

OOPS Project on Banking Management System

Submitted by:

BIVAS DUTTA (RA1511008010714)

Submitted to:

Mr. P. Savaridassan

ACKNOLEDGEMENT

I take this opportunity to present my votes of thanks to all those guidepost who really acted as lightening pillars to enlighten our way throughout this project that has led to successful and satisfactory completion of this study.

I am highly thankful to Mr. P. Savaridassan for his support, valuable time and advice, whole-hearted guidance, sincere cooperation and pains-taking involvement during the study and in completing the assignment of preparing the said project within the time stipulated.

Bivas Dutta

(RA1511008010714)

PROPOSED SYSTEM

The following documentation is a project the **“BANKING SYSTEM”**. It is a detailed summary of all the drawbacks of the old system and how the new proposed system overcomes these shortcomings. The new system takes into account the various factors while designing a new system. It keeps into the account the Economical bandwidth available for the new system. The foremost thing that is taken care of is the Need and Requirements of the User.

PROBLEM STATEMENT:

* Problem statement was to design a module:
* Which is user friendly
* Which will restrict the user from accessing other user’s data.
* Which will help user in viewing his data and privileges.
* Which will help the administrator to handle all the changes.

FUNCTIONS TO BE PROVIDED:

The system will be user friendly and completely menu driven so that the users shall have no problem in using all options.

* The system will be efficient and fast in response.
* The system will be customized according to needs.
* Create new account
* Search ,edit and display customer details
* Display account holders
* Deposit money
* Withdraw money
* Balance enquiry

INTRODUCTION

In the existing system, most of the records are maintained on paper. It becomes very inconvenient to modify the data. In the existing system, here is a possibility that the same data in different registers may have different values which means the entries of the same data do not match. This inconsistent state does not supply the concrete information which poses a problem in the case information related to particular search record .Our project is very useful. User is no longer required to check his register in search of records, as now it can be searched over the software by choosing some options. The user need not to type in most of the information. He/she is just required to enter the desired options. On the whole it liberates the user from keeping lengthy manual records. In a nutshell, it abates the work load of an organization .In today’s world, no one likes to perform calculations on calculator or manually when computer is there. Everyone wants his/her work to be done by computer automatically and displaying the result for further manipulations. This term paper project is just an application of the language C in developing software’s .This is the program for keeping records of the bank ,that is the entire details of the customer ,savings ,how much money he/she can deposit and withdraw from the bank ,to open the new account ,create a new account ,display all account holders, balance enquiry etc.

**REQUIREMENT ANALYSIS**

This process is adopted when management of the system development, Personnel decide that the particular system needs improvement. The system development life cycle is the set of activities, carried out by the analyst, designers and users to develop and implement a system. The systems that are present in the nature follow common life cycle pattern. For example consider the raining system. Initially the rain falls into the river, river flows into sea, the sea water evaporates to form vapour, the vapors form clouds which again bring rain. Similarly consider a manmade system initially a system is analysed, designed and made operational by the efforts of system analysis. After successful operation or a number of users, the system becomes less and less effective by change in theenvironment. So these changes have to be incorporated in to the system by minor modifications. So the general activities from the life cycle of the system are given below:

Select ion and identification of the system to be studied

* Preliminary study
* Defining the system
* Design and development of the system
* Implementation of the system

SOURCE CODE

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// HEADER FILE USED IN PROJECT

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#include<iostream>

#include<fstream>

#include<cctype>

#include<iomanip>

#include<stdlib.h>

using namespace std;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// CLASS USED IN PROJECT

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

class account

{

int acno;

char name[50];

int deposit;

char type;

public:

void create\_account(); //function to get data from user

void show\_account() const; //function to show data on screen

void modify(); //function to add new data

void dep(int); //function to accept amount and add to balance amount

void draw(int); //function to accept amount and subtract from balance amount

void report() const; //function to show data in tabular format

int retacno() const; //function to return account number

int retdeposit() const; //function to return balance amount

char rettype() const; //function to return type of account

}; //class ends here

void account::create\_account()

{

cout<<"\nEnter The account No. : ";

cin>>acno;

cout<<"\n\nEnter The Name of The account Holder : ";

cin.ignore();

cin.getline(name,50);

cout<<"\nEnter Type of The account (C/S) : ";

cin>>type;

type=toupper(type);

cout<<"\nEnter The Initial amount(>=500 for Saving and >=1000 for current ) : ";

cin>>deposit;

cout<<"\n\n\nAccount Created..";

}

void account::show\_account() const

{

cout<<"\nAccount No. : "<<acno;

cout<<"\nAccount Holder Name : ";

cout<<name;

cout<<"\nType of Account : "<<type;

cout<<"\nBalance amount : "<<deposit;

}

void account::modify()

{

cout<<"\nAccount No. : "<<acno;

cout<<"\nModify Account Holder Name : ";

cin.ignore();

cin.getline(name,50);

cout<<"\nModify Type of Account : ";

cin>>type;

type=toupper(type);

cout<<"\nModify 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;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// function declaration

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void write\_account(); //function to write record in binary file

void display\_sp(int); //function to display account details given by user

void modify\_account(int); //function to modify record of file

void delete\_account(int); //function to delete record of file

void display\_all(); //function to display all account details

void deposit\_withdraw(int, int); // function to desposit/withdraw amount for given account

void intro(); //introductory screen function

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// THE MAIN FUNCTION OF PROGRAM

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

int main()

{

char ch;

int num;

intro();

do

{

system("cls");

cout<<"\n\n\n\tMAIN MENU";

cout<<"\n\n\t01. NEW ACCOUNT";

cout<<"\n\n\t02. DEPOSIT AMOUNT";

cout<<"\n\n\t03. WITHDRAW AMOUNT";

cout<<"\n\n\t04. BALANCE ENQUIRY";

cout<<"\n\n\t05. ALL ACCOUNT HOLDER LIST";

cout<<"\n\n\t06. CLOSE AN ACCOUNT";

cout<<"\n\n\t07. MODIFY AN ACCOUNT";

cout<<"\n\n\t08. EXIT";

cout<<"\n\n\tSelect Your Option (1-8) ";

cin>>ch;

switch(ch)

{

case '1':

write\_account();

break;

case '2':

cout<<"\n\n\tEnter The account No. : "; cin>>num;

deposit\_withdraw(num, 1);

break;

case '3':

cout<<"\n\n\tEnter The account No. : "; cin>>num;

deposit\_withdraw(num, 2);

break;

case '4':

cout<<"\n\n\tEnter The account No. : "; cin>>num;

display\_sp(num);

break;

case '5':

display\_all();

break;

case '6':

cout<<"\n\n\tEnter The account No. : "; cin>>num;

delete\_account(num);

break;

case '7':

cout<<"\n\n\tEnter The account No. : "; cin>>num;

modify\_account(num);

break;

case '8':

cout<<"\n\n\tThanks for using bank managemnt system";

break;

default :cout<<"\a";

}

cin.ignore();

cin.get();

}while(ch!='8');

return 0;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// function to write in file

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

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();

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// function to read specific record from file

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

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<<"\nBALANCE 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\nAccount number does not exist";

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// function to modify record of file

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

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\nEnter The New Details of account"<<endl;

ac.modify();

int pos=(-1)\*static\_cast<int>(sizeof(account));

File.seekp(pos,ios::cur);

File.write(reinterpret\_cast<char \*> (&ac), sizeof(account));

cout<<"\n\n\t Record Updated";

found=true;

}

}

File.close();

if(found==false)

cout<<"\n\n Record Not Found ";

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// function to delete record of file

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

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\n\tRecord Deleted ..";

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// function to display all accounts deposit list

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void display\_all()

{

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();

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// function to deposit and withdraw amounts

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

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\tTO DEPOSITE AMOUNT ";

cout<<"\n\nEnter The amount to be deposited";

cin>>amt;

ac.dep(amt);

}

if(option==2)

{

cout<<"\n\n\tTO WITHDRAW AMOUNT ";

cout<<"\n\nEnter The amount to be withdraw";

cin>>amt;

int bal=ac.retdeposit()-amt;

if((bal<500 && ac.rettype()=='S') || (bal<1000 && ac.rettype()=='C'))

cout<<"Insufficience balance";

else

ac.draw(amt);

}

int pos=(-1)\*static\_cast<int>(sizeof(ac));

File.seekp(pos,ios::cur);

File.write(reinterpret\_cast<char \*> (&ac), sizeof(account));

cout<<"\n\n\t Record Updated";

found=true;

}

}

File.close();

if(found==false)

cout<<"\n\n Record Not Found ";

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// INTRODUCTION FUNCTION

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void intro()

{

cout<<"BANK\n\n";

cout<<"MANAGEMENT\n\n";

cout<<"SYSTEM";

cout<<"\n\n\n\nMADE BY :SHUBHENDU & BIVAS";

cout<<"\n\nCOLLEGE :SRM UNIVERSITY";

cin.get();

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// END OF PROJECT

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

OUTPUT