SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
SCHOOL OF COMPUTING
DEPARTMENT OF COMPUTING TECHNOLOGIES
18CSE357T - BIOMETRICS - MINI PROJECT

ATM MACHINE SIMULATOR

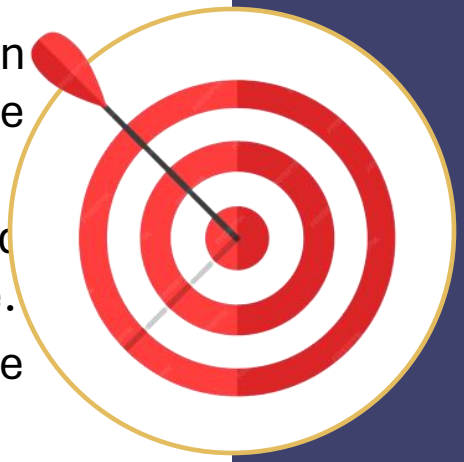
RA2211003010625
JHANVI SINGH

RA2211003010632
VARUN PRAKASH

RA2211003010630
AMAN JAIN

Abstract

- User-friendly interface with four key options: Cash Withdraw, Deposit, Check Balance, and Exit.
- **Option 1 :** Cash Withdraw allows users to specify an amount, verifies sufficient funds, and deducts the specified amount.
- **Option 2 :** Deposit enables users to input the desired amount, promptly adding it to their account balance.
- **Option 3 :** Check Balance provides immediate access to the current account balance.
- **Option 4 :** Exit gracefully closes the program with a polite farewell message.
- Simplifies cash transactions, balance checks, and deposits for user convenience.



SRM INSTITUTE OF SCIENCE & TECHNOLOGY
Deemed to be University u/s 3 of UGC Act, 1956

Introduction

Functional ATM Simulation : To create a functional shell script that simulates basic ATM operations, including cash withdrawal, deposits, and checking account balances.

User Interaction : To provide a user-friendly menu-driven interface that allows users to easily interact with the script.

Account Management : To accurately manage the account balance, ensuring that withdrawals and deposits are reflected correctly.

Usability : To ensure that the code is easy to understand and use, even for individuals with minimal scripting knowledge.

SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
Deemed to be University u/s 3 of UGC Act, 1956

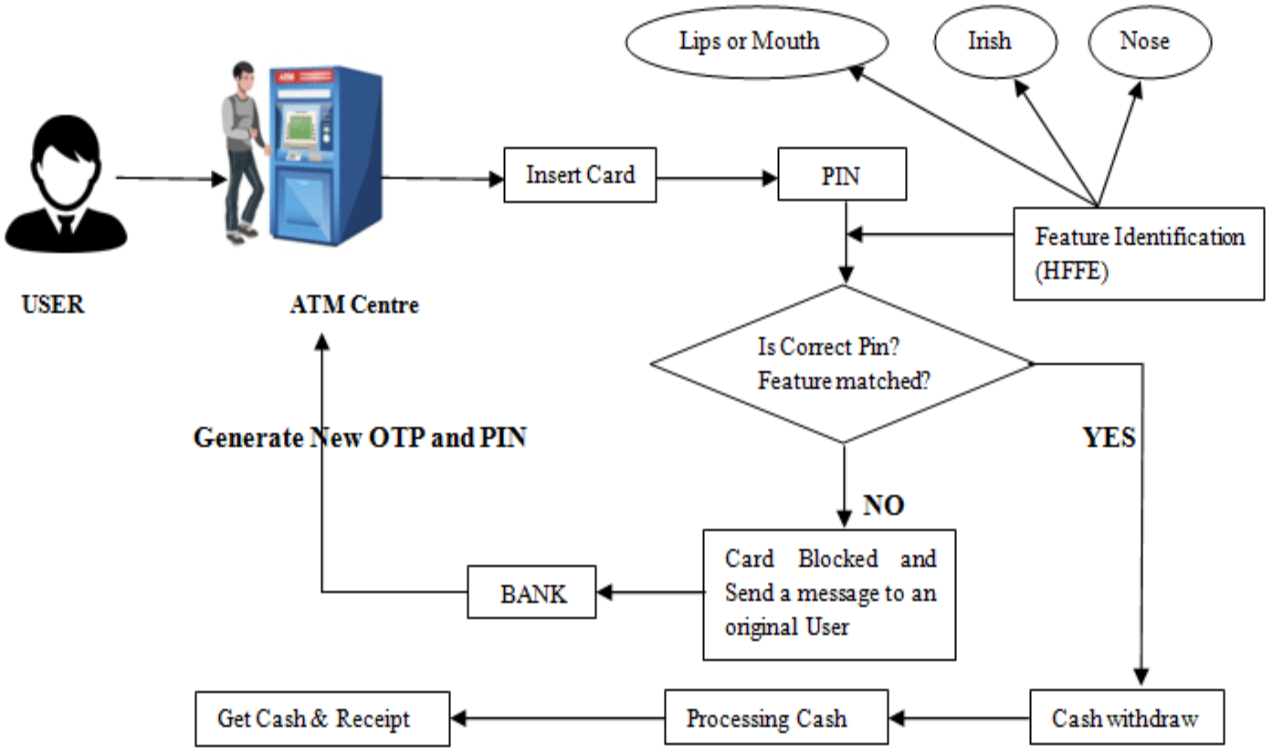
Software Requirement

The software our project is dealing with is shell scripting.

Shell scripting is like having a virtual assistant for your computer. It breaks down complex tasks into smaller, easier-to-handle steps, making it simpler to manage and control different aspects of your program. It essentially acts as a guide, allowing you to organize and automate various functions within your software project.

SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
Deemed to be University u/s 3 of UGC Act, 1956

Architecture Diagram



Modules

User Authentication Module

Validate user identity using a combination of PIN, card number, and possibly biometric information..

Account Information Module

Retrieve and display account balance, transaction history, and other account details.

Transaction Processing Module

Handle various transaction types, such as withdrawals, deposits, transfers, balance inquiries, and bill payments.

Cash Dispensing Module

Simulate the physical process of dispensing cash from the ATM. Manage the cash inventory within the ATM.

Receipt Generation Module

Generate transaction receipts for successful transactions, which can be printed or sent via email or SMS.

SRM INSTITUTE OF SCIENCE & TECHNOLOGY
Deemed to be University u/s 3 of UGC Act, 1956

The Code

```
#!/bin/sh

WITHDRAW=0;
BALANCE=5000;
DEPOSIT=0;
AMOUNT=0;


echo "Select your options...

1.Cash Withdraw

2.Deposit

3.Check Balance"


while :

do

read INPUT_INTEGER

case $INPUT_INTEGER in

        1)

echo "Enter the amount"

read AMOUNT
```

```
if [ "$AMOUNT" -gt "$BALANCE" ]; then

{

echo "Insufficient Funds"

}


else


{

BALANCE=$(($BALANCE - $AMOUNT));

echo "Your current balance is ${BALANCE}"

} fi

;;


        2)

                echo "Enter the amount"

                read AMOUNT;

                BALANCE=$(($AMOUNT + $BALANCE))
```

```
                read AMOUNT;

                BALANCE=$(($AMOUNT + $BALANCE))

                echo "Your current balance is ${BALANCE}"

;;

        3)

                echo "YOur current balance is ${BALANCE}"

;;

        *)

                echo "Sorry, I don't Understand"

;;

esac

done

done
```

SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
Deemed to be University u/s 3 of UGC Act, 1956

Output

```
Welcome to Simple ATM
1. Check Balance
2. Deposit Money
3. Withdraw Money
4. Exit
============================
Enter your choice:
1
Your current balance is: 2655377
============================
Welcome to Simple ATM
1. Check Balance
2. Deposit Money
3. Withdraw Money
4. Exit
============================
Enter your choice:
```

DISCUSSION

1. Accessibility and Convenience

The script offers a straightforward way to perform basic banking operations via a command-line interface.Users can access and execute these operations without the need for a dedicated application or graphical user interface.This simplicity can be advantageous for those comfortable with the command line and interested in managing their finances efficiently.

2. Limited Functionality

While the script provides essential functionalities like withdrawals, deposits, and balance checks, it's quite limited in its scope.A real-world banking application would require additional features such as user authentication, transaction history, and error handling, which are not present in this basic script.

3. Ease of Use

The script's menu-based approach makes it relatively user-friendly for those familiar with command-line interactions.Users can quickly select their desired operation by entering a corresponding number.

4. Error Handling

The script lacks robust error handling. For instance, it doesn't handle cases where the user inputs non-integer values.A more sophisticated script would incorporate error-checking to ensure the user provides valid input.

5. Simplicity of Code

The script's code is relatively simple and easy to understand, which can be advantageous for educational purposes or as a starting point for more complex projects.

6. Lack of Data Persistence

This script doesn't store account information or transaction history. A complete banking system would need a data storage mechanism to maintain account balances and transaction records.

7. Security Considerations:

The script doesn't address security concerns related to financial data. For a real-world banking application, it's essential to consider data encryption, access control, and secure communication protocols.

8. Future Development:

To make this script more practical and secure, future developments could involve integrating it with a database to store account information securely.Enhancements for user authentication and authorization, error handling, and transaction logging would be essential for a production-ready system.

SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
Deemed to be University u/s 3 of UGC Act, 1956

Conclusion

- User Interface & Option Handling:
- **Menu options :** Cash Withdraw, Deposit, Check Balance, and Exit.
- Each option executes a corresponding code block.
- **Option 1 : Cash Withdraw -** User specifies amount, balance check, and deduction if sufficient.
- **Option 2 : Deposit -** User inputs amount, added to account balance.
- **Option 3 : Check Balance -** Displays current balance.
- **Option 4 : Exit -** Offers goodbye message and program exit.

SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
Deemed to be University u/s 3 of UGC Act, 1956

References

1. https://onecompiler.com/
2. https://www.onlinegdb.com/
3. https://www.geeksforgeeks.com/