Fatima Jinnah Women University

Department Of Software Engineering

PROJECT

Course Title
Object Oriented Programming

Submitted To
Dr. Bushra Basheer

Submitted By
Mariam Fatima
Registration No: 2021-BSE-020

Date of Submission
June 04, 2022

Table of Contents

REPORT

1. Introduction

ATM (automated teller machine) is an electronic banking outlet. This ATM management system enables the persons of financial institutions like banks to carry out money transactions such as cash withdrawal, and balance inquiries. This program helps to do all the functionalities more quickly and accurately. This system is designed to easily deposit, withdraw and save money. It isa more efficient way rather than using a cheque. A customer will use a plastic card with a magnetic strip or with a chip, known as an ATM card or credit card to withdraw money. When this is implemented, the user who uses this will be able to see all the services and information provided by the ATM. The data will be stored in a database and will be retrieved whenever it is required.

Purpose

- The main purpose of an ATM is that it allows customers to complete basic transactions without the help of a branch teller.
- Anyone with a credit card can access the cash at any time.
- This will allow the customers to perform a quick self-transaction such as deposits, cash withdrawals, bill payments, and transfers between accounts.
- It will also help an operator to manage cash more efficiently, analyze transaction data, maintain transaction logs, and coordinate tickets and services.
- ATM management system will reduce the pressure on banks staff and will also reduce the time of customers.
- ATMs are also great for travelers. They can withdraw cash from any country or state across the country or outside the country.

2. Functions used in this system

Classes

We have used classes in this project. We have made an “account” name class. From parent class we derived two classes “current account” and “saving account”.

Inheritance

We have used inheritance in this program. Inheritance means when a class drives from another class. The child class “current account” and “saving account” will inherit all the properties and methods from a parent class “account”.

Access specifier

We have used access specifiers **public**, **protected,** and **private.** In the public specifier, we added functions of pin details, display details, get function, and display all customer details Public

specifiers are accessible by all the members and objects. Private are accessible within the class. Protected are those which are not accessible outside of the class but are accessed in the inherited class. Protected specifiers in this program are saving balance, current balance, deposit, and withdraw.

Files

We have used files in this project. In file we keep the record of customer and how much he deposit, withdraw and transfer all that information would be saved in files.

Pin code

We have made a function "pn" in the public access specifier. In this you have to enter your pin code to access the money transaction process. When you call the pin function, it will display a message" Please Enter Your P.I.N" and if you enter the wrong pin it will exit. You can enter your pin 3 time. If your number exceeds the 3 it willexit from the program.

Get account detail

In this function we get the information of customer his/her name, account type and account number.

Display detail

This function will display the details of Customers name, Account type and Account number **Current Account**

We make a child class "current account" derived from a parent class "account". In public access specifier we have made a constructor with a functions of display, deposit, withdraw and transfer.

Display

This will display the current balance of the account.

Deposit

When you will deposit the money, it will ask you "Enter amount to deposit:" The deposit amount will be added into the current balance and it will show us the amount after deposit.

Withdraw

You can withdraw your money when you do transaction. You will enter the amount you want to withdraw but in the multiples of 500. If you will add the amount in the multiples of 500 or equal to 500, your amount will be withdrawn. Now the total amount left would be the **current balance-withdraw** and after withdraw it will display the message "THANKYOU".

If the amount you enter is not in the multiples of 500 then it will ask you “do you want to do another transaction”. If you choose yes then you can do another transaction but in the multiples of 500.

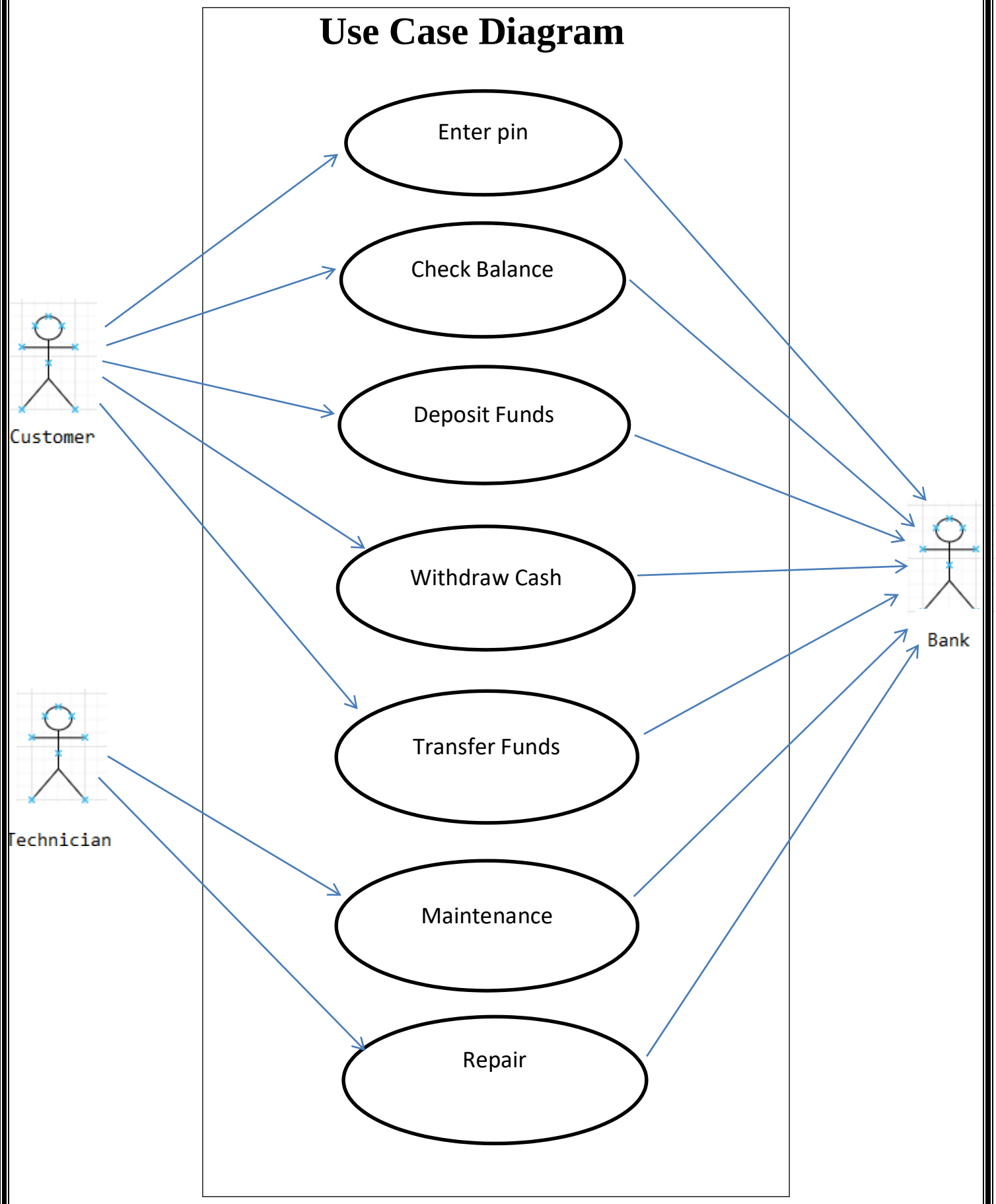
Transfer

You can also transfer the amount from your account to others. We have created a new account variable to which you will transfer your amount. When you will transfer your amount it will be deducted from your account balance and will be added in to a new account balance. And after transfer it will show you your remaining balance.

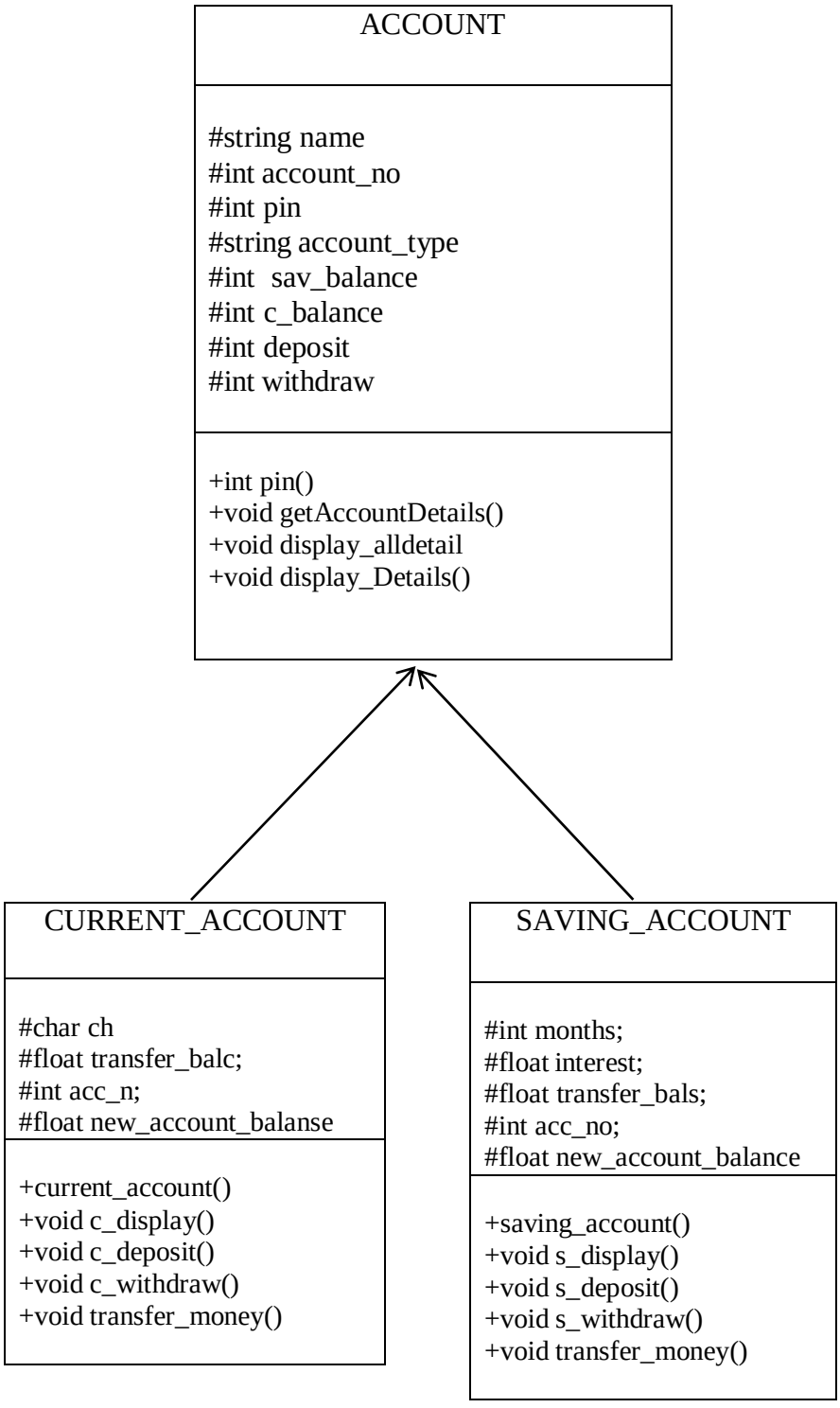
3. Saving Account

In saving account deposit, transfer and withdraw will be done as same as current account. But in saving account if your months exceeds from five months after saving balance in saving account then interest will be added in your account.

Use Case Diagram



Class Diagram



Program

```
#include<iostream>
#include<string>
#include<fstream>
using namespace std;
class ACCOUNT
{
private:
string  name;
int account_no;
int pin;
string account_type;
protected:
int sav_balance;
int c_balance;
int deposit;
int withdraw;
public:
int pn()
{
int PIN=12;
cout<<"PLEASE ENTER YOUR P.I.N"<<endl;
for (int i=0; i<3; i++)
{
cin>>pin;
```

```
if(pin==PIN)
{
cout << "\n
        **********WELCOME**********\n\n";
return 0;
}
else
{
cout<<"WRONG PIN"<<endl;
}
}
exit(0);
}
void  get_AccountDetails()
{
system("cls");
fstream atm;
atm.open("D:Customers detail.txt",ios::app);
cout<<"\nEnter Customer Name : ";
cin>>name;
cout<<"Enter Account Number  : ";
cin>>account_no;
cout<<"Enter Account Type    : ";
cin>>account_type;
if(atm.is_open())
```

```
{
atm<<"Customer name: "<<name<<endl;
atm<<"Account no: "<<account_no<<endl;
atm<<"Account type: "<<account_type<<endl;
}
atm.close();
}
void display_alldetail()
{
fstream atm;
atm.open("D:Customers detail.txt",ios::in);
if(atm.is_open())
{
string ch;
while(!atm.eof())
{
getline(atm,ch);
cout<<"\t"<<ch<<endl;
}
cout<<endl;
}
atm.close();
}
void display_Details()
{
```

```
system("cls");
cout<<"\n\nCustomer Name : "<<name;
cout<<"\nAccount Number  : "<<account_no;
cout<<"\nAccount Type    : "<<account_type;
}
};
class CURRENT_ACCOUNT : public ACCOUNT
{
public:
CURRENT_ACCOUNT(int c)
{
c_balance=c;
}
void c_display()
{
cout<<"\nCurrent Balance :"<<c_balance;
}
void c_deposit()
{
system("cls");
fstream atm;
atm.open("D:Customers detail.txt",ios::app);
cout<<"\nEnter amount to Deposit:";
cin>>deposit;
if(atm.is_open())
```

```
{
atm<<"Deposit : "<<deposit<<endl;
}
atm.close();
c_balance = c_balance + deposit;
cout<<"Amount after deposit "<<c_balance;
cout<<"\n";
cout<<"\n\t\tThank you...!\n";
cout<<"\tFor using"<<endl;
cout<<"\t\tOur ATM Service"<<endl;
}
void c_withdraw()
{
system("cls");
fstream atm;
atm.open("D:Customers detail.txt",ios::app);
cout<<"\n\nBalance : "<<c_balance;
cout<<"\nEnter amount to be withdraw in a multiple of 500:";
cin>>withdraw;
if(atm.is_open())
{
atm<<"Withdraw: "<<withdraw<<endl;
}
atm.close();
if(withdraw %500==0)
```

```
{
c_balance=c_balance-withdraw;
cout<<"\nBalance Amount After Withdraw:"<<c_balance;
cout<<"\n";
cout<<"\n\t\tThank you...!\n";
cout<<"\tFor using"<<endl;
cout<<"\t\tOur ATM Service"<<endl;
}
else if(withdraw%500!=0)
{
cout<<"you can't withdraw"<<endl;
char ch;
cout<<"\nDo you want to do an other transaction "<<endl;
cin>>ch;
if(ch=='y')
{
c_withdraw();
}
else if(ch=='n')
{
exit(0);
}
}
}
void transfer_money()
```

```
{
system("cls");
fstream atm;
atm.open("D:Customers detail.txt",ios::app);
float transfer_balc;
int acc_n;
float new_account_balanse=0;
cout<<"Enter account number in which you want to transfer money"<<endl;
cin>>acc_n;
cout<<"Please enter the amount you want to transfer"<<endl;
cin>>transfer_balc;
if(atm.is_open())
{
atm<<"Transfer amount "<<transfer_balc<<endl;
}
atm.close();
if(transfer_balc>c_balance)
{
cout<<"Inufficient account balance"<<endl;
}
else
{
c_balance-=transfer_balc;
cout<<transfer_balc<<" Amount transferred"<<endl;
cout<<"The operation is successful,your remaining balance is: "<<c_balance<<endl;
```

```
new_account_balanse+=transfer_balc;
cout<<"Now,amount in new account is "<<new_account_balanse<<endl;
}
}
};
class SAVING_ACCOUNT : public ACCOUNT
{
public:
SAVING_ACCOUNT(int s)
{
sav_balance=s;
}
void s_display()
{
cout<<"\nSaving Balance :"<<sav_balance;
}
void s_deposit()
{
system("cls");
fstream atm;
atm.open("D:Customers detail.txt",ios::app);
cout<<"\nEnter amount to Deposit : ";
cin>>deposit;
if(atm.is_open())
{
```

```
atm<<"Deposit : "<<deposit<<endl;
}
atm.close();
sav_balance = sav_balance + deposit;
cout<<"Amount after deposit "<<sav_balance;
cout<<"\n";
cout<<"\n\t\tThank you...!\n";
cout<<"\tFor using"<<endl;
cout<<"\t\tOur ATM Service"<<endl;
}
void s_withdraw()
{
system("cls");
fstream atm;
int months;
float interest;
atm.open("D:Customers detail.txt",ios::app);
cout<<"\nBalance :- "<<sav_balance;
cout<<"\nEnter amount to be withdraw in a multiple of 500: ";
cin>>withdraw;
if(atm.is_open())
{
atm<<"Withdraw : "<<withdraw<<endl;
}
atm.close();
```

```
if(withdraw %500==0)
{
sav_balance=sav_balance-withdraw;
cout<<"\nBalance Amount After Withdraw: "<<sav_balance<<endl;
cout<<"\n";
cout<<"\n\t\tThank you...!\n";
cout<<"\tFor using"<<endl;
cout<<"\t\tOur ATM Service"<<endl;
}
else if(withdraw%500!=0)
{
cout<<"You can't withdraw"<<endl;
char ch;
cout<<"\nDo you want to do an other transaction "<<endl;
cin>>ch;
if(ch=='y')
{
s_withdraw();
}
}
cout<<"\nYour interest in saving account "<<endl;
cout<<"Enter months"<<endl;
cin>>months;
if(months>=5)
{
```

```
interest=(sav_balance*months)/100;
cout<<"Interest"<<interest<<endl;
sav_balance+=interest;
cout<<"Your saving balance after adding interest: "<<sav_balance<<endl;
atm.open("D:Customers detail.txt",ios::app);
if(atm.is_open())
{
atm<<"Interest: "<<interest<<endl;
atm<<"Your saving balance after adding interest: "<<sav_balance<<endl;
}
atm.close();
}
}
void transfer_moneysaving()
{
system("cls");
fstream atm;
atm.open("D:Customers detail.txt",ios::app);
float transfer_bals;
int acc_no;
float new_account_balance=0;
cout<<"Enter account number in which you want to transfer money"<<endl;
cin>>acc_no;
cout<<"Please enter the amount you want to transfer"<<endl;
cin>>transfer_bals;
```

```
if(atm.is_open())
{
atm<<"Transfer amount "<<transfer_bals<<endl;
}
atm.close();
if(transfer_bals>sav_balance)
{
cout<<"Inufficient account balance"<<endl;
}
else
{
sav_balance-=transfer_bals;
cout<<transfer_bals<<" Amount transferred"<<endl;
cout<<"The operation is successful,your remaining balance is: "<<sav_balance<<endl;;
new_account_balance+=transfer_bals;
cout<<"Now,amount in new account is "<<new_account_balance<<endl;
}
}
};
int main()
{
account a1;
CURRENT_ACCOUNT c1(2000);
SAVING_ACCOUNT s1(3000);
char type;
```

```
cout << "\n                ****************************ATM MACHINE
SOFTWARE****************************\n\n";

cout<<"ENTER YOUR ATM CARD "<<endl;

a1.pn();

cout<<"Select account type"<<endl;

cout<<"\t\t\t 1.saving"<<endl;

cout<<"\t\t\t 2.current"<<endl;

cout<<"\nEnter S for saving customer and C for current account customer : ";

cin>>type;

int choice;

if(type=='s' || type=='S')

{

s1.get_AccountDetails();

system("cls");

while(1)

{

cout<<"\n\t\t\t\t**********************************";

cout<<"\n\t\t\t     Choose Your Choice"<<endl;

cout<<"\t\t\t\t1) Deposit"<<endl;

cout<<"\t\t\t\t2) Withdraw"<<endl;

cout<<"\t\t\t\t3)  Display Balance"<<endl;

cout<<"\t\t\t\t4)  Transfer Money"<<endl;

cout<<"\t\t\t\t5) Display Detail of customer"<<endl;

cout<<"\t\t\t\t6)  Display Details of all customer"<<endl;

cout<<"\t\t\t\t7)  Exit"<<endl;
```

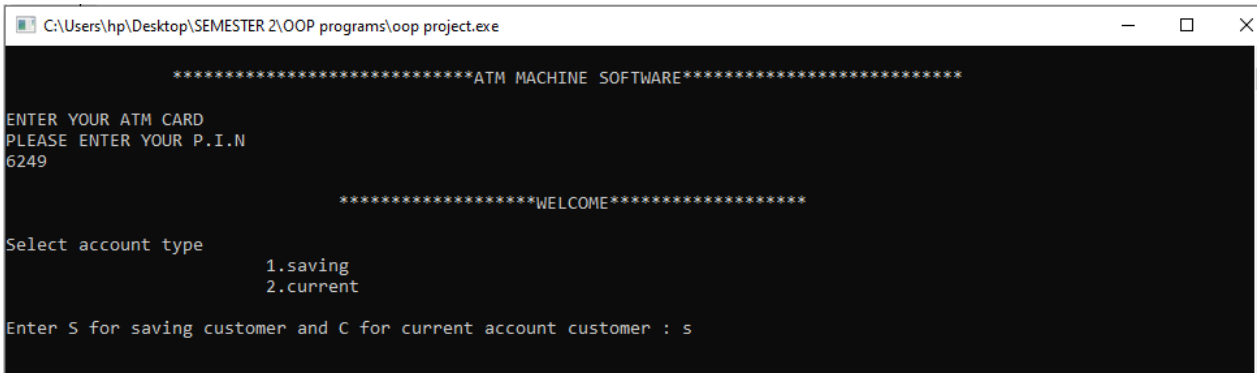
```
cout<<"\t\t\t\t**********************************"<<endl;
cout<<"Enter Your choice: ";
cin>>choice;
switch(choice)
{
case 1 :
s1.s_deposit();
break;
case 2 :
s1.s_withdraw();
break;
case 3 :
s1.s_display();
break;
case 4 :
s1.transfer_moneysaving();
break;
case 5 :
s1.display_Details();
s1.s_display();
break;
case 6:
s1.display_alldetail();
break;
case 7:
```

```
exit(0);
}
}
}
else if(type=='c' || type=='C')
{
c1.get_AccountDetails();
system("cls");
while(1)
{
cout<<"\n\t\t\t\t**********************************";
cout<<"\n\t\t\t     Choose Your Choice"<<endl;
cout<<"\t\t\t\t1) Deposit"<<endl;
cout<<"\t\t\t\t2) Withdraw"<<endl;
cout<<"\t\t\t\t3)  Display Balance"<<endl;
cout<<"\t\t\t\t4)  transfer money"<<endl;
cout<<"\t\t\t\t5) Display Details of customer"<<endl;
cout<<"\t\t\t\t6) Display Details of all customer"<<endl;
cout<<"\t\t\t\t7) Exit"<<endl;
cout<<"\t\t\t\t**********************************"<<endl;
cout<<"Enter Your choice: ";
cin>>choice;
switch(choice)
{
case 1 :
```

```
c1.c_deposit();
break;
case 2 :
c1.c_withdraw();
break;
case 3 :
c1.c_display();
break;
case 4 :
c1.transfer_money();
break;
case 5 :
c1.display_Details();
c1.c_display();
break;
case 6:
c1.display_alldetail();
break;
case 7:
exit(0);
}
}
}
return 0;
}
```

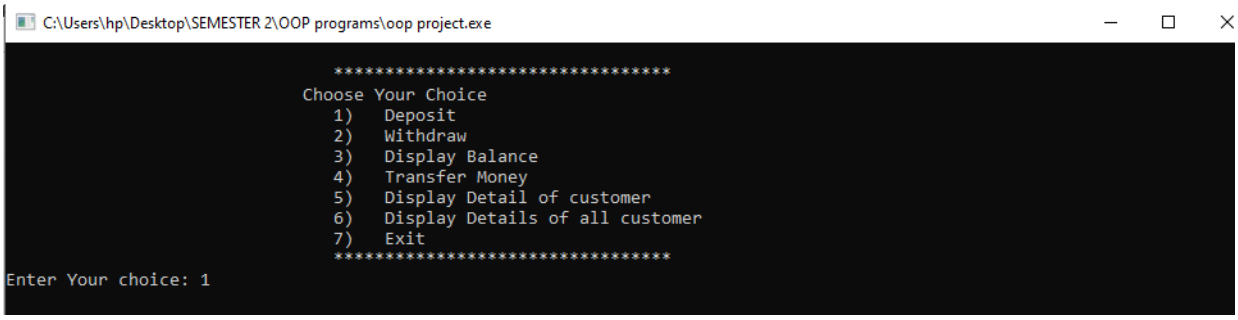
Results

Main screen

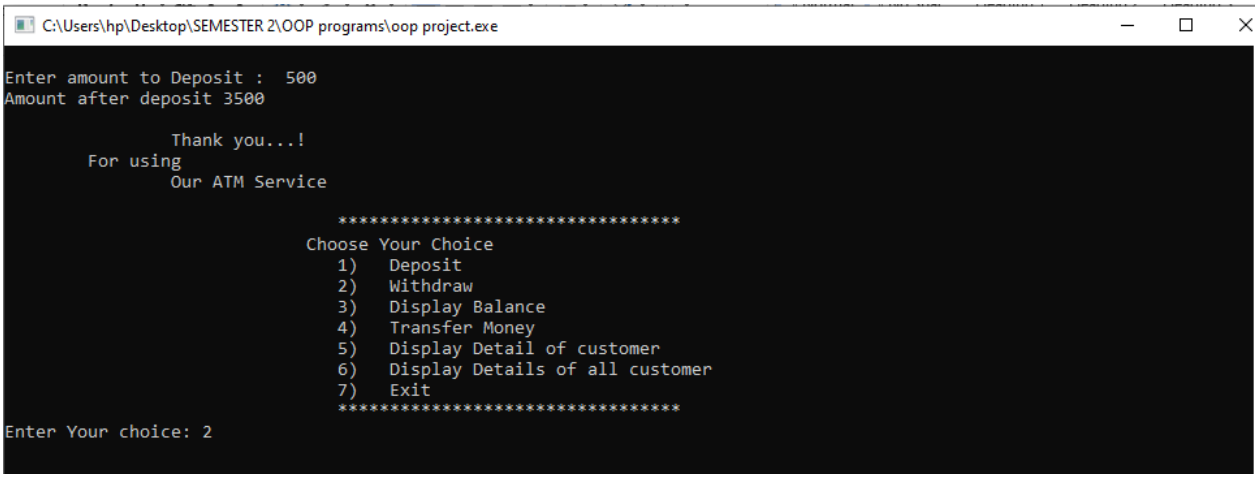


Enter customer detail

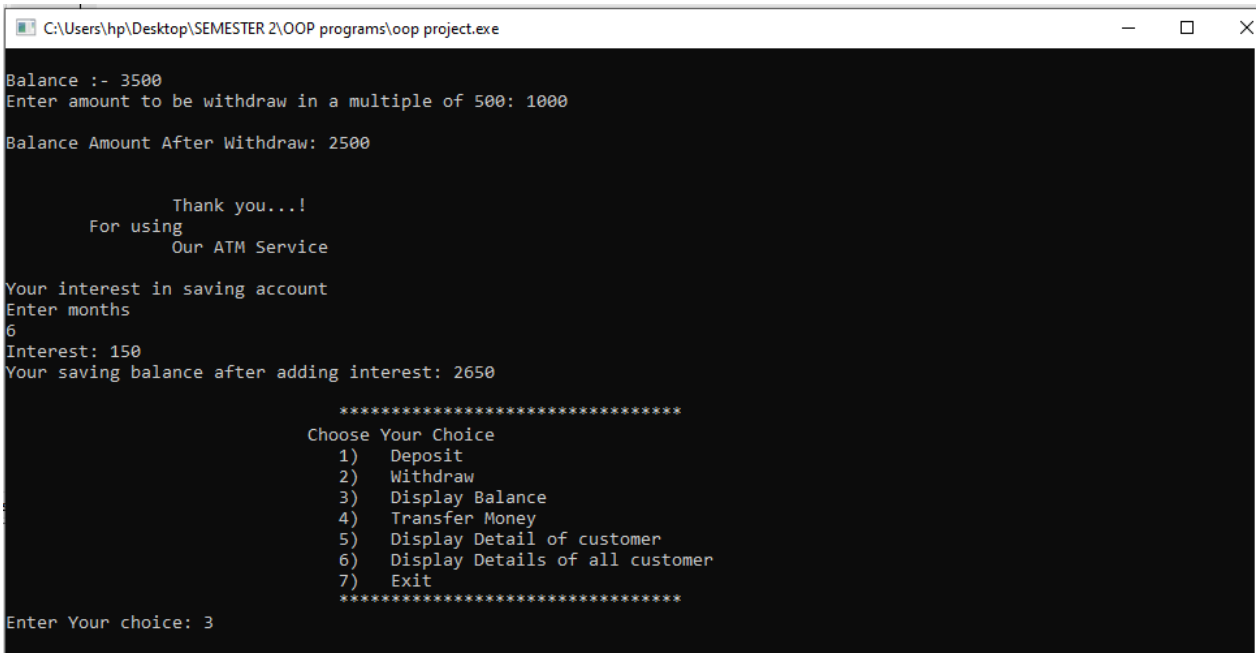
Enter choice



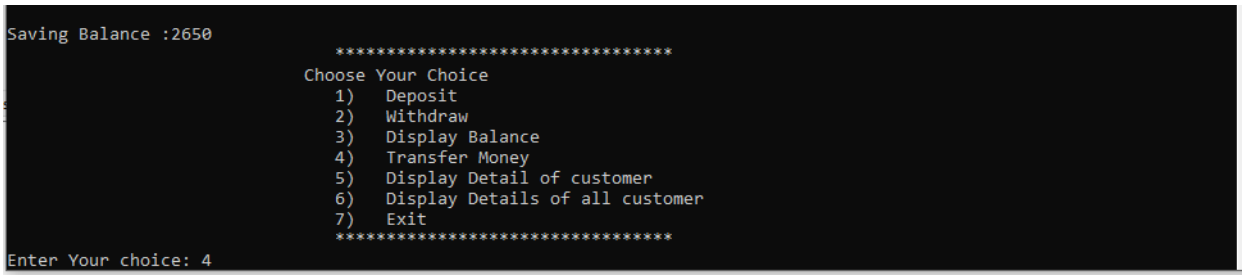
For deposit



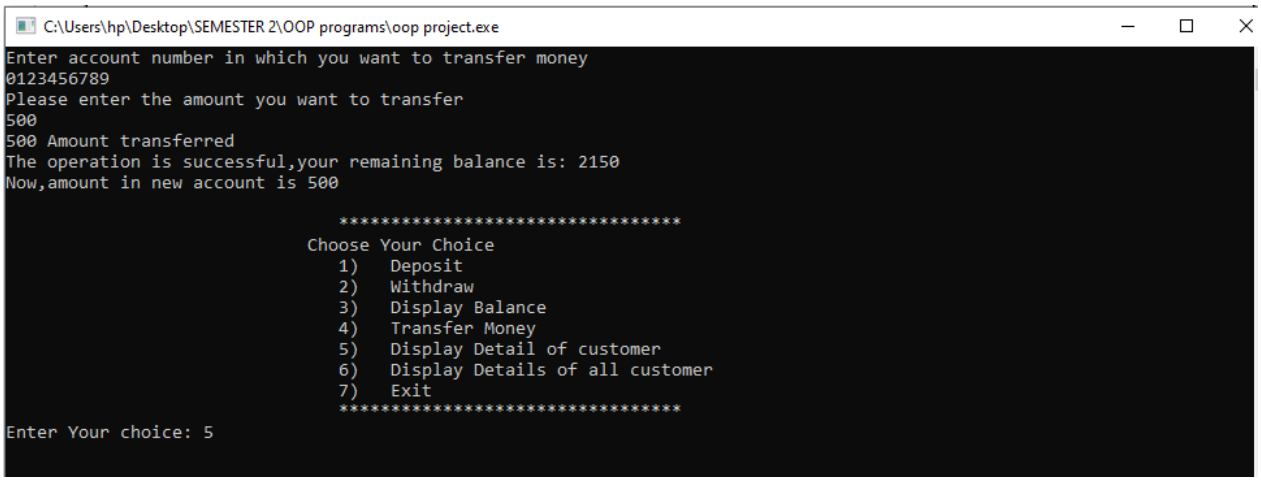
For withdraw



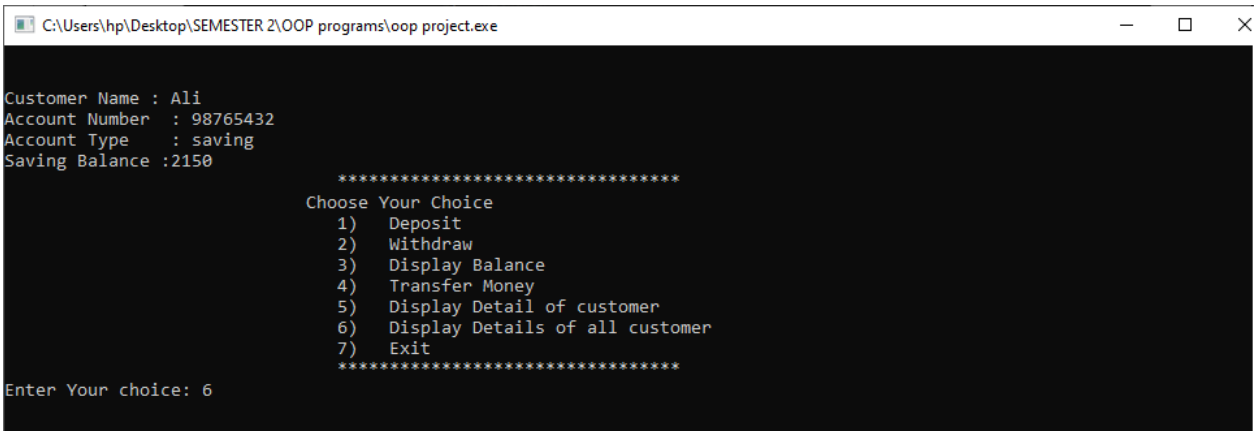
Display saving balance



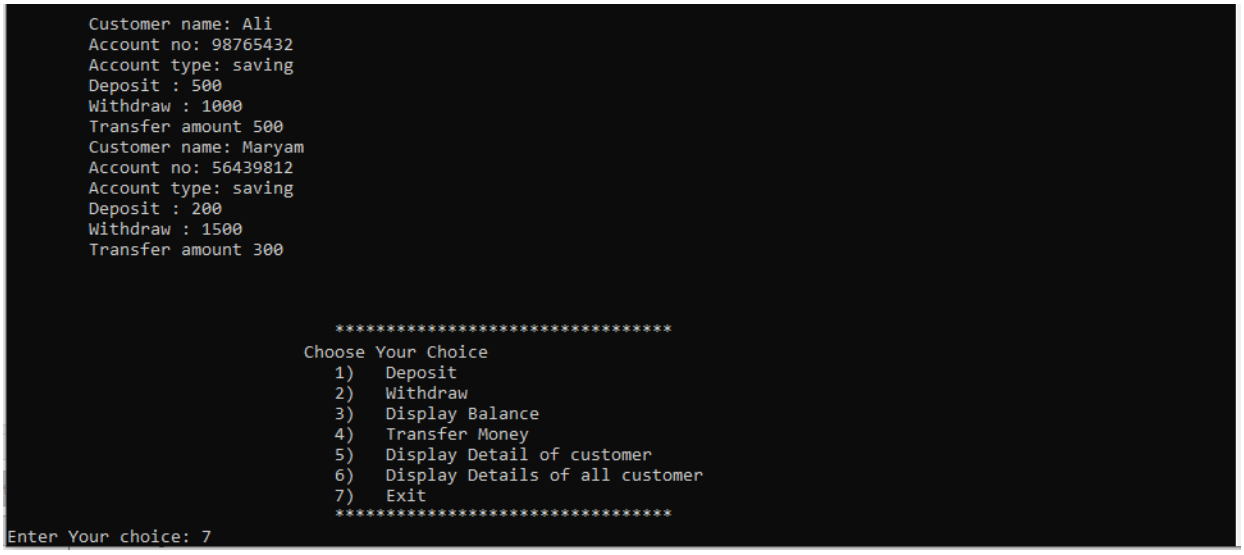
For transfer money



Display detail of customer



Display all customer details



For exit

```
                              **********************************
                           Choose Your Choice
                              1)   Deposit
                              2)   Withdraw
                              3)   Display Balance
                              4)   Transfer Money
                              5)   Display Detail of customer
                              6)   Display Details of all customer
                              7)   Exit
                              **********************************
Enter Your choice: 7

--------------------------------
```

In a file