```c
#include <stdio.h>

#define MAX_CUSTOMERS 3

int accountNumbers[MAX_CUSTOMERS];
short int customerIds[MAX_CUSTOMERS];
char customerNames[MAX_CUSTOMERS][50];
int ages[MAX_CUSTOMERS];
float balances[MAX_CUSTOMERS];
float loanAmounts[MAX_CUSTOMERS];

int numberOfCustomers = 0;


void createAccount();
void checkBalance();
void deposit();
void withdraw();
void applyForLoan();
void makePayment();
void displayMenu();

int main() {
    int choice;

    printf("Welcome to the Bank Management System!\n");

    while (1) {
        displayMenu();

        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                createAccount();
                break;
            case 2:
                checkBalance();
                break;
            case 3:
                deposit();
                break;
            case 4:
                withdraw();
                break;
            case 5:
                applyForLoan();
                break;
```

```
            case 6:
                makePayment();
                break;
            case 0:
                printf("Exiting the Bank Management System. Have a great
day!\n");
                return 0;
            default:
                printf("Invalid choice. Please enter a valid option.\n");
                break;
        }
    }

    return 0;
}

void displayMenu() {
    printf("\n1. Create Account\n2. Check Balance\n3. Deposit\n4. Withdraw\n5.
Apply for a Loan\n");
    printf("6. Make Payment\n0. Exit\n");
}

void createAccount() {
    if (numberOfCustomers >= MAX_CUSTOMERS) {
        printf("Maximum number of customers reached. Cannot create more
accounts.\n");
        return;
    }

    FILE *file = fopen("accounts.txt", "w");
    if (file == NULL) {
        printf("Error opening file.\n");
        return;
    }

    printf("Enter customer details:\n");
    printf("Customer ID : ");
    scanf("%hd", &customerIds[numberOfCustomers]);

    printf("Customer Name : ");
    scanf(" %[^\n]s", customerNames[numberOfCustomers]);

    printf("Age : ");
    scanf("%d", &ages[numberOfCustomers]);

    accountNumbers[numberOfCustomers] = numberOfCustomers + 1;
    balances[numberOfCustomers] = 0.0;
    loanAmounts[numberOfCustomers] = 0.0;
```

```
    fprintf(file, "%d %hd %s %d %.2f %.2f\n",
accountNumbers[numberOfCustomers], customerIds[numberOfCustomers],
customerNames[numberOfCustomers], ages[numberOfCustomers],
balances[numberOfCustomers], loanAmounts[numberOfCustomers]);

    printf("Account created successfully. Account Number: %d\n",
accountNumbers[numberOfCustomers]);
    numberOfCustomers++;

    if (numberOfCustomers < MAX_CUSTOMERS - 1) {
        createAccount();
    }

    fclose(file);
}

void checkBalance() {
    int accountNumber;
    printf("Enter account number: ");
    scanf("%d", &accountNumber);

    int customerIndex = -1;
    for (int i = 0; i < numberOfCustomers; i++) {
        if (accountNumbers[i] == accountNumber) {
            customerIndex = i;
            break;
        }
    }

    if (customerIndex!= -1) {
        printf("Account Number: %d\n", accountNumbers[customerIndex]);
        printf("Customer ID: %hd\n", customerIds[customerIndex]);
        printf("Customer Name: %s\n", customerNames[customerIndex]);
        printf("Age: %d\n", ages[customerIndex]);
        printf("Balance: %.2f\n", balances[customerIndex]);
        printf("Loan Amount: %.2f\n", loanAmounts[customerIndex]);
    } else {
        printf("Account not found.\n");
        char response;
        printf("Do you want to exit the program? (y/n) ");
        scanf(" %c", &response);
        if (response == 'y' || response == 'Y') {
            return;
        }
        checkBalance();
    }
}
```

```
void deposit() {
    int accountNumber;
    printf("Enter account number: ");
    scanf("%d", &accountNumber);

    int customerIndex = -1;
    for (int i = 0; i < numberOfCustomers; i++) {
        if (accountNumbers[i] == accountNumber) {
            customerIndex = i;
            break;
        }
    }

    if (customerIndex != -1) {
        float depositAmount;
        printf("Enter the deposit amount: ");
        scanf("%f", &depositAmount);
        balances[customerIndex] += depositAmount;
        printf("Deposit successful. New balance: %.2f\n",
balances[customerIndex]);
    } else {
        printf("Account not found.\n");
    }
}

void withdraw() {
    int accountNumber;
    printf("Enter account number: ");
    scanf("%d", &accountNumber);

    int customerIndex = -1;
    for (int i = 0; i < numberOfCustomers; i++) {
        if (accountNumbers[i] == accountNumber) {
            customerIndex = i;
            break;
        }
    }

    if (customerIndex != -1) {
        float withdrawalAmount;
        printf("Enter the withdrawal amount: ");
        scanf("%f", &withdrawalAmount);
        if (withdrawalAmount <= balances[customerIndex]) {
            balances[customerIndex] -= withdrawalAmount;
            printf("Withdrawal successful. New balance: %.2f\n",
balances[customerIndex]);
        } else {
            printf("Insufficient funds. Withdrawal failed.\n");
```

```
        }
    } else {
        printf("Account not found.\n");
    }
}

void applyForLoan() {
    int accountNumber;
    printf("Enter account number: ");
    scanf("%d", &accountNumber);

    int customerIndex = -1;
    for (int i = 0; i < numberOfCustomers; i++) {
        if (accountNumbers[i] == accountNumber) {
            customerIndex = i;
            break;
        }
    }

    if (customerIndex != -1) {
        float loanAmountInput;
        printf("Enter the loan amount: ");
        scanf("%f", &loanAmountInput);

        if (loanAmountInput <= 0 || loanAmounts[customerIndex] > 0 ||
loanAmountInput > balances[customerIndex] * 2 || ages[customerIndex] < 18) {
            printf("Loan application rejected. Please check eligibility.\n");
        } else {
            loanAmounts[customerIndex] = loanAmountInput;
            balances[customerIndex] += loanAmounts[customerIndex];
            printf("Loan approved. New balance: %.2f\n",
balances[customerIndex]);
        }
    } else {
        printf("Account not found.\n");
    }
}

void makePayment() {
    int accountNumber;
    printf("Enter account number: ");
    scanf("%d", &accountNumber);

    int customerIndex = -1;
    for (int i = 0; i < numberOfCustomers; i++) {
        if (accountNumbers[i] == accountNumber) {
            customerIndex = i;
            break;
```

```
        }
    }

    if (customerIndex != -1) {
        printf("Select Payment Mode:\n");
        printf("1. Cheque\n2. Debit/Credit Card\n3. UPI Payment Mode\n");
        printf("Enter your choice: ");
        int paymentModeChoice;
        scanf("%d", &paymentModeChoice);

        float paymentAmount;
        printf("Enter the payment amount: ");
        scanf("%f", &paymentAmount);

        if (paymentAmount <= 0) {
            printf("Invalid payment amount.\n");
            return;
        }

        switch (paymentModeChoice) {
            case 1:
                if (paymentAmount > balances[customerIndex]) {
                    printf("Insufficient funds. Payment failed.\n");
                } else {
                    balances[customerIndex] -= paymentAmount;
                    printf("Payment of %.2f using Cheque successful. New
balance: %.2f\n", paymentAmount, balances[customerIndex]);
                }
                break;
            case 2:
                if (paymentAmount > balances[customerIndex]) {
                    printf("Insufficient funds. Payment failed.\n");
                } else {
                    balances[customerIndex] -= paymentAmount;
                    printf("Payment of %.2f using Debit/Credit Card
successful. New balance: %.2f\n", paymentAmount, balances[customerIndex]);
                }
                break;
            case 3:
                if (paymentAmount > balances[customerIndex]) {
                    printf("Insufficient funds. Payment failed.\n");
                } else {
                    balances[customerIndex] -= paymentAmount;
                    printf("Payment of %.2f using UPI Payment Mode successful.
New balance: %.2f\n", paymentAmount, balances[customerIndex]);
                }
                break;
            default:
```

```
                printf("Invalid payment mode. Payment failed.\n");
                break;
        }
    } else {
        printf("Account not found.\n");
    }
}
```

```
1. Create Account
2. Check Balance
3. Deposit
4. Withdraw
5. Apply for a Loan
6. Make Payment
0. Exit
Enter your choice: 1
Enter customer details:
Customer ID : 123456
Customer Name : qwerty
Age : 12
Account created successfully. Account Number: 1
Enter customer details:
Customer ID : 23664
Customer Name : werty
Age : 123
Account created successfully. Account Number: 2

1. Create Account
2. Check Balance
3. Deposit
4. Withdraw
5. Apply for a Loan
6. Make Payment
0. Exit
Enter your choice: 789456
Invalid choice. Please enter a valid option.
```

```
1. Create Account
2. Check Balance
3. Deposit
4. Withdraw
5. Apply for a Loan
6. Make Payment
0. Exit
Enter your choice: 3
Enter account number: 1
Enter the deposit amount: 123456
Deposit successful. New balance: 123456.00

1. Create Account
2. Check Balance
3. Deposit
4. Withdraw
5. Apply for a Loan
6. Make Payment
0. Exit
Enter your choice: 2
Enter account number: 1
Account Number: 1
Customer ID: -7616
Customer Name: qwerty
Age: 12
Balance: 123456.00
Loan Amount: 0.00
```