Instructor: Dr. pranshant Trivedi

C PROGRAMMING REPORT
ATM SIMULATOR

Name: Rupesh kumar

SAP ID: 590028036

Batch 59

Abstract

This project presents a simple ATM (Automated Teller Machine) simulator created using the C programming language. The system mimics essential ATM operations such as PIN verification, balance inquiry, deposits, withdrawals, and exiting the system. The program incorporates decision-making, loops, and modular design through user-defined functions. The goal is to simulate real-world ATM functionality while demonstrating fundamental programming skills. Testing results show that the system operates correctly under valid and invalid inputs, ensuring reliability and user-friendliness.

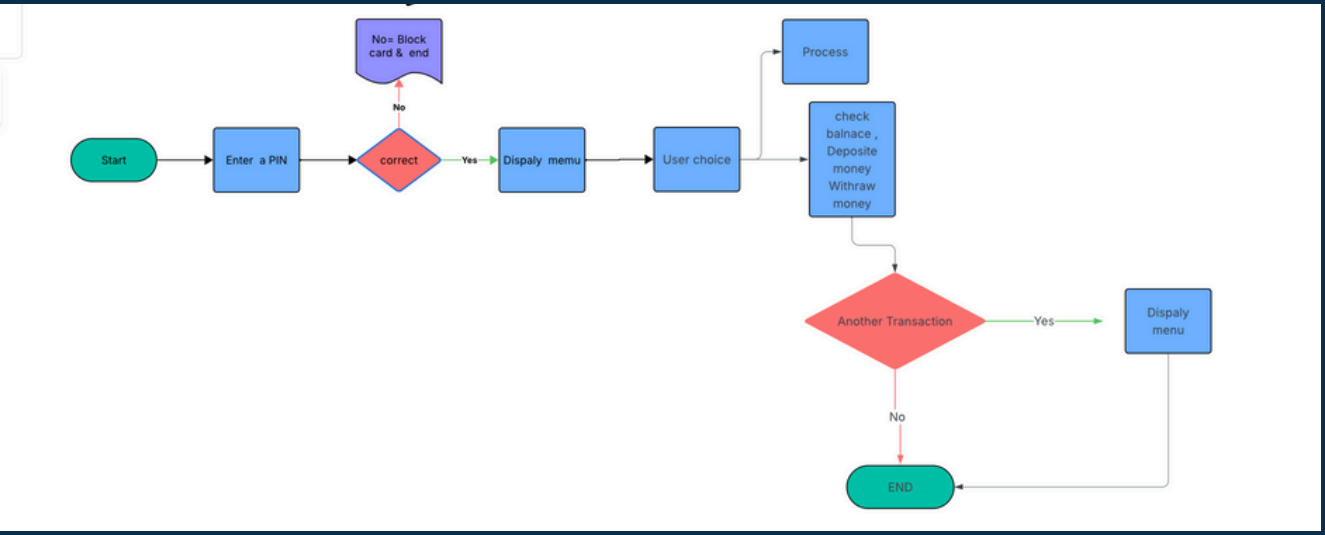
Problem Definition

TM machines serve vital roles in modern banking by providing secure and quick access to financial transactions. The problem addressed in this project is to build a simplified ATM system that performs:

- Secure PIN verification
- Balance checking
- Deposit operations
- Withdrawal operations
- User-controlled exit
- Blocking after three incorrect PIN attempts

The system must ensure correct user interaction, validate input, and handle errors such as insufficient funds or invalid amounts

System Design



Algorithms

Algorithm: PIN Verification

1. **Set correct PIN = 1234.**
2. **Allow user three attempts.**
3. **If PIN matches → proceed to menu.**
4. **If all attempts fail → display "Card blocked" and exit.**
5. **Algorithm: Deposit**
6. **Ask user for deposit amount.**
7. **If amount > 0 → balance = balance + amount.**
8. **Else → show "Invalid amount".**
9. **Display updated balance.**
10. **Algorithm: Withdraw**
11. **Ask user for withdrawal amount.**
12. **If amount ≤ 0 → invalid.**
13. **If amount > balance → insufficient funds.**
14. **Else subtract amount from balance and display new balance.**
15. **Algorithm: Menu**
16. **Display choices 1–4.**
17. **Perform selected operation.**
18. **Ask if user wants another transaction.**
19. **Loop until user exits.**

Implementation Details

The ATM Simulator program is implemented in C using a modular programming approach. The program begins by defining essential functions such as displayMenu(), checkBalance(), depositMoney(), and withdrawMoney(), ensuring the code remains organized and readable. The main function manages program control, initializes variables including the account balance and PIN, and handles the PIN verification system using a loop that limits the user to three attempts. After successful authentication, the program displays a menu from which users can select various operations. Each operation calls the corresponding function to perform tasks like checking the balance or handling money transactions. The deposit function accepts a positive amount and adds it to the balance, while the withdrawal function checks the requested amount against the available balance before deducting it. After each transaction, the system asks whether the user wants to continue. This structured and function-based approach makes the system efficient, easy to understand, and similar to real ATM functionality.

Testing & Results

The program was tested through multiple scenarios to ensure correct functionality. The PIN system worked correctly by granting access upon correct input and blocking the card after three consecutive incorrect attempts. Each menu option responded accurately: the balance inquiry consistently showed the correct amount, the deposit function updated the balance correctly for valid inputs, and the withdrawal function allowed transactions only when sufficient funds were available. Invalid inputs such as negative deposit amounts or excessive withdrawal amounts were handled with proper error messages. Overall, all components of the ATM Simulator performed as expected, confirming the success of the program.

Conclusion

The ATM Simulator project successfully demonstrates the basic operations of an ATM using C programming. It enhances understanding of loops, conditional statements, functions, and user input handling. The project provides a simple yet effective simulation of a banking environment, making it ideal for learning fundamental programming concepts.