OOP PROJECT

E WALLET MANAGEMENT SYSTEM

GUIDED BY VIVEKRAJ SIR

NAME:VISHNUKUMAR P

ROLLNO:20BCS138

PROBLEM STATEMENT:

THIS IS THE SYSTEM, WHERE WE CAN STORE MONEY IN ONLINE WALLET AND USE IT FOR EVERY PURCHASES USING WALLET MONEY DIRECTLY WITHOUT USING ANY CARDS.

* 1. WE CAN SEND MONEY TO OTHER USERS WHO HAS WALLET OF THE SAME SYSTEM AND WE CAN ADD ACCOUNT OR DELETE ACCOUNT.
  2. WE CAN CREATE NEW WALLET AND SET AN WALLET ID WHICH WILL BE USED FOR YOUR TRANSACTIONS.

WE CAN ADD MONEY TO THE WALLET FROM THE WALLET AND VICE VERSA.

REQUIREMENTS:

* CREATE ACCOUNT/WALLET
* SEND MONEY
* LOAD WALLET
* DELETE WALLET
* DELETE ACCOUNT
* DISPLAY PARTICULAR ACCOUNT
* DISPLAY PARTICULAR WALLET
* DISPLAY ALL ACCOUNTS

DISPLAY ALL WALLETS

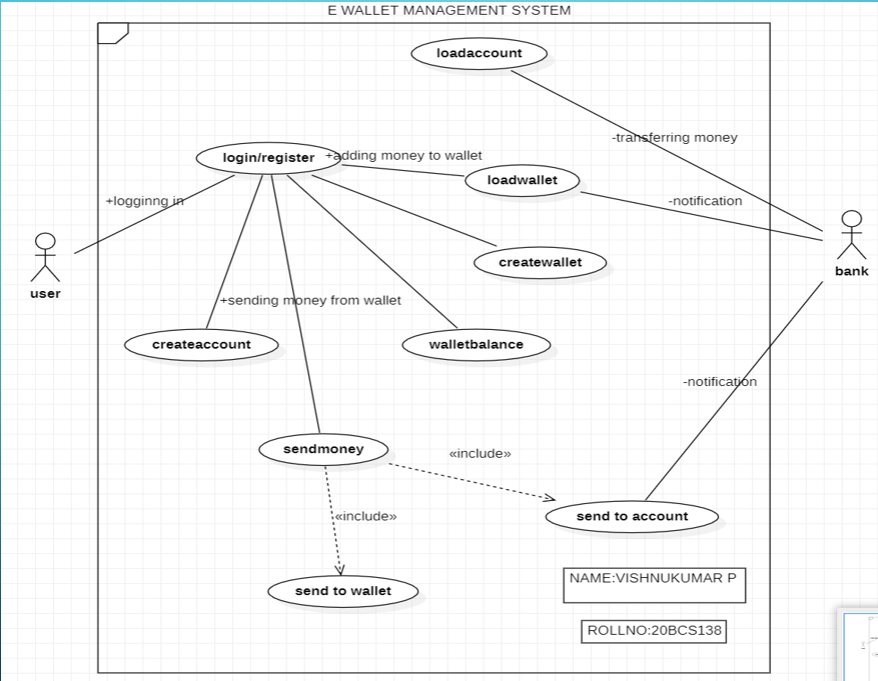
AIMS:

TO CREATE A ONLINE WALLET WHICH STORES THE USER’S MONEY AND USED FOR THE TRANSACTIONS ON PURHASING PRODUCTS WITHOUT USING ANY CARDS

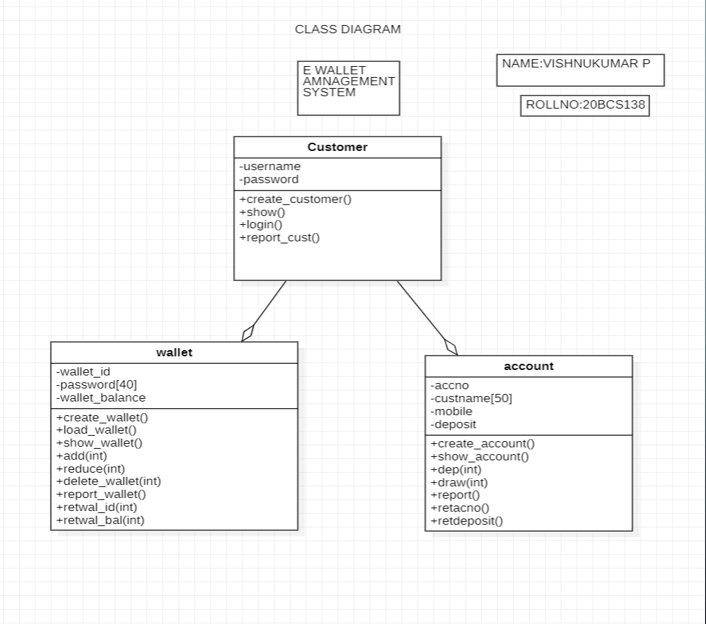
PROCEDURE:

WE ARE GOING OT USE OBJECT ORIENTED PROGRAMMING AND CREATING THE CLASSES REALTED THE REQUIREMENTS WHICH CONTAINS ATTRIBUTES AND FUNCTIONS WHICH WILL BE OPERATED ON THOSE ATTRIBUTES.

USE CASE DIAGRAM:



CLASS DIAGRAM:



CODE:

#include<iostream>

#include<fstream>

#include<cctype>

#include<iomanip>

using namespace std;

class account

{

protected:

int accno;

char custname[50];

long int mobile;

int deposit;

public:

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

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

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(); //function to show data in tabular format

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

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

};

void account::create\_account()

{

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

cin>>accno;

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

cin.ignore();

cin.getline(custname,50);

cout<<"enter your mobile number "<<endl;

cin>>mobile;

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. : "<<accno;

cout<<"\nAccount Holder Name : "<<custname;

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

}

void account::dep(int x)

{

deposit+=x;

}

void account::draw(int x)

{

deposit-=x;

}

void account::report()

{

cout<<accno<<setw(32)<<custname<<setw(30)<<mobile<<setw(20)<<deposit<<endl;

}

int account::retacno() const

{

return accno;

}

int account::retdeposit() const

{

return deposit;

}

class wallet

{

int wallet\_id;

char password[40];

int wallet\_balance;

public:

void create\_wallet();

void load\_wallet(int);

void show\_wallet();

void add(int);

void reduce(int);

void delete\_wallet(int);

void report\_wallet();

int retwal\_id();

int retwal\_bal();

};

void wallet::create\_wallet()

{

cout<<"\nEnter the wallet id :";

cin>>wallet\_id;

cout<<"\n\nEnter The password : ";

cin.ignore();

cin.getline(password,40);

cout<<"enter your initial wallet\_balance"<<endl;

cin>>wallet\_balance;

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

}

void wallet::show\_wallet()

{

cout<<"wallet\_id :"<<wallet\_id<<endl;

cout<<"wallet\_balance:"<<wallet\_balance<<endl;

}

void wallet::add(int x)

{

wallet\_balance+=x;

}

void wallet::reduce(int x)

{

wallet\_balance-=x;

}

void wallet::report\_wallet()

{

cout<<wallet\_id<<setw(30)<<password<<setw(20)<<wallet\_balance<<endl;

}

int wallet::retwal\_id()

{

return wallet\_id;

}

int wallet::retwal\_bal()

{

return wallet\_balance;

}

class customer

{

char username[20];

char password[20];

public:

void create\_customer();

void show(char username,char password);

void login(char username,char password);

void report\_cust();

};

void customer::create\_customer()

{

cout<<"\nEnter The username :";

cin.ignore();

cin.getline(username,20);

cout<<"\n\nEnter The Password : ";

cin.ignore();

cin.getline(password,20);

}

void customer::show(char username,char password)

{

cout<<"username:"<<username<<endl;

cout<<"password:"<<password<<endl;

}

void customer::login(char username,char password)

{

customer c;

char un;

char pw;//last two are comparison string

cout << "Enter username :";

cin >> un;

cout << "Enter password :";

cin >> pw;

ifstream inFile;

inFile.open("customer.txt",ios::binary);

if(!inFile)

{

cout<<"File could not be open !! Press any Key...";

}

while(inFile>>username>>password)

{

if (username == un)

{

if(password == pw)

{

int count =1;

}

else

{

cout<<"login error"<<endl;

}

}

}

inFile.close();

int count;

if(count==1)

{

cout<<"your login is successful"<<endl;

}

else

{

cout<<"register first"<<endl;

}

}

void customer::report\_cust()

{

cout<<username<<setw(30)<<password<<endl;

}

void write\_account();

void write\_wallet();

void deposit\_withdraw(int, int);

void deposit\_withdraw\_w(int, int);

void delete\_account(int);

void delete\_wallet(int);

void display\_sp(int);

void display\_all();

void display\_spw(int);

void display\_allw();

void create\_customer();

void display\_allc();

int main()

{

int cho;

char un;

char pw;

cout<<"if you are a old customer press 1 or press 2"<<endl;

cin>>cho;

customer c;

if(cho == 1)

{

cout<<"Successfully Logged in"<<endl;

cout<<"if you already have an account and wallet:then choose the option you needed:"<<endl;

cout<<"orlelse first create your account and open the wallet"<<endl;

int ch;

int num;

do{

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. NEW WALLET";

cout<<"\n\n\t08. LOAD WALLET";

cout<<"\n\n\t09. SEND MONEY TO WALLET";

cout<<"\n\n\t10. BALANCE ENQUIRY";

cout<<"\n\n\t11. ALL WALLET HOLDER LIST";

cout<<"\n\n\t12. CLOSE AN wallet";

cout<<"\n\n\t13. SEND MONEY TO ACCOUNT FROM WALLET";

cout<<"\n\n\t14. ALL CUSTOMER DETAILS";

cout<<"\n\n\t15. SEND MONEY TO WALLET FROM ACCOUNT";

cout<<"\n\n\t16. EXIT";

cout<<"\n\n\tSelect Your Option (1-15) ";

cin>>ch;

system("cls");

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:

write\_wallet();

break;

case 8:

cout<<"\n\n\tEnter the wallet\_id: "; cin>>num;

deposit\_withdraw\_w(num, 1);

break;

case 9:

cout<<"\n\n\t Enter the wallet\_id of your's and recipient wallet id:";

cin>>num;

deposit\_withdraw\_w(num,2);

cout<<"\n\n\t Enter the wallet\_id of your's and recipient wallet id:";

cin>>num;

deposit\_withdraw\_w(num,1);

break;

case 10:

cout<<"Enter your wallet\_id:"; cin>>num;

display\_spw(num);

break;

case 11:

display\_allw();

break;

case 12:

cout<<"\n\n\tEnter The WALLET ID. : "; cin>>num;

delete\_wallet(num);

break;

case 13:

cout<<"\n\n\t Enter the wallet\_id of your's and recipient wallet id:";

cin>>num;

deposit\_withdraw\_w(num,2);

cout<<"\n\n\t Enter the wallet\_id of your's and recipient wallet id:";

cin>>num;

deposit\_withdraw(num,1);

break;

case 14:

display\_allc();

break;

case 15:

cout<<"\n\n\t Enter the wallet\_id of your's and recipient wallet id:";

cin>>num;

deposit\_withdraw(num,2);

cout<<"\n\n\t Enter the wallet\_id of your's and recipient wallet id:";

cin>>num;

deposit\_withdraw\_w(num,1);

break;

case 16:

cout<<"thanks you"<<endl;

break;

default :cout<<"\a";

}

cin.ignore();

cin.get();

}while(ch!= 16);

return 0;

}

else if(cho == 2)

{

customer c;

char username;

char password;

c.create\_customer();

ofstream f1("customer.txt",ios::app);

f1<<username<<setw(20)<<password<<endl;

system("cls");

cout<<"successfully registered"<<endl;

}

else

{

cout<<"wrong put"<<endl;

}

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 write\_wallet()

{

wallet wal;

ofstream outFilew;

outFilew.open("wallet.dat",ios::binary|ios::app);

wal.create\_wallet();

outFilew.write(reinterpret\_cast<char \*> (&wal), sizeof(wallet));

outFilew.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<<"\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";

}

void display\_spw(int n)

{

wallet wal;

bool flag=false;

ifstream inFilew;

inFilew.open("wallet.dat",ios::binary);

if(!inFilew)

{

cout<<"File could not be open !! Press any Key...";

return;

}

cout<<"\nBALANCE DETAILS\n";

while(inFilew.read(reinterpret\_cast<char \*> (&wal), sizeof(wallet)))

{

if(wal.retwal\_id() == n)

{

wal.show\_wallet();

flag=true;

}

}

inFilew.close();

if(flag==false)

cout<<"\n\nAccount number does not exist";

}

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 mobileno Balance\n";

cout<<"====================================================\n";

while(inFile.read(reinterpret\_cast<char \*> (&ac), sizeof(account)))

{

ac.report();

}

inFile.close();

}

void display\_allw()

{

wallet wal;

ifstream inFilew;

inFilew.open("wallet.dat",ios::binary);

if(!inFilew)

{

cout<<"File could not be open !! Press any Key...";

return;

}

cout<<"\n\n\t\tWALLET HOLDER LIST\n\n";

cout<<"====================================================\n";

cout<<"wallet\_id password Balance\n";

cout<<"====================================================\n";

while(inFilew.read(reinterpret\_cast<char \*> (&wal), sizeof(wallet)))

{

wal.report\_wallet();

}

inFilew.close();

}

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 ..";

}

void delete\_wallet(int n)

{

wallet wal;

ifstream inFilew;

ofstream outFilew;

inFilew.open("wallet.dat",ios::binary);

if(!inFilew)

{

cout<<"File could not be open !! Press any Key...";

return;

}

outFilew.open("Temp.dat",ios::binary);

inFilew.seekg(0,ios::beg);

while(inFilew.read(reinterpret\_cast<char \*> (&wal), sizeof(wallet)))

{

if(wal.retwal\_id() != n)

{

outFilew.write(reinterpret\_cast<char \*> (&wal), sizeof(wallet));

}

}

inFilew.close();

outFilew.close();

remove("wallet.dat");

rename("Temp.dat","wallet.dat");

cout<<"\n\n\tRecord Deleted ..";

}

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<0))

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 ";

}

void deposit\_withdraw\_w(int n, int option)

{

int amt;

bool found=false;

wallet wal;

fstream Filew;

Filew.open("wallet.dat", ios::binary|ios::in|ios::out);

if(!Filew)

{

cout<<"File could not be open !! Press any Key...";

return;

}

while(!Filew.eof() && found==false)

{

Filew.read(reinterpret\_cast<char \*> (&wal), sizeof(wallet));

if(wal.retwal\_id() == n)

{

wal.show\_wallet();

if(option==1)

{

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

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

cin>>amt;

wal.add(amt);

}

if(option==2)

{

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

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

cin>>amt;

int bal=wal.retwal\_bal() - amt;

if(bal<0)

cout<<"Insufficience balance";

else

wal.reduce(amt);

}

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

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

Filew.write(reinterpret\_cast<char \*> (&wal), sizeof(wallet));

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

found=true;

}

}

Filew.close();

if(found==false)

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

}

void display\_allc()

{

customer c;

ifstream inFilec;

inFilec.open("customer.txt",ios::binary);

if(!inFilec)

{

cout<<"File could not be open !! Press any Key...";

return;

}

cout<<"\n\n\t\tCUSTOMER LIST\n\n";

cout<<"====================================================\n";

cout<<"NAME PASSWORD\n";

cout<<"====================================================\n";

while(inFilec.read(reinterpret\_cast<char \*> (&c), sizeof(customer)))

{

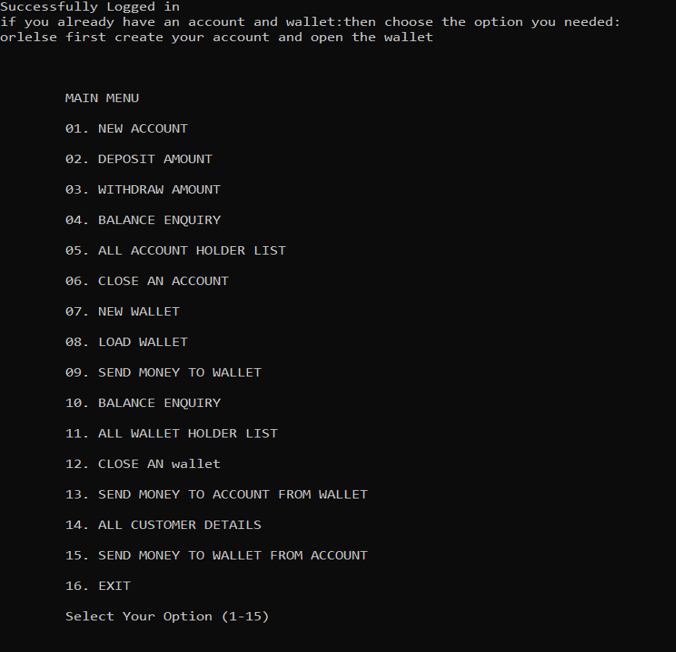
c.report\_cust();

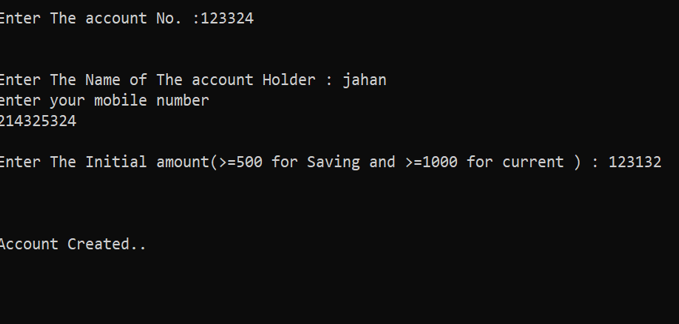
}

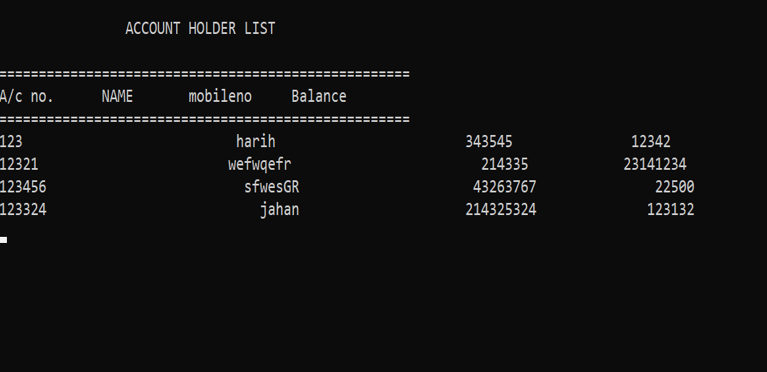
inFilec.close();

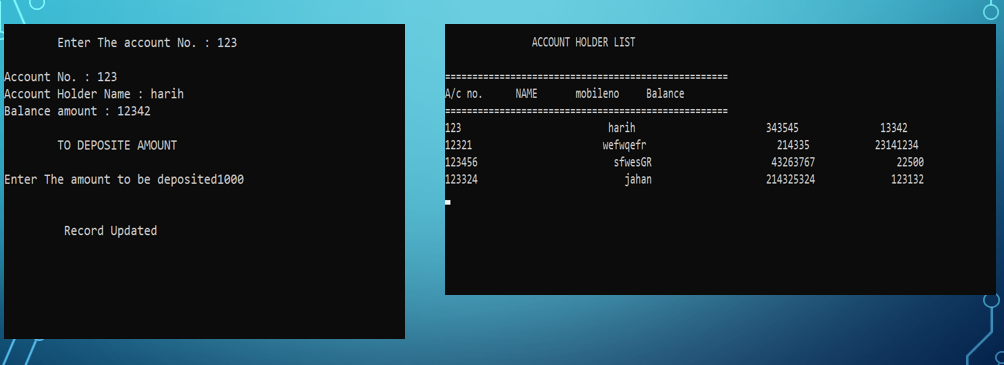
}

RESULT:









CONCLUSION:

HEREBY, WE HAVE CREATED THE E WALLET WHICH WILL STORE THE MONEY IN ITS WALLET AND IT WILL BE USED FOR THE EVERY PURCHASE, THANKING YOU , SIR.