North South University

Department of Electrical and Computer Engineering

Project Title : Bank Management System in C Programming

Course Name : Programming language I lab (CSE 115L)

Submitted To : Zobaer Ahammod Zamil & Fowzia Farzana

Submission Date : 21-08 - 2025

==========**PROJECT MEMBER**==========

Name : Mazba Uddin Saif	Name : Md. Sagor	Name : Ajwad Abrar
Stu. ID : 2522638642	Stu. ID : 2524569643	Stu. ID : 2521178642
Section: 09	Section: 09	Section: 09

Table of Contents:

Bank Management System in C Programming

1. Problem Statement:

Manual banking processes are slow and error-prone. To manage accounts, deposits, withdrawals, and transaction records more efficiently, a computerized system is needed. This project implements a bank management system in C where users can register, login, and perform transactions, while admin can monitor users and transaction history through a simple menu-driven program.

2. Project Description:

The “Bank Management System” is a C programming project developed to simulate the core operations of a banking system. It allows both administrator and users to perform various banking activities, such as account creation, deposit, withdrawal, balance inquiry, and transaction history management. The system makes use of structured programming concepts, including structures, file handling, and modular design with functions.

3.Objectives:

- To implement a simple banking system using C programming.
- To apply **structures** for handling user and transaction data.
- To use **file handling** (users.txt, transaction_history.txt) for permanent storage of customer information.
- To demonstrate modular programming by dividing the system into multiple Functions.
- To simulate both **Admin** and **User** functionalities in one system.

4. System Features:

1. **Admin Features:**
    - Login using admin credentials.
    - View all users.
    - View specific user details.
    - View all transaction history.
    - Search bank statements by username.
    - Delete user account.

2. **User Features:**
    - New user registration.
    - Login with username and password.
    - Deposit money into account.
    - Withdraw money (with balance validation).
    - View balance.
    - View account details.
    - Reset password.

5. Methodology:

1. **Programming Language :** C
2. **Compiler/IDE:** Code blocks **/** VS code
3. **Menu-Driven Interface:**
    - The main menu allows users and administrator to easily select options through switch–case based navigation.
    - This ensures the program is user-friendly and well-structured.
4. **Data Structures Used:**
    - struct user { name, password, balance, date of birth }
    - struct transaction { name, deposit, withdraw, time }

5. **File Handling:**
    - `users.txt` → Stores user details
    - `transaction_history.txt` → Stores deposits & withdrawals with timestamps
6. **Functions:**
    - **Admin:**`adminLogin()`,`adminWiew()`,`viewAllUsers()`, `viewUserDetails()`,`transactionHistory()`,`deleteUser()`, `searchBankStatements()`.
    - **User:**`userRegister()`,`userLogin()`,`deposit()`,`withdraw()`, `showUserBalance()`,`showUserDetails()`,`resetUserPassword()`.
    - **Utility:** `readUsers()`, `writeUsers()`, `readStatements()`.
7. **Error Handling & Validation:**
    - Conditional checks are used to handle invalid inputs (negative deposits, wrong passwords).

6. Code Implementation:

The project is implemented in multiple functions to separate concerns. Below is a small snippet:

```
// All structures
typedef struct
{
    char name[100];
    char password[25];
    float balance;
    char DOB[11]; // dd-mm-yyyy

} user;

typedef struct
{
    char name[100];
    float deposit;
    float withdraw;
    char time[30];
} transaction;

```

- **File operations:** `fprintf()` and `fscanf()` are used to store and retrieve user data.

- **Time function:** `ctime()` is used to store timestamps for transactions.
- **Admin & User menus:** Implemented with `switch-case` inside infinite loops until logout.

7. Sample Input/Output:

- **Main Menu:**

```
125         system("cls");
126         printf("-----------------------------------\n");
127         printf("\tBANK MANAGEMENT SYSTEM\n");
128         printf("-----------------------------------\n");
129         printf("\t1.Admin Login\n");
130         printf("\t2.New User Registration\n");
131         printf("\t3.User Login\n");
132         printf("\t4.Exit\n");
133
134         printf("\n\tEnter your choice:");
135         scanf("%d", &choice);
136         getchar();
```

```
-----------------------------------
       BANK MANAGEMENT SYSTEM
-----------------------------------
       1.Admin Login
       2.New User Registration
       3.User Login
       4.Exit

       Enter your choice:
```

- **Admin Login:**

```
162     system("cls");
163     char name[25], pass[25];
164     printf("------------------------------\n");
165     printf("\tADMIN LOGIN\n");
166     printf("------------------------------\n");
167     printf("\n\tEnter Admin Username: ");
168     gets(name);
169     fflush(stdin);
170     printf("\n\tEnter Password: ");
171     gets(pass);
```

```
-----------------------------
       ADMIN LOGIN
-----------------------------

       Enter Admin Username: admin

       Enter Password: admin
```

- **Admin Panel:**

```
191         system("cls");
192         printf("-----------------------------\n");
193         printf("\tWELCOME TO ADMIN PANEL\n");
194         printf("-----------------------------\n");
195         printf("\t1.View All Users\n");
196         printf("\t2.View User Details\n");
197         printf("\t3.View All Transactions\n");
198         printf("\t4.Search Bank Statements\n");
199         printf("\t5.Delete user\n");
200         printf("\t6.Logout\n");
201
202         printf("\n\tEnter your choice:");
203         scanf("%d", &choice);
204         getchar();
```

```
-----------------------------
       WELCOME TO ADMIN PANEL
-----------------------------
       1.View All Users
       2.View User Details
       3.View All Transactions
       4.Search Bank Statements
       5.Delete user
       6.Logout

       Enter your choice:
```

- **View All User:**

```
236  void viewAllUsers()
237  {
238      system("cls");
239      user users[MAX_USERS];
240      int numUsers = readUsers(users, "users.txt");
241
242      printf("List of all users:\n");
243      for (int i = 0; i < numUsers; i++)
244      {
245          printf("User %d: %s\n", i + 1, users[i].name);
246      }
247      printf("\nPress enter to go back to admin menu.\n");
248      getchar();
249  }
```

```
List of all users:
User 1: sagor
User 2: mezba
User 3: ajwad

Press enter to go back to admin menu.
```

- **View User Details:**

```
    for (int i = 0; i < numUsers; i++)
    {
        if (strcmp(users[i].name, name) == 0)
        {
            printf("Account Holder Name: %s\n", users[i].name);
            printf("Date of Birth: %s\n", users[i].DOB);
            printf("Current Balance: %.2f\n", users[i].balance);
            printf("\nPress enter to go back to admin menu.\n");
            getchar();
            return;
        }
    }
    printf("User not found! Press enter to try again.\n");
    getchar();
    viewUserDetails();
```

```
Enter the name of the user to view details: sagor
Account Holder Name: sagor
Date of Birth: 20-05-2003
Current Balance: 0.00

Press enter to go back to admin menu.
```

- **View All Transactions:**

```
    printf("-------------------------------------------------------------------------\n");
    printf("S.No   %-20s %-10s %-10s %-25s\n", "Name", "Deposit", "Withdraw", "Time");
    printf("-------------------------------------------------------------------------\n");

    for (int i = 0; i < numTrans; i++)
    {
        printf("%-6d %-20s %-10.2f %-10.2f %-25s\n",
               i + 1, trans[i].name, trans[i].deposit, trans[i].withdraw, trans[i].time);
        totalDeposit += trans[i].deposit;
        totalWithdraw += trans[i].withdraw;
    }

    printf("-------------------------------------------------------------------------\n");
    printf("TOTAL  %-20s %-10.2f %-10.2f\n", "", totalDeposit, totalWithdraw);
    printf("-------------------------------------------------------------------------\n");
```

```
-------------------------------------------------------------------------
S.No   Name                 Deposit    Withdraw   Time
-------------------------------------------------------------------------
1      sagor                2000.00    0.00       Wed Aug 20 02:33:08 2025
2      sagor                0.00       500.00     Wed Aug 20 02:33:18 2025
3      mezba                50000.00   0.00       Wed Aug 20 02:34:05 2025
-------------------------------------------------------------------------
TOTAL                       52000.00   500.00
-------------------------------------------------------------------------
```

- **Search Bank Statements:**

```
system("cls");
transaction trans[MAX_USERS];
int numTrans = readStatements(trans, "transaction_history.txt");

char name[100];
printf("Enter the name of the user to view statement: ");
gets(name);

float totalDeposit = 0, totalWithdraw = 0;
int found = 0;

printf("-------------------------------------------------------------------------\n");
printf("S.No   %-20s %-10s %-10s %-25s\n", "Name", "Deposit", "Withdraw", "Time");
printf("-------------------------------------------------------------------------\n");
```

```
Enter the name of the user to view statement: sagor
-------------------------------------------------------------------------
S.No   Name                 Deposit    Withdraw   Time
-------------------------------------------------------------------------
1      sagor                2000.00    0.00       Wed Aug 20 02:33:08 2025
2      sagor                0.00       500.00     Wed Aug 20 02:33:18 2025
-------------------------------------------------------------------------
TOTAL                       2000.00    500.00
-------------------------------------------------------------------------

Press enter to go back to admin menu.
```

- **Delete User:**

```
system("cls");
user users[MAX_USERS];
int numUsers = readUsers(users, "users.txt");

char name[100];
printf("Enter the name of the user to delete: ");
gets(name);
```

```
Enter the name of the user to delete: mezba
User deleted successfully! Press enter to go back to admin menu.
```

- **Log Out:**

```
223
224             case 6:
225                 printf("\nLogging out...\n");
226                 return;
227
```

```
------------------------------
        WELCOME TO ADMIN PANEL
------------------------------
        1.View All Users
        2.View User Details
        3.View All Transactions
        4.Search Bank Statements
        5.Delete user
        6.Logout

        Enter your choice:6
```

- **New User Registration:**

```
388 void userRegister()
389 {
390     system("cls");
391     user newUser;
392     user users[MAX_USERS];
393     int numUsers = readUsers(users, "users.txt");
394
395     printf("Enter your name: ");
396     gets(newUser.name);
397
398
399     for (int i = 0; i < numUsers; i++)
400     {
401         if (strcmp(users[i].name, newUser.name) == 0)
402         {
403             printf("\nThis username is already registered! Please choose another name.\n");
404             getchar();
405             userRegister();
406             return;
407         }
408     }
409
410     printf("Enter password: ");
411     gets(newUser.password);
412     newUser.balance = 0.0;
413     printf("Enter date of birth (dd-mm-yyyy): ");
414     gets(newUser.DOB);
415
416
417     FILE *fp = fopen("users.txt", "a");
418     fprintf(fp, "Account holder name - %s\nPassword - %s\nDate of birth - %s\nBalance - %.2f\n\n",
419             newUser.name, newUser.password, newUser.DOB, newUser.balance);
420     fclose(fp);
421
422     printf("\nRegistration successful! Press enter to continue.\n");
423     getchar();
424     return;
425 }
```

```
Enter your name: human
Enter password: 5555
Enter date of birth (dd-mm-yyyy): 20-05-2003

Registration successful! Press enter to continue.
```

- **User Login:**

```
430     system("cls");
431     user editedUser[50];
432     int numUsers = readUsers(editedUser, "users.txt");
433     int userExists = 0;
434     int userIndex;
435     int choice;
436
437     char name[100];
438     char password[25];
439
440     printf("Enter your name:");
441     gets(name);
442     printf("Enter your password:");
443     gets(password);
444
```

```
Enter your name:ajwad
Enter your password:3333
```

- **User Login Panel:**

```
457         {
458             system("cls");
459             printf("-------------------------------------\n");
460             printf("\tSUCCESSFULLY LOGGED IN\n");
461             printf("-------------------------------------\n");
462
463             printf("\t1.Deposit\n");
464             printf("\t2.Withdraw\n");
465             printf("\t3.Show Balance\n");
466             printf("\t4.Show Account Details\n");
467             printf("\t5.Reset Password\n");
468             printf("\t6.Logout\n");
469
470             printf("\n\tEnter your choice:");
471             scanf("%d", &choice);
472             getchar();
473
```

```
-------------------------------------
        SUCCESSFULLY LOGGED IN
-------------------------------------
        1.Deposit
        2.Withdraw
        3.Show Balance
        4.Show Account Details
        5.Reset Password
        6.Logout

        Enter your choice:
```

- **Deposit:**

```
    printf("Enter amount to deposit: ");
    scanf("%f", &amount);
    getchar();

    editedUser[userIndex].balance += amount;

    printf("Deposit successful! New balance: %.2f\n", editedUser[userIndex].balance);
    writeUsers(editedUser, numUsers, "users.txt");
    printf("\nPress enter to go back to previous menu.\n");
```

```
Enter amount to deposit: 500
Deposit successful! New balance: 500.00

Press enter to go back to previous menu.
```

- **Withdraw:**

```
    do
    {
        printf("Enter amount to withdraw: ");
        scanf("%f", &amount);
        getchar();
        if (editedUser[userIndex].balance < amount)
        {
            printf("Withdraw insuccessful! Current balance: %.2f\n", editedUser[userIndex].balance);
        }
    } while (editedUser[userIndex].balance < amount);

    editedUser[userIndex].balance -= amount;
    printf("Withdraw successful! New balance: %.2f\n", editedUser[userIndex].balance);
    writeUsers(editedUser, numUsers, "users.txt");
    printf("\nPress enter to go back to previous menu.\n");
```

```
Enter amount to withdraw: 200
Withdraw successful! New balance: 300.00

Press enter to go back to previous menu.
```

- **Show Balance:**

```
void showUserBalance(user editedUser[], int numUsers, int userIndex)
{
    system("cls");
    printf("Your current balance is: %.2f\n", editedUser[userIndex].balance);
    printf("\nPress enter to go back to previous menu.\n");
    getchar();
}
```

```
Your current balance is: 300.00

Press enter to go back to previous menu.
```

- **Show Account Details:**

```c
void showUserDetails(user editedUser[], int numUsers, int userIndex)
{
    system("cls");
    printf("Account Holder Name: %s\n", editedUser[userIndex].name);
    printf("Date of Birth: %s\n", editedUser[userIndex].DOB);
    printf("Current Balance: %.2f\n", editedUser[userIndex].balance);
    printf("\nPress enter to go back to previous menu.\n");
    getchar();
}
```

```
Account Holder Name: ajwad
Date of Birth: 05-03-2004
Current Balance: 300.00

Press enter to go back to previous menu.
```

- **Reset Password:**

```c
void resetUserPassword(user editedUser[], int numUsers, int userIndex)
{
    system("cls");
    char newPassword[25];
    fflush(stdin);
    printf("Enter new password: ");
    gets(newPassword);
    strcpy(editedUser[userIndex].password, newPassword);
    writeUsers(editedUser, numUsers, "users.txt");
    printf("Password reset successful! Press enter to go back to previous menu.\n");
    getchar();
}
```

```
Enter new password: 5555
Password reset successful! Press enter to go back to previous menu.
```

8. Discussion:

This project demonstrates how file handling and structures can be effectively used in C to build real-world applications. It provides separate interfaces for admin and users, mimicking an actual banking system. Error handling is included for invalid credentials and insufficient balances.

9. Conclusion:

The project successfully implements a simplified banking system. It provides a practical demonstration of structured programming, modular design, and persistent data storage. Future improvements may include encryption of passwords, graphical user interface (GUI), and database integration (MySQL).

10. References:

- Class Lecture Notes, CSE115.
- Class Lecture Notes, CSE115 Lab
- Book: Problem Solving and Program Design in C by J Hanly and E Koffman
- Online resorses : W3School.

THANK YOU