BANK MANAGEMENT SYSTEM

Project File

(IT Workshop)

Submitted By :-

Paramveer Singh (2102212)

Nitesh Chugh (2102209)

Submitted To :-

Neha Bansal

(Assistant Professor)

Chandigarh Engineering College – CGC
Landran, Mohali, Punjab, 140307

CONTENTS

ABSTRACT

The Bank Account Management System is an application for maintaining a person's account in a bank. In this project I tried to show the working of a banking account system and cover the basic functionality of a Bank Account Management System. To develop a project for solving financial applications of a customer in banking environment in order to nurture the needs of an end banking user by providing various ways to perform banking tasks. Also to enable the user's workspace to have additional functionalities which are not provided under a conventional banking project. The main aim of this project is to develop software for Bank Account Management System. This project has been developed to carry out the processes easily and quickly, which is not possible with the manuals systems, which are overcome by this software. The project analyzes the system requirements and then comes up with the requirements specifications. It studies other related systems and then come up with system specifications. The system is then designed in accordance with specifications to satisfy the requirements. The content management system deals with data entry, validation confirm and updating whiles the interactive system deals with system interaction with the administration and users. Thus, above features of this project will save transaction time and therefore increase the efficiency of the system.

INTRODUCTION

The “Bank Account Management System” project is a model Internet Banking. This site enables the customers to perform the basic banking transactions by sitting at their office or at homes through PC or laptop. The system provides the access to the customer to create an account, deposit/withdraw the cash from his account, also to view reports of all accounts present. The customers can access the banks website for viewing their Account details and perform the transactions on account as per their requirements. With Internet Banking, the brick and mortar structure of the traditional banking gets converted into a click and portal model, thereby giving a concept of virtual banking a real shape.

The primary motive of this “Bank Account Management System” is to provide an improved design and methodology, which envisages the future expansion, and modification, which is necessary for a core sector like banking. This necessitates the design to be expandable and modifiable and so a modular approach is used in developing the application software. Thus, considering above necessities, the software for bank management has became necessary which would be useful in managing the bank more efficiently. All transactions are carried out online by transferring from accounts in the same Bank or international bank. The software is meant to overcome the drawbacks of the manual system.

FEASIBILITY STUDY

Understanding Feasibility :

Feasibility study means the analysis of problem to determine if It can be solved effectively. In, other words it is the study of the possibilities of the proposed system it studies the work ability, impact on the organization ability to meet user's need and efficient use of resources. Three aspects in which the system has to be feasible are:-

➢ ECONOMICAL FEASIBILITY:

The economical analysis checks for the high investment incurred on the system. It evaluates development &implementing charges for the proposed "Banking Project". The S/W used for the development is easily available at minimal cost & the database applied is freely available hence it results in low cost implementation.

➢ TECHNICAL FEASIBILITY:

This aspect concentrates on the concept of using Computer Meaning, "Mechanization" of human works. Thus, the automated solution leads to the need for a technical feasibility study. The focus on the platform used C ++ &users for that S/W.

➢ BEHAVIOURAL FEASIBILITY:

Behavioral feasibility deals with the runtime performance of the S/W the proposed system must score higher than the present in the behavioral study. The S/W should have end user in mind when the system is designed while designing s/w the programmer should be aware of the conditions user's Knowledge input, output, calculations etc..

METHODOLOGY

Modules Description:

The Modules description of Bank Account Management System project. These Modules include :-

1. Create New Account: A customer who having the account in the world can create a virtual account through this module. This module receives the customer profile details and the bank account details with the proof of the ownership of the bank account.

2. Login: Virtual account holders can login in to the system using this module. Thus this is the secured login page for the customers in the website.

3. Deposit Money: You can deposit your money any amount whenever, you want from your saving account or current account just by logging in with your login details.

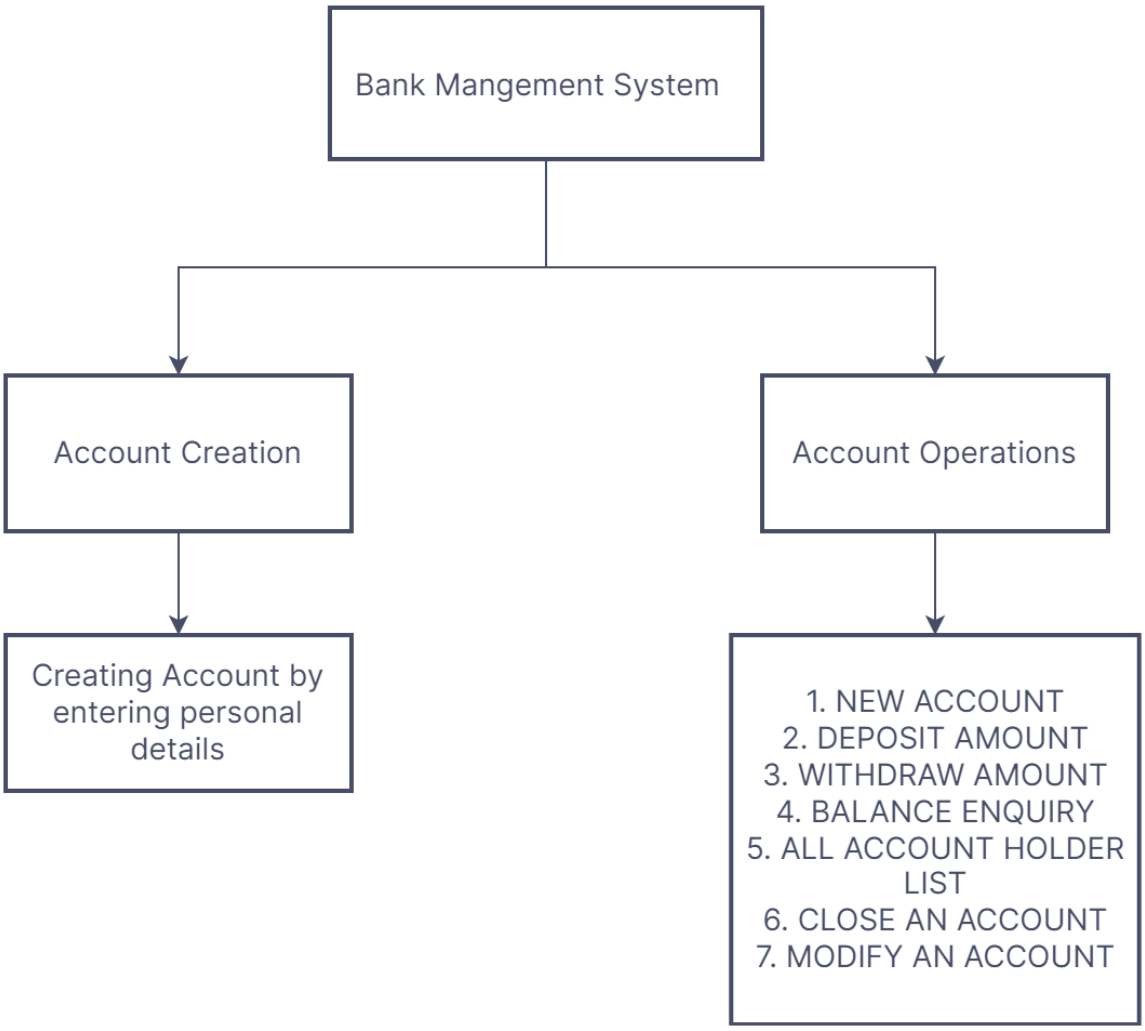
4. Withdraw Money: You can withdraw your money from your saving account.

5. Balance-Enquiry: You can check your current balance in your bank account with our application whenever you deposit or withdraw money.

6. Beneficiary: Beneficiary is a person who receives money. Here the customer can add the beneficiaries to make fund transfer in the future.

7. Closing of Account: This feature will close the account created by the user.

8. Modifying the Account: This feature will help you modify their account details



LANGUAGE USED : C++

The c++ code used for banking management system includes

File system access, so all the data gets stored in external '.dat' Binary file that will be created during runtime.

IMPLEMENTATION

Source Code :-

```
#include<iostream>
#include<fstream>
#include<cctype>
#include<iomanip>
using namespace std;
class account
{
   int acno;
   char name[50];
   int deposit;
   char type;
public:
   void create_account();
   void show_account() const;
   void modify();
   void dep(int);
   void draw(int);
   void report() const;
   int retacno() const;
   int retdeposit() const;
   char rettype() const;
};
```

```
void account::create_account()
{
   system("CLS");
   cout<<"\n\t\t\tEnter the Account No. : ";
   cin>>acno;
   cout<<"\n\n\t\t\tEnter the Name of the Account holder : ";
   cin.ignore();
   cin.getline(name,50);
   cout<<"\n\t\t\tEnter Type of the Account (C/S) : ";
   cin>>type;
   type=toupper(type);
   cout<<"\n\t\t\tEnter The Initial amount : ";
   cin>>deposit;
   cout<<"\n\n\t\t\tAccount Created..";
}

void account::show_account() const
{
   cout<<"\n\t\t\tAccount No. : "<<acno;
   cout<<"\n\t\t\tAccount Holder Name : ";
   cout<<name;
   cout<<"\n\t\t\tType of Account : "<<type;
   cout<<"\n\t\t\tBalance amount : "<<deposit;
}

void account::modify()
```

```
{
   cout<<"\n\t\t\tAccount No. : "<<acno;
   cout<<"\n\t\t\tModify Account Holder Name : ";
   cin.ignore();
   cin.getline(name,50);
   cout<<"\n\t\t\tModify Type of Account : ";
   cin>>type;
   type=toupper(type);
   cout<<"\n\t\t\tModify Balance amount : ";
   cin>>deposit;
}

void account::dep(int x)
{
   deposit+=x;
}

void account::draw(int x)
{
   deposit-=x;
}

void account::report() const
{
   cout<<acno<<setw(10)<<" "<<name<<setw(10)<<"
"<<type<<setw(6)<<deposit<<endl;
}
```

```
int account::retacno() const
{
   return acno;
}

int account::retdeposit() const
{
   return deposit;
}

char account::rettype() const
{
   return type;
}


void write_account();
void display_sp(int);
void modify_account(int);
void delete_account(int);
void display_all();
void deposit_withdraw(int, int);

int main()
{
   char ch;
   int num;
   do
```

```
{
system("CLS");
cout<<"\n\n\t\t\t\t======================\n";
cout<<"\t\t\t\tBANK MANAGEMENT SYSTEM";
cout<<"\n\t\t\t\t======================\n";

   cout<<"\t\t\t\t    ::MAIN MENU::\n";
   cout<<"\n\t\t\t\t1. NEW ACCOUNT";
   cout<<"\n\t\t\t\t2. DEPOSIT AMOUNT";
   cout<<"\n\t\t\t\t3. WITHDRAW AMOUNT";
   cout<<"\n\t\t\t\t4. BALANCE ENQUIRY";
   cout<<"\n\t\t\t\t5. ALL ACCOUNT HOLDER LIST";
   cout<<"\n\t\t\t\t6. CLOSE AN ACCOUNT";
   cout<<"\n\t\t\t\t7. MODIFY AN ACCOUNT";
   cout<<"\n\t\t\t\t8. EXIT";
   cout<<"\n\n\t\t\t\tSelect Your Option (1-8): ";
   cin>>ch;

   switch(ch)
   {
   case '1':
      write_account();
      break;
   case '2':
      system("CLS");
      cout<<"\n\n\t\t\tEnter The account No. : "; cin>>num;
      deposit_withdraw(num, 1);
```

```
    break;
case '3':
    system("CLS");
    cout<<"\n\n\t\t\tEnter The account No. : "; cin>>num;
    deposit_withdraw(num, 2);
    break;
case '4':
    system("CLS");
    cout<<"\n\n\t\t\tEnter The account No. : "; cin>>num;
    display_sp(num);
    break;
case '5':
    display_all();
    break;
case '6':
    system("CLS");
    cout<<"\n\n\t\t\tEnter The account No. : "; cin>>num;
    delete_account(num);
    break;
 case '7':
    system("CLS");
    cout<<"\n\n\t\t\tEnter The account No. : "; cin>>num;
    modify_account(num);
    break;
 case '8':
    system("CLS");
    cout<<"\n\n\t\t\tBrought To You By code-projects.org";
```

```
            break;
         default :cout<<"\a";
        }
        cin.ignore();
        cin.get();
    }while(ch!='8');
    return 0;
}

void write_account()
{
    account ac;
    ofstream outFile;
    outFile.open("account.dat",ios::binary|ios::app);
    ac.create_account();
    outFile.write(reinterpret_cast<char *> (&ac), sizeof(account));
    outFile.close();
}

void display_sp(int n)
{
    account ac;
    bool flag=false;
    ifstream inFile;
    inFile.open("account.dat",ios::binary);
    if(!inFile)
    {
```

```
        cout<<"File could not be open !! Press any Key...";
        return;
    }
    cout<<"\n\t\t\tBALANCE DETAILS\n";
        while(inFile.read(reinterpret_cast<char *> (&ac), sizeof(account)))
    {
        if(ac.retacno()==n)
        {
            ac.show_account();
            flag=true;
        }
    }
    inFile.close();
    if(flag==false)
        cout<<"\n\n\t\t\tAccount number does not exist";
}


void modify_account(int n)
{
    bool found=false;
    account ac;
    fstream File;
    File.open("account.dat",ios::binary|ios::in|ios::out);
    if(!File)
    {
        cout<<"File could not be open !! Press any Key...";
        return;
```

```
    }
    while(!File.eof() && found==false)
    {
        File.read(reinterpret_cast<char *> (&ac), sizeof(account));
        if(ac.retacno()==n)
        {
            ac.show_account();
            cout<<"\n\n\t\t\tEnter The New Details of account"<<endl;
            ac.modify();
            int pos=(-1)*static_cast<int>(sizeof(account));
            File.seekp(pos,ios::cur); //fncallat1353
            File.write(reinterpret_cast<char *> (&ac), sizeof(account));
            cout<<"\n\n\t\t\tRecord Updated";
            found=true;
          }
    }
    File.close();
    if(found==false)
        cout<<"\n\n\t\t\tRecord Not Found ";
}


void delete_account(int n)
{
    account ac;
    ifstream inFile;
    ofstream outFile;
    inFile.open("account.dat",ios::binary);
```

```
    if(!inFile)
    {
       cout<<"File could not be open !! Press any Key...";
       return;
    }
    outFile.open("Temp.dat",ios::binary);
    inFile.seekg(0,ios::beg);
    while(inFile.read(reinterpret_cast<char *> (&ac), sizeof(account)))
    {
       if(ac.retacno()!=n)
       {
          outFile.write(reinterpret_cast<char *> (&ac), sizeof(account));
       }
    }
    inFile.close();
    outFile.close();
    remove("account.dat");
    rename("Temp.dat","account.dat");
    cout<<"\n\nRecord Deleted ..";
}

void display_all()
{
    system("CLS");
    account ac;
    ifstream inFile;
    inFile.open("account.dat",ios::binary);
```

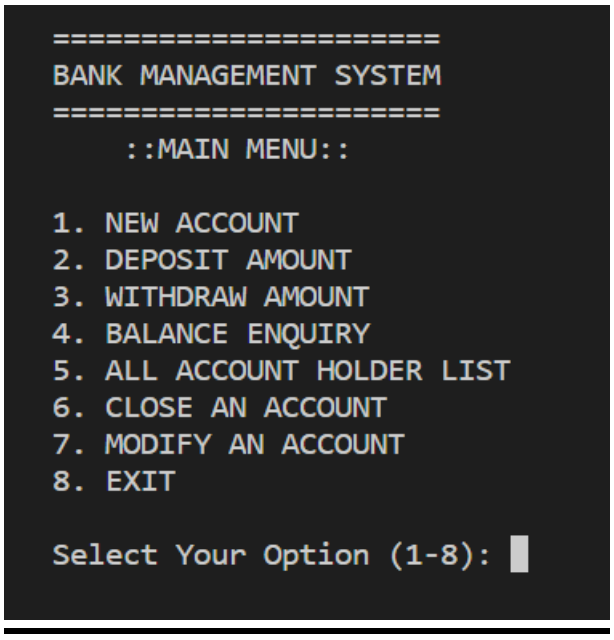
```
    if(!inFile)
    {
        cout<<"File could not be open !! Press any Key...";
        return;
    }
    cout<<"\n\n\t\tACCOUNT HOLDER LIST\n\n";
    cout<<"==========================================
===\n";
    cout<<"A/c no.      NAME          Type  Balance\n";
    cout<<"==========================================
===\n";
    while(inFile.read(reinterpret_cast<char *> (&ac), sizeof(account)))
    {
        ac.report();
    }
    inFile.close();
}

void deposit_withdraw(int n, int option)
{
    int amt;
    bool found=false;
    account ac;
    fstream File;
    File.open("account.dat", ios::binary|ios::in|ios::out);
    if(!File)
    {
        cout<<"File could not be open !! Press any Key...";
```

```
    return;
}
while(!File.eof() && found==false)
{
    File.read(reinterpret_cast<char *> (&ac), sizeof(account));
    if(ac.retacno()==n)
    {
        ac.show_account();
        if(option==1)
        {
            cout<<"\n\n\t\t\tTO DEPOSITSS AMOUNT";
            cout<<"\n\n\t\t\tEnter The amount to be deposited: ";
            cin>>amt;
            ac.dep(amt);
        }
        if(option==2)
        {
            cout<<"\n\n\t\t\tTO WITHDRAW AMOUNT";
            cout<<"\n\n\t\t\tEnter The amount to be withdraw: ";
            cin>>amt;
            int bal=ac.retdeposit()-amt;
            if(bal<0)
                cout<<"Insufficience balance";
            else
                ac.draw(amt);
          }
        int pos=(-1)*static_cast<int>(sizeof(ac));
```

```
            File.seekp(pos,ios::cur);//fn1353
            File.write(reinterpret_cast<char *> (&ac), sizeof(account));
            cout<<"\n\n\t\t\tRecord Updated";
            found=true;
           }
         }
    File.close();
    if(found==false)
       cout<<"\n\n\t\t\tRecord Not Found ";
}
```

IMPLEMENTED RESULTS :-

- **Main Menu :**



➢ Creating an Account :-

```
Enter the Account No. : 20210011

Enter the Name of the Account holder : Nitesh Malhotra

Enter Type of the Account (C/S) : S

Enter The Initial amount : 5000

Account Created..
```

➢ Deposit Amount :-

```
Enter The account No. : 20210011

Account No. : 20210011
Account Holder Name : Nitesh Malhotra
Type of Account : S
Balance amount : 5000

TO DEPOSITSS AMOUNT

Enter The amount to be deposited: 7655

Record Updated
```

➢ Withdraw Amount :-

```
Enter The account No. : 20210011

Account No. : 20210011
Account Holder Name : Nitesh Malhotra
Type of Account : S
Balance amount : 12655

TO WITHDRAW AMOUNT

Enter The amount to be withdraw: 500

Record Updated
```

- **All Account Holders List :-**

```
                ACCOUNT HOLDER LIST

=====================================================
A/c no.      NAME           Type  Balance
=====================================================
20210011          Nitesh Malhotra          S 12155
21020014          Paramveer          S 50000
```

- **Close an Account :-**

```
                        Enter The account No. : 20210011


 Record Deleted ..
```

- **Modify an Account :-**

```
Enter The account No. : 21020014

Account No. : 21020014
Account Holder Name : Paramveer
Type of Account : S
Balance amount : 50000

Enter The New Details of account

Account No. : 21020014
Modify Account Holder Name : Nitesh Malhotra

Modify Type of Account : C

Modify Balance amount : 5000


Record Updated
```

CONCLUSION

This project is developed to nurture the needs of user in a banking sector by embedding all the tasks of transactions taking place in a bank.

Future version of this software will still be much enhanced than the current version. Thus, the Bank Management System it is developed and executed successfully.

REFERENCES

The project file is prepared with C++ programming language using object oriented programming. The source code written by worth of understanding about the programming language. We took some references and ideas about the project and their significance from the described sources :-

- **GeeksForGeeks**
- **TutorialsPoint**
- **Wikipedia**