**BANK MANAGEMENT SYSTEM**

**Submitted by**

**Name of the Students:** SHUBHAM ACHARYA

**Enrolment Number:** 120222002003044

**Section:** I

**Class Roll Number:** 01  
**Stream:** ECE

**Subject:** Programming for Problem Solving using C

**Subject Code:** ESC103(Pr)

**Department:** Basic Science and Humanities

Under the supervision of

Swarnendu Ghosh

**Academic Year: 2022-26**

PROJECT REPORT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE FIRST SEMESTER



**DEPARTMENT OF BASIC SCIENCE AND HUMANITITES**

**INSTITUTE OF ENGINEERING AND MANAGEMENT, KOLKATA**



**CERTIFICATE OF RECOMMENDATION**

We hereby recommend that the project prepared under our supervision by **Shubham Acharya,** entitled Bank Management System be accepted in partial fulfillment of the requirements for the degree of partial fulfillment of the first semester.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Head of the Department Project Supervisor

Basic Sciences and Humanities

IEM, Kolkata

# Introduction

The bank management system is a software program that allows banks to efficiently manage their operations and customer accounts. This program is typically used by banks to manage customer account information, process transactions, and generate reports.

## Objective

The bank management system implemented in C provides a basic framework for managing bank accounts.

## Organization of the Project

This project has been organized using different elements like user name, account number, balance, deposit amount, withdraw amount, etc.

# Dataset Descriptions

* “Number of customers records to enter” stores and updates data about the number of customers.
* “Display all records” stores and updates data about all the financial records of customers.
* “Search a record” search the records about a particular customer.
* “Deposit amount” enables to enter deposit amount.
* “Withdraw amount” enables to enter withdraw amount

# Variable declaration

|  |  |  |
| --- | --- | --- |
| Variable Name | Datatype | Purpose |
| N | int | To enter a number |
| Choice | int | To enter a choice of the user |
| Account No. | int | To enter account number |
| Amount | int | To enter an amount |
| I | int | To enter a number |
| Index | Int | To enter index value |

# Function declaration

|  |  |
| --- | --- |
| Function Name | Description |
| Printf() | To print a statement, number, etc. |
| Scanf() | To take value from the user |
| Gets() | To read a string or a text line |
| Display Account | To display account details |
| Deposit | To deposit money into an account |
| Withdraw | To withdraw money from an account |

# Features

The bank management system implemented in C provides the following functionality:

- Creation of new bank accounts

- Displaying account details

- Depositing money into accounts

- Withdrawing money from accounts

- Error handling for insufficient account balance

# Programs

#include <stdio.h>

struct customer

{

    int account\_no;

    char name[80];

    int balance;

};

void accept(struct customer[], int);

void display(struct customer[], int);

int search(struct customer[], int, int);

void deposit(struct customer[], int, int, int);

void withdraw(struct customer[], int, int, int);

int main()

{

    struct customer data[20];

    int n, choice, account\_no, amount, index;

    printf("Banking System\n\n");

    printf("Number of customer records you want to enter? : ");

    scanf("%d", &n);

    accept(data, n);

    do

    {

        printf("\nBanking System Menu :\n");

        printf("Press 1 to display all records.\n");

        printf("Press 2 to search a record.\n");

        printf("Press 3 to deposit amount.\n");

        printf("Press 4 to withdraw amount.\n");

        printf("Press 0 to exit\n");

        printf("\nEnter choice(0-4) : ");

        scanf("%d", &choice);

        switch (choice)

        {

            case 1:

                display(data, n);

                break;

            case 2:

                printf("Enter account number to search : ");

                scanf("%d", &account\_no);

                index = search(data, n, account\_no);

                if (index ==  - 1)

                {

                    printf("Record not found : ");

                }

                else

                {

                    printf("A/c Number: %d\nName: %s\nBalance: %d\n",

                        data[index].account\_no, data[index].name,

                        data[index].balance);

                }

                break;

            case 3:

                printf("Enter account number : ");

                scanf("%d", &account\_no);

                printf("Enter amount to deposit : ");

                scanf("%d", &amount);

                deposit(data, n, account\_no, amount);

                break;

            case 4:

                printf("Enter account number : ");

                scanf("%d", &account\_no);

                printf("Enter amount to withdraw : ");

                scanf("%d", &amount);

                withdraw(data, n, account\_no, amount);

        }

    }

    while (choice != 0);

    return 0;

}

void accept(struct customer list[80], int s)

{

    int i;

    for (i = 0; i < s; i++)

    {

        printf("\nEnter data for Record #%d", i + 1);

        printf("\nEnter account\_no : ");

        scanf("%d", &list[i].account\_no);

        fflush(stdin);

        printf("Enter name : ");

        gets(list[i].name);

        list[i].balance = 0;

    }

}

void display(struct customer list[80], int s)

{

    int i;

    printf("\n\nA/c No\tName\tBalance\n");

    for (i = 0; i < s; i++)

    {

        printf("%d\t%s\t%d\n", list[i].account\_no, list[i].name,

            list[i].balance);

    }

}

int search(struct customer list[80], int s, int number)

{

    int i;

    for (i = 0; i < s; i++)

    {

        if (list[i].account\_no == number)

        {

            return i;

        }

    }

    return  - 1;

}

void deposit(struct customer list[], int s, int number, int amt)

{

    int i = search(list, s, number);

    if (i ==  - 1)

    {

        printf("Record not found");

    }

    else

    {

        list[i].balance += amt;

    }

}

void withdraw(struct customer list[], int s, int number, int amt)

{

    int i = search(list, s, number);

    if (i ==  - 1)

    {

        printf("Record not found\n");

    }

    else if (list[i].balance < amt)

    {

        printf("Insufficient balance\n");

    }

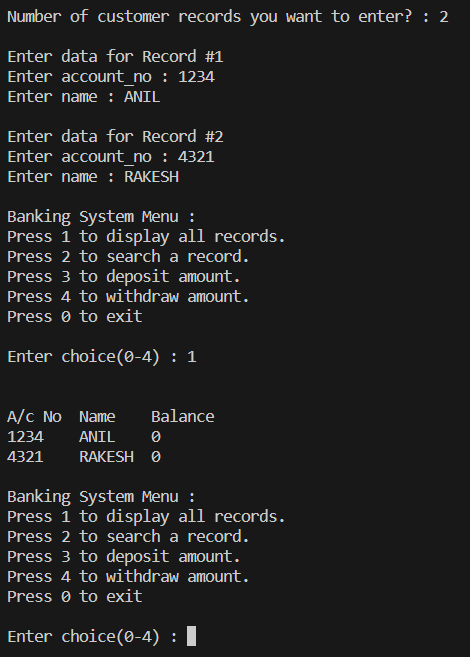
    else

    {

        list[i].balance -= amt;

    }}

# Outputs



The user will first have to enter the number of customer records to be entered. Then the number of desired account number and customer name has to be entered. User can get through all the details of the customers and get the information on all the records of the customer. Alongside a new amount can be deposited and withdrawn by using the mentioned options.