**PROECT REPORT**

**ATM SERVICES**



**BHAVIT MAHAJAN\_20103240\_B8**

**YASH JAUHARI\_20103219\_B8**

**ANKIT TYAGI\_20103230\_B8**

**TANISHQ SHARMA\_20103214\_B8**

**SUBMITTED TO:**

**DR. ANITA SAHOO**

**INTRODUCTION**

ATM service application is primarily meant for performing card less transactions of various types. ATM service application will provide the basic set of features of performing various types of **transaction**s, **updating records**, **checking balance** and **generating mini statement**. This mini project in C++ ATM service allows to perform regular ATM operations like using an actual **ATM kiosk**. One can update pin, mobile number and surf other account holder related records. Concept of classes, objects, constructor, virtual functions, abstract class, inheritance, friend function, operator overloading, structure and exception handling have been extensively used.

ATM service in C++ is a console application without graphics. This is a **menu driven program**. The source code is complete and totally error-free. It is compiled in Code Blocks with GCC compiler. Functions and data structure have been used. Withdrawing, depositing and transferring money to another account are the basic functions which make up the main menu of this ATM service application. The mini statement function displays the account holder’s name, account number and phone number, current balance and the transaction performed during the current usage. This code has been made as simple as possible.

**Menu** is an abstract class from which account holder class is inherited. The **account holder class** contains records of transactions performed by the customer.

**OBJECTIVE**

The main objective of this ATM system which is developed in C++ is to provide a flexible solutions for banking sector. With this system, bank customers will have ease in using this system and will be able to perform all their operations.

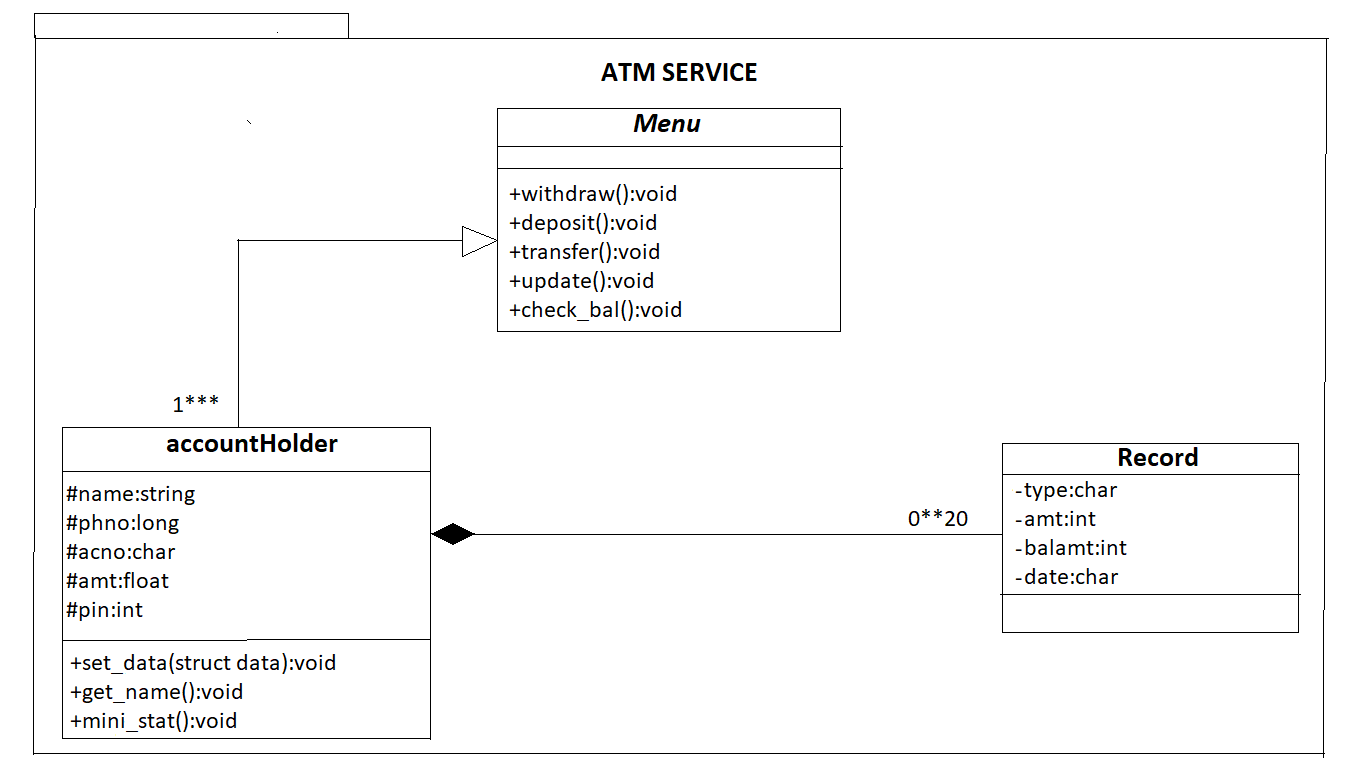
This system enable its users:

* Withdrawal of Money
* Deposition of Money
* Transfer of Money to another Account
* Update

1. Phone Number
2. Pin Number

* Check Current Balance
* Print Mini Statement

**CLASS DIAGRAM**



**CODE**

#include<stdio.h>

#include <stdlib.h>

#include<string.h>

#include<time.h>

#include<iostream>

using namespace std;

int n1=0,ind;

struct data

{

string name;

long phno;

char \*acno=new char[11];

float amt;

int pin;

};

data d[4];

time\_t curtime;

class Menu

{

public:

virtual void withdraw()=0;

virtual void deposit()=0;

virtual void transfer()=0;

virtual void update()=0;

virtual void check\_bal()=0;

};

class record

{

public:

char type[10];

int amt;

float balamt;

char date[40];

};

class accountholder : public Menu

{

protected:

string name;

long phno;

char \*acno;

float amt;

int pin;

class record r1[20];

public:

accountholder()

{

phno=0;

acno=new char[11];

amt=0.0;

pin=0;

}

friend ostream &operator<<(ostream &output,accountholder &);

void set\_data(struct data s1);

string get\_name();

void withdraw();

void deposit();

void transfer();

void update();

void check\_bal();

void mini\_stat(accountholder &a);

};

string accountholder::get\_name()

{

return name;

}

void accountholder::set\_data(struct data s1)

{

name=s1.name;

strcpy(acno,s1.acno);

phno=s1.phno;

amt=s1.amt;

pin=s1.pin;

}

void accountholder::withdraw()

{

time(&curtime);

int note500,note200,note100;

int withdraw,f=0;

cout<<"Welcome to the Withdraw Section\n";

cout<<"Your Current Balance is Rs" <<amt<<endl;

cout<<" Note: 1.Amount should be multiple of hundred\n 2.Amount should be less than 20,000(Twenty Thousand)\n";

cout<<"Enter Amount: ";

cin>>withdraw;

if(withdraw%100==0)

{

if(amt-2000<withdraw)

{

cout<<"\tInsufficient Balance\n\tPlease Try Again with amount less than your current balance, i.e.,\nAvailable Balance= "<<amt<<endl;

}

else

{

if(withdraw>20000)

{

cout<<"\tWithdrawal Limit Exceeded\n\tTransaction Failed!\n";

}

else

{

amt-=withdraw;

r1[n1].balamt=amt;

r1[n1].amt=withdraw;

strcpy(r1[n1].date,ctime(&curtime));

strcpy(r1[n1++].type,"Withdrawed");

note500=(withdraw/500)-1;

if(note500<=0)

note500=0;

withdraw=withdraw-(note500\*500);

note200=(withdraw/200)-1;

if(note200<=0)

note200=0;

withdraw=withdraw-(note200\*200);

note100=withdraw/100;

f=1;

}

}

}

else

{

cout<<"\tEnter the amount in multiples of 100 only!\n";

}

if(f==1)

{

cout<<"\tDispensing the following notes:\n\t1.Rs500->"<<note500 <<"\n\t2.Rs200-> "<<note200<<"\n\t3.Rs100-> "<<note100;

cout<<endl;

cout<<"Your Current Balance is Rs "<<amt;

cout<<endl;

}

}

void accountholder::deposit()

{

time(&curtime);

int deposit,n500,n200,n100;

cout<<"Welcome to the Deposit Section\n";

cout<<"Note: 1.Amount should be multiple of 100.\n 2.Amount should be less than 10,000(Ten Thousand).\n";

cout<<"Enter Amount: ";

cin>>deposit;

cout<<"Enter the number of Rs 500, Rs 200 and Rs 100 note(s)\n";

cout<<"Enter no of 500 note(s)\n";

cin>>n500;

cout<<"Enter no of 200 note(s)\n";

cin>>n200;

cout<<"Enter no of 100 note(s)\n";

cin>>n100;

int mo=n500\*500+n100\*100+n200\*200;

if(mo==deposit)

{

if(deposit>10000)

{

cout<<"\tDeposited amount exceeded Rs.10000\n\t\tTransaction Failed!\n";

}

else

{

strcpy(r1[n1].date,ctime(&curtime));

amt=amt+deposit;

r1[n1].balamt=amt;

r1[n1].amt=deposit;

strcpy(r1[n1++].type,"Deposited");

cout<<"\tTransaction Successful\n\tAvailable Balance:Rs "<<amt<<endl;

}

}

else

cout<<"\tAmount entered is not equal to Amount Deposited\n\t\tTransaction Failed!\n";

}

void accountholder::transfer()

{

int transfer;

char acc[11];

cout<<"Welcome to the Transfer Section\n";

cout<<"Enter the Beneficiary Account Number: ";

fflush(stdin);

gets(acc);

cout<<"Note: Amount should be less than 20,000(Twenty Thousand)\n";

cout<<"Enter Amount: ";

cin>>transfer;

if(transfer>20000)

{

cout<<"\tTransfer amount exceeded Rs.20000\n\tTransaction Failed!\n";

}

else if(amt-2000<transfer)

{

cout<<"\tInsufficient Balance in your account \nPlease Try Again with amount less than your current balance, i.e.,\n\tBalance= "<<amt<<"\n\t\tTransaction Failed\n";

}

else

{

amt=amt-transfer;

strcpy(r1[n1].date,ctime(&curtime));

r1[n1].balamt=amt;

r1[n1].amt=transfer;

strcpy(r1[n1++].type,"Transfered");

cout<<"\tTransaction Successful\n\tAvailable Balance:Rs "<<amt<<endl;

}

}

void accountholder::check\_bal()

{

cout<<"\nYour Current Account Balance is Rs "<<amt<<endl<<endl;

}

void accountholder::update()

{

int choice,new\_pin,tempin; int f=0;

char new\_phno[11];

cout<<"Welcome to the Update Section\n";

cout<<"Enter your choice\n";

b:

cout<<"1.Update Phone number\n2.Update PIN\n3.Return to main menu \n";

cin>>choice;

switch(choice)

{

case 1:

try{

cout<<"Enter new phone number\n"; //phone no

cin>>new\_phno;

int l=strlen(new\_phno);

if(l!=10)

throw 1;

for(int i=0;i<10;i++)

{

if(new\_phno[i]<48 || new\_phno[i]>57)

throw 1;

}

}

catch(int x)

{

cout<<"Enter valid Phone Number"<<endl;

f=1;

}

if(f==0)

{

phno=stoi(new\_phno);

cout<<"Updated phone number is "<<phno<<endl;

}

break;

case 2:

c:

cout<<"Enter a new four digit PIN.\n"; //pin

cin>>new\_pin;

cout<<"Confirm your PIN.\n";

cin>>tempin;

if(tempin!=new\_pin)

{

cout<<"\tPIN did not match!\n";

goto c;

}

else

{

pin=new\_pin;

cout<<"\tPIN Updated Successfully!\n";

}

break;

case 3:

break;

default :

cout<<"\tEnter correct choice!\n\n";

goto b;

}

}

ostream &operator<<( ostream &output,accountholder &a)

{

time\_t t;

time(&t);

cout<<" Day Date Time \n "<< ctime(&t);

cout<<"Account Number: "<<a.acno<<"\nAccount Holder: "<<a.name<<"\nPhone Number : "<<a.phno;

if(n1!=0)

{

if(n1==1)

cout<<"\n STATEMENT OF LAST "<<n1<<" RECORD is as follows:\n";

else

cout<<"\n STATEMENT OF LAST "<<n1<<" RECORDS is as follows:\n";

cout<<"Amount\t\t Type\t\tAvailable Balance\tDate and Time\n";

for(int i=0;i<n1;i++)

{

cout<<"Rs."<<a.r1[i].amt<<"\t\t"<<a.r1[i].type<<"\t"<<a.r1[i].balamt<<"\t\t\t"<<a.r1[i].date<<endl;

}

}

else

cout<<"\n\tNo Transaction History Available\n";

cout<<"The current Account Balance is Rs."<<a.amt<<endl<<endl;

cout<<"\*\*\*For any dispute/query, call us at \n\t toll-number +91 9473030300.\n\t For complaints, don't call us.\n";

cout<<"#########Thank you for using our ATM Services#########\n ^.^.^.^.^Hope to serve you again^.^.^.^.^\n\t\tVisit us soon\n";

return output;

}

void accountholder::mini\_stat(accountholder &a)

{

cout<<"#The JIIT ATM Services#\nJIIT Sector-62 Premises\n\n";

cout<<"MINI STATEMENT\n\n";

cout<<a;

}

int main()

{

int ch;

char cont;

d[0].amt=200000; d[0].name="BHAVIT MAHAJAN"; d[0].phno=9911111111; d[0].pin=1234; strcpy(d[0].acno,"JIT1123456");

d[1].amt=300000; d[1].name="YASH JAUHARI"; d[1].phno=9922222222; d[1].pin=1235; strcpy(d[1].acno,"JIT1123457");

d[2].amt=500000; d[2].name="ANKIT TYAGI"; d[2].phno=9933333333; d[2].pin=1236; strcpy(d[2].acno,"JIT1123458");

d[3].amt=400000; d[3].name="TANISHQ SHARMA"; d[3].phno=9944444444; d[3].pin=1237; strcpy(d[3].acno,"JIT1123459");

cout<<"#Welcome to the JIIT ATM Services#\nJIIT Sector-62 Premises\n\n";

char ac[11];

int pin,flag=0,flag2=0;

cout<<"Enter Acc no\n";

gets(ac);

revisit:

for(int i=0;i<4;i++)

{

if(strcmp(ac,d[i].acno)==0)

{ind=i;

flag2=1;

break;

}

}

if(flag2==0)

{

cout<<"You have entered Wrong Account Number\nRe-enter the Account Number\n";

gets(ac);

goto revisit;

}

cout<<"Enter PIN\n";

for(int i=0;i<3;i++)

{

cin>>pin;

if(d[ind].pin==pin)

{

flag=1;

break;

}

else

{

if(i!=2)

cout<<"Wrong PIN!!\tRe-enter the PIN \t";

}

}

if(flag==1)

{

Menu \*m1;

accountholder a;

m1=&a;

a.set\_data(d[ind]);

do

{

cout<<"\t\tWElCOME "<<a.get\_name()<<endl;

cout<<"Enter \n1.Withdraw\n2.Deposit\n3.Transfer\n4.Update Record\n5.Check Balance\n6.Mini Statement\n";

cin>>ch;

switch(ch)

{

case(1): m1->withdraw();break;

case(2): m1->deposit();break;

case(3): m1->transfer();break;

case(4): m1->update();break;

case(5): m1->check\_bal();break;

case(6): a.mini\_stat(a);break;

default: cout<<"Wrong Choice\n";

}

fflush(stdin);

cout<<"enter Y to continue and N to exit.\n";

cin>>cont;

}while(cont=='y'||cont=='Y');

}

else

{ cout<<"You have exceeded the number of attempts to enter the correct PIN.";

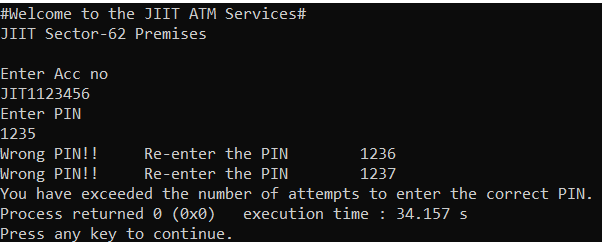
exit(0);

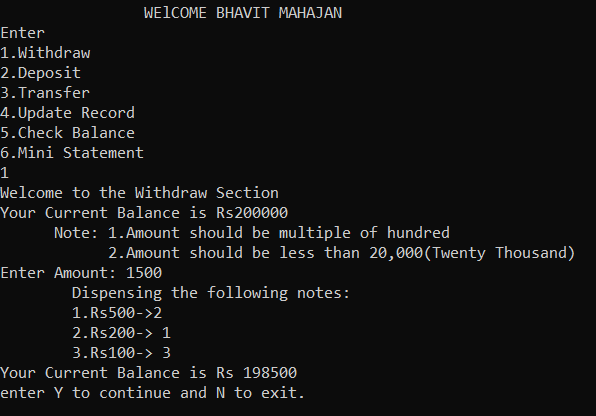
}

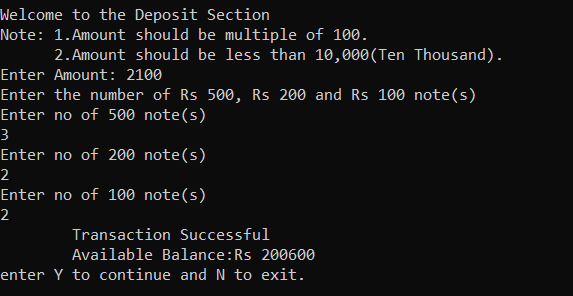
return 0;

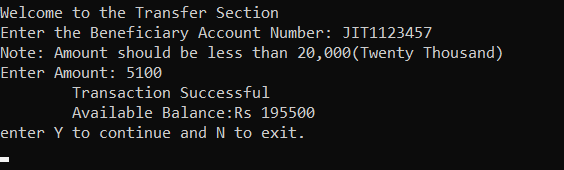
}

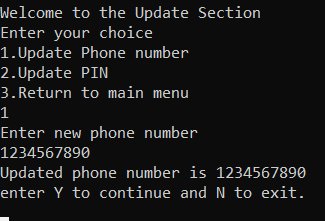
**RESULTS**

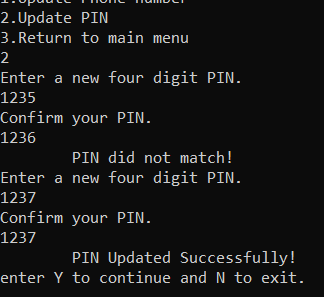
****

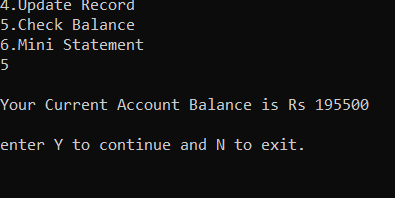
****

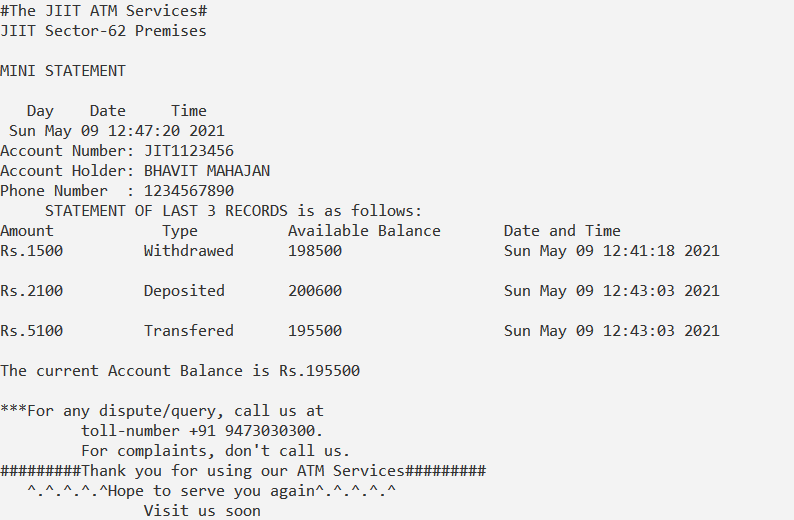
****

****

****

****

****

****

**CONCLUSION**

The application software has been implemented successfully by using test cases. The coding language used is C++ language. This application is used to

* Withdraw Money
* Deposit Money
* Transfer Money
* Update Records
* Check Account Balance
* Mini Statement

This complete project has been coded on **codeblocks** platform .

**REFERENCES**

1. E Balagurusamy, Computer

Programming, McGraw Hill Education, 1st Edition

1. Geeks for Geeks
2. C++: The Complete Reference 4th Edition By Schildt Herbert