

ATM-Simulator Project

Presented by -Prakhar Goyal
SAP ID:590027648
Batch-59
Semester-1
Course Name-Programming In C
SOCS,UPES



Introduction:

This project on Automated Teller Machine(ATM) is made using C Programming language.

This is a simulation of real ATM,it allows a single user to perform various everyday needed operations like ,logging in via a pin(with max 3 wrong attempts for Security feature),checking account balance,depositng and withdrawing money,changing atm pin,changing username,prinntng receipt and mini transaction history (on screen) using file handling concepts.

Objectives

- To build a simple and easy to use ATM Simulator in C that feels similar to a real ATM experience.
 - To apply the concepts taught by our professor in the semester
- To allow the user to perform everyday ATM operations tasks like logging in an ATM with secure PIN,checking current balance,deposit and withdraw money,change PIN,etc
- To use the concepts of file handling so the ATM can remember acc details even after the program is terminated.
 - To use the concept of modular programming approach and breaking the code into clean and reusable functions that are easy to understand.
 - To get a clearer idea how real ATM machine work behind the scene
- To implement a basic easy security mechanism (3 wrong PIN attempts) to make ATM feel more like real ATM.
 - To improve my understanding and logic building by handling different types of user input cases.

Problem Definition

- In everyday life,ATM is used by people everyday but very few know what happens inside the machine's software, how it updates the account balance and info.
- There is a need to simulate how an ATM performs tasks like login,transactions and error handling
 - The system must allow secure login using a 4 digit PIN with only 3 attempt allowed.
- It should support basic banking operations like checking balance,deposit and withdraw money,change PIN,username,Mini Transaction History.
 - User data must be stored in a file using file handling .
 - The system must handle invlalid inputs and maintain errorless operations
 - The interface must be like real ATM

Solution Approach

- Used structure to store account details.
- The program loads saved account data using the function loadFromFile() and creates a default account if no file exists.
- Used a secure login system which verifies 4-digit with a limit of 3 attempts.
- After the user login, a menu driven system (with loop and switch case) allows the user to choose different ATM tasks and operations.
- Each operation is implemented by using **modular functions** approach like depositMoney(), withdrawMoney(), changePIN(), checkBalance(), etc.
- Files (**account.txt** and **history.txt**) are used to permanently store account and transaction history even after the program termination implementing File Handling concepts.
- A reset function that is used to clear all data and restoring the details to a preset default values for testing and demonstration purposes.
- All inputs are validated (wrong PIN, invalid amounts and empty names and buffer cleaning) to check for smooth and error free execution. Buffer Cleaning using fgets() along with NULL, also used fgetc().

Algorithm of main() function

1. Start the program
2. Load the saved account data; if file not found it creates a default account
3. Initialize attempt counter=0
4. Loop until correct PIN entered by user or attempts =3
 - i. Ask for 4 digit PIN
 - ii. Validate numeric input
 - iii. If PIN matches then show welcome and break
 - iv. else it increases attempts count and show attempts left
5. If attempts reached 3 then show that Account locked and exit the program
6. Loop until user selects Exit
 - a. Read menu choice
 - b. Call the required function
 - i. checkBalance()
 - ii. depositMoney()
 - iii. withdrawMoney()
 - iv. changePIN()
 - v. showHistory()
 - vi. changeName()
 - vii. resetAccount()
 - c. Handle invalid choices
7. On exit show Thank You message
8. End

Scope of The Project

- Simulates essential ATM functions for everyday banking
 - Allows secure login using a PIN with limited attempts
 - Supports balance check, deposit, withdrawal, PIN change, name update and reset account
 - Provides a mini transaction feature
 - Maintains data using file handling for storage
- Focusses on reusability of functions and understanding C programming concepts taught to us

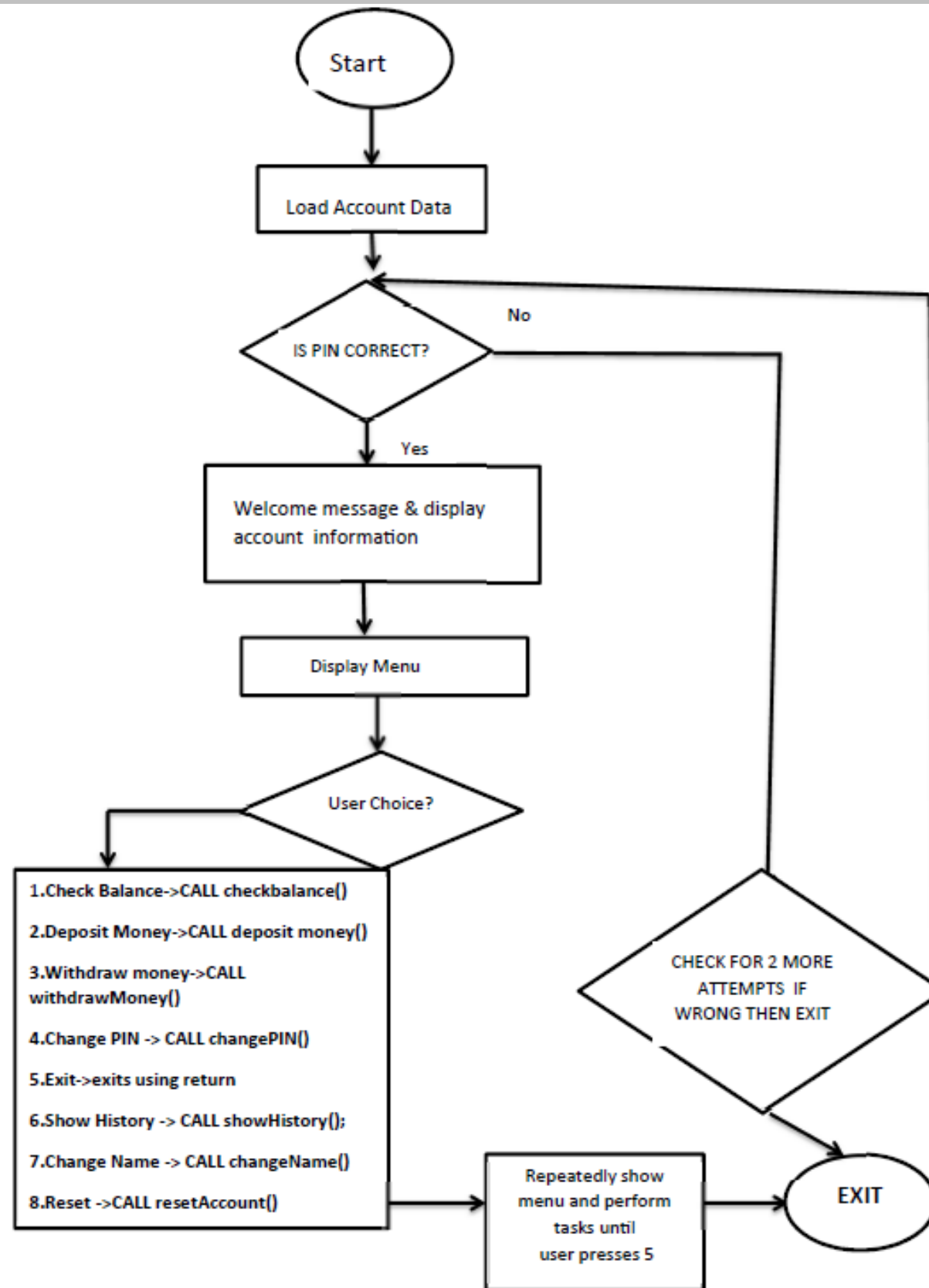


Major Functions Used

- `checkBalance()`-displays current account balance
- `depositMoney()` -adds money to balance and updates history
- `withdrawMoney()`-it deducts the amount and checks sufficient balance
- `changePin()`-updates PIN with confirmation
- `changeName()`-used to update account holder's name
- `showHistory()`-to show mini transaction history
- `saveToFile()`-to save account data permanently to account.txt file
- `loadFromFile()`- to load the saved data and if not present it loads default values
- `resetAccount()`-clears data of account.txt & deletes all transactions and resets to default



A short Flowchart of main() function



TESTING

Test Case	What I tested	What I expected to Happen	What Actually Happened	Result
1.Correct PIN Entry	PIN=correct value	Login successful	Login successful	Pass
2.Incorrect PIN Entry	PIN=wrong value (3 times)	Access Denied after 3 attempts	Access denied	Pass
3.Deposit Valid Amount	Deposit=500	Balance increases by 500	Balance updated correctly	Pass
4. Deposit Invalid Amount	Deposit=-50 or 0	Show error message	Error shown	Pass
5.Withdraw Valid amount	Withdraw= amount<= balance	Balance reduced correctly	Works as expected	Pass
6.Withdraw More than Balance	Withdraw> balance	Show insufficient balance	Error Shown	Pass
7.Check Balance	Select balance option	Display current balance	Balance displayed	Pass
8.Change PIN(valid)	Enter new Valid PIN	PIN updated	PIN updated	Pass
9.Change PIN (invalid)	Entering mismatched pin in retyping pin	Should show error	Error message shown	Pass
10.Save Data to file	Select save option	File should be created/updated	File saved successfully	Pass
11.Load Data from file	Select Load option	Data should load correctly	Data loaded successfully	Pass
12.Mini Transaction History	Performing few operations	Should show last few transactions	Correct list displayed	Pass
13.Change account holder name	Entering a new valid name	Should update the name when a valid name is entered and save to file	Name updated to new name and saved in account.txt	Pass
14.Reset Account	Select Reset option	Balance and PIN reset to default	Reset successful	Pass
15.Exit	Selecting exit option from main menu	Program shouls terminate gracefully with a thank you message	Program exited successfully and displayed expected message	Pass

Some screenshots of the outputs (1)

Test case-1 screenshot : Correct PIN Entry

```
- Welcome to State Bank ATM -  
Enter your 4 digit pin: 1234  
Welcome, rohan  
Login successful.  
Bank:State Bank of India  
Branch:Dehradun Main Branch  
Account Type:Savings  
  
-- MENU --  
1.Check Balance  
2.Deposit Money  
3.Withdraw Money  
4.Change PIN  
5.Exit  
6.Transaction History  
7.Change Account Holder Name  
8.Reset Account  
Enter choice: █
```

Explanation: The user entered the correct PIN .The system verified it successfully and displayed the welcome message along with account details.

Some screenshots of the outputs (2)

Test case-2 screenshot : Incorrect PIN Entry

```
- Welcome to State Bank ATM -  
Enter your 4 digit pin: 2025  
Wrong pin entered...Attempts left: 2  
Enter your 4 digit pin: 2024  
Wrong pin entered...Attempts left: 1  
Enter your 4 digit pin: 2023  
Wrong pin entered...Attempts left: 0  
You tried wrong pin as too many times.  
Account locked for safety.  
PS C:\Users\pc\Desktop\C lang project\C_Project> |
```

Explanation: The user entered an incorrect PIN three times. After the third attempt, the system denied access and displayed an account lock warning.

Some screenshots of the outputs (3)

Test case-5 screenshot : Withdraw Valid amount

Explanation: The user a valid amount within the available balance. The balance decreased accordingly and a receipt was optionally displayed.

```
- Welcome to State Bank ATM -  
Enter your 4 digit pin: 1234  
Welcome, rohan  
Login successful.  
Bank:State Bank of India  
Branch:Dehradun Main Branch  
Account Type:Savings  
  
-- MENU --  
1.Check Balance  
2.Deposit Money  
3.Withdraw Money  
4.Change PIN  
5.Exit  
6.Transaction History  
7.Change Account Holder Name  
8.Reset Account  
Enter choice: 3  
Enter amount to withdraw: 500  
Please collect Rs 500.00  
Do you want a receipt?(y/n):y  
  
----Receipt-----  
Transaction :withdrawal  
Amount:Rs 500.00
```

Some screenshots of the outputs (4)

Test case-7 screenshot : Check Balance

Explanation: The User selected the Check Balance option and the current was displayed correctly.

```
- Welcome to State Bank ATM -  
Enter your 4 digit pin: 1234  
Welcome, rohan  
Login successful.  
Bank:State Bank of India  
Branch:Dehradun Main Branch  
Account Type:Savings
```

```
-- MENU --  
1.Check Balance  
2.Deposit Money  
3.Withdraw Money  
4.Change PIN  
5.Exit  
6.Transaction History  
7.Change Account Holder Name  
8.Reset Account  
Enter choice: 1
```

```
Your balance: Rs 10500.00
```


Working Code Demonstration

- Run ATM program live on VS code
- Key Functions to demonstrate
 - ☐ withdrawMoney() and DepositMoney()
 - ☐ checkBalance(),resetAccount(),changePIN(),changeName()
 - ☐ Login/PIN validation
- Input/Output behaviour

```
PS C:\Users\pc\Desktop\C lang project> cd "c:\Users\pc\Desktop\C lang project\C_Project\" ; if ($?) { gcc atm_simulator.c -o atm_simulator } ; if ($?) { .\atm_simulator.exe }

- Welcome to State Bank ATM -
Enter your 4 digit pin: 1234
Welcome, rohan
Login successful.
Bank:State Bank of India
Branch:Dehradun Main Branch
Account Type:Savings

-- MENU --
1.Check Balance
2.Deposit Money
3.Withdraw Money
4.Change PIN
5.Exit
6.Transaction History
7.Change Account Holder Name
8.Reset Account
Enter choice: 2
Enter amount to deposit: 500
Deposited Rs 500.00
Do you want a receipt?(y/n):y

----Receipt-----
Transaction :deposited
Amount: Rs 500.00
New balance:Rs 11000.00
```

Novelty and Contributions

- I added a Reset Account feature that clears all the data and transaction history ,and put default values for easy testing.
- I also with help of file handling implemented a Mini Transaction History using history.txt,similar to a real ATM mini statement.
- I added an optional receipt printing for deposits and withdrawal to make it feel real life based.
- I used file handling so ATM remembers the user detail even after prog. terminates.
- I added strong input validation like invalid amount checks and PIN mismatching handling.
- Included low balance warnings on screen
- Used modular functions for clear and easy to understand code.





Conclusion

- This ATM simulator helped me understand real-world banking systems
 - Breaking the program into reusable functions emphasized modular programming
 - I implemented features: balance check, deposit/withdrawal, PIN change, mini transaction history, file handling and reset account
 - Tried to apply all concepts taught so far
- This project boosted my confidence and showed how programming solves real-life problems
- Thanks to our Professor Dr. Prashant Trivedi for his continuous guidance and support.

References:

- Let Us C Book by Yashavant Kanetkar
- Class materials and lecture notes provided by my Professor
- GeeksforGeeks



Appendix:code of some functions

```
void saveToFile(struct account acc)
{
    FILE *fp = fopen("account.txt", "w");
    if (fp == NULL)
    {
        printf("Error saving data!\n");
        return;
    }
    //printing information to file account.txt
    fprintf(fp, "%s\n%d\n%.2f\n%s\n%s\n%s\n",acc.name,acc.pin,acc.balance,acc.bankName,acc.branch,acc.type);
    fclose(fp); //closing file
}
```

```
void checkBalance(struct account acc)
{
    printf("\nYour balance: Rs %.2f\n", acc.balance);//printing acc balance
    if (acc.balance < 1000)
    { printf("\n WARNING:Your balance is very low.Please deposit soon.\n");
    }
}
```

“All other test cases and full source code are available in the project report.”