ZZZ Bank
ATM Software System
Final Report

Corbin Kiefner
5/13/2022
EKU CSC 340
Ethics and Software Engineering

Table of Contents

I. Introduction

A. Problem Statement

A new local bank ZZZ asks for your help to develop a software system for their multiple ATM machines. The machines will provide basic operations to their customers, including depositing money, withdrawing money, checking balances, and transferring money from one account to another. Each machine will be refilled to hold $100,000 cash daily for possible withdrawals. For security reasons, each account can have at most $3000 in total for all the transactions (except for checking balance) through ATM machines each day. Effectiveness and efficiency are their primary requirements.

B. System Proposal

We propose an ATM software system that will allow ZZZ Bank to enact desired functions on customers bank accounts at the request of the customer.

II. System Description

The proposed ATM Software will allow its users to withdraw, deposit, and transfer funds. It will also allow users to check the balance of their accounts. The system will store all of this information using a MySQL Database, in which it will store transaction information alongside a daily limit function to ensure users do not exceed their daily $3,000 cash limit.

III. System Requirements

A. Functional Requirements

1. The system shall allow a user to log in.
    1. The system shall display a login menu to the user.
    2. The system shall prompt the user to enter their credentials.
    3. The user shall first enter their credit card.
    4. The system shall switch to allow the user to enter their pin.
    5. The user shall enter their pin and click the “Enter” button.
    6. The system shall validate whether the login credentials are correct.
        a. If the login credentials are correct, continue to step 7.
        b. If the login credentials are incorrect, the system shall display an error message, “Credentials invalid.” and return to step 2.
    7. The system shall log the user in, and display the main menu.
2. The system shall allow a user to deposit money.
    1. The user shall select the “Deposit” button from the main menu.
    2. The system shall display the deposit menu.
    3. The user shall select which account they would like to deposit funds into.
    4. The user shall input the funds into the machine. *(Type in the desired amount.)*
    5. The user shall select the “Enter” button.
        a. If the amount entered is $0.00, the system shall display an error, “No funds inserted!” and return to step 4.
        b. If the amount entered sets the user over their $3,000 daily limit, the system shall display an error, “Funds inserted exceed the daily limit.” and return to step 4.
        c. If the amount entered is valid, the system shall display a confirmation message, “*X.XX* funds deposited.” and move to step 6.

6. The system shall add the inserted funds to the selected accounts balance.
7. The system shall update the daily transaction limit.

3. The system shall allow a user to withdraw money.
    1. The user shall select the “Withdraw” button from the main menu.
    2. The system shall display the withdrawal menu.
    3. The user shall select the account they wish to withdraw funds from.
    4. The user shall type the amount of funds they wish to withdraw.
    5. The user shall select the “Enter” button.
        a. If the amount entered is greater than the funds in the account, the system shall display an error message, “Insufficient funds.” and return to step 4.
        b. If the amount entered is greater than the user's remaining daily limit, then the system shall display an error message, “Amount over daily transaction limit.” and return to step 4.
        c. If the amount entered is valid, the system shall display a confirmation message, “*X.XX* funds withdrawn.” and move to step 6.
    6. The system shall subtract the selected amount from the accounts balance.
    7. The system shall update the daily transaction limit.
4. The system shall allow a user to transfer funds between accounts.
    1. The user shall select the “Transfer Funds” button from the main menu.
    2. The system shall display the transfer funds menu.
    3. The user shall select the account they wish to withdraw from.
    4. The user shall select the account they wish to deposit to.
    5. The user shall type the amount of funds they wish to transfer
    6. The user shall select the “Enter” button
        a. If the amount entered is greater than the funds in the withdrawal account, the system shall display an error message, “Insufficient funds.” and return to step 5.
        b. If the amount entered is valid, the system shall display a confirmation message, “*X.XX* funds transferred.” and move to step 7.
    7. The system shall subtract the selected amount from the withdrawal accounts balance.
    8. The system shall add the selected amount to the deposit accounts balance.
5. The system shall allow a user to view the balance of a selected account.
    1. The user shall select the “View Balance” button from the main menu.
    2. The system shall display the view balance menu.
    3. The user shall select the account they wish to view.
    4. The system shall display the balance of the selected account.
6. The system shall allow the user to log out.
    1. The user shall select the “Log Out” button.
    2. The system shall log the user out and lock their information.
    3. The system shall display the login menu.

B. Non-Functional Requirements

1. The daily limit of $3,000 shall take into account only deposits and withdrawals, as they are the only two functions that add or subtract money to/from the machine itself.

2. The user should be able to return to the main menu or log out from any point in the system, allowing for ease of access.

C. Domain Requirements

1. The system must be able to access customer and account data via a MySQL Database.

IV. Use Case Diagrams

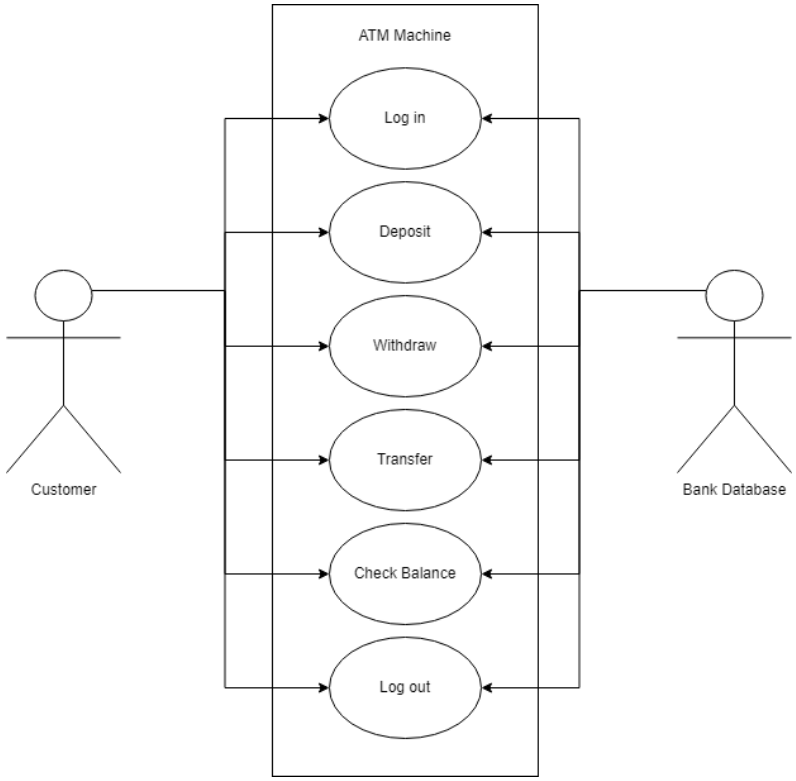

Figure 1. Use Case Diagram

The customer interacts with every function the ATM has, while the database interacts with the transactions made in order to log them.

V. Class Diagrams

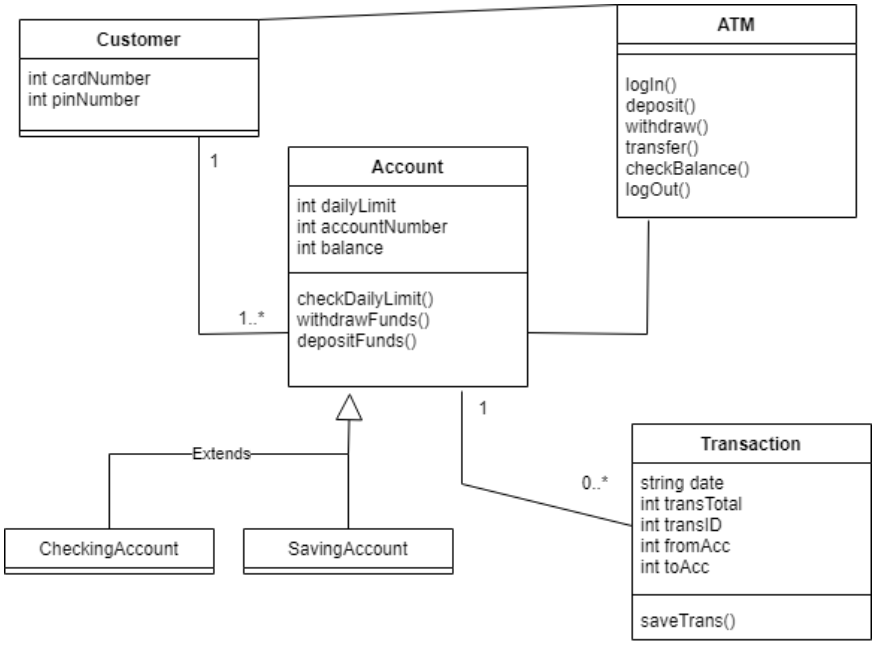

Figure 2. Class Diagram

A customer may have several accounts, checking or savings, that may enact none to several transactions. The account is displayed on the ATM for the user and interacts directly with the database via the ATM.

VI. Sequence Diagrams

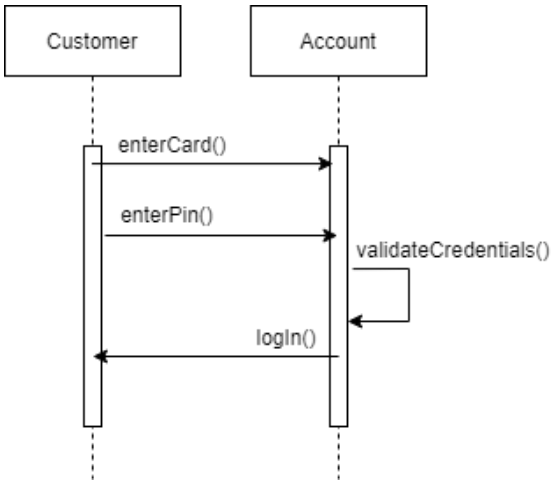

Figure 3. Login Success Sequence Diagram

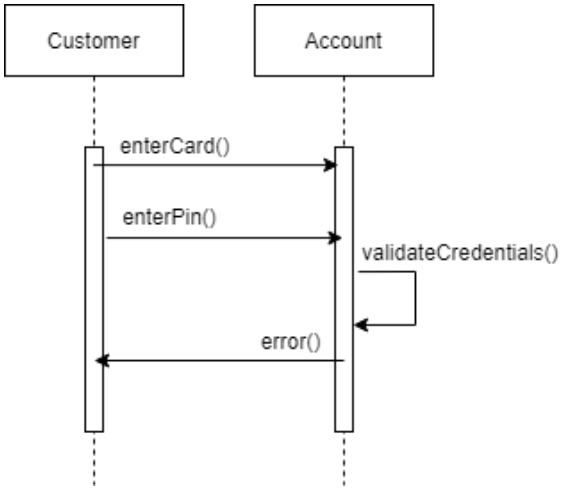

Figure 4. Login Failure Sequence Diagram

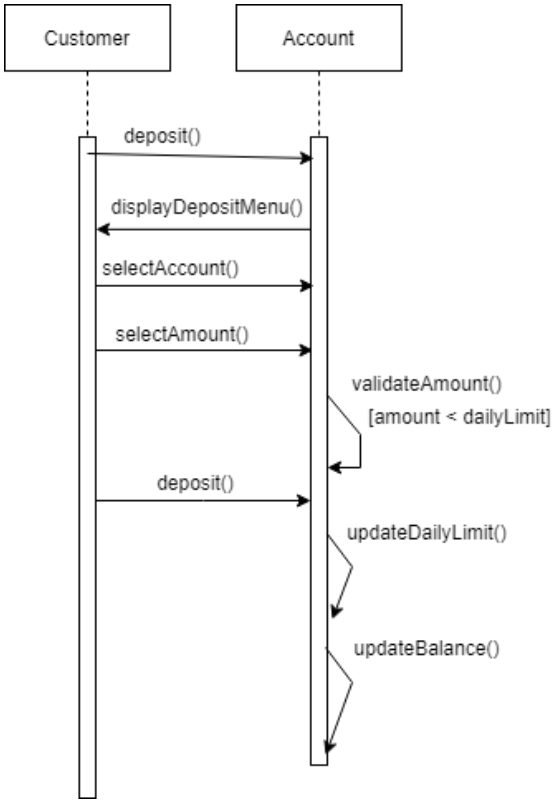

Figure 5. Deposit Success Sequence Diagram

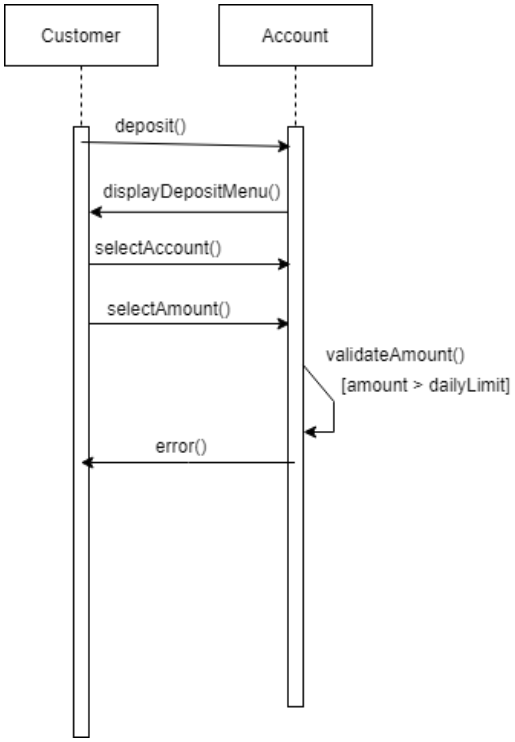

Figure 6. Deposit Failure Sequence Diagram

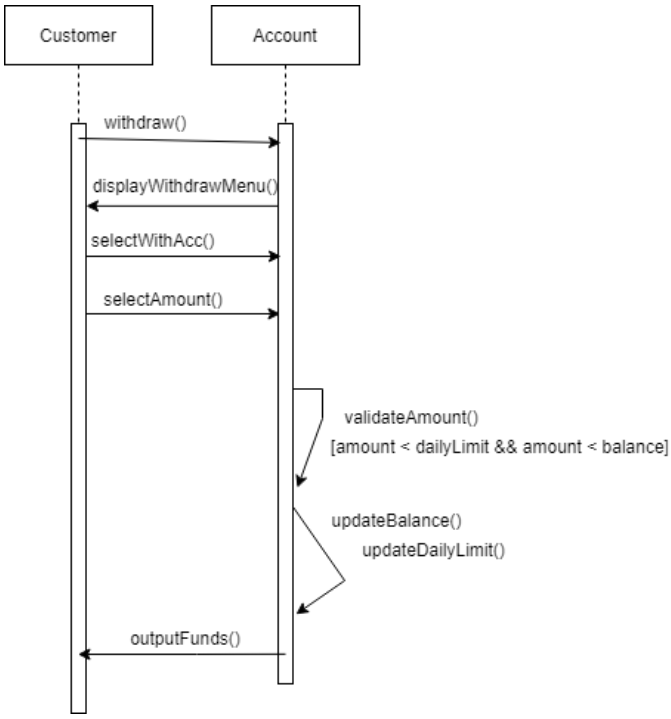

Figure 7. Withdrawal Success Sequence Diagram

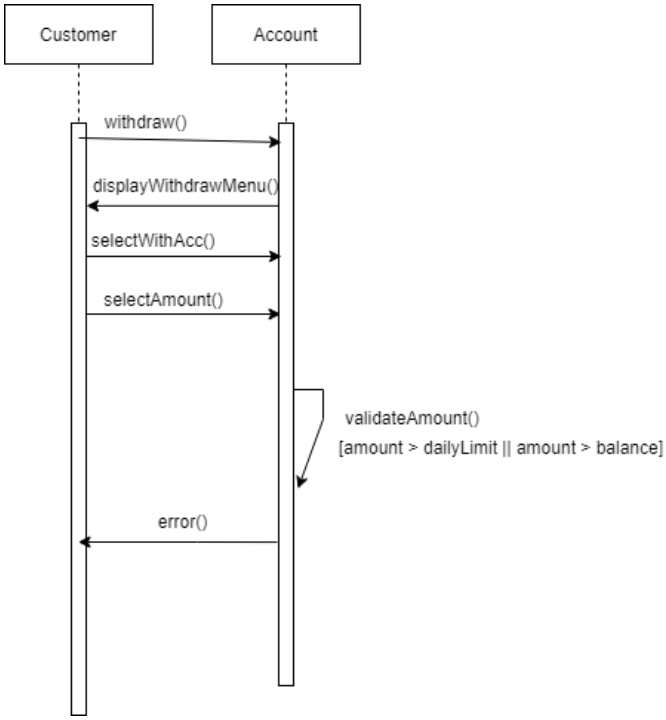

Figure 8. Withdraw Failure Sequence Diagram

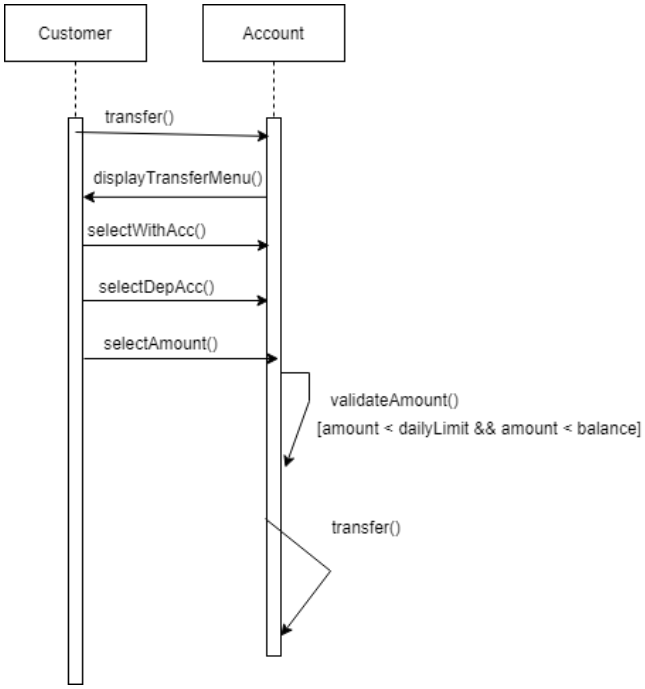

Figure 9. Transfer Success Sequence Diagram

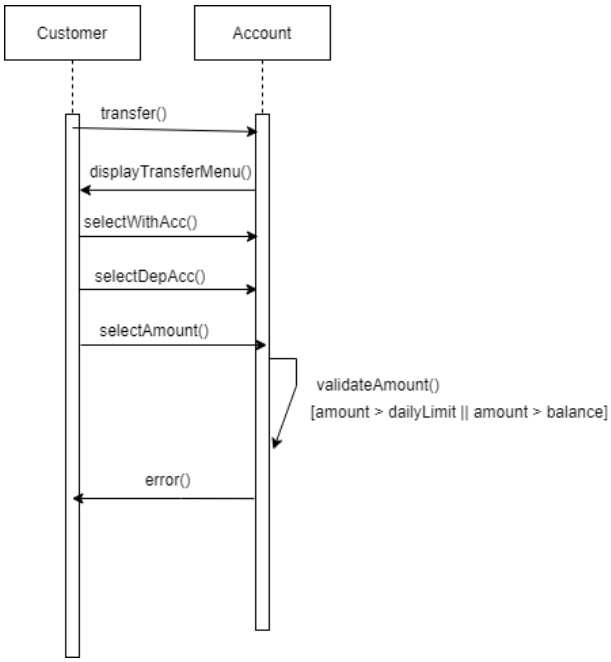

Figure 10. Transfer Failure Sequence Diagram

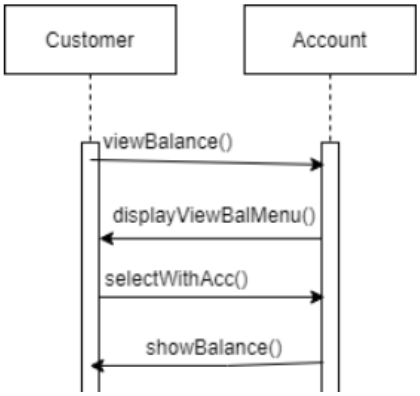

Figure 11. Check Balance Sequence Diagram

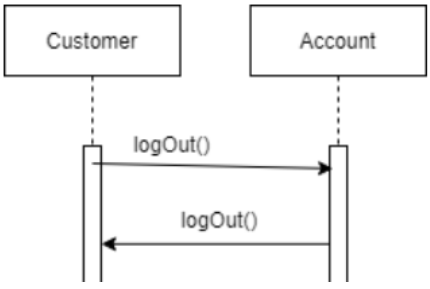

Figure 13. Log Out Sequence Diagram

VII. State Diagrams

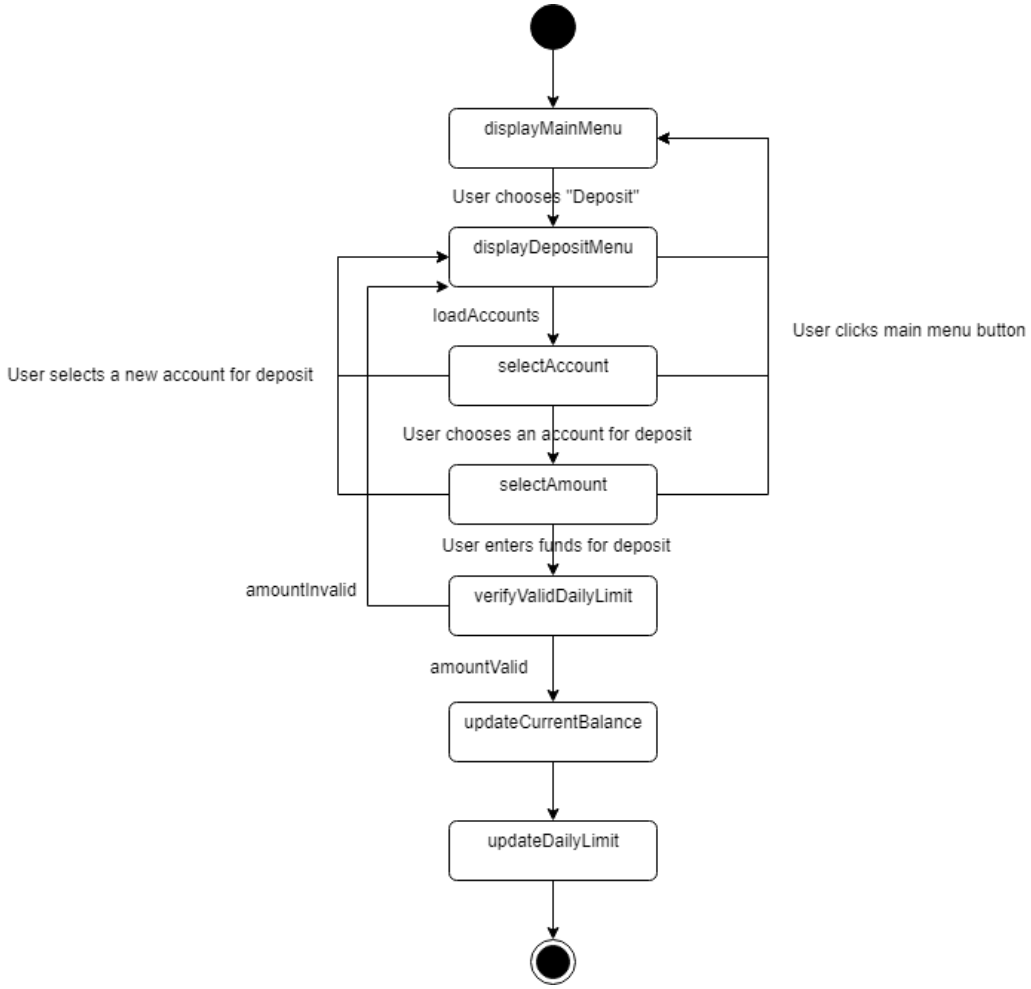

Figure 14. Deposit State Diagram

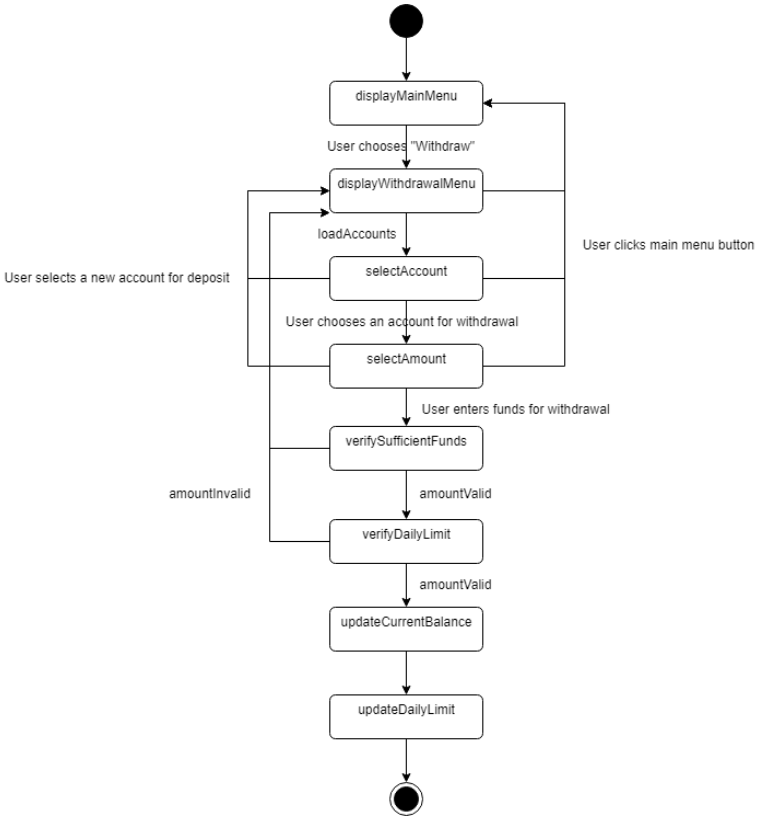

Figure 15. Withdraw State Diagram

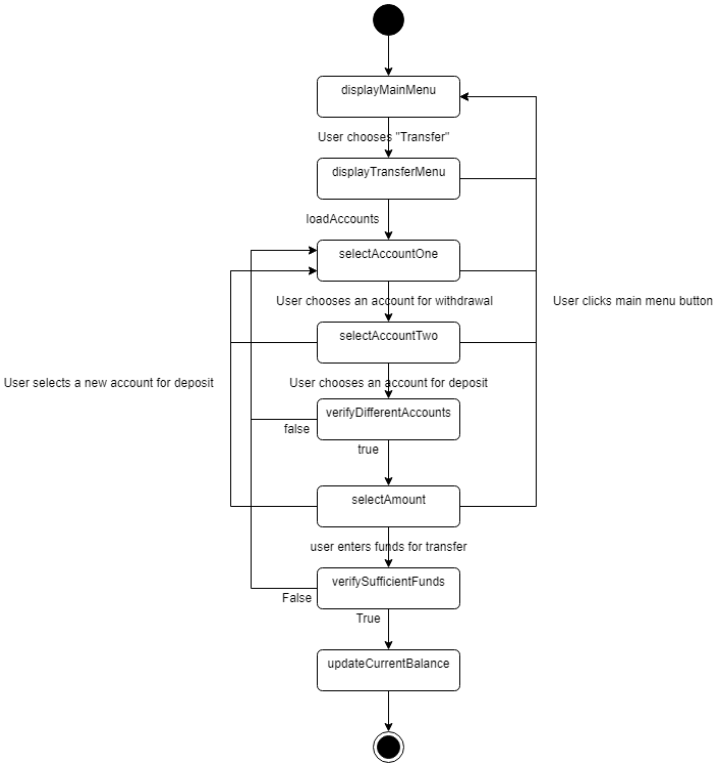

Figure 16. Transfer State Diagram

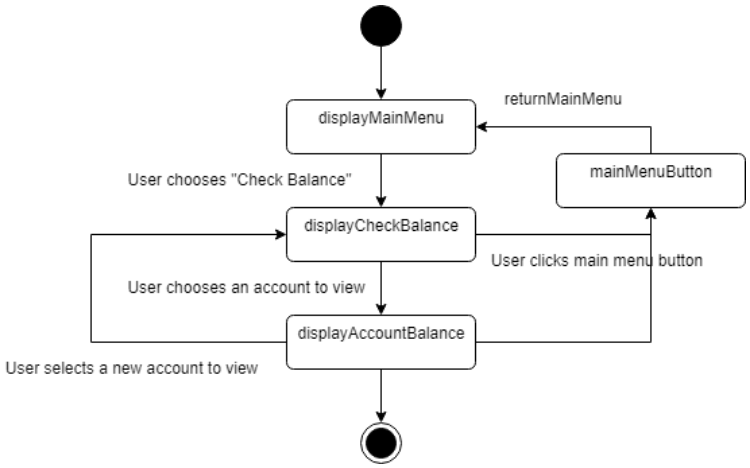

Figure 17. View Balance State Diagram

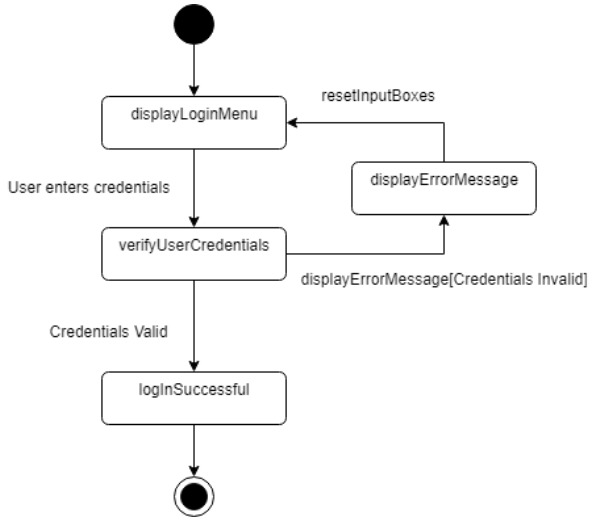

Figure 18. Log In State Diagram

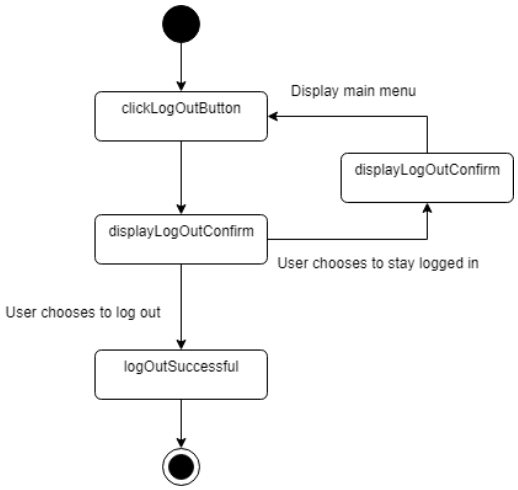

Figure 19. Log Out State Diagram

VIII. Database Design

A. ER Diagram

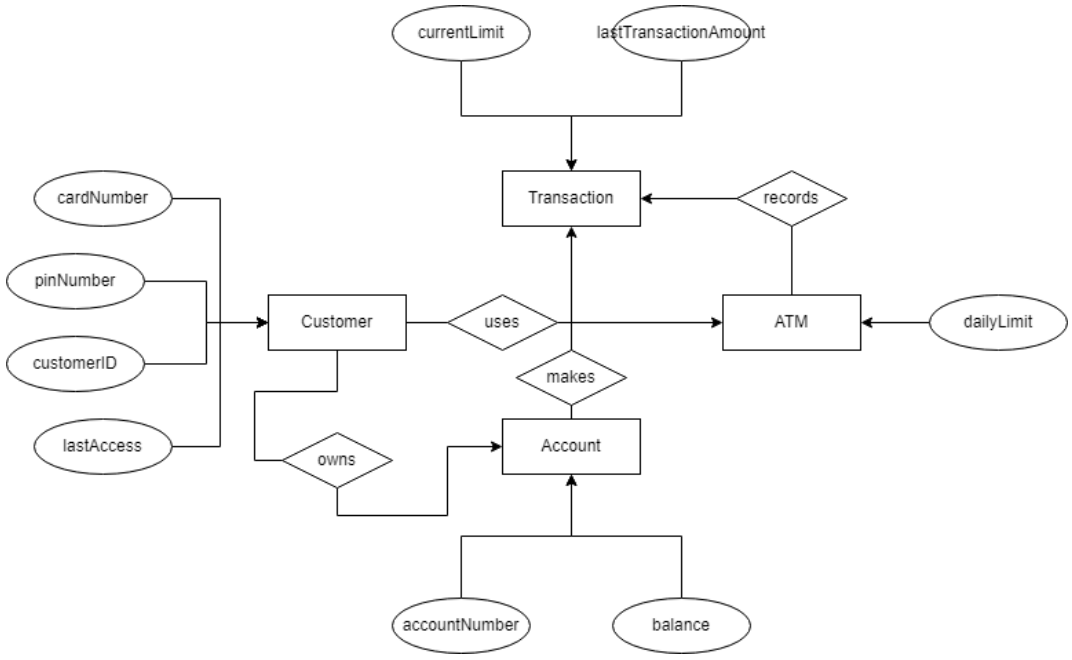

Figure 20. ATM ER Diagram

B. Table Schema

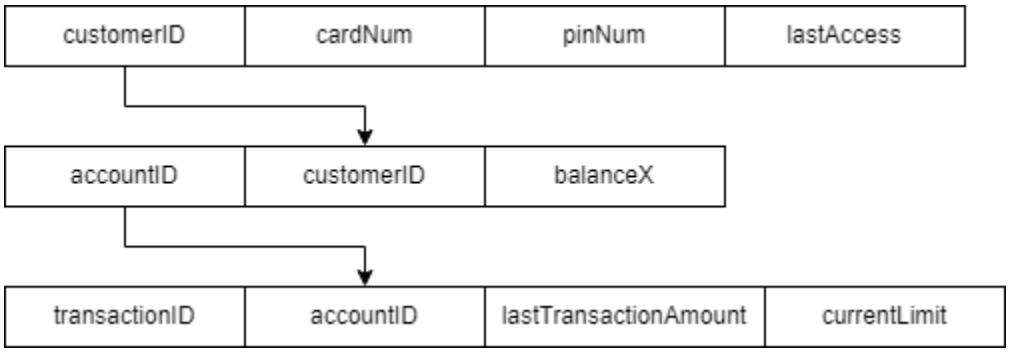

Figure 21. ATM Database Table Schema

IX. Conclusion

An ATM System was successfully created that follows all of the criteria set in the functional and non-functional requirements. The system is able to accurately retrieve data from the database and allow the customer to manipulate it through dedicated functions listed in the Use Case Diagram. The data is then accurately sent back to the database for storage upon later request.

X. Data Dictionary

Actor *- Objects that represent things that act upon systems such as employees, customers, and databases. These are what trigger use-cases.*

Class Diagram *- A diagram to represent the overview of the system. They describe the properties and relationships between systems.*
Sequence Diagram *- A sequence diagram is a flowchart that shows the order in which functions run.*

State *- A state is a function or purpose that the system is capable of doing. A system may have several states in which it can perform different tasks in different orders.*

State Diagram *- A diagram showing the different states that the system can enter and how an actor can navigate through them.*

Table Schema *- A table that represents how a database is structured in order to provide clear feedback on how to navigate it.*

Use Case *- A use case is a task or procedure that is activated by an actor, or outside input.*

Use Case Diagram *- A diagram designed to show all use cases that an actor can cause and how actors interact with the system as a whole. This is meant to show the relationship between the system and outside input.*
