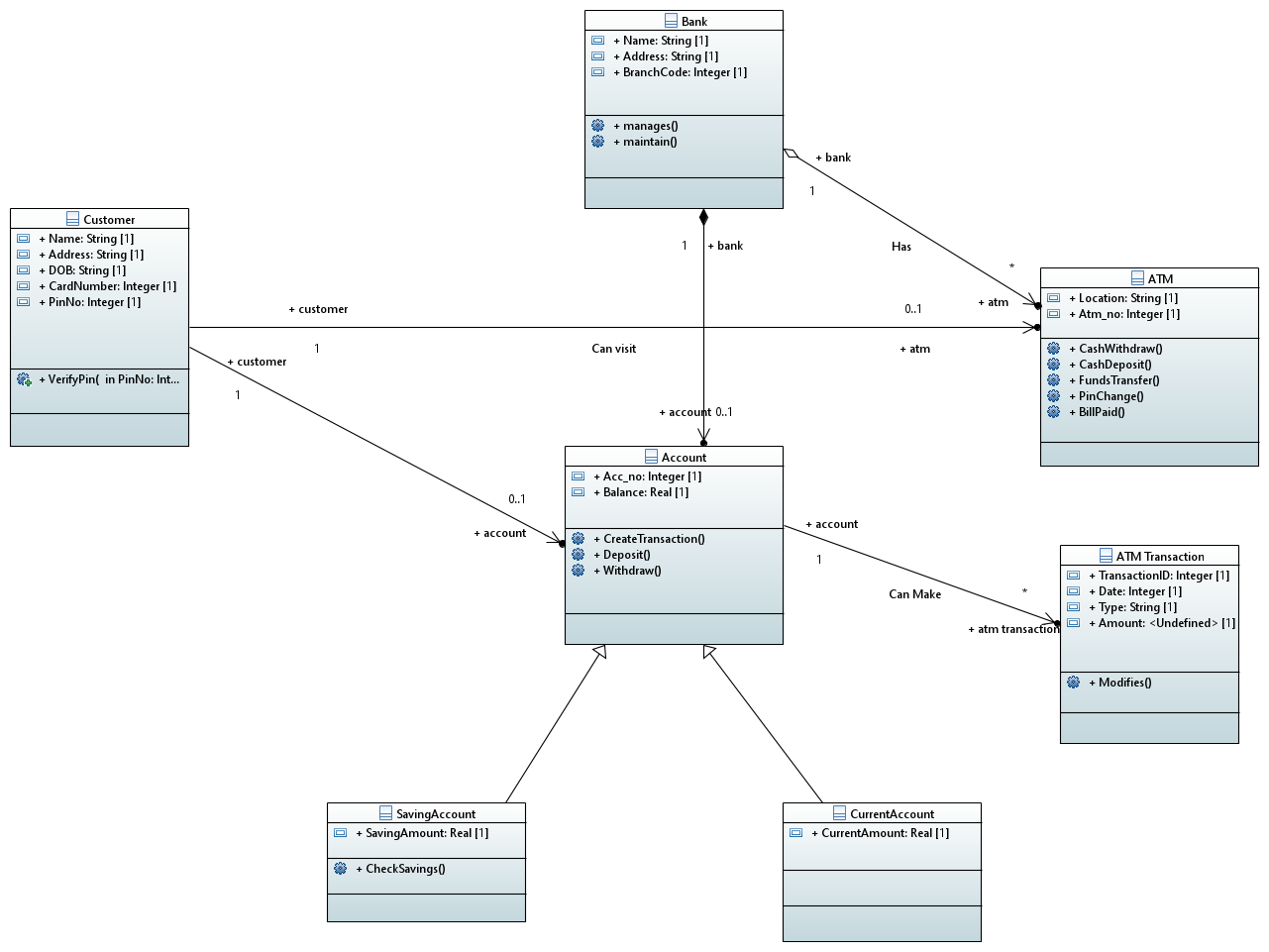
## 6.2. Software Level Architecture:

**Component Diagram.**



# 7. Detailed System Design

## 7.1.Class Diagram:



### 7.1.1Class Diagram Description:

**Bank Class:**

The Bank class has 3 attributes:

* + 1. Public String Name
    2. Public String Address
    3. Public Integer BranchCode

The name attribute is use to store the name of the bank whose atm is it along with the address attribute holding the address of the main branch of bank where as BranchCode attribute hold the codes of all the branches of the bank . The bank class contain manages() and maintain functions. It has composite association with account class with the multiplicity of 1 to many and Aggregation association with Atm class.

**Account Class:**

The Account class has 2 attributes:

1. Public Integer Acc\_no
2. Public Real Balance

The Acc\_no attribute holds the account number of the customer who has opened his account in the bank. Secondly, the Balance attribute holds the balance of the customer's account/accounts. Furthermore, there are three methods in the Account class: Firstly, CreateTransaction() will call the ATMTransaction class’s object. Secondly, Deposit() method deposits the amount deposited into the account of the customer. And lastly, Withdraw() method withdraws the amount Withdrawn from the account of the customer. Account class has composite association with Bank, has dependent association with Customer and ATMTransaction classes and generalization relationship with SavingAccount and CurrentAccount.

**ATMTransaction Class:**

The ATMTransaction class has 4 attributes:

1. Public Integer TransactionID
2. Public Integer Date
3. Public String Type
4. Public Integer Amount

The Attribute TransactionID stores unique transaction ID of every transaction. Secondly, Date attribute stores the date of the day transaction was done. Thirdly, attribute type stores the type of transaction customer will be performing. Lastly, attribute amount stores the amount that has been transacted. Moreover, there is only one method in the ATMTransaction class which is Modifies(). It modifies the account details of the customer that has just performed the transaction. ATMTransaction dependency association with Account claas

**CurrentAccount Class:**

The CurrentAccount class has 1 attribute:

1. Public Real CurrentAccount

CurrentAccount class is a child class of Account class and inherits it. It describes the type, ‘current’ of the account.

**SavingAccount Class:**

The SavingAccount class has 1 attribute:

1. Public Real SavingAccount

SavingAccount class is a child class of Account class and inherits it. It describes the type, ‘saving’ of the account. It’s only method CheckSavings() Checks the amount(savings) in the account.

**Customer Class:**

The Customer class has 5 attributes:

1. Public String Name
2. Public String Address
3. Public String DOB
4. Public Integer CardNumber
5. Public Integer PinNo

The attribute Name stores the name of the customer. Secondly, attribute Address stores the address of the customer. Thirdly, DOB(Date Of Birth) stores the date of birth of the customer. Fourthly, attribute CardNumber stores the unique verification card number of the customer. And lastly, PinNo stores the ATM pin of the customer. Moreover, the method VerifyPin(int PinNo) verifies if the PIN is correct. Customer’s class has dependency association with Account and ATM class.

**ATM Class:**

The ATM class has 2 attributes:

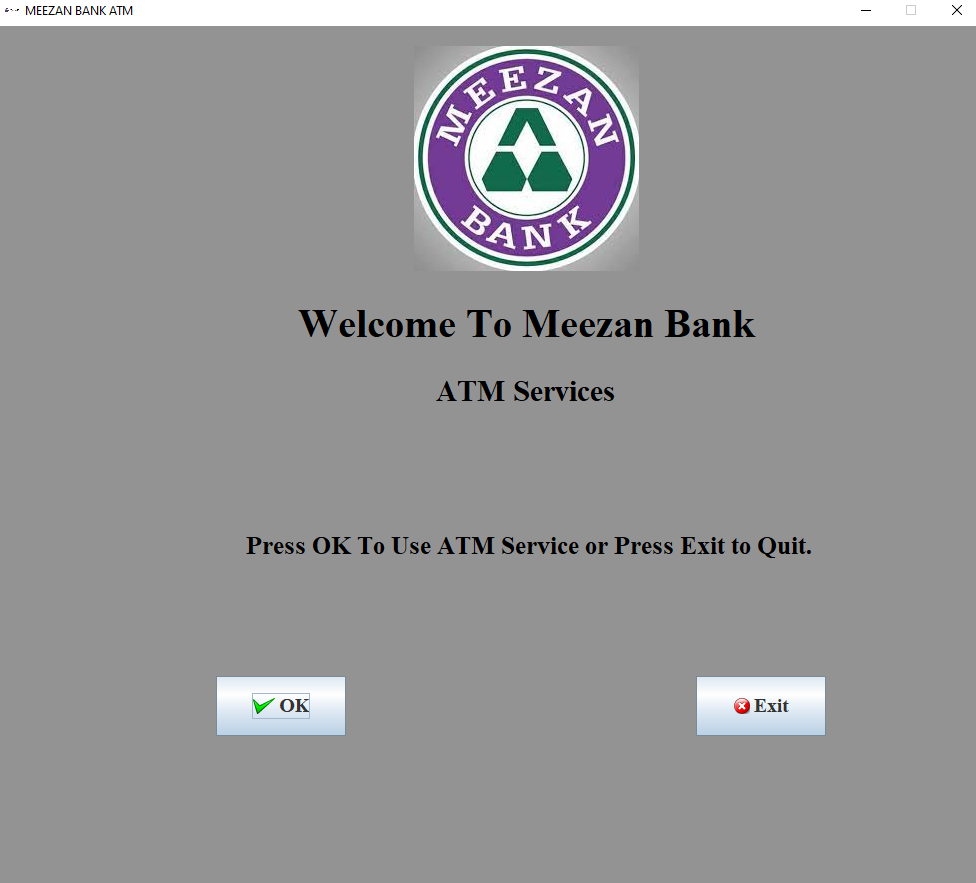
1. Public String Location
2. Public Integer Atm\_no

The attribute Location stores the location (area name) of the ATM and secondly, Atm\_no stores the unique identification number of the ATM. It has four methods: firstly, CashWithdraw() performs the withdrawal transaction from the atm. Secondly, CashDepsoit() performs the deposit transaction from the atm. Thirdly, FundsTransfer() transfers funds from the customer's account to another customer's account. Fourthly, PinChange() allows customers to change their ATM pin number. Lastly, BillPaid() allows customers to pay their electricity bills through the ATM.Furthermore, ATM class has dependency association with Bank and Customer classes.

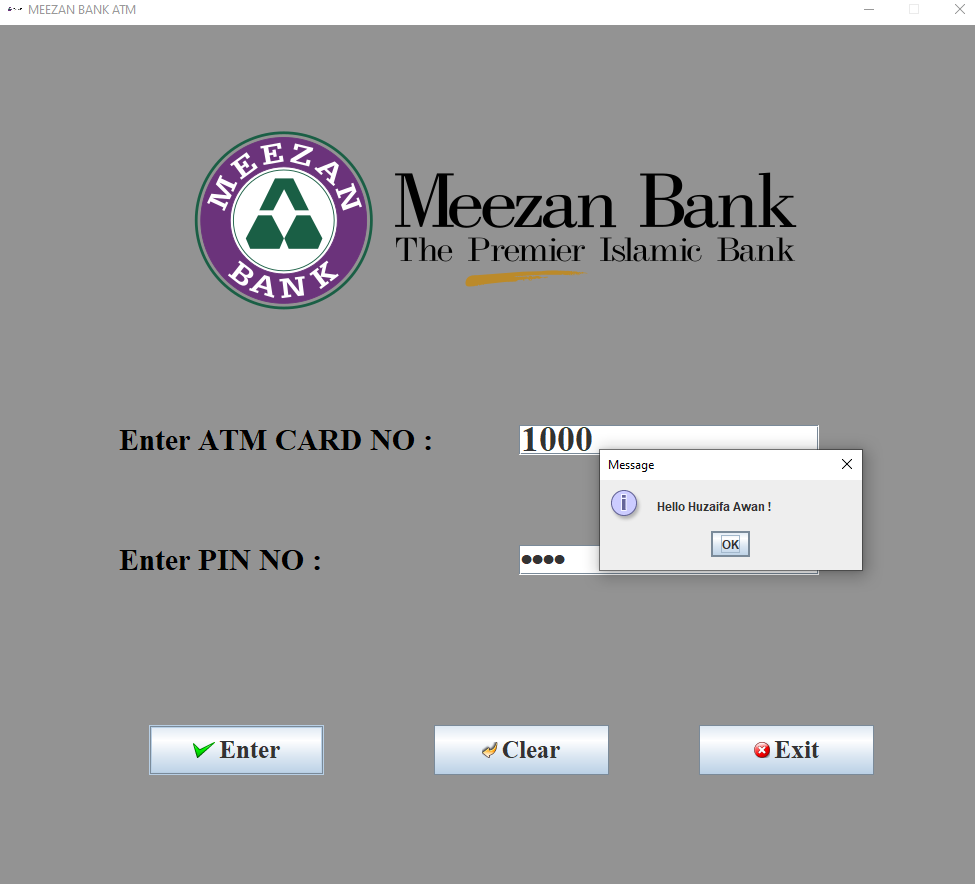
## 7.2. Detailed GUI

Input Screens using sample data:

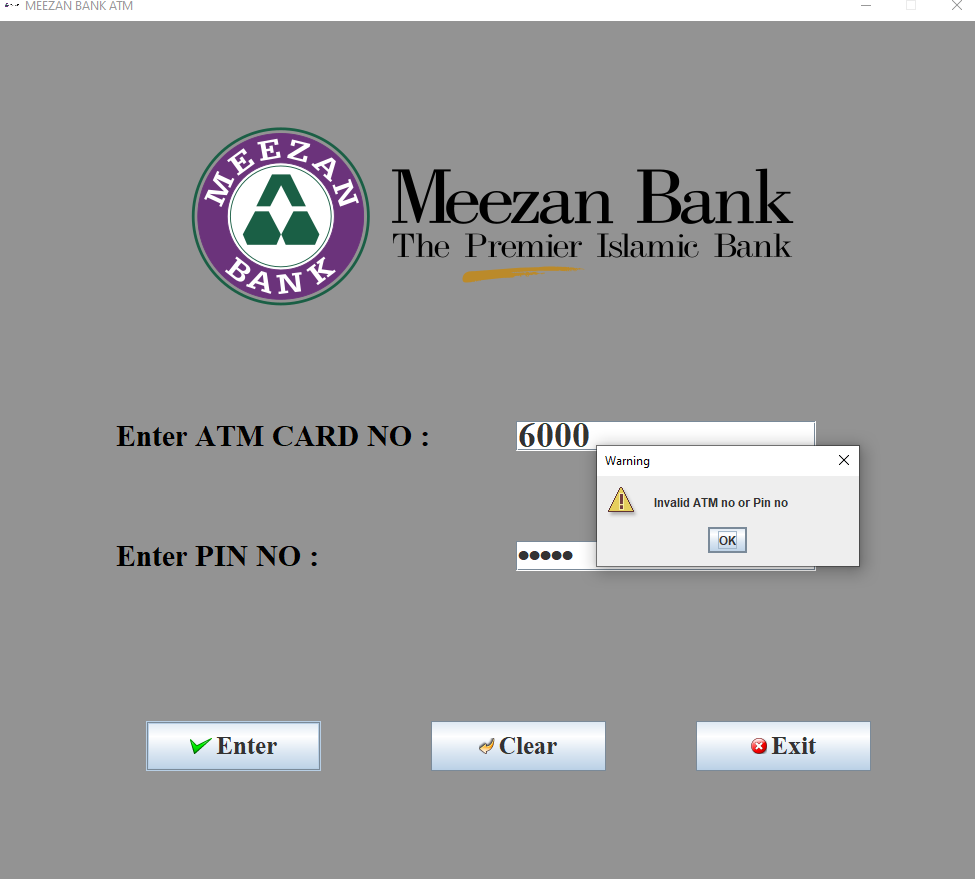
1. **Welcome screen:**



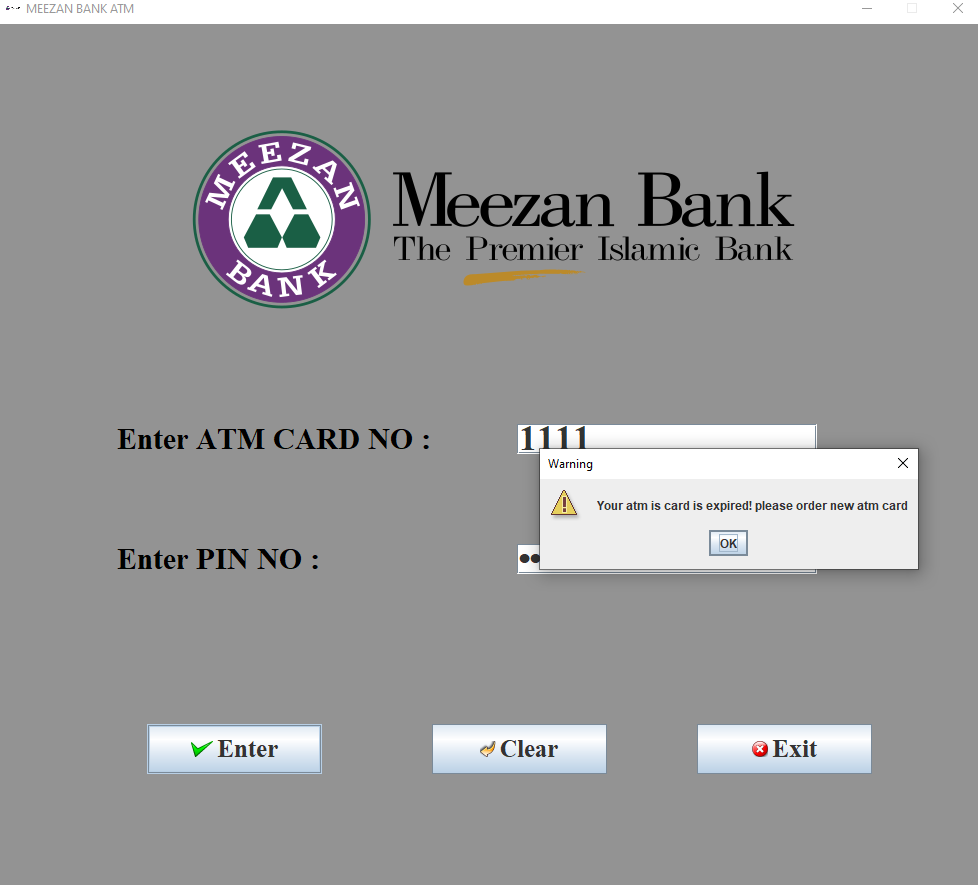
1. **ATM card no screen**: on success then below screen.



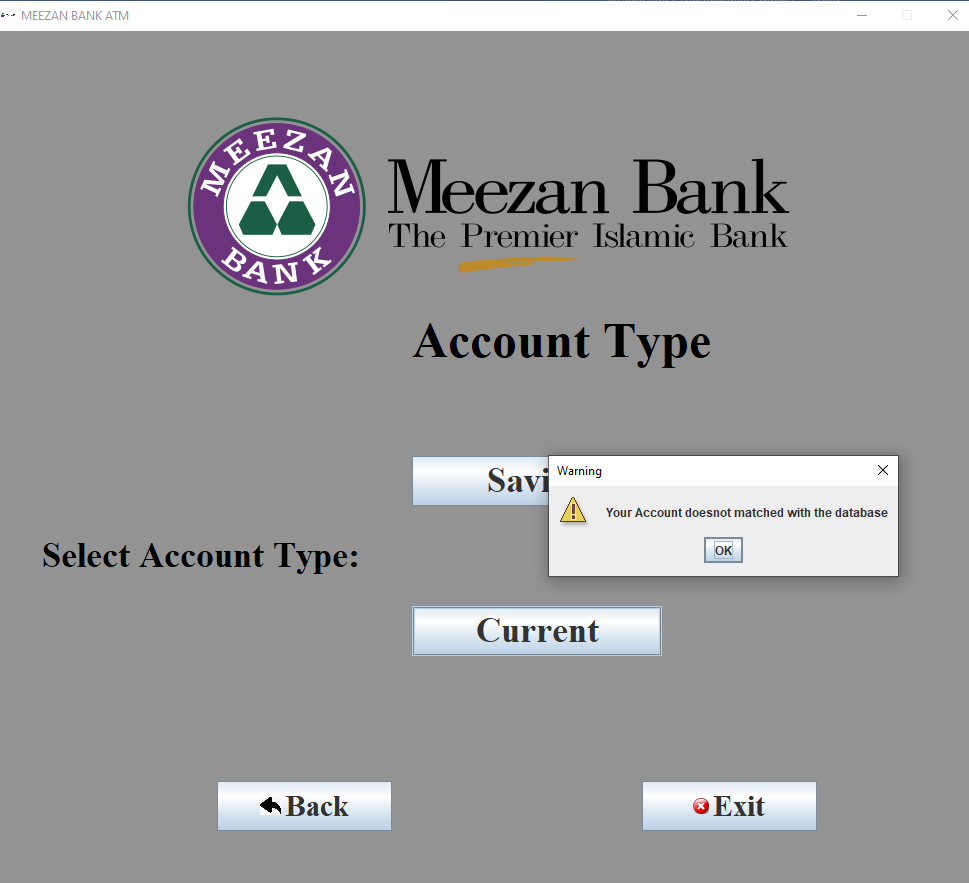
1. **ATM card no screen:** if enter wrong ATM no or PIN no.



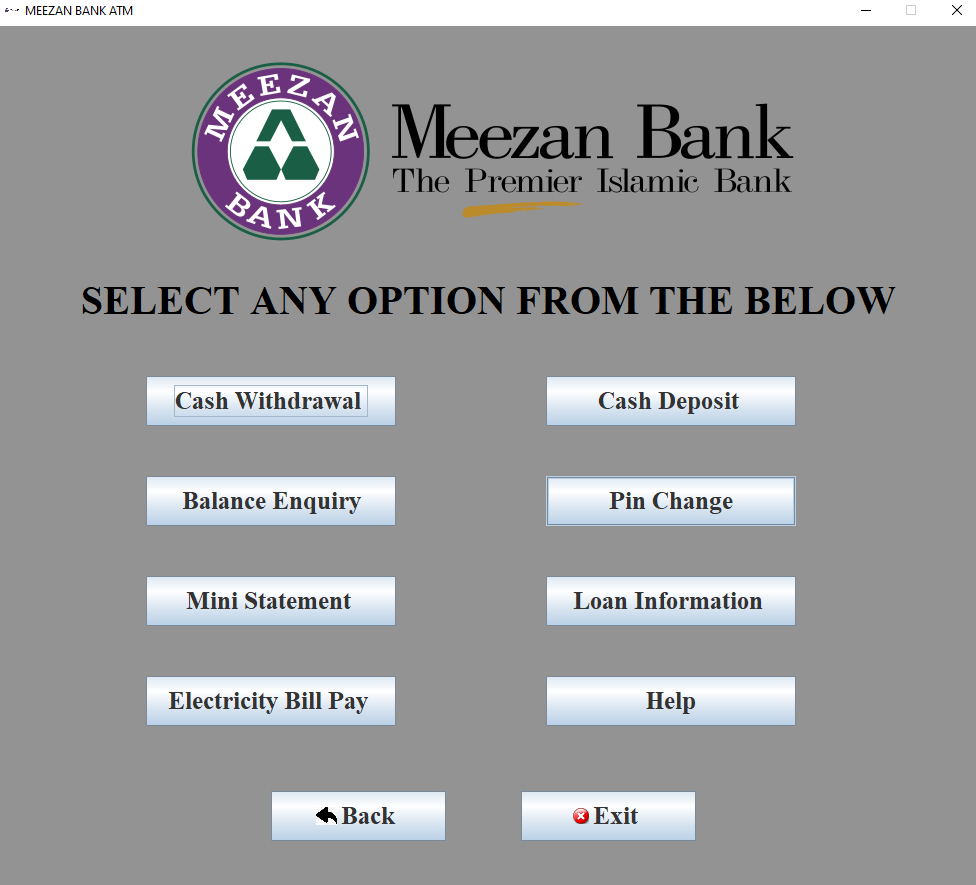
1. **ATM card no screen:** if ATM card is out expiry date



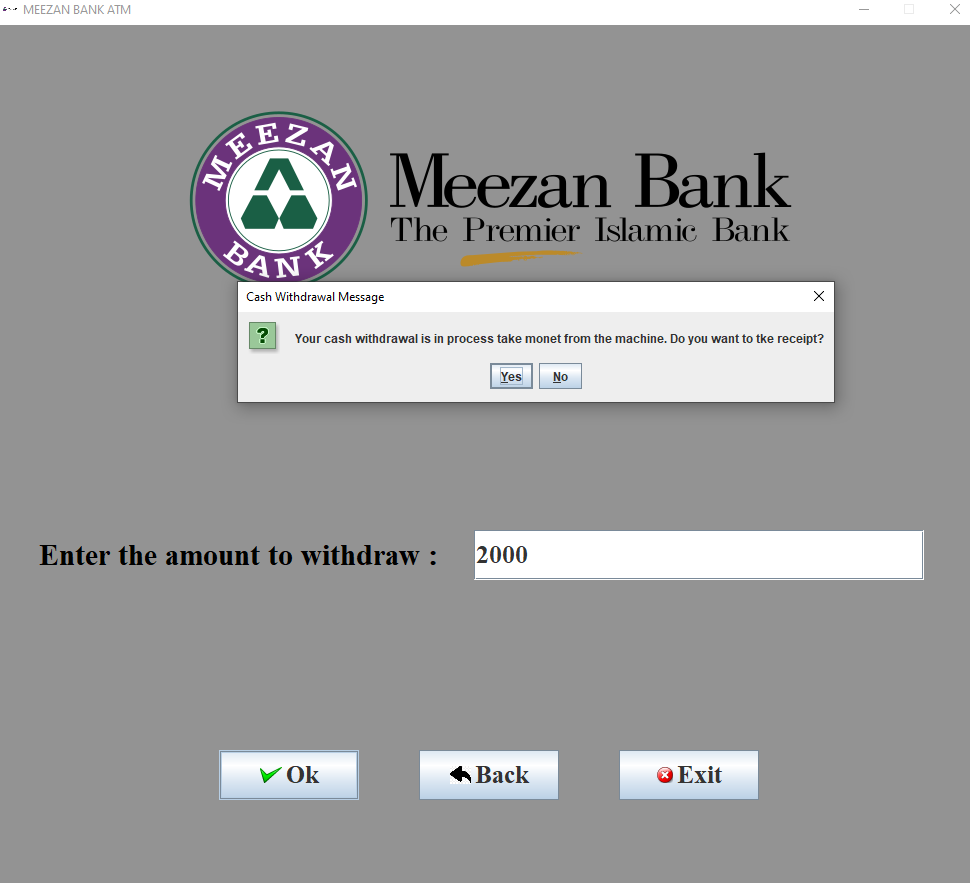
1. **Account type screen:** if account type is correct then go to transaction menu & wrong then appear below screen & go to welcome screen.



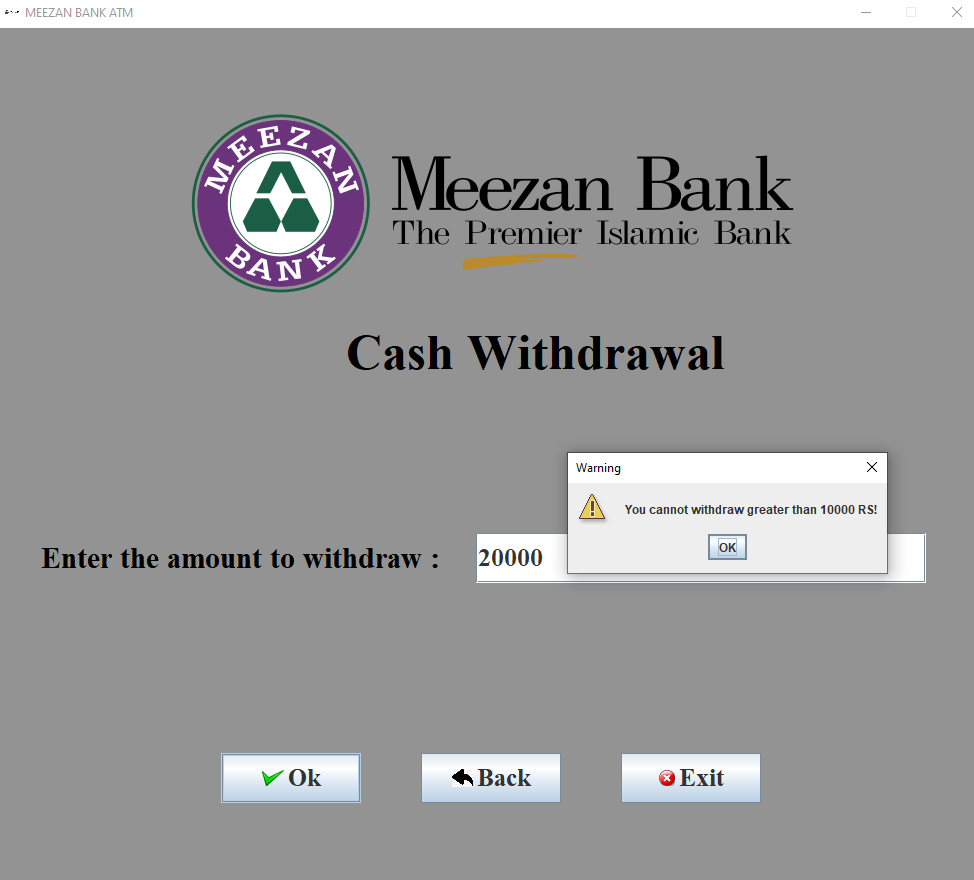
1. **Transaction menu screen:**



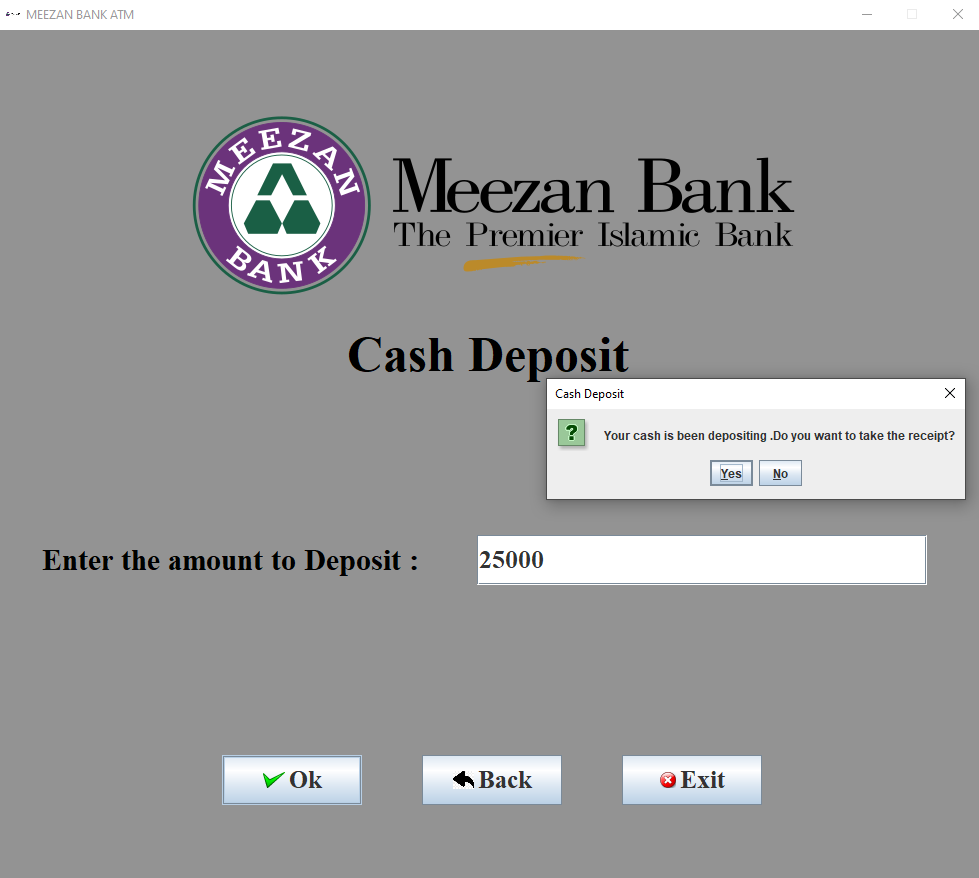
1. **Cash withdrawal screen:** if cash withdrawal is success then appear below screen, if we click on YES then go to balance enquiry screen, if NO then show available balance.



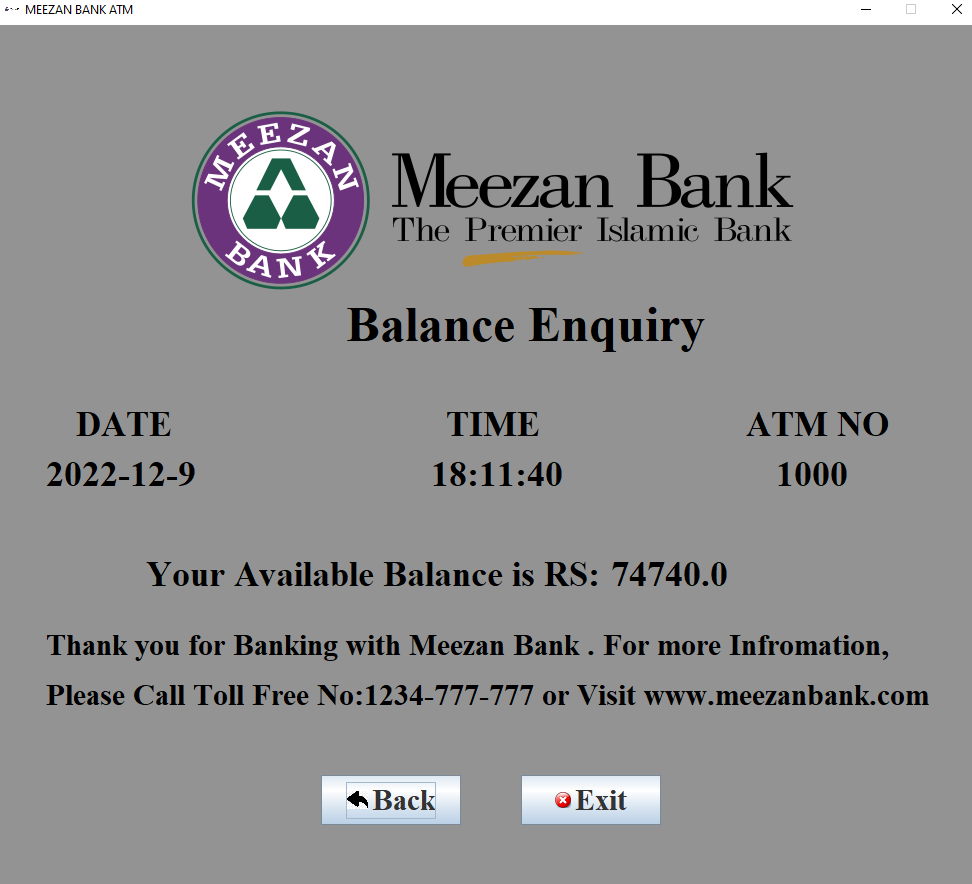
1. **Cash withdrawal screen:** if we enter amount 10000 then below appear screen.



1. **Cash deposit screen:** if cash deposit is success then appear below screen, if we click on YES then go to balance enquiry screen, if NO then show available balance.



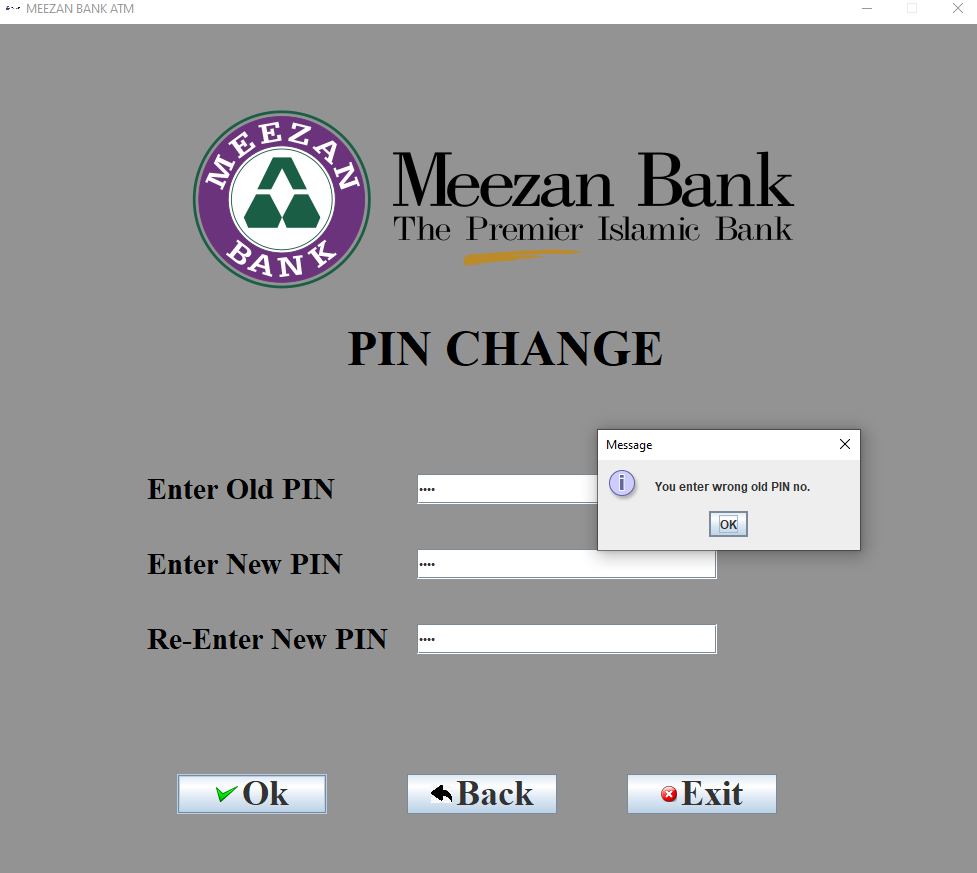
1. **Balance Enquiry screen:**



1. **PIN change screen:** if we enter old pin no correct and new & confirm pin no same then appear below screen.

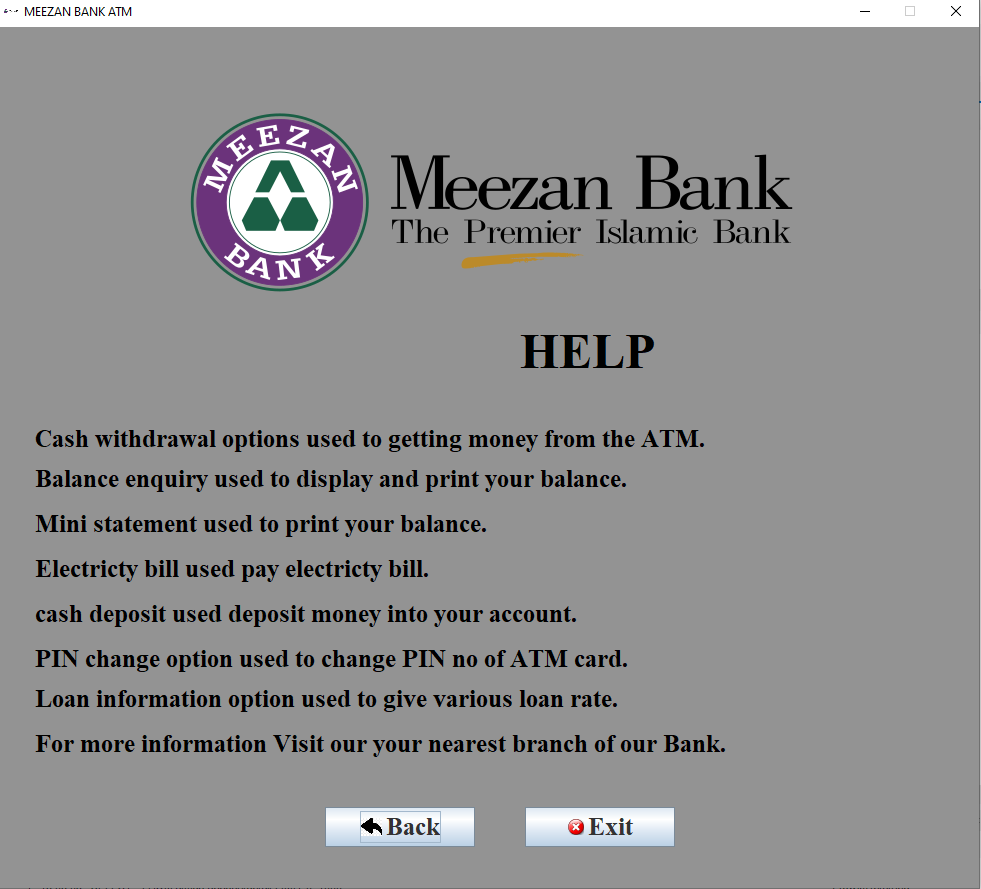


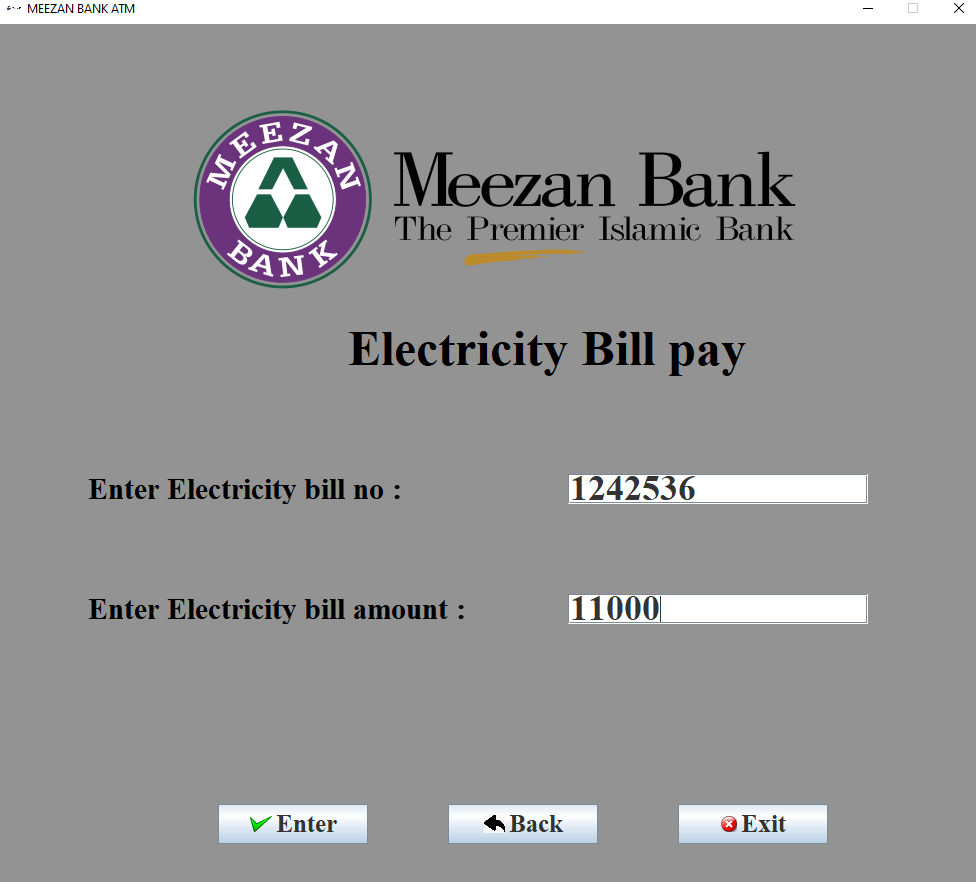
1. **PIN change screen:** if we enter old pin no wrong and new & confirm pin no same then appear below screen.



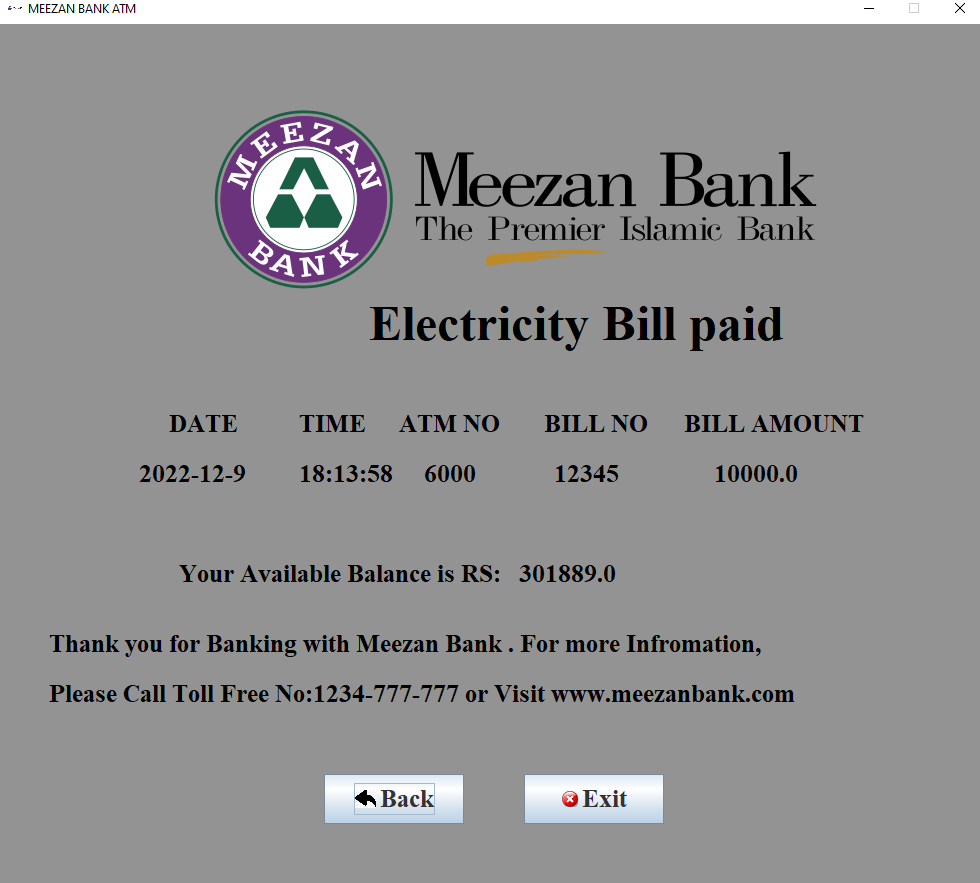
1. **Loan Information:**



1. **Help screen:**
2. **Electricity Bill Pay Screen:**



1. **Electricity Bill Paid :** Entering the bill no and the amount results updating the database and give you screen telling the time and remaining balance .mini



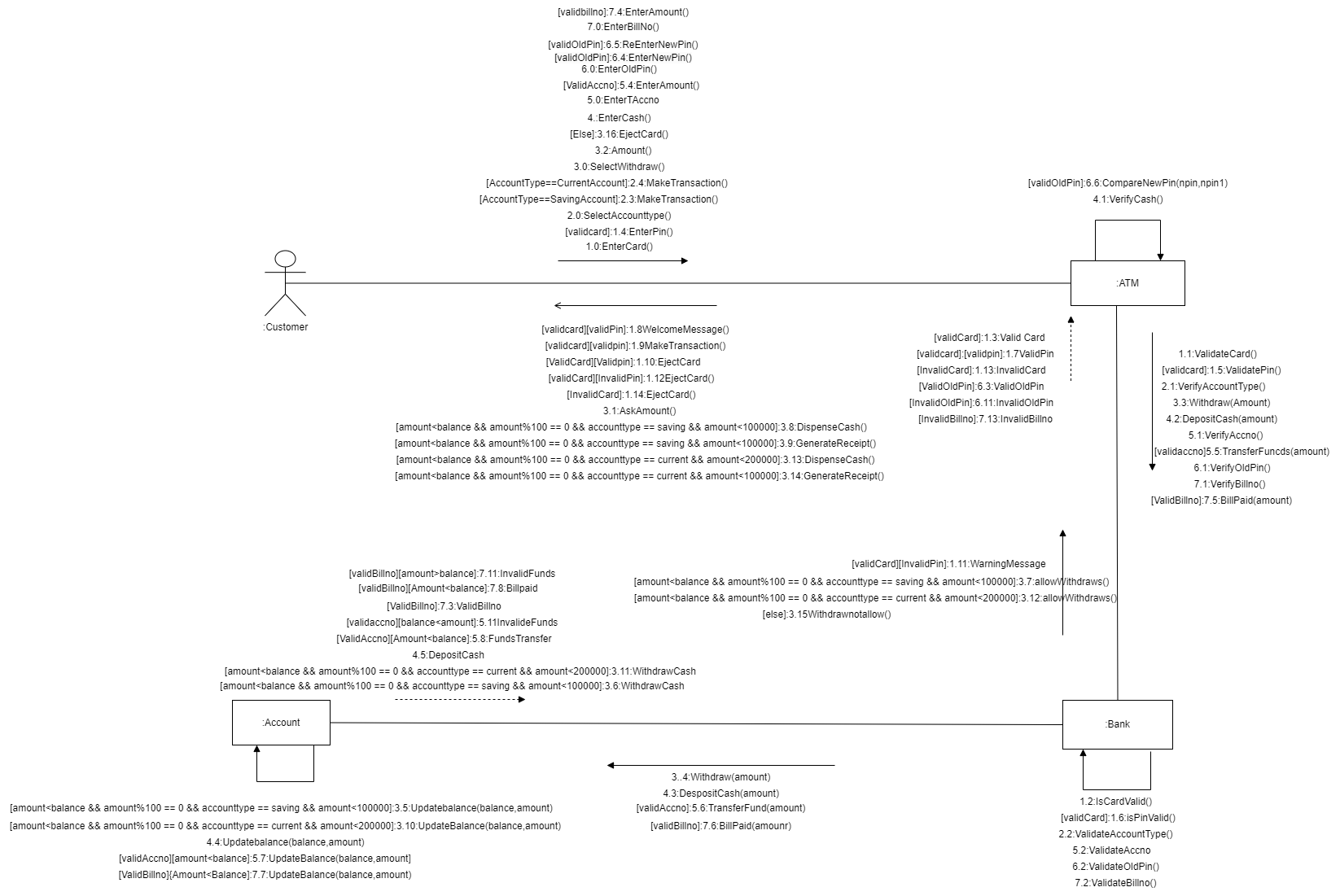
1. **Mini Statement Screen:**



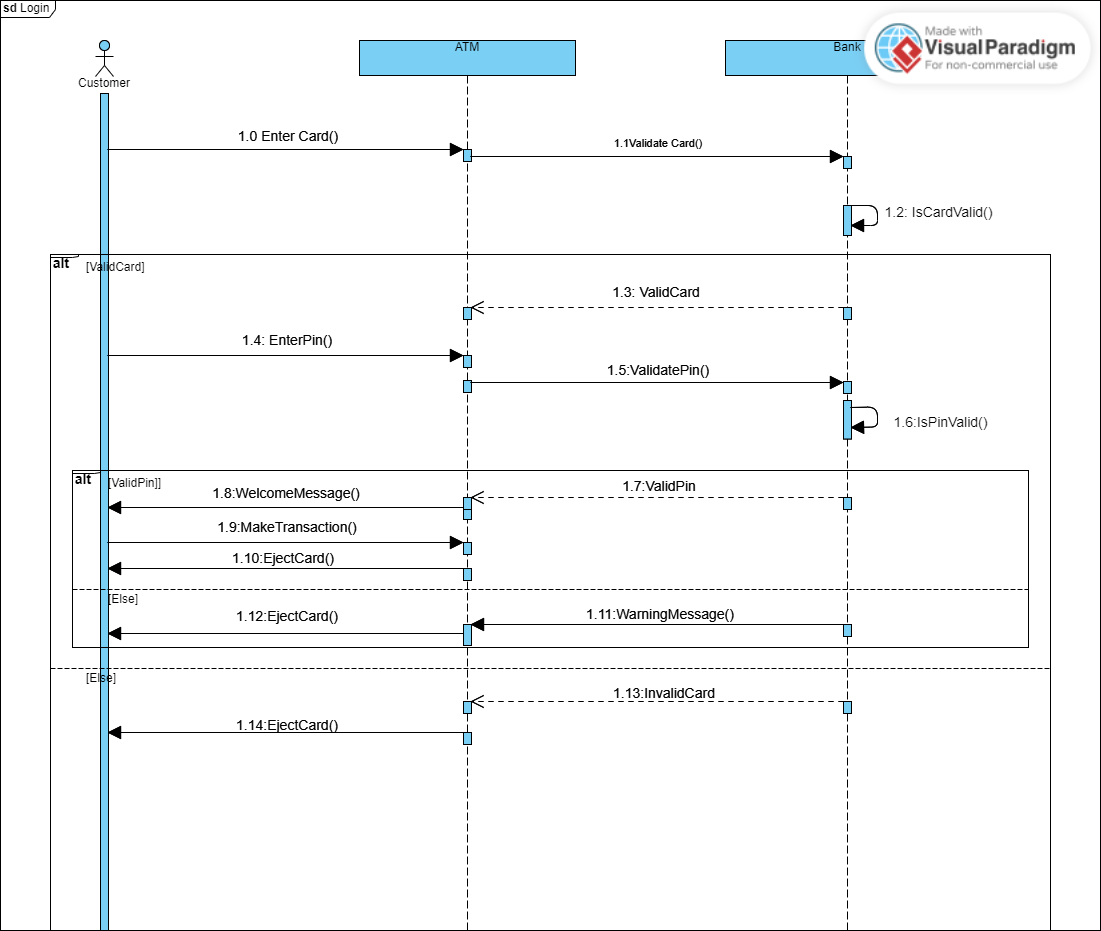
# 8. Application Design

All the diagrams have already been explained in the use-case description.

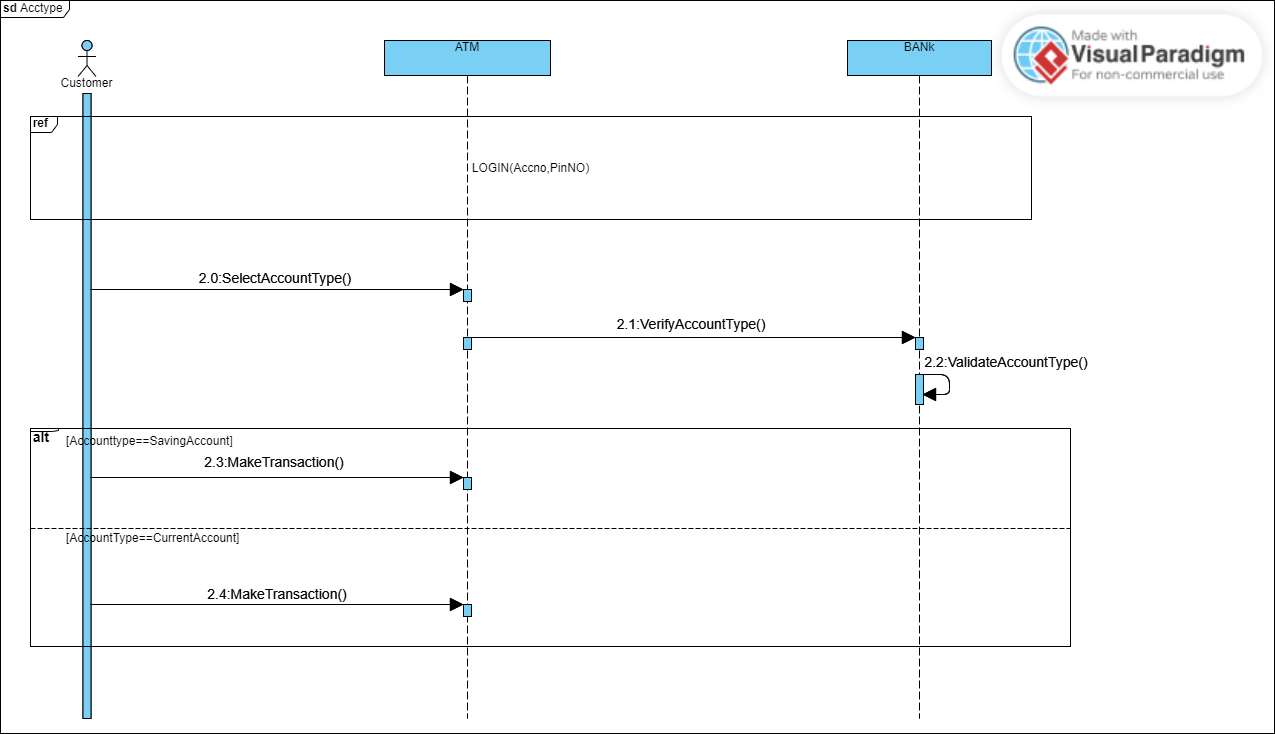
## 8.1 Collaboration Diagram



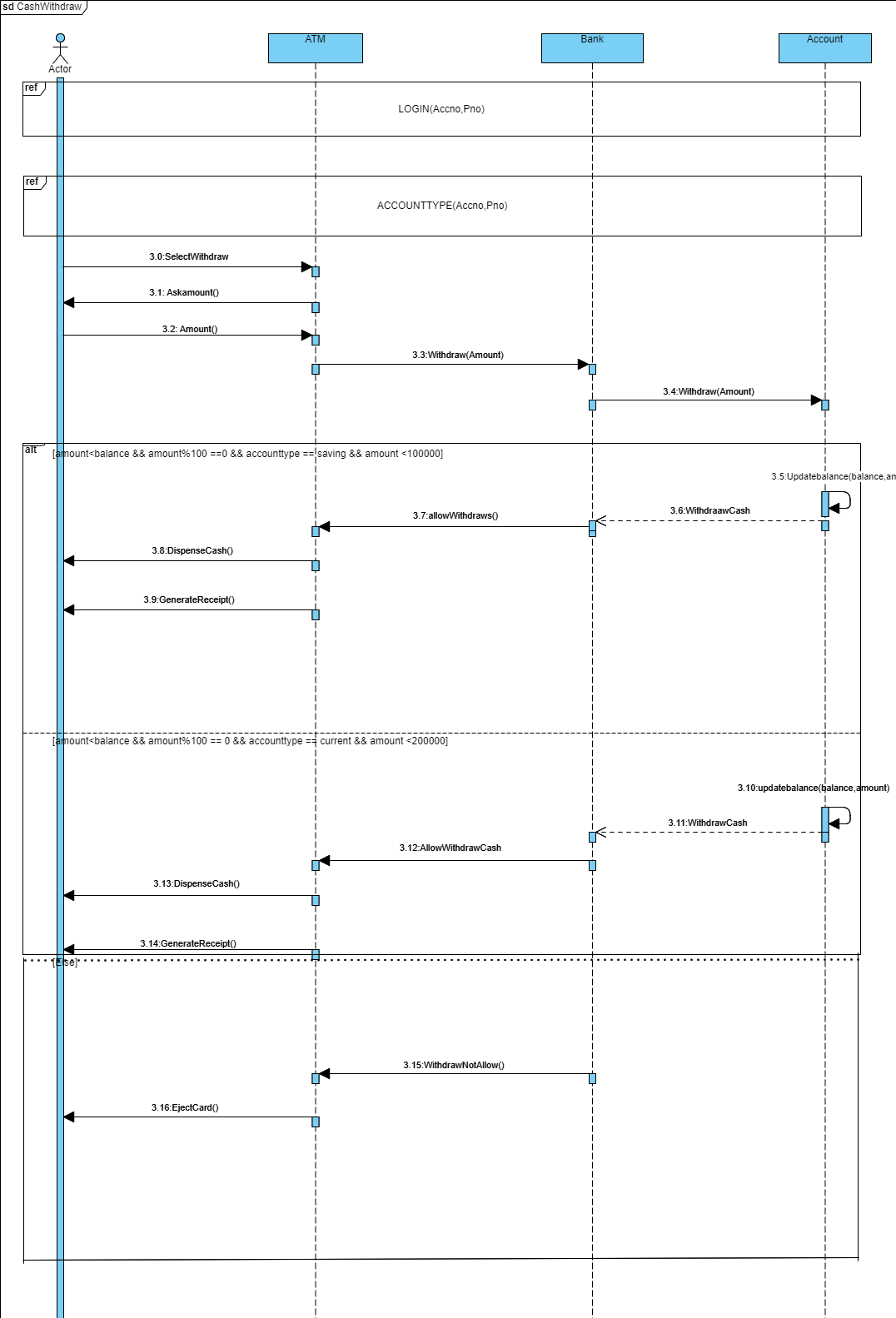
## 8.2 Sequence Diagram

8.2.1 Login

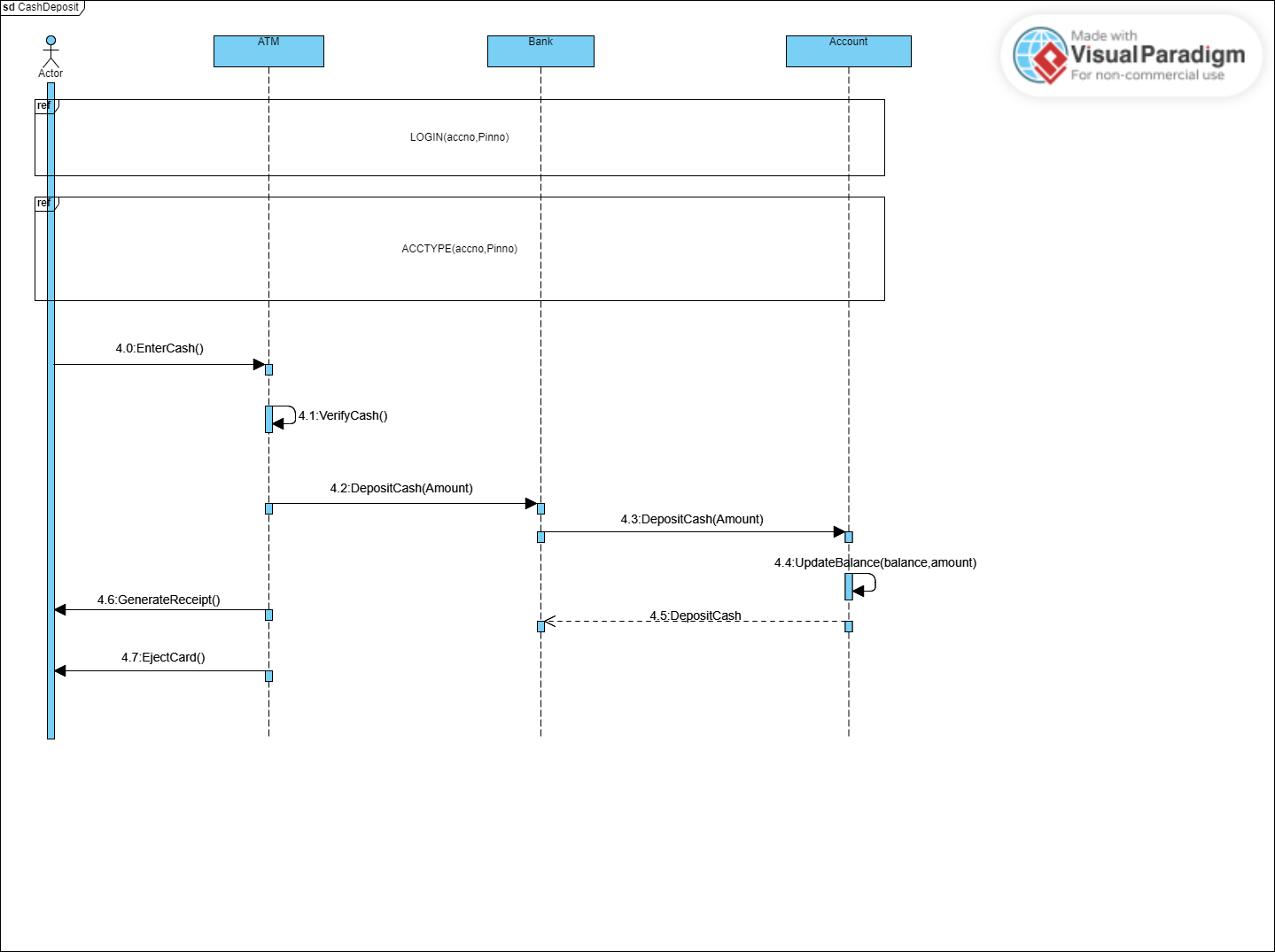
### 8.2.2 Account Type



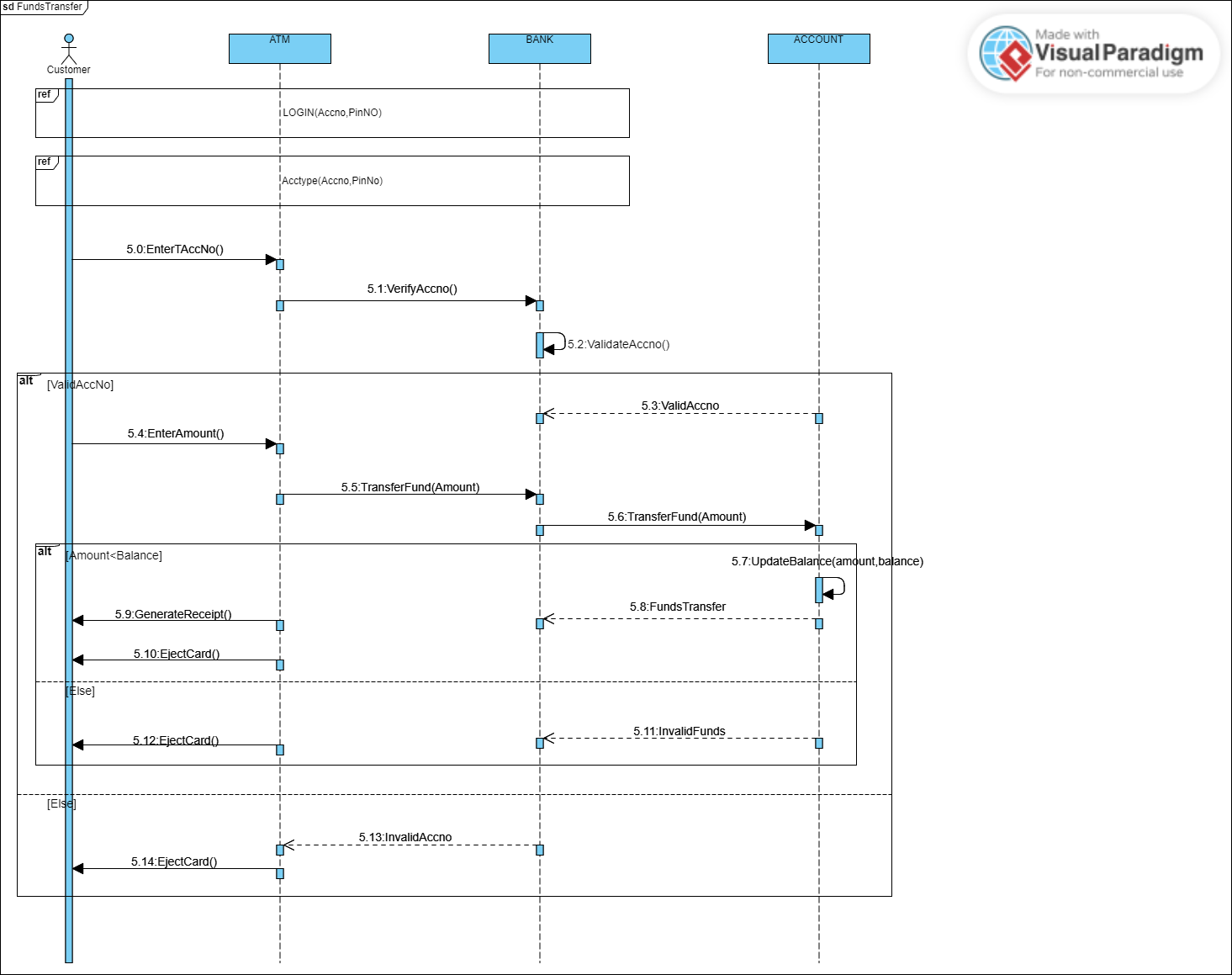
### 8.2.3 CashWithdraw



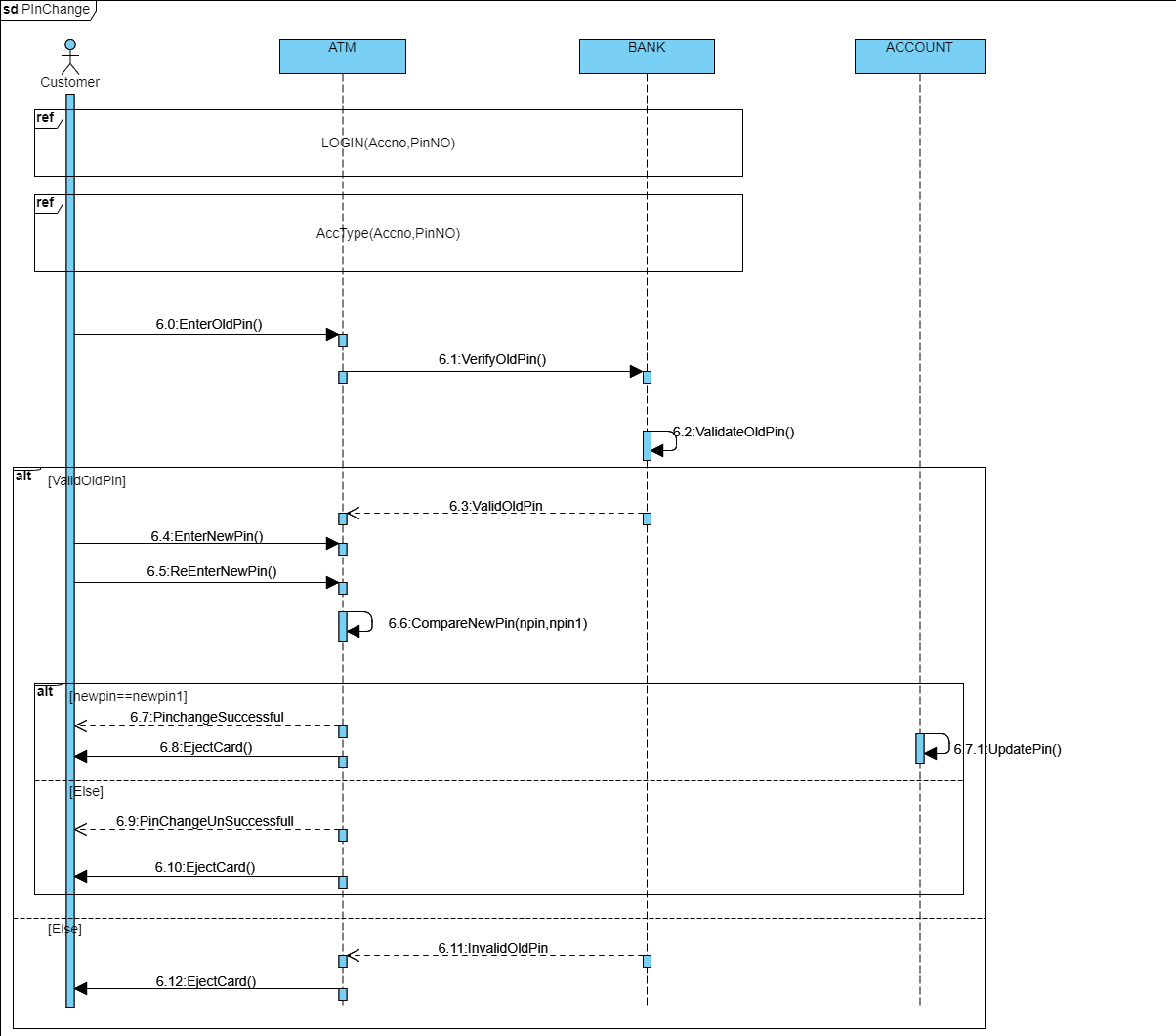
### 8.2.4 Cash Deposit



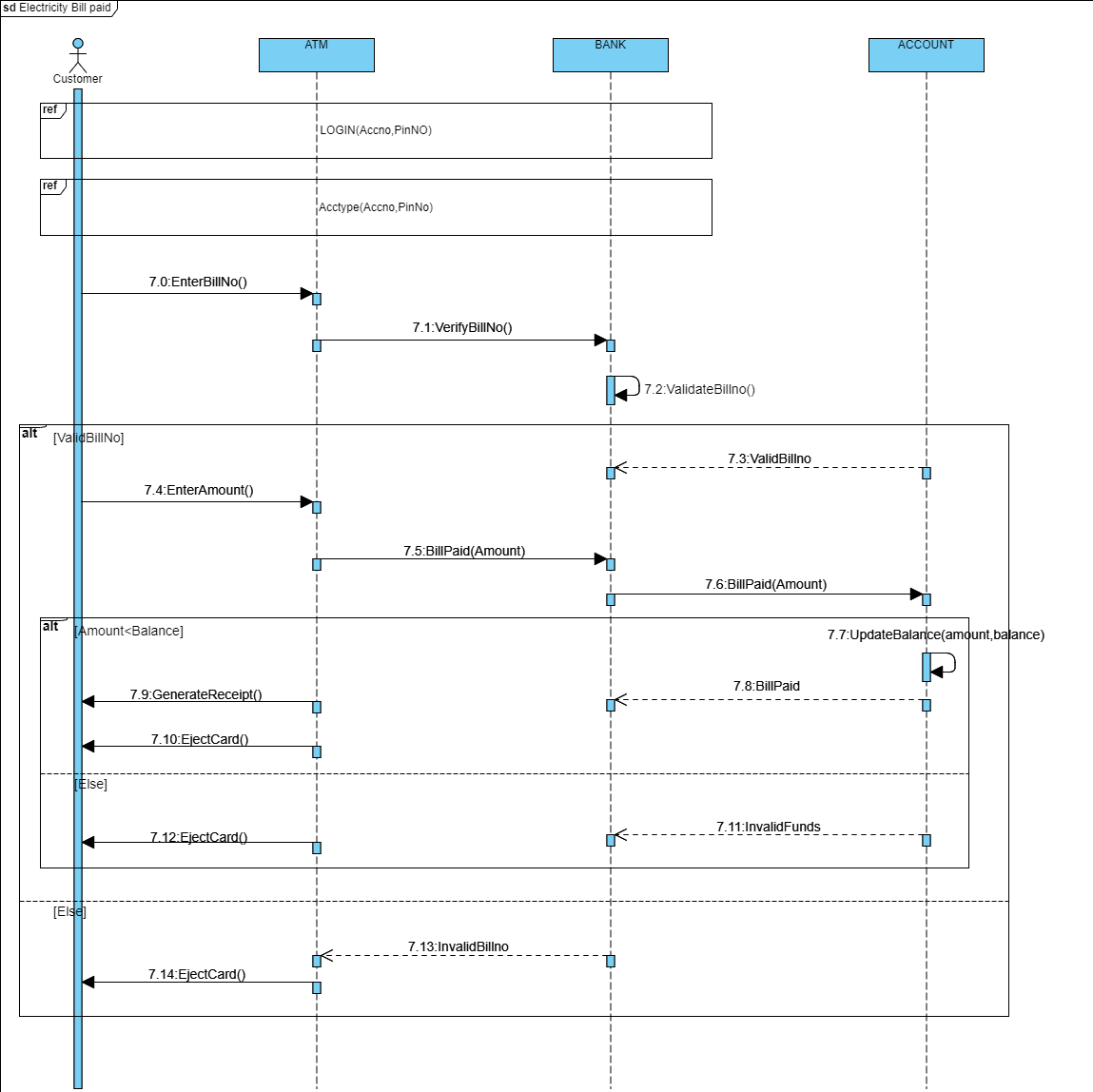
### 8.2.5 Funds Transfer



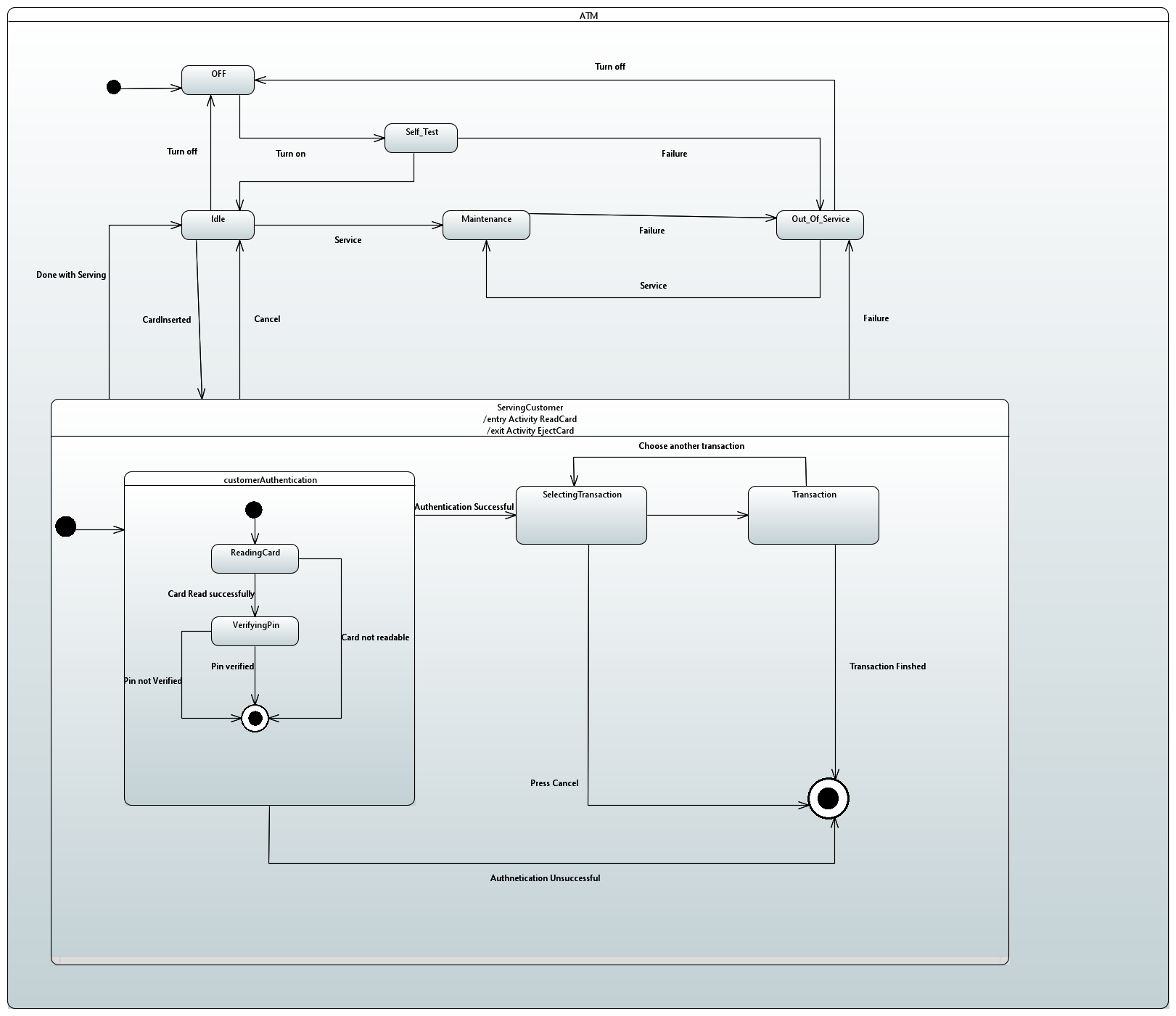
### 8.2.6 Pin change



### 8.2.7 Electricity Bill paid

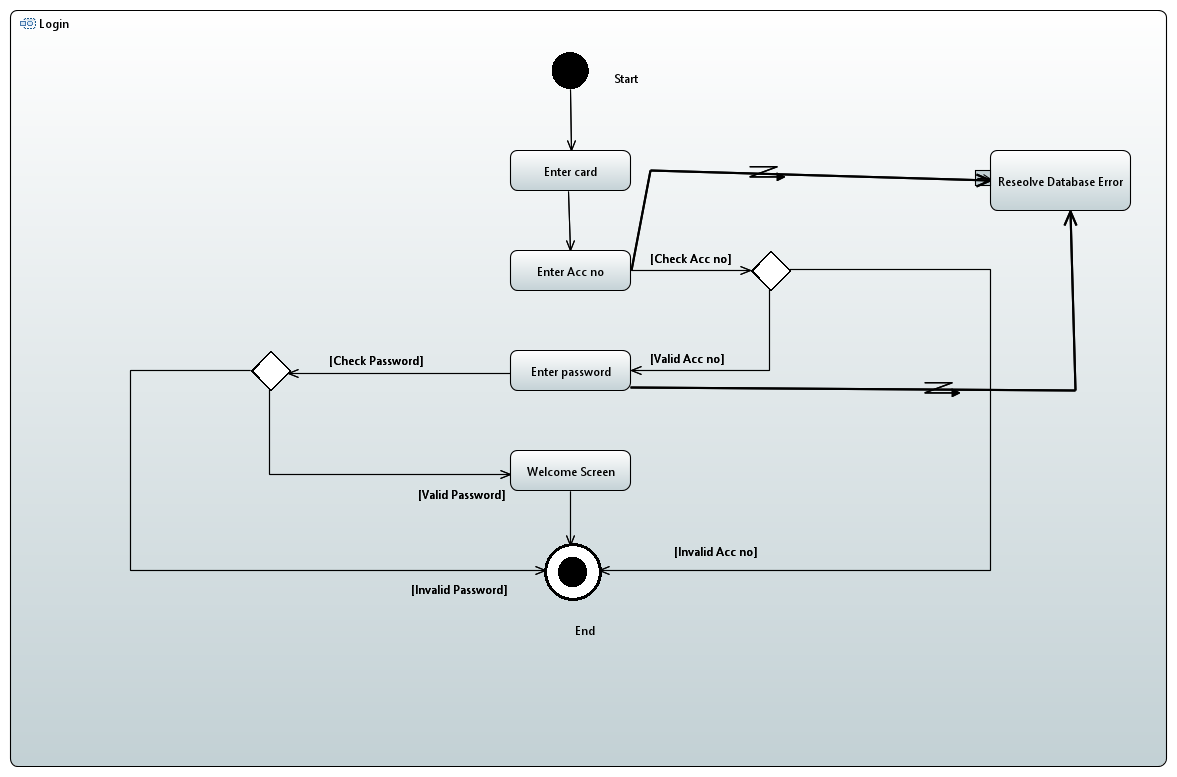


## 8.3. State Transition Diagram

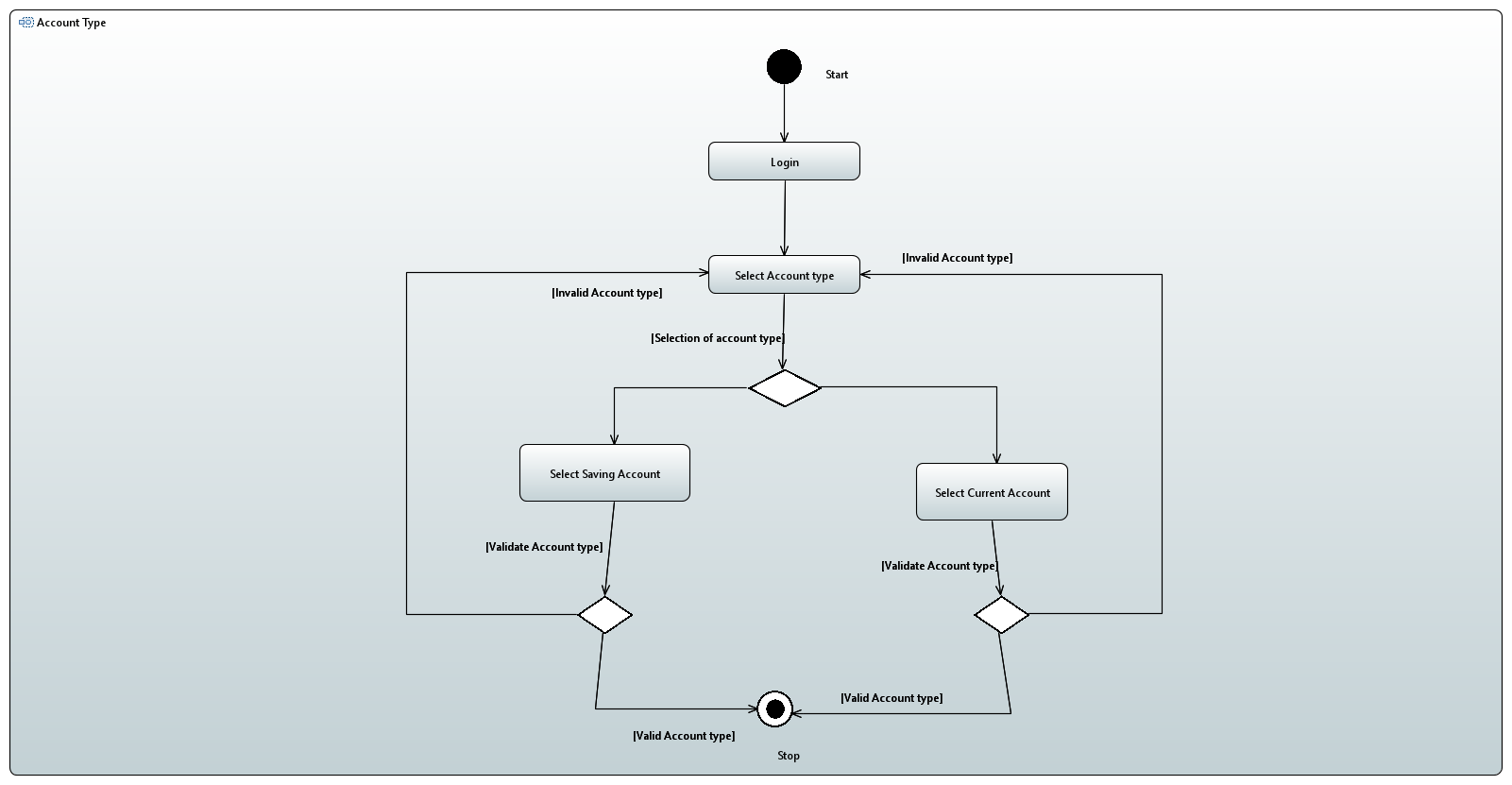


## 8.4. Activity Diagram

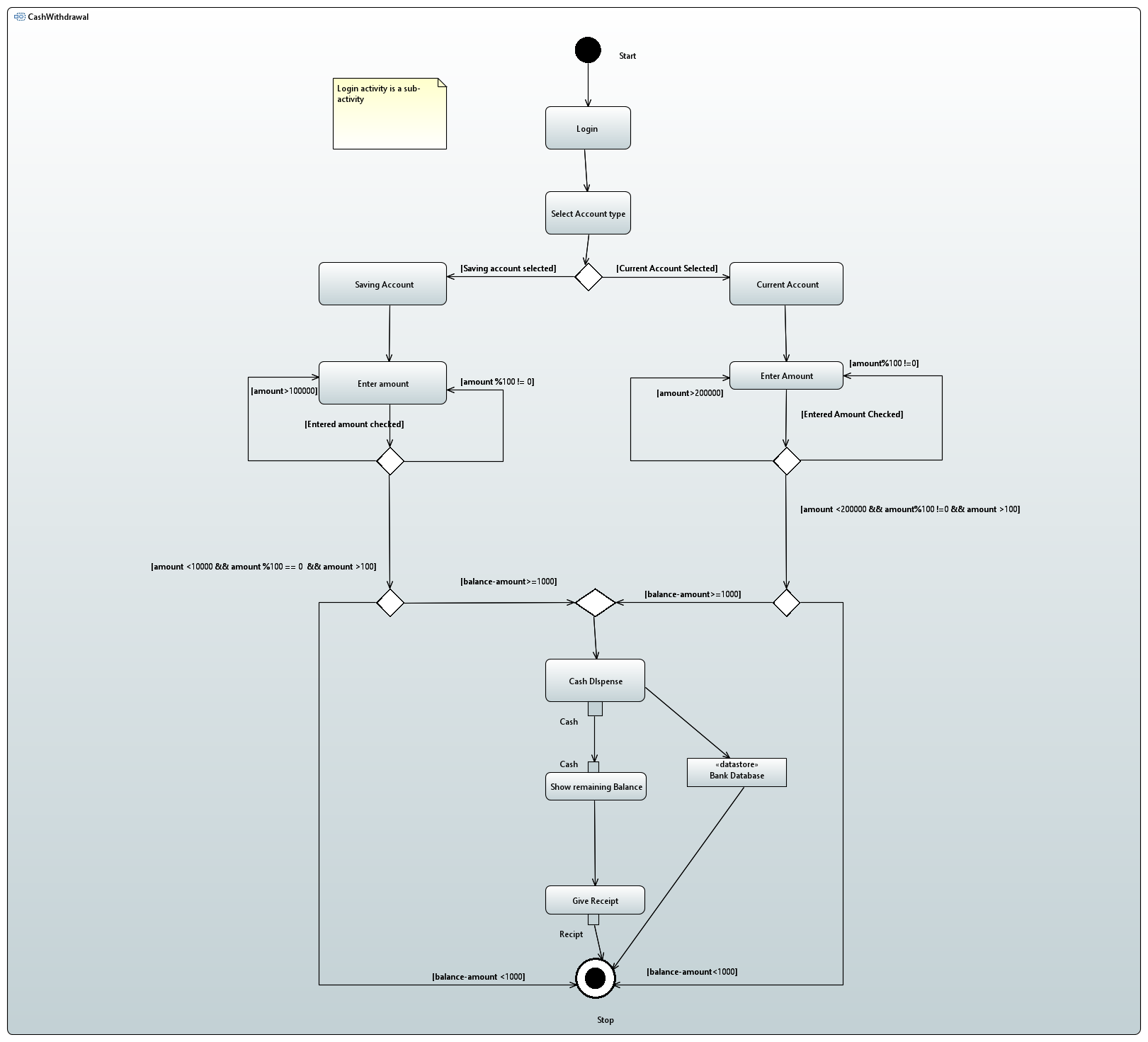
### 8.4.1 Login



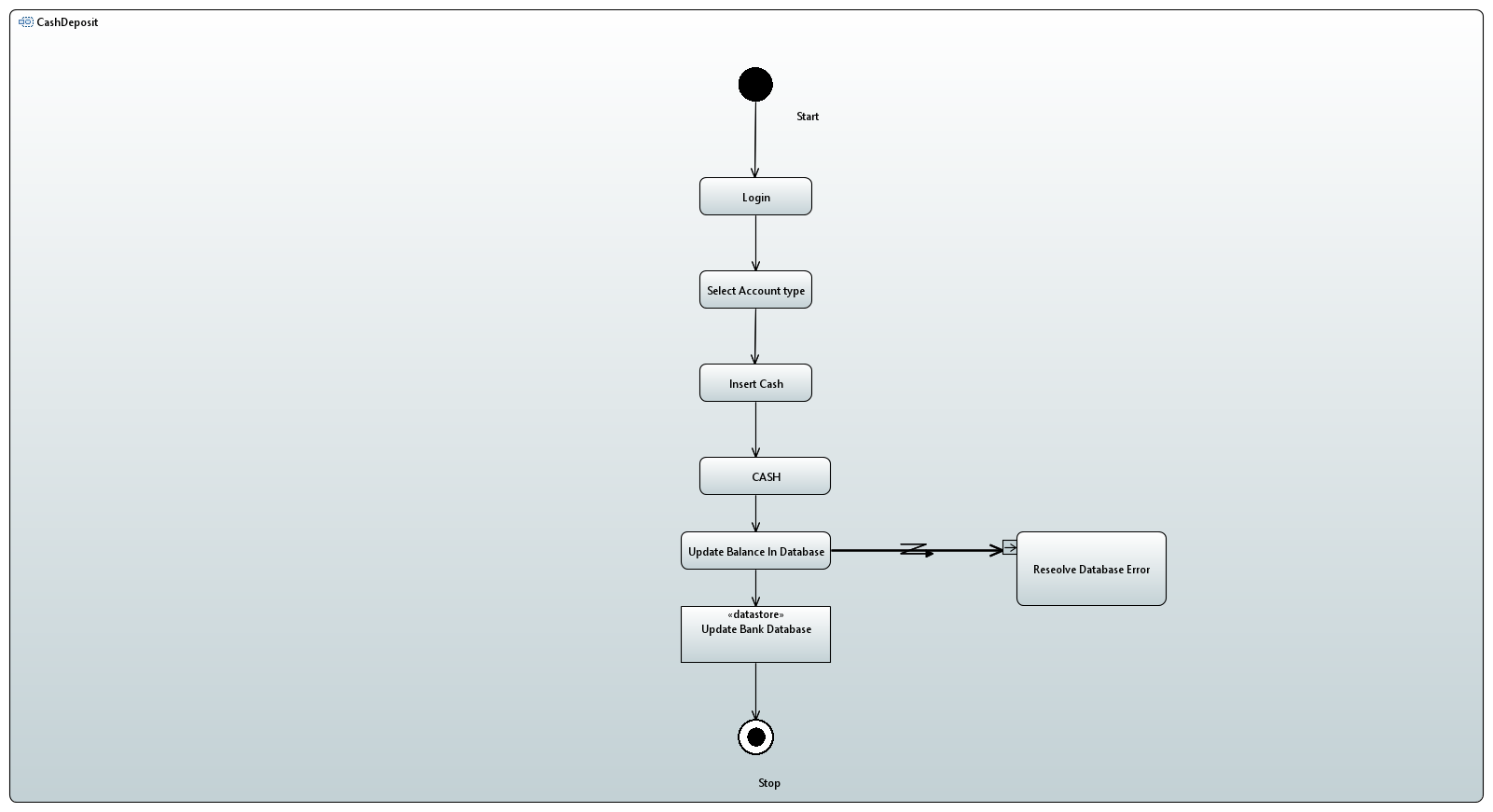
### 8.4.2 Account type



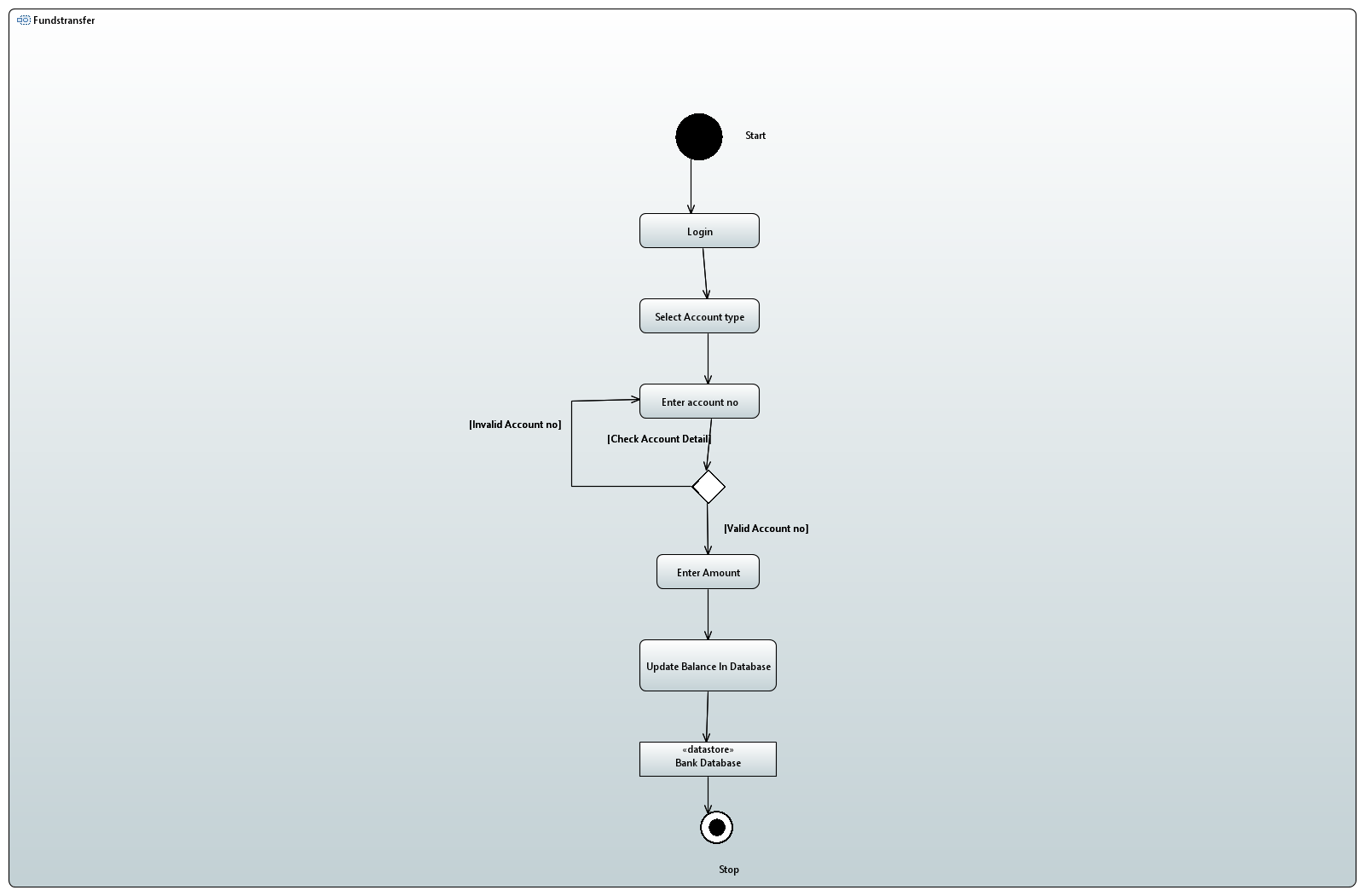
### 8.4.3 Cash Withdraw



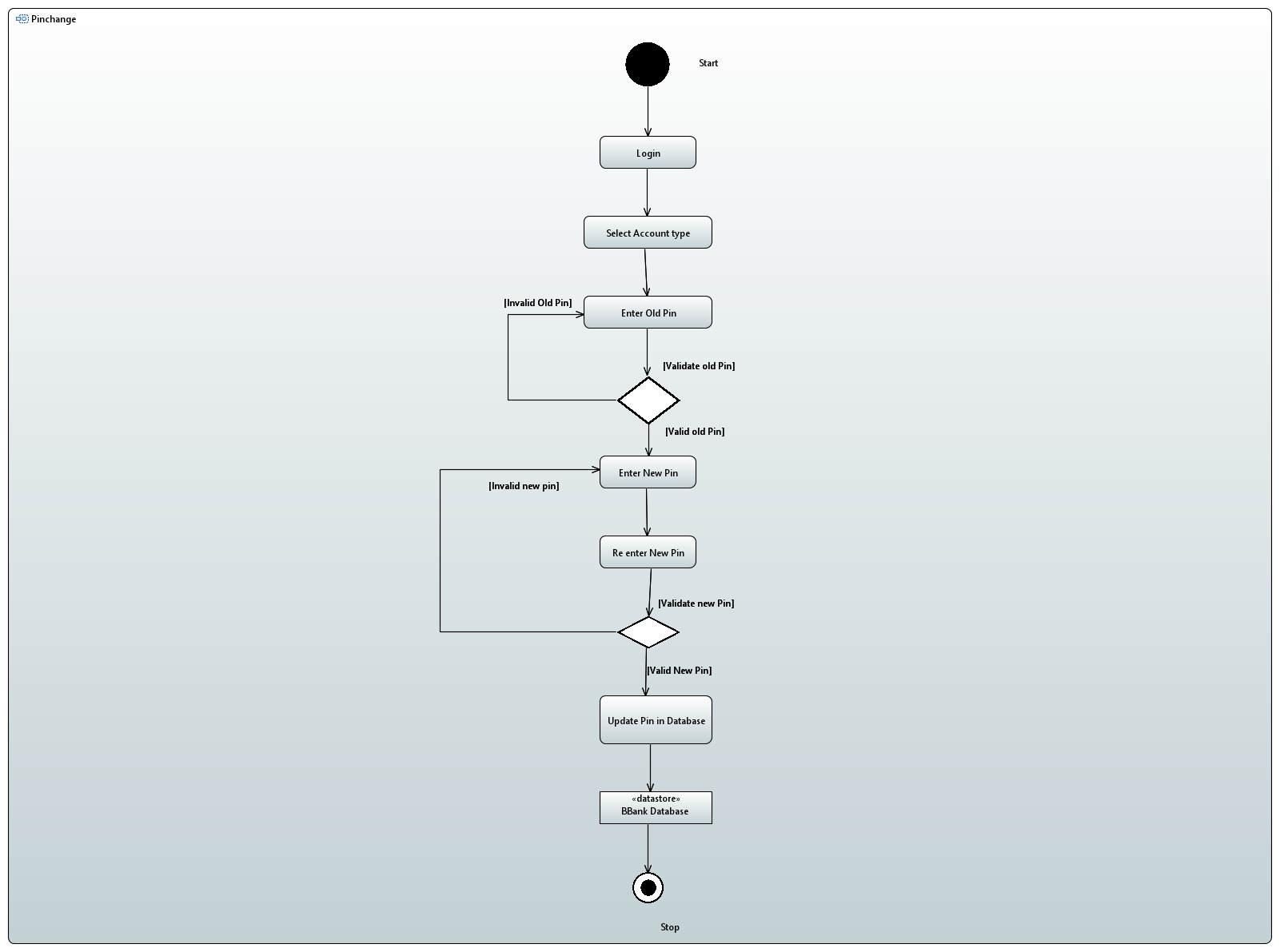
### 8.4.4 Cash Deposit



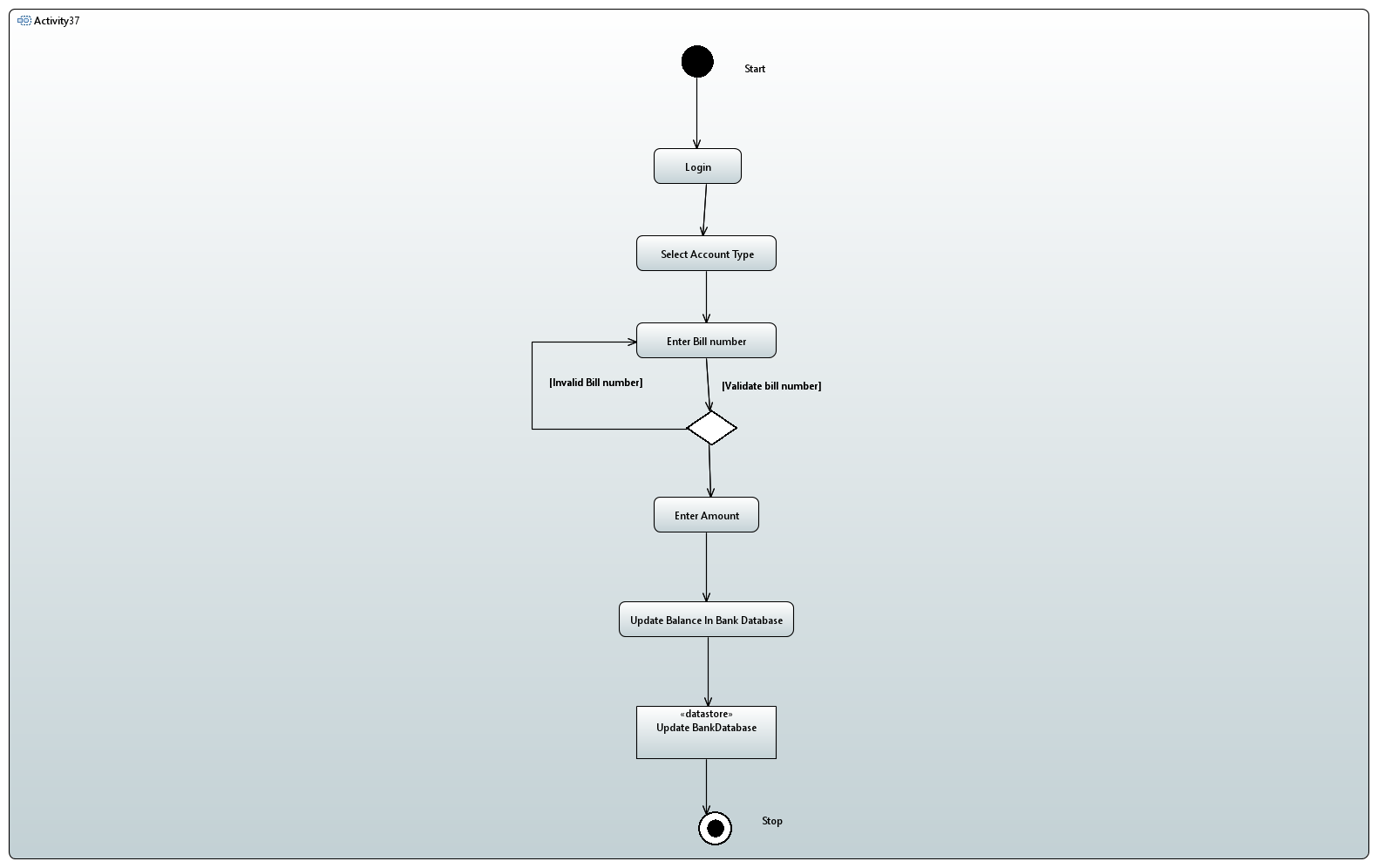
### 8.4.5 Funds Transfer



### 8.4.6 Pin Change



### 8.4.7 Electricity Bill Paid



# 9.References

## 9.1. Websites

<https://online.visual-paradigm.com>

<https://en.wikipedia.org>

<https://creately.com>

<www.javaworld.com>

<https://www.meezanbank.com>

<https://personal.utdallas.edu/~chung/RE/IEEE830-1993.pdf>

## 9.2 Books

* Software engineering body of knowledge.
* Requirements Engineering: A Good Practice Guide.
* Requirements Engineering: Processes and Techniques.

# 10. Appendices

## 10.1. Appendix A: Glossary

1. Account

* A single account in a bank against which transactions can be applied. Accounts

may be of various types with at least checking and savings. A customer can

hold more than one account.

1. ATM

* An electronic telecommunications device that enables the customers of a financial

institution to perform transactions without the need for a human cashier, clerk,

or bank teller.

1. Bank

* A financial institution that holds accounts for customers and issues account

numbers and passwords authorizing access to accounts over the ATM network.

1. Bank computer

* The computer is owned by a bank that interfaces with the ATM network and the

banks own cashier stations. A bank may have its internal network of computers to

process accounts, but we are only concerned with the one that interacts with the

network.

1. Customer

* The holder of one or more accounts in a bank. A customer can consist of one or

more people or corporations and the correspondence is not relevant to this

problem. The same person holding an account at a different bank is considered a

different customer.

1. Transaction

* A single integral request for operations on the accounts of a single customer.

We only specified that ATMs must dispense cash, accept cash deposits, and view

balances. The ATM does not give a print of these but rather displays them on the

screen.

1. Database

* A database is an organized collection of data stored on a computer.