Bank Management System

Introduction

In bank management system there is

Functions in the Bank Management System

The source code for the Bank Management System for Customer Accounts is reasonably short and simple to comprehend. This C mini project is separated into many functions, most of which are connected to various financial operations. Some of the most critical functionalities are listed below to assist you better understand the project.

1)linked list

void addCustomer(struct Customer\*\* head, struct Customer\* newCustomer) {

    if (\*head == NULL) {

        \*head = newCustomer;

    } else {

        struct Customer\* current = \*head;

        while (current->next != NULL) {

            current = current->next;

        }

        current->next = newCustomer;

        newCustomer->prev = current;

    }

}

//linked list

void addLoan(struct Loan\*\* front, struct Loan\*\* rear, struct Loan\* newLoan) {

    if (\*front == NULL) {

        \*front = \*rear = newLoan;

    } else {

        (\*rear)->next = newLoan;

        \*rear = newLoan;

    }

}

2)Stack

void transferMoney(struct Customer\* sender, struct Customer\* receiver, double amount) {

    if (sender->balance >= amount) {

        sender->balance -= amount;

        receiver->balance += amount;

        receiver->transferredAmount += amount;

        sender->transferredAmount -= amount;

        printf("Transfer of %.2f successful from %s to %s.\n", amount, sender->name, receiver->name);

    } else {

        printf("Transfer from %s to %s failed: Insufficient balance for the transfer.\n", sender->name, receiver->name);

    }

}

3)doubly linked list

void displayCustomers(struct Customer\* head, struct Loan\* loanFront) {

struct Customer\* current = head;

printf("Account Number\tName\tBalance\tTransferred Amount\tTotal Loan Amount\n");   while (current != NULL) {

double totalLoan = 0; // Initialize the total loan amount to 0 printf("%d\t\t%s\t%.2f\t%.2f\t", current->accountNumber, current->name, current->balance, current->transferredAmount);

if (current->transferredAmount > 0) {

printf("    Transferred from:\n");

struct Customer\* sender = head;

while (sender != NULL) {

if (sender->transferredAmount < 0) {

printf("    Sender: %s (Account Number: %d) - %.2f\n", sender->name, sender->accountNumber, -sender->transferredAmount);

}

sender = sender->next;

}

}

 printf("    Transferred to:\n");

struct Customer\* receiver = head;

while (receiver != NULL) {

if (receiver->transferredAmount > 0) {

printf("    Receiver: %s (Account Number: %d) - %.2f\n", receiver->name, receiver->accountNumber, receiver->transferredAmount);

}

receiver = receiver->next;

}

struct Loan\* loan = loanFront;

while (loan != NULL) {

if (loan->accountNumber == current->accountNumber) {

totalLoan += loan->amount;

            }

loan = loan->next;

        }

printf("Total Loan Amount: %.2f\n", totalLoan);

current = current->next;

    }

}

4)Queue

struct Customer\* loadFromTextFile(const char\* fileName) {

    FILE\* file = fopen(fileName, "r");

    if (file == NULL) {

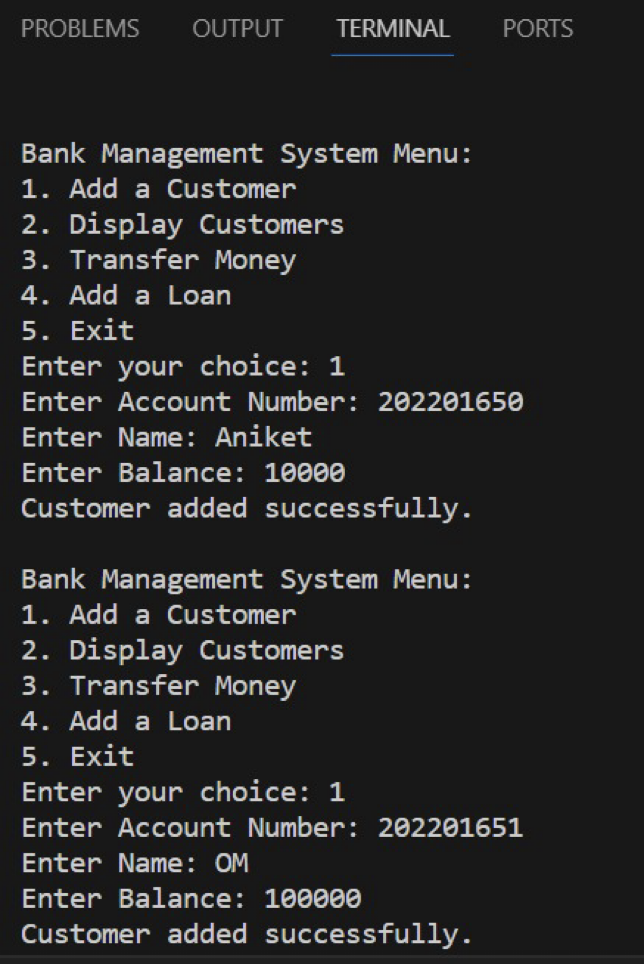
        perror("Unable to open file for reading");

        return NULL;

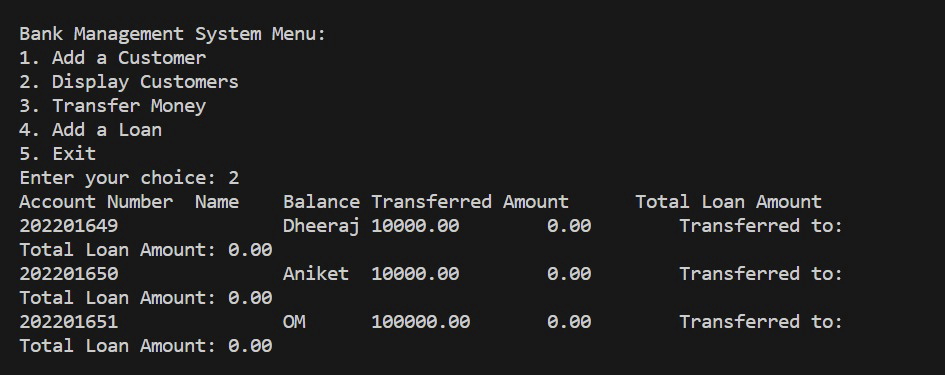
    }

Code of bank management system

1)Add a customer: This method takes details from the customer like name, address, type of account, and depositing money and creating a new account.

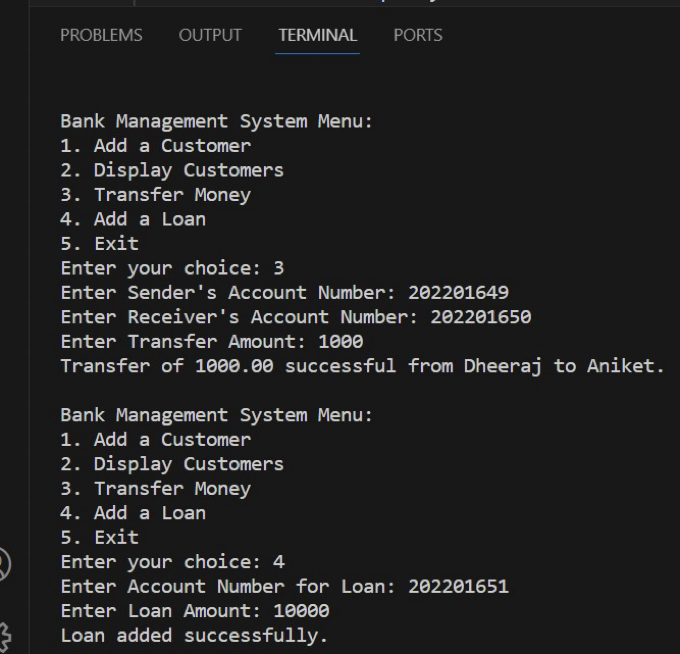


2)Display customers-Display Account function will show the details of the customer like name, address, type of account and available balance.

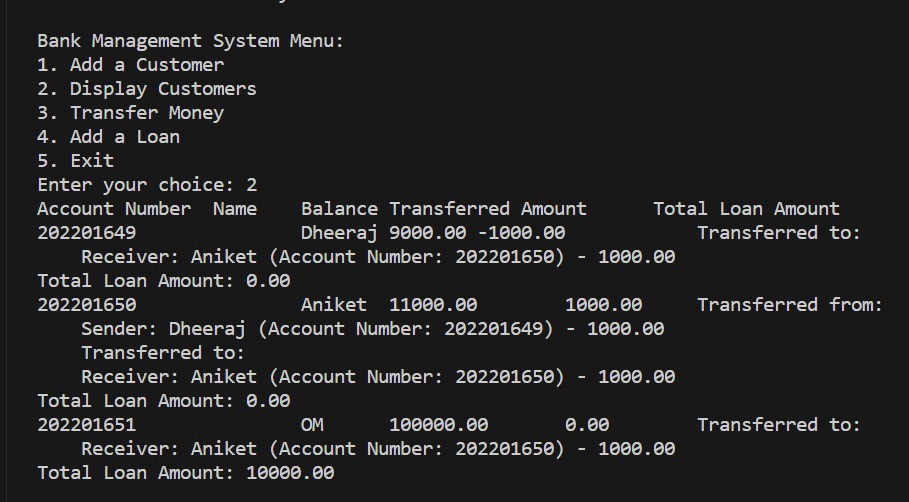


3)Transfer Money-This method will help to transfer money from one account number to another.

4)Add a loan-This method will help to add a loan in a specificied account and display in the loan amount field.



5)Overall transactions of two customers-



Conclusion

After finishing the Bank Management System project using the C programming language, it is clear that this system is a useful tool for managing the transactions and operations of a bank. Users can do many things with the system, like set up accounts, deposit and withdraw money, and check their account balances. This project shows how powerful and flexible the C language is, as well as how it can handle complicated tasks in a clear and efficient way. Overall, putting the Bank Management System into place using C was a success, and it is expected to be a useful tool for managing how a bank works