



UNIVERSITY OF PETROLEUM ENGINEERING AND STUDIES

PROJECT REPORT ON BANK ACCOUNT MANAGEMENT SYSTEM IN C

Submitted By:

Name: Ohmsai Nirmalkar

Roll Number: 590025872

Submitted To: Mohsin Dar

Subject: Programming in C

Date: 2 December , 2025

ABSTRACT

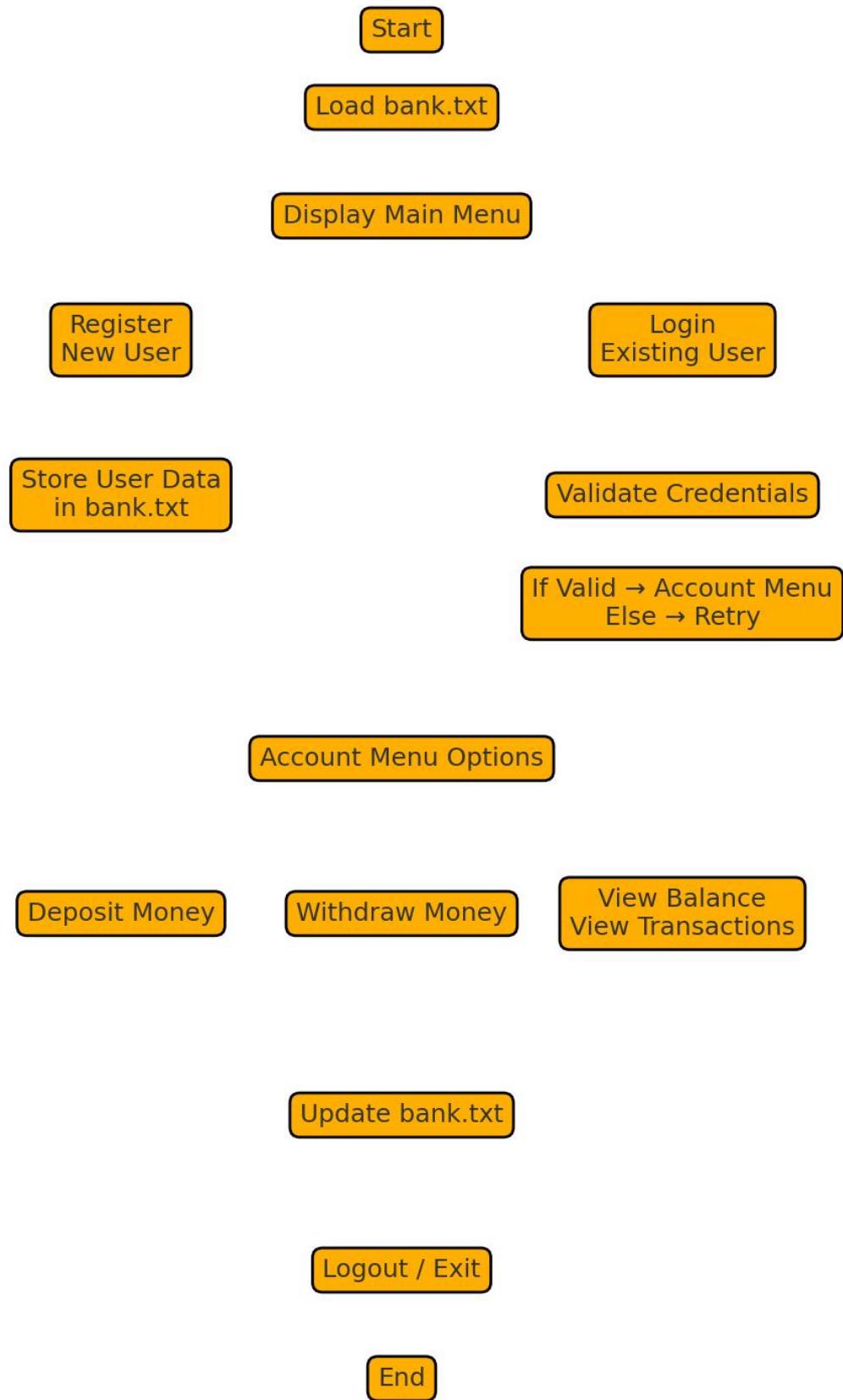
This project presents a comprehensive Bank Account Management System implemented in the C programming language. The system is designed to allow users to register securely, authenticate through password-based login, and perform essential operations like depositing funds, withdrawing funds, checking account balance, and accessing complete transaction history. All user information is securely stored in a structured plain-text file for ease of readability. The project demonstrates fundamental concepts of modular programming, file handling, data structures, and menu-driven program design.

PROBLEM DEFINITION

The primary objective is to create a secure and reliable banking system simulation in C that supports multiple user accounts and provides authenticated access. The system must ensure data privacy, integrity, and correct transaction handling. Additionally, it should demonstrate practical implementation of structures, functions, and file handling in C programming.

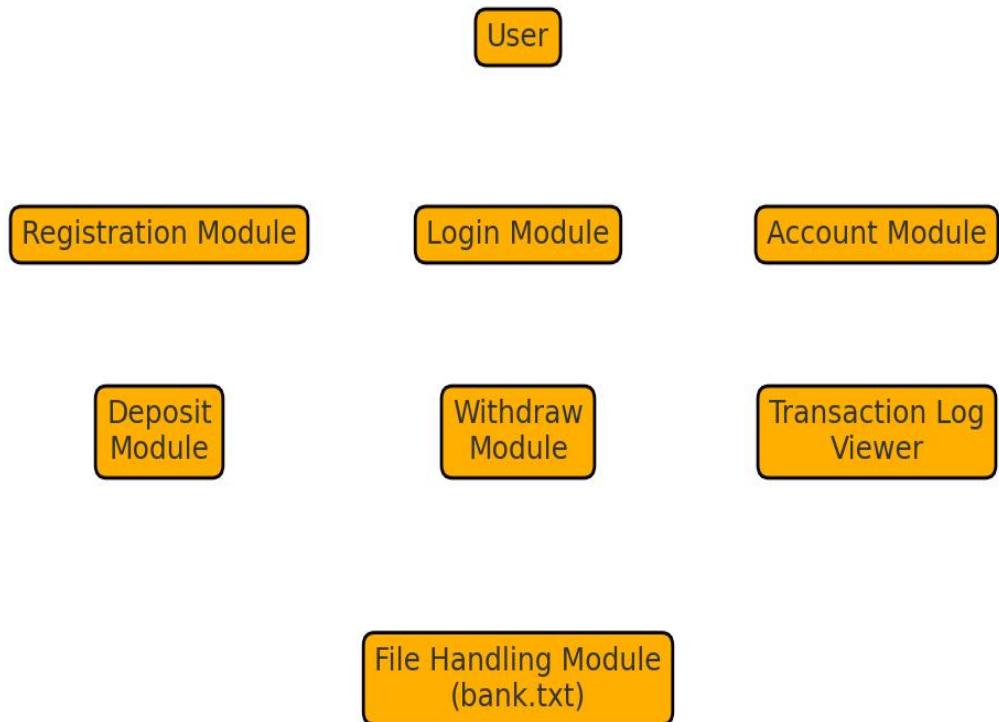
FLOWCHART

A detailed flowchart representing the entire workflow of the banking system is shown below:



SYSTEM ARCHITECTURE DIAGRAM

The system architecture provides a clear overview of how modules interact with each other:



DATA FLOW DIAGRAM (LEVEL 0)

This DFD Level 0 diagram provides a high-level flow of information between the system and the data store:

User

Bank System
(Program)

Operations:
Register, Login,
Deposit, Withdraw,
View Balance,
View Transactions

bank.txt
(Data Store)

ALGORITHMS

The following are the key algorithms used in this project:

1. **User Registration Algorithm***

- Input user details.
- Validate email uniqueness.
- Save user data to file.

2. **Login Algorithm***

- Ask user for email and password.
- Validate credentials with stored data.
- Grant or deny access.

3. **Deposit Algorithm**

- Input deposit amount.
- Add to user's balance.
- Save updated balance to file.

4. **Withdraw Algorithm**

- Input withdrawal amount.
- Check if sufficient balance exists.
- Deduct and update file.

5. **Transaction Log Algorithm**

- Timestamp each transaction.
- Save details in structured format.

IMPLEMENTATION DETAILS

The system is implemented using C programming concepts such as structures, functions, loops, conditional statements, arrays, and file handling using text files. Below are the main modules developed in the implementation:

- **User Module** — Handles registration, login, and profile verification.
- **Transaction Module** — Manages deposits, withdrawals, and log creation.
- **File Handling Module** — Reads and writes all data to bank.txt.
- **Menu Controller Module** — Provides user interface for operations.

TESTING AND RESULTS

Multiple test cases were executed to ensure smooth functionality. The key results include:

- Successful creation of multiple user accounts.

- Proper verification of login credentials.
- Accurate balance updates after deposit and withdrawal.
- Correct chronological transaction log storage.
- Stability during repeated operations.

OUTPUTS

1) Registering a new bank account.

```
vboxuser@Ubuntu1404:~/Bank$ gcc *.c -o bank
vboxuser@Ubuntu1404:~/Bank$ ./bank
Loaded 1 user(s) from 'bank.txt'.

== Main Menu ==
1) Register
2) Login
3) Exit
Choose: 1
Enter name: Sai
Enter email: ohmsai965@gmail.com
Enter phone: 9810274982
Set password: 1234
Registration successful! Your ID: 2
```

2) Logging in to an existing bank account.

```
== Main Menu ==
1) Register
2) Login
3) Exit
Choose: 2
Enter email: ohmsainirmalkar@gmail.com
Enter password: 0406$xyZ
Login successful! Welcome Ohmsai Nirmalkar.
```

3) Viewing the profile of an account.

```
== Account Menu (ohmsainirmalkar@gmail.com) ==
1) View Profile
2) View Balance
3) Deposit
4) Withdraw
5) Transactions
6) Logout
Choose: 1

--- Profile ---
ID: 1
Name: Ohmsai Nirmalkar
Email: ohmsainirmalkar@gmail.com
Phone: 9667148381
Balance: 400.00
```

4) Viewing the Balance.

```
== Account Menu (ohmsainirmalkar@gmail.com) ==
1) View Profile
2) View Balance
3) Deposit
4) Withdraw
5) Transactions
6) Logout
Choose: 2
Current balance: 400.00
```

5) Depositing Money.

```
== Account Menu (ohmsainirmalkar@gmail.com) ==
1) View Profile
2) View Balance
3) Deposit
4) Withdraw
5) Transactions
6) Logout
Choose: 3
Enter amount to deposit: 500
Deposited 500.00. New balance: 900.00
```

6) Withdrawing Money.

```
== Account Menu (ohmsainirmalkar@gmail.com) ==
1) View Profile
2) View Balance
3) Deposit
4) Withdraw
5) Transactions
6) Logout
Choose: 4
Enter amount to withdraw: 200
Withdrawn 200.00. New balance: 700.00
```

7) Viewing all the transactions.

```
== Account Menu (ohmsainirmalkar@gmail.com) ==
1) View Profile
2) View Balance
3) Deposit
4) Withdraw
5) Transactions
6) Logout
Choose: 5

--- Transactions (4) ---
Timestamp          Type      Amount      Balance
2025-12-01 08:20:37 DEPOSIT   500.00    500.00
2025-12-01 08:20:42 WITHDRAW  100.00    400.00
2025-12-01 10:48:35 DEPOSIT   500.00    900.00
2025-12-01 10:49:17 WITHDRAW  200.00    700.00
```

8) Logging out of the account.

```
== Account Menu (ohmsainirmalkar@gmail.com) ==
1) View Profile
2) View Balance
3) Deposit
4) Withdraw
5) Transactions
6) Logout
Choose: 6

== Main Menu ==
1) Register
2) Login
3) Exit
Choose:
```

9) Exiting the code.

```
== Main Menu ==
1) Register
2) Login
3) Exit
Choose: 3
Exiting...
```

10) Text File Containing all the information.

```
Open ▾ +1
● bank.txt x bank.h file_ops.h

2
ID:1
Name:Ohmsai Nirmalkar
Email:ohmsainirmalkar@gmail.com
Phone:9667148381
Password:0406$xyZ
Balance:700.00
Transactions:4
1764577237,DEPOSIT,500.00,500.00
1764577242,WITHDRAW,100.00,400.00S
1764586115,DEPOSIT,500.00,900.00
1764586157,WITHDRAW,200.00,700.00
END
ID:2
Name:Sai
Email:ohmsai965@gmail.com
Phone:9810274982
Password:1234
Balance:0.00
Transactions:0
END
```

CONCLUSION AND FUTURE WORK

The Bank Account Management System developed in C demonstrates structured programming, secure authentication, and persistent data storage. The project successfully achieves reliability, modularity, and readability. In future versions, enhancements such as encrypted password storage, GUI interface, cloud storage, database integration, and multi-user online access can be added.

REFERENCES

1. E. Balagurusamy – Programming in ANSI C
2. Yashavant Kanetkar – Let Us C
3. ISO/IEC 9899 C Standard Documentation
4. Online C File Handling Documentation