ICS Major Project Report

ATM (Automated Teller Machine)

Team name : Classic Four

(B24CS1074) Suvarn Vijay Patil

(B24ME1018) Brijesh Thakkar

(B24CM1027) Harsh Harish Chaudhari

(B24BB1010) Faiz Raise

Overview

This report documents the design, functionality, and implementation of a simulated ATM system. The system replicates the basic features of an ATM, including cash withdrawal, cash deposit, balance check, PIN change, and transaction logging. The system uses C programming language and handles file-based data storage for user balance and PIN.

Objective

The primary goal of this ATM system is to provide an easy-to-use simulation of a real-world ATM, with the added features of logging transactions and providing a typewriter effect for a more realistic experience. It simulates typical user operations like depositing cash, withdrawing money, and checking balances while maintaining security via PIN verification.

Functionalities

1. User Authentication

- PIN Verification: The system prompts the user to enter a 4-digit PIN to authenticate access. There are 3 attempts allowed before the system exits. If the user enters the correct PIN, they are granted access to the ATM menu.

2. ATM Menu Options

After successful PIN authentication, the user is presented with the following menu options:

- Cash Withdrawal: Withdraw funds from the account while respecting a daily withdrawal limit of 2500.00.
- Cash Deposit: Deposit funds into the user's account.
- Check Balance: View the current balance in the account.
- Change PIN: Modify the PIN for additional security.
- Exit: End the session and exit the program.

3. Balance Management

- Account Balance: The system loads the balance from a file called balance.txt. If the file doesn't exist or is empty, the system defaults to an initial balance of 10,000.

- File Operations: Balance changes are stored back in the balance.txt file after each transaction.

4. PIN Management

- PIN Storage: The system stores the user's PIN in a file called pin.txt. If the file doesn't exist or is empty, the system uses a default PIN of 0000.
- PIN Change: Users can change their PIN through the "Change PIN" option in the menu. The new PIN is confirmed before being saved.

5. Transaction Logging

Every transaction (deposit, withdrawal, balance check, PIN change) is logged with details such as the transaction type, amount, balance after the transaction, and timestamp. The log is stored in a file called transaction_log.txt.

6. Visual Effects

- Typewriter Text Effect: To enhance the user experience, text is printed character by character, simulating a typewriter effect using the Typewriter_Text() function.

System Design

1. File Handling

The system uses file handling to store and retrieve the user's balance and PIN:

- balance.txt: Stores the user's account balance.
- pin.txt: Stores the user's 4-digit PIN.
- transaction_log.txt: Stores all transaction logs, including date, time, transaction type, amount, and balance.

2. Data Types

- Balance: A floating-point value that holds the account balance.
- PIN: A string of 4 characters representing the user's PIN.
- Transaction Logs: A textual record of every interaction made by the user.

3. Error Handling

- If a file (e.g., balance.txt or pin.txt) cannot be opened, the program uses default values for balance and PIN.
- Invalid input, such as entering a non-numeric value for a transaction, is handled through user prompts, ensuring that the system does not crash.
- If the user enters an incorrect PIN too many times, the system exits gracefully.

Security Measures

- PIN Authentication: Every transaction, except for checking the balance, requires the user to authenticate with their 4-digit PIN.
- Daily Withdrawal Limit: The user can only withdraw a maximum of 2500.00 per day. Any attempt to exceed this limit will result in an error message.

- Logging: All transactions are logged for security and auditing purposes.

System Flow

1. Startup: When the user runs the program, they are prompted to enter their 4-digit PIN.
2. PIN Verification: The system checks the entered PIN against the one stored in pin.txt. If it matches, the user proceeds to the ATM menu; otherwise, the system exits after three failed attempts.
3. ATM Menu: The user is shown the ATM menu and selects an option. After each operation (withdrawal, deposit, balance check, PIN change), the system updates the balance, logs the transaction, and asks if the user wants to continue or exit.
4. Exit: If the user selects the "Exit" option or fails to authenticate within three attempts, the program exits.

Code Structure

Main Program

The main program initializes the system by loading the balance and PIN. It then enters a loop, where the user can select from the available menu options. Based on the user's choice, the corresponding function is executed, and the system either continues or exits based on user input.

Helper Functions

Several helper functions handle the core functionality, including balance and PIN management, transaction logging, and user interaction. These functions encapsulate individual operations to keep the main program flow clear and organized.

Conclusion

The ATM system implemented here provides a simple but effective simulation of an ATM's core functions. The system ensures secure user authentication, keeps track of transactions, and provides realistic user feedback through a typewriter effect and sound notifications. This project demonstrates the practical use of C for simulating real-world applications, including file handling, user input validation, and transaction logging.