PART B:

Microproject Report

1.0 Summary

Bank Management System is based on a concept of recording customer's account details. Here the user can perform all the tasks like creating an account, deposit amount, withdraw amount, check balance, view all account holders detail, close an account and modify an account. There's no login system for this project

2.0 Course action addressed

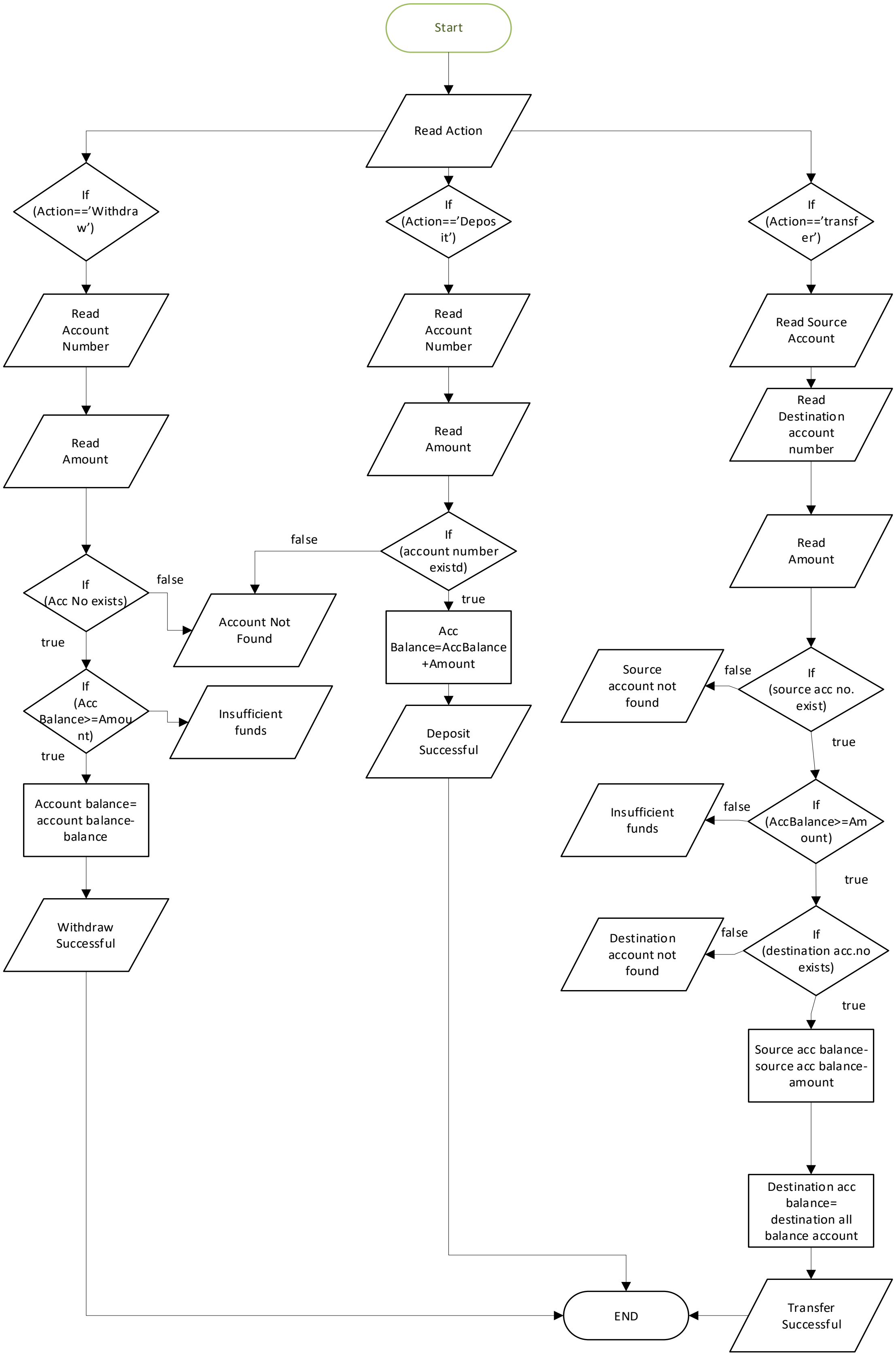
a. Develop C++ programs to solve problems using Procedure Oriented Approach.
b. Develop C++ programs using classes and objects.
c. Implement Inheritance in C++ program.
d. Use Polymorphism in C++ program.
e. Develop C++ programs to perform file operations.

3.0 Actual methodology

a. Algorithm:

step 1 : Create a class for BankAccount which would have the attributes like accountNumber, accountHolderName, accountBalance
step 2 : Create functions for opening a bank account, withtaking money, depositing money, displaying bank account details
step 3 : Create a menu for the user to select the desired operation for managing their bank account
step 4 : Create another class for Bank which would have the attributes like branchName, branchAddress, branchManager
step 5 : Create functions for creating a new bank branch, displaying the bank branch details, adding new customers
step 6 : Create a main() function in the program to call the functions of both the classes
step 7 : Create a loop to allow the user to select the desired operation and perform the desired operations on the bank account
step 8 : Exit the program after the user has completed all the operations.

b. Flowchart:


Start
Read Action
If (Action=='Withdraw')
If (Action=='Deposit')
If (Action=='transfer')
Read Account Number
Read Amount
If (Acc No exists)
false
true
Account Not Found
If (Acc Balance>=Amount)
Insufficient funds
true
Account balance= account balance-balance
Withdraw Successful
Read Account Number
Read Amount
If (account number existd)
false
true
Acc Balance=AccBalance +Amount
Deposit Successful
Read Source Account
Read Destination account number
Read Amount
If (source acc no. exist)
false
Source account not found
true
If (AccBalance>=Amount)
false
Insufficient funds
true
If (destination acc.no exists)
false
Destination account not found
true
Source acc balance-source acc balance-amount
Destination acc balance= destination all balance account
Transfer Successful
END

c. Source code:

```
 #include<iostream>
#include<fstream>
#include<conio.h>
#include<stdlib.h>
#include<string.h>
using namespace std;
class bank

{
 int accno;
 char name[50];
 long 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 customer
 void dep(long int);     //function to accept amount and add to balance amount
 void draw(long 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
 long int retdeposit() const;     //function to return balance amount
 char rettype() const;   //function to return account type
};        //class ends here
void bank::create_account()
{
 cout<<"\nEnter The account No. :";
 cin>>accno;
 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 bank::show_account() const
{
 cout<<"\nAccount No. : "<<accno;
 cout<<"\nAccount Holder Name : ";
 cout<<name;
 cout<<"\nType of Account : "<<type;
 cout<<"\nBalance amount : "<<deposit;
}

void bank::modify()
{
 cout<<"\nAccount No. : "<<accno;
 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 bank::dep(long int x)
{
 deposit+=x;
}

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

void bank::report() const
{
 cout<<accno<<"\t\t"<<name<<"\t\t"<<type<<"\t\t"<<deposit<<"\n";
}

int bank::retacno() const
{
 return accno;
}

long int bank::retdeposit() const
{
 return deposit;
}

char bank::rettype() const
```

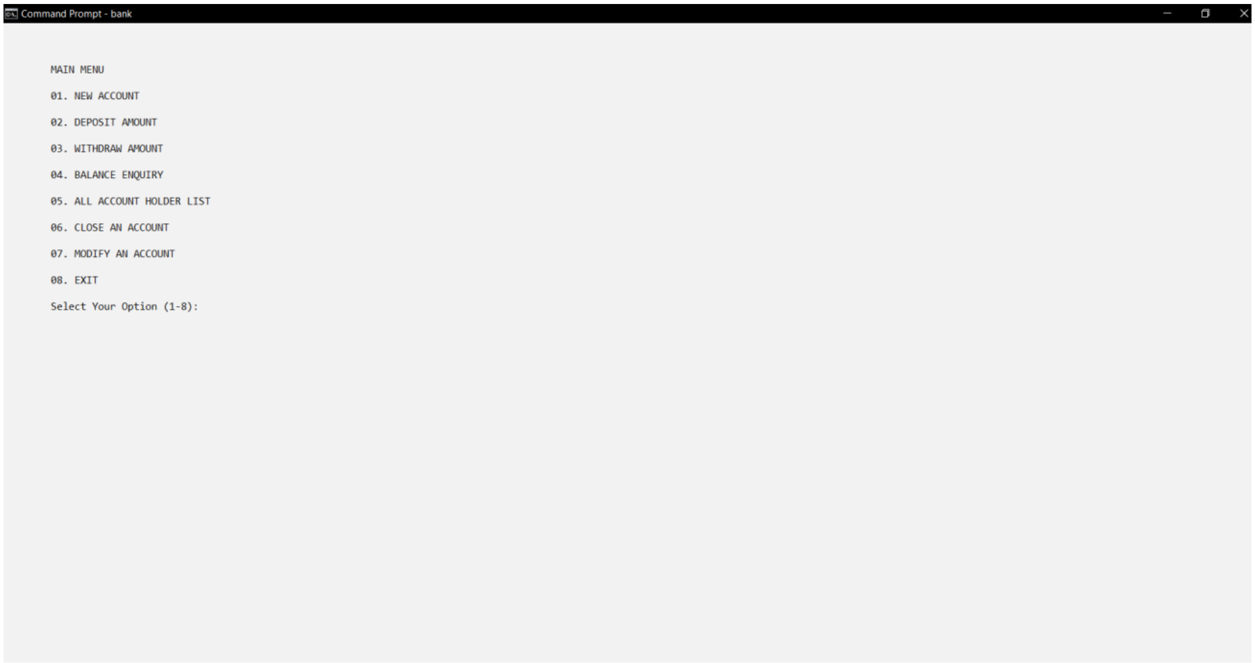
```
{
 return type;
}

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

//***********************************************************************
//      function to read specific record from file
//************************************************************************

void display_sp(int n)
{
        bank 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(bank)))
        {
                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)
```

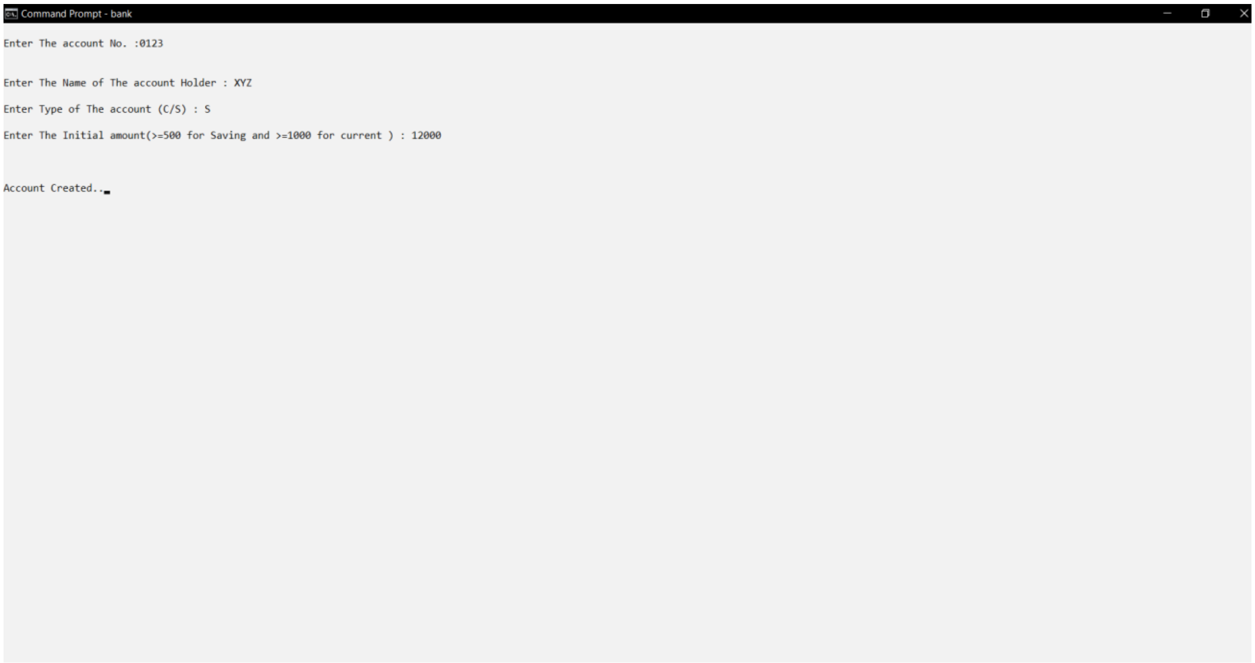
```
{
        bool found=false;
        bank ac;
        fstream File;
   File.open("account.dat",ios::binary|ios::in|ios::out);
        if(!File)
        {
                cout<<"File could not be open !! Press any Key...";
                return;
        }
   while(!File.eof() && found==false)
        {
                File.read(reinterpret_cast<char *> (&ac), sizeof(bank));
                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(bank));
                   File.seekp(pos,ios::cur);
                   File.write(reinterpret_cast<char *> (&ac), sizeof(bank));
                   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)
{
        bank 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(bank)))
        {
                if(ac.retacno()!=n)
                {
                        outFile.write(reinterpret_cast<char *> (&ac), sizeof(bank));
                }
        }
   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()
{
        bank 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(bank)))
        {
                ac.report();
        }
        inFile.close();
}

//**************************************************************
//      function to deposit and withdraw amounts
```

```
//**************************************************************

void deposit_withdraw(int n, int option)
{
        int amt;
        bool found=false;
        bank 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(bank));
                if(ac.retacno()==n)
                {
                        ac.show_account();
                        if(option==1)
                        {
                                cout<<"\n\n\tTO DEPOSIT 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  withdraw : ";
                                cin>>amt;
                                int bal=ac.retdeposit()-amt;
                                if((bal<500 && ac.rettype()=='S') || (bal<1000 &&
ac.rettype()=='C'))
                                        cout<<"Minimun Account balance of 1000 is required,
Please enter a valid amount to withdraw";
                                else
                                        ac.draw(amt);
                    }
                        int pos=(-1)*static_cast<int>(sizeof(ac));
                        File.seekp(pos,ios::cur);
                        File.write(reinterpret_cast<char *> (&ac), sizeof(bank));
                        cout<<"\n\n\t Record Updated";
                    found=true;
            }
        }
    File.close();
        if(found==false)
                cout<<"\n\n Record Not Found ";
}


//************************************************************
//      INTRODUCTION FUNCTION
//*************************************************************

void intro()

{
        cout<<"\n\n\n\t  BANK";
        cout<<"\n\n\tMANAGEMENT";
        cout<<"\n\n\t  SYSTEM";
        cout<<"\n\n\n\n By Group Number : 4";
    cout<<"\nPress Enter to continue....";
        cin.get();
}

//************************************************************
//                      END OF PROJECT
//************************************************************
```

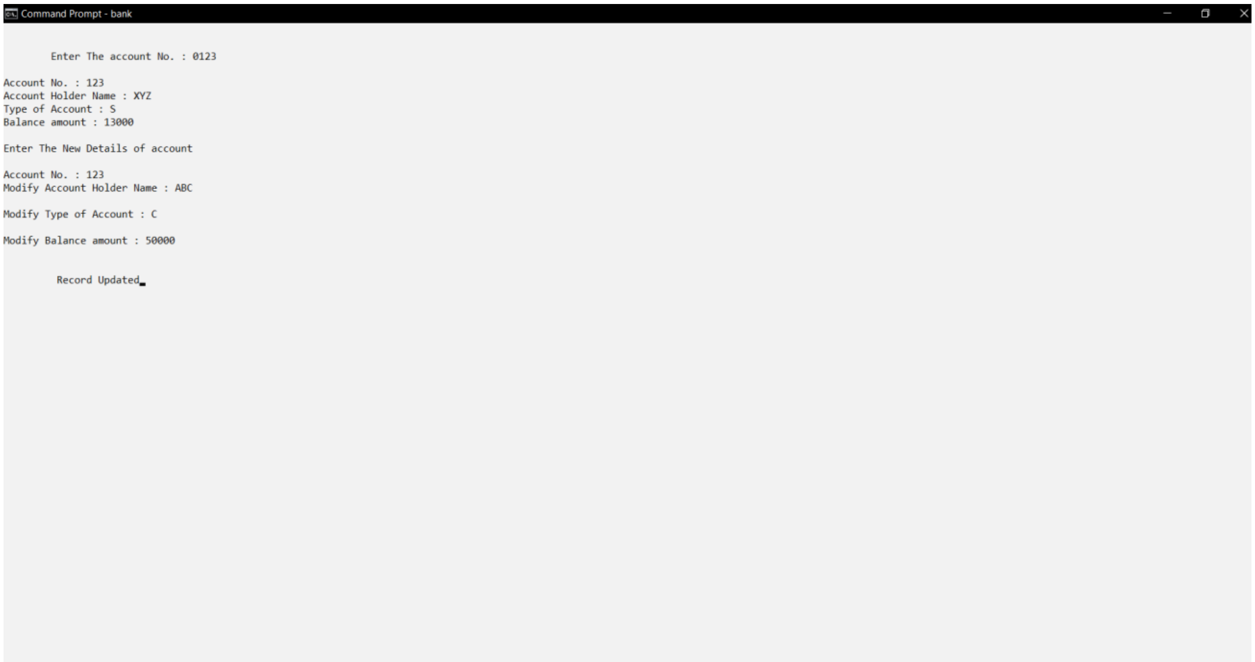
4.0 Microproject output



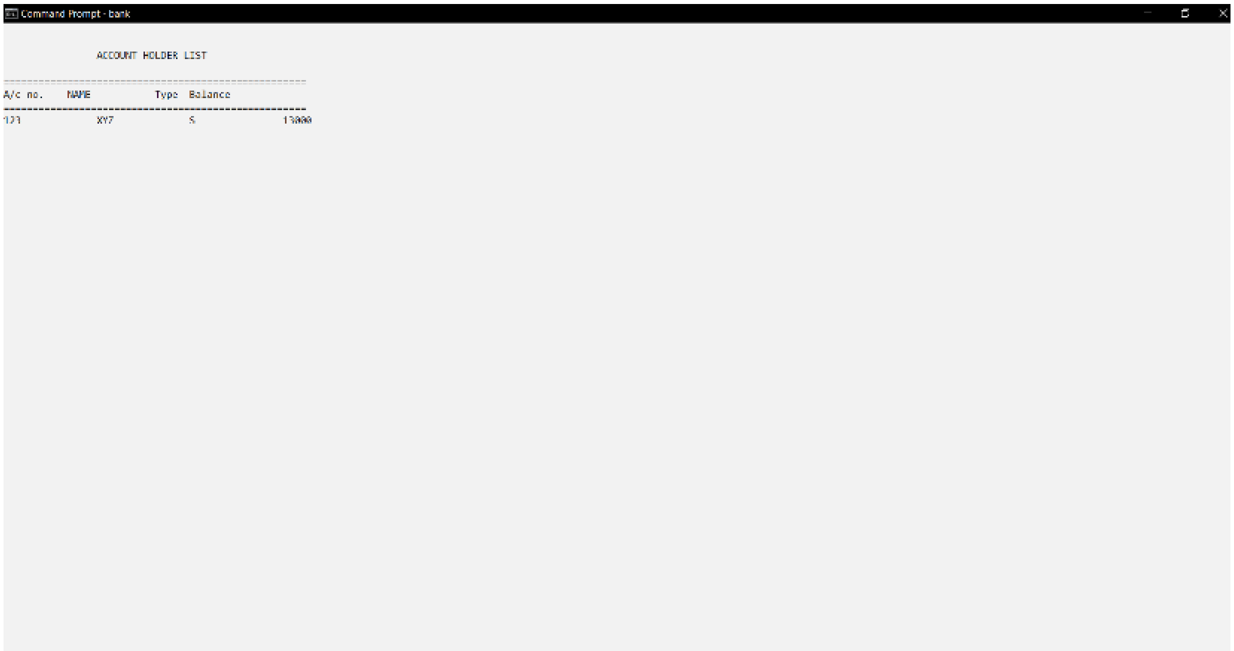
```
        MAIN MENU

        01. NEW ACCOUNT

        02. DEPOSIT AMOUNT

        03. WITHDRAW AMOUNT

        04. BALANCE ENQUIRY

        05. ALL ACCOUNT HOLDER LIST

        06. CLOSE AN ACCOUNT

        07. MODIFY AN ACCOUNT

        08. EXIT

        Select Your Option (1-8):
```



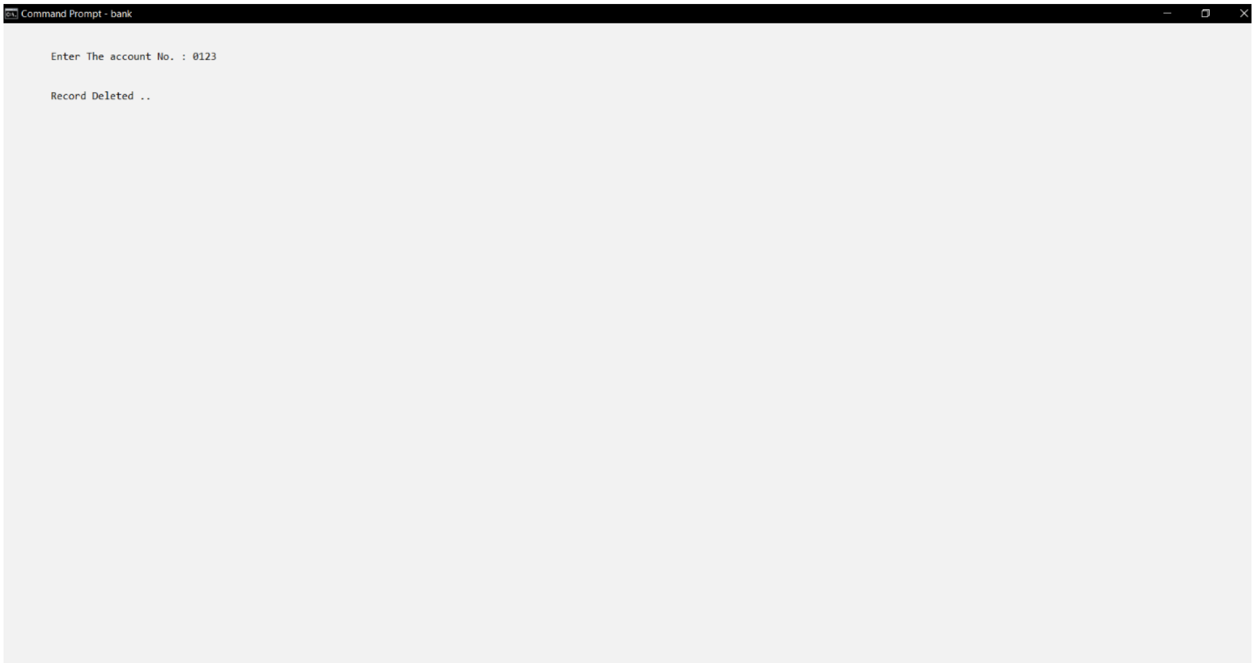
```
Enter The account No. :0123


Enter The Name of The account Holder : XYZ

Enter Type of The account (C/S) : S

Enter The Initial amount(>=500 for Saving and >=1000 for current ) : 12000



Account Created..
```



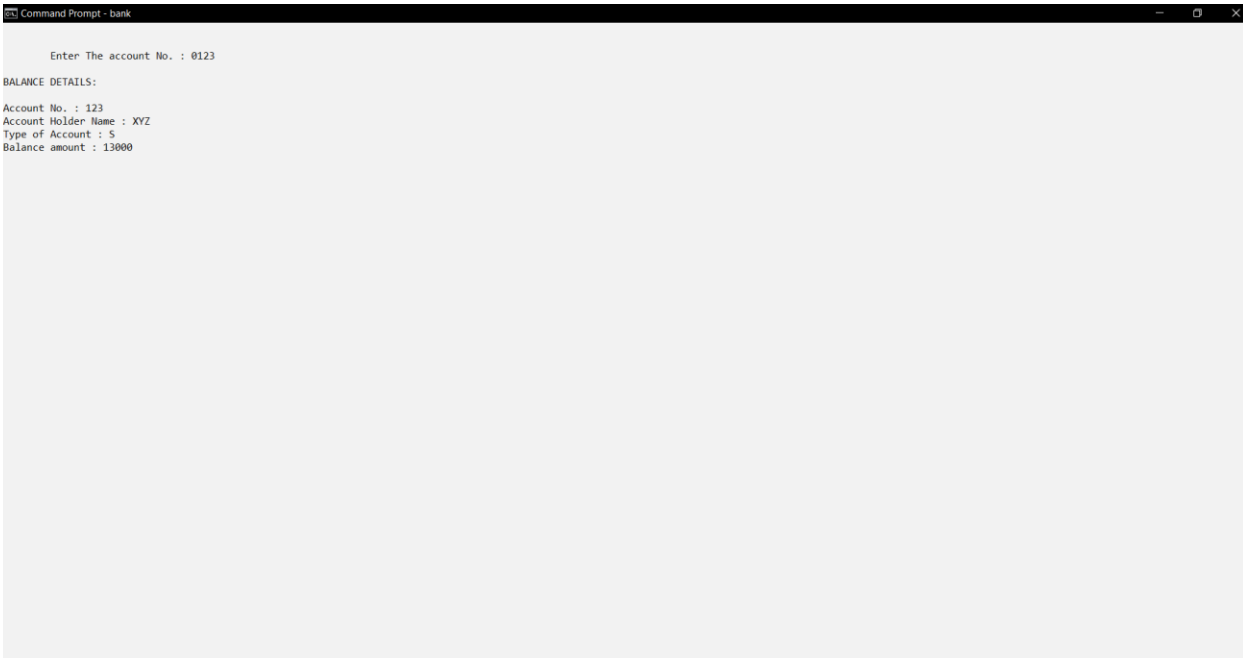
```
        Enter The account No. : 0123

Account No. : 123
Account Holder Name : XYZ
Type of Account : S
Balance amount : 13000

Enter The New Details of account

Account No. : 123
Modify Account Holder Name : ABC

Modify Type of Account : C

Modify Balance amount : 50000


         Record Updated_
```



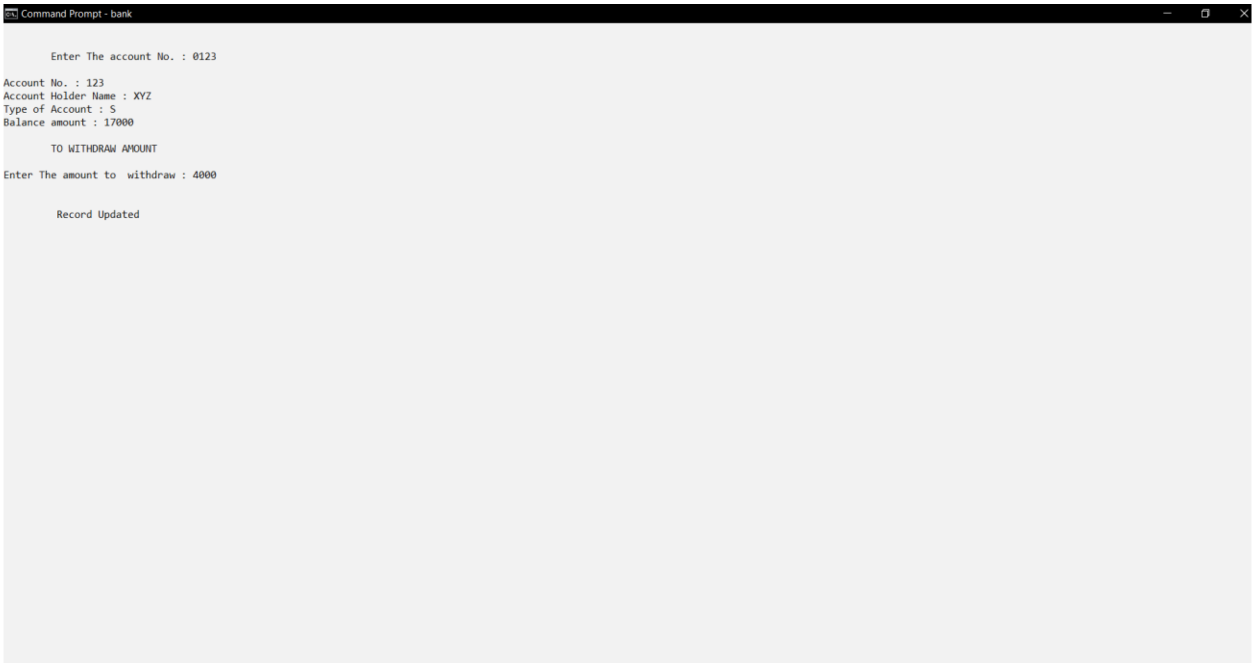
```
                ACCOUNT HOLDER LIST

===================================================
A/c no.   NAME           Type  Balance
---------------------------------------------------
123             XYZ            S               13000
```



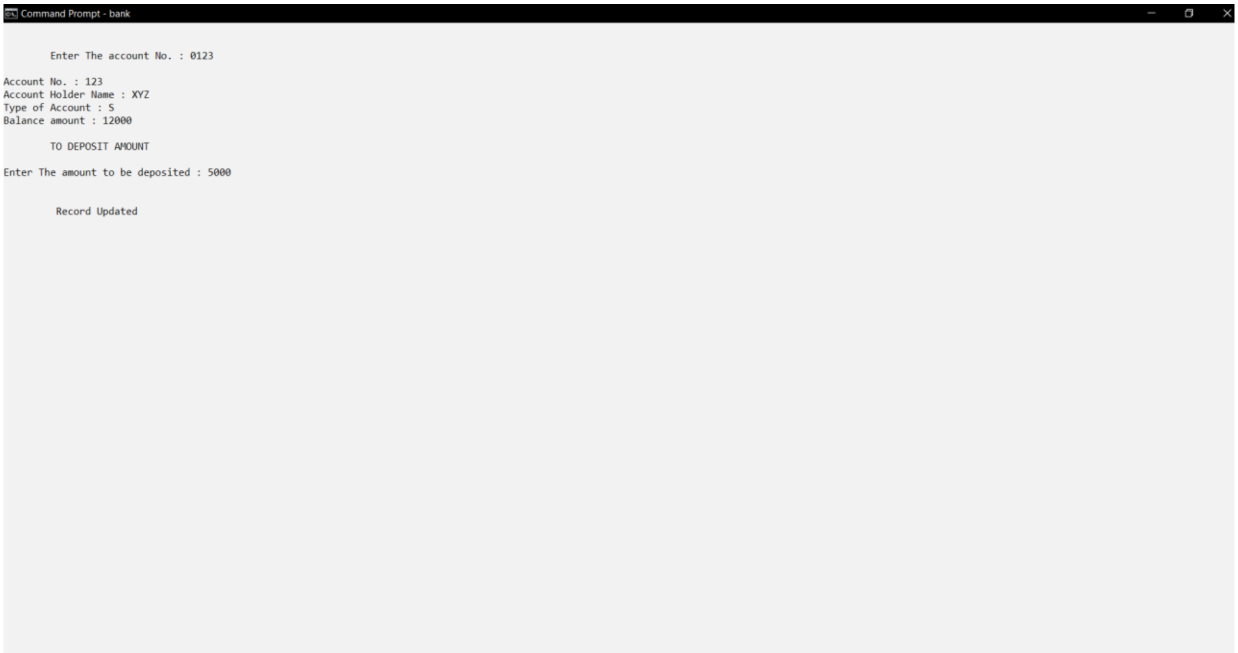
```
        Enter The account No. : 0123


        Record Deleted ..
```



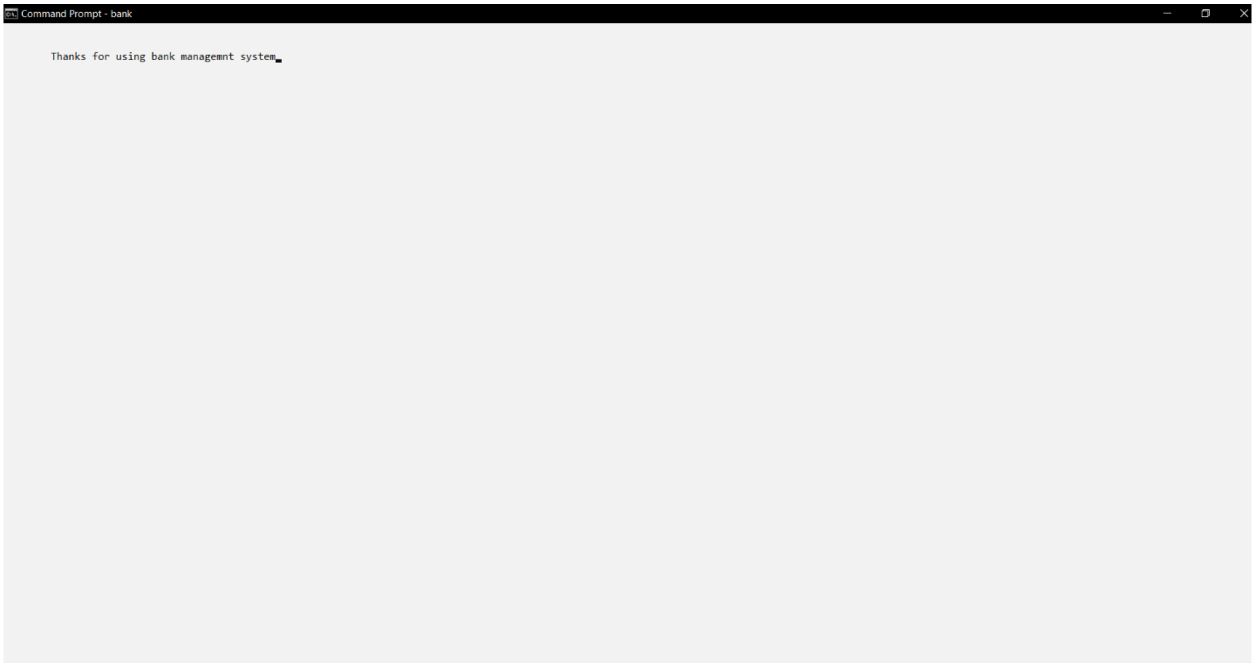
```
        Enter The account No. : 0123

BALANCE DETAILS:

Account No. : 123
Account Holder Name : XYZ
Type of Account : S
Balance amount : 13000
```



```
        Enter The account No. : 0123

Account No. : 123
Account Holder Name : XYZ
Type of Account : S
Balance amount : 17000

        TO WITHDRAW AMOUNT

Enter The amount to  withdraw : 4000


         Record Updated
```



```
        Enter The account No. : 0123

Account No. : 123
Account Holder Name : XYZ
Type of Account : S
Balance amount : 12000

        TO DEPOSIT AMOUNT

Enter The amount to be deposited : 5000


         Record Updated
```



```
    Thanks for using bank managemnt system
```

5.0 Actual resources used

i. Visual Studios for coding
ii. MingW G++ compiler
iii. Google for Reference

6.0 Skills developed

We developed programming skills
We developed searching skills
We developed out team work skills

7.0 Application of Microproject

- Creating the bank account for the customer.
- Deposit amount for the customer.
- Withdraw amount for the customer.
- Balance Enquiry by the customer.
- Show account holder's full detail.
- Closing or terminating a bank account.
- Updating the bank account**.**