A

Project report

On

Bank management system in c++

Submitted

In partial fulfilment of requirement for the degree of

Bachelor of technology

In

computer science and engineering

Submitted by:

Mohd Suhaib Khan(1902250100086)

Rohit Rathaur(1902250100123)

Syed Rahib(1902250100148)

Suhail Khan(1902250100143)

Department of Computer Science and Engineering

Accurate Institute Of Management and Technology,Greater Noida

DR. A.P.J. ABDUL KALAM TECHNICAL UNIVERSITY, LUCKNOW,

UTTAR PRADESH ,

November, 2021

CONTENT: Page No.

List of figures: *page no.*

DECLARATION

I hereby declare that this submission is our own work and that to the best of our knowledge and beliefs. It contains no material previously published or written by neither any person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgement has been made in the text.

Name: MOHD SUHAIB KHAN

Roll no.:1902250100086

Date:9/12/2021

Name: SUHAIL KHAN

Roll no:1902250100143

Date:9/12/2021

Name: ROHIT RATHAUR

Roll no:1902250100123

Date:9/12/2021

Name: SYED RAHIB

Roll no:1902250100148

Date:9/12/2021

CERTIFICATE

This is to certify that Project report entitled "bank management system in c++", submitted by Mohd Suhaib khan , Suhail khan , Rohit Rathaur and Syed Rahib for partial fulfilment of the requirement for the award of degree Bachelors of Technology in Department of Computer Science & Engineering of Dr. A.P.J Abdul Kalam University, Lucknow is a record of the candidates' own work carried out by them under my supervision. The matter embodied in this report is original and has not been submitted for the award of any other degree.

Date: 9.12.2021 **Supervisor:** ASHISH JAIN SIR

ACKNOWLEDGEMENT

In completing this project, we have been fortunate enough to have help, support and encouragement from many people. I would like to acknowledge them for their cooperation.

Firstly, we would like to thank MR ASHISH JAIN SIR from Department of Computer Science & Engineering, A.I.M.T for guiding us through each and every step of the process with knowledge and support. His thoughts have been a constant source of inspiration for us.

We would also like to acknowledge the contribution of all faculty members of the department for their kind assistance, suggestions and cooperation throughout the development of the project.

Finally, we would like to thank our classmates for the encouragement and help during the project.

Name: MOHD SUHAIB KHAN

Roll no.:1902250100086

Date:9/12/2021

Name: SUHAIL KHAN

Roll no:1902250100143

Date:9/12/2021

Name: ROHIT RATHAUR

Roll no:1902250100123

Date:9/12/2021

Name: SYED RAHIB

Roll no:1902250100148

Date:9/12/2021

INTRODUCTION

Bank management system can keep the information of account type, account opening form, deposit , and searching the transaction, transaction report, individual account opening form, group account as a record.

it displays records of transaction reports, statistical summary of account type and interest information. this helps to provide the flexible solution to the user. here the chance of occurrence of error is less when compared with the existing system.

it is fast, efficient and reliable. easy accessibility of data and avoids data redundancy and inconsistency.

banking record system project in C++ is a simple console application developed without the use of graphics component.

It is more of a database project in C++, and is built using the language's file handling mechanism.

It is suitable for beginners who want to learn how to add, edit, search, delete or modify records in a file, and how to use file as database overall .

BANK STATEMENTS CAN ALSO BE USEFUL TO ANALYZE THE CREDIT WORTHINESSOF THE ACCOUNT HOLDER. MOST BANKS AND FINANCIAL INSTITUTIONS REQUIREVERIFICATION OF BANK STATEMENTS FOR THE LAST 2-5 YEARS BEFORE GIVINGLOANS TO INDIVIDUAL CLIENTS. BANKS USE THE INDIVIDUAL'S BANK STATEMENTS AND OTHER CREDIT

DOCUMENTS TO ANALYZE THE CREDITWORTHINESS OF THE BORROWER. ITAPPLIES TO MOST TYPES OF LOANS, INCLUDING RESIDENTIAL_MORTGAGES, STUDENT LOANS, AND LOANS FOR SMALL BUSINESSES.

WHAT IS A BANK ACCOUNT?

A bank account is a financial account maintained by a bank or other financial institution in which the financial transactions between the bank and a customer are recorded.

Each financial institution sets the terms and conditions for each type of account it offers, which are classified in commonly understood types , such as deposit accounts, credit card accounts, current accounts, loan accounts or many other types of account.

A customer may have more than one account. Once an account is opened, funds entrusted by the customer to the financial institution on deposit are recorded in the account designated by the customer Funds can be withdrawn from loan loaders.

This **banking record system** project in C++ is a simple console application developed without the use of graphics component. It is more of a database project in C++, and is built using the language's file handling mechanism. It is suitable for beginners who want to learn how to add, edit, search, delete or modify records in a file, and how to use file as database overall .

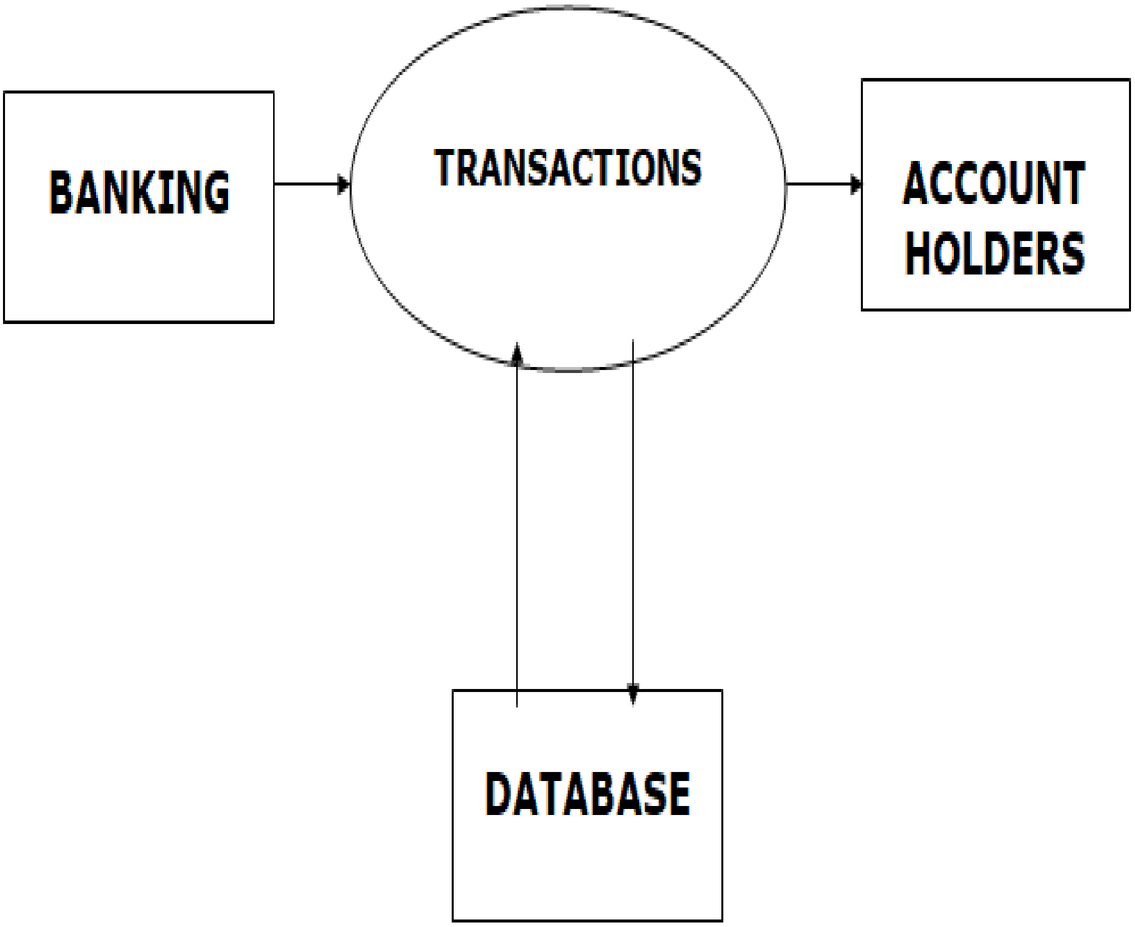


Fig1.1: Transaction between bank and account holders stored in DataBase

WHAT IS A BANKING RECORD?

The bank account record stores all bank account information you need to track and manage, such as account and routing numbers, current and minimum balances, bank details, adjustment categories, as well as any notes you want to associate with the bank account.

From the bank account record, you can add alerts and adjustments, open the register, reconcile, and close accounts.

We can use the bank record to keep track of our bank activity ,reconciliations and how the bank account is performing.

To view the bank record go to Banking then click bank account.

Each financial institution sets the terms and conditions for each type of account it offers, which are classified in commonly understood types,such as deposit accounts, credit card accounts,_current accounts, loan accounts or many other types of account.

A customer may have more than one account. Once an account is opened, funds entrusted by the customer to the financial institution on deposit are recorded in the account designated by the customer Funds can be withdrawn from loan loaders.

WHY ARE BANK STATEMENTS IMPORTANT?

1 BUDGETING AND FINANCIAL PLANNING:

a bank statement is like a personal p & l statement. it allows account holders to keep track of their finances and plan for future expenditures. bank statements are also extremely helpful for budgeting, as they allow account holders to decipher how much they are spending on different categories.

2 RECONCILIATION AND IDENTIFICATION:

once the bank prepares a bank statement or e-statement at the end of the month, account holders are usually given 30-60 days to analyze the charges and reconcile their cash balance .since the bank statement contains all charges, along with the corresponding dates and payees, it can help account holders identify any fraudulent activity.

3 CREDIT VERIFICATION:

bank statements can also be useful to analyze the credit worthiness of the account holder. Most banks and financial institutions require verification of bank statements for the last 2-5 years before giving loans to individual clients. banks
use the individual’s bank statements and other credit
documents to analyze the creditworthiness of the borrower. It applies to most types of loans, including

Residential_mortgages, student loans, and loans for small businesses

INTRODUCTION OF THE PROJECT:

bank record system can keep the information of account type, account opening form, deposit, and searching the transaction, transaction report, individual account opening form, group account as a record. it displays records of transaction reports, statistical summary of account type and interest information.

this helps to provide the flexible solution to the user. here the chance of occurrence of error is less when compared with the existing system. it is fast, efficient and reliable. easy accessibility of data and avoids data redundancy and inconsistency.

a bank statement is like a personal p&l statement. it allows account holders to keep track of their finances and plan for future expenditures. bank statements are also extremely helpful for budgeting, as they allow account holders to decipher how much they are spending on different categories.

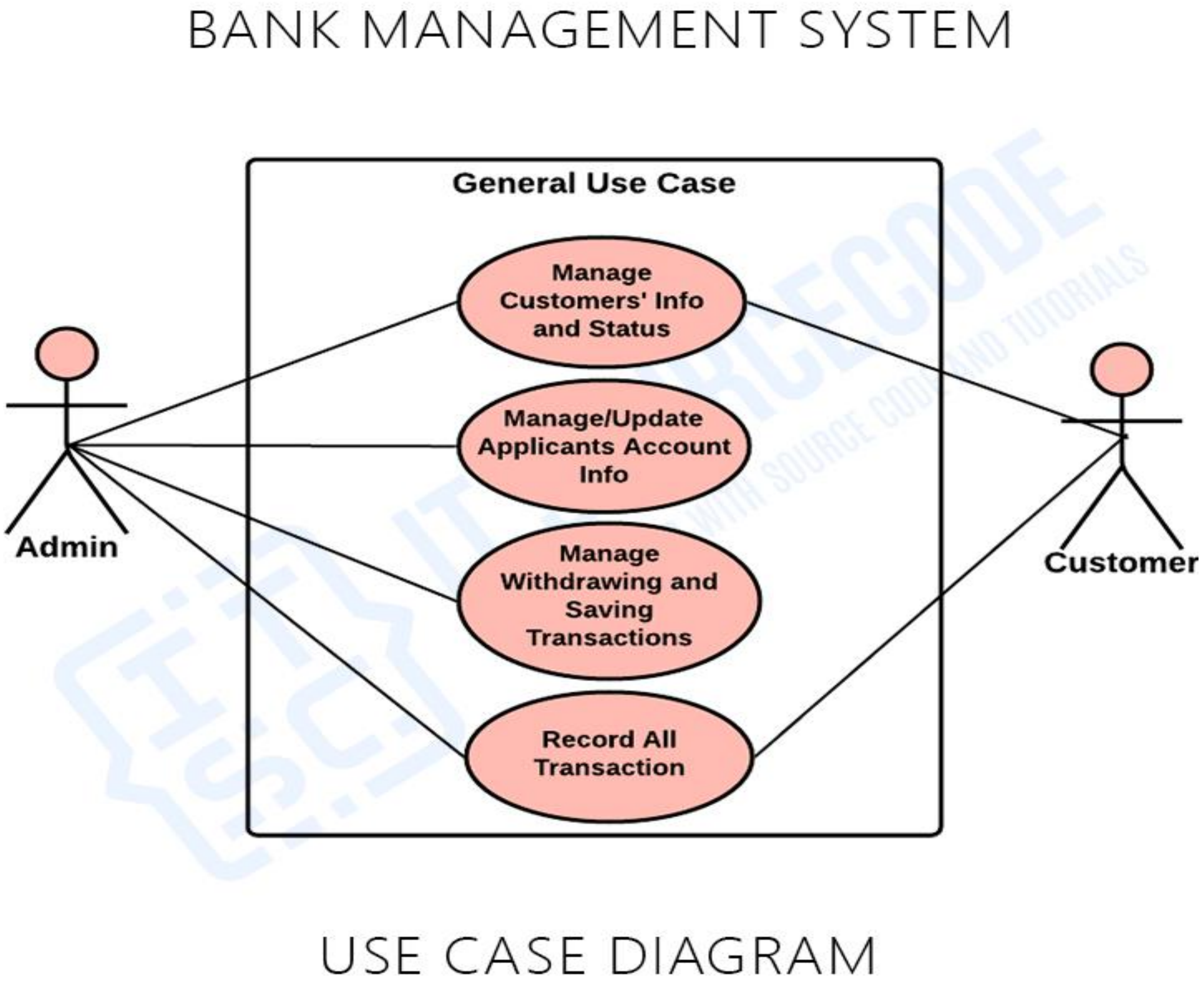

BANK MANAGEMENT SYSTEM
General Use Case
Manage Customers' Info and Status
Manage/Update Applicants Account Info
Manage Withdrawing and Saving Transactions
Record All Transaction
Admin
Customer
USE CASE DIAGRAM

Fig1.2: use of case diagram

INTRODUCTION OF BANKING RECORD SYSTEM:

This **banking record system** project in C++ is a simple console application developed without the use of graphics component. It is more of a database project in C++, and is built using the language's file handling mechanism.

It is suitable for beginners who want to learn how to add, edit, search, delete or modify records in a file, and how to use file as database overall .

The **source code** for this project is short–just over 300 lines. The coding has been presented in a very understandable manner. The source code needs to be compiled in **Dev-C++** IDE Version 5.11.

You can use this application to keep the records such as Account number, First Name, Last Name, Balance etc. of your regular costumer. Moreover, if you have a new customer, you can add and edit the account at any time. Many banks need an effective and accurate record system to be able to assure their records.

The record involves receiving banking records from various systems, determining the record rates associated with the customer's records, calculating the amount foreach customers, aggregating these records periodically to generate invoices, showing invoices to the customer, and collecting balance received from the customer Banking record System application is so simple to use.

PROJECT CATEGORY: Language description The project is based on the concepts of C++ Programming.

ABOUT THE PROGRAMMING LANGUAGE:

C++ is a general purpose programming language that was developed as an enhancement of the C language to include object-oriented paradigm. It is an imperative and a compiled language.

C++ is a middle-level language rendering it the advantage of programming low-level (drivers, kernels) and even higher-level applications (games, GUI, desktop apps etc.).

The basic syntax and code structure of both C and C++ are the same. Some of the features & key-points to note about the programming language are as follows:

Simple: It is a simple language in the sense that programs can be broken down into logical units and parts, has a rich library support and a variety of data-types.

MID-LEVEL LANGUAGE: it is a mid-level language as we can do both systems-programming (drivers, kernels, networking etc.) and build large-scale user applications (media players, photoshop, game engines etc.)

RICH LIBRARY SUPPORT: has a rich library support (both standard ~ built-in data structures, algorithms etc.) as well3rd party libraries (e.g. boost libraries) .

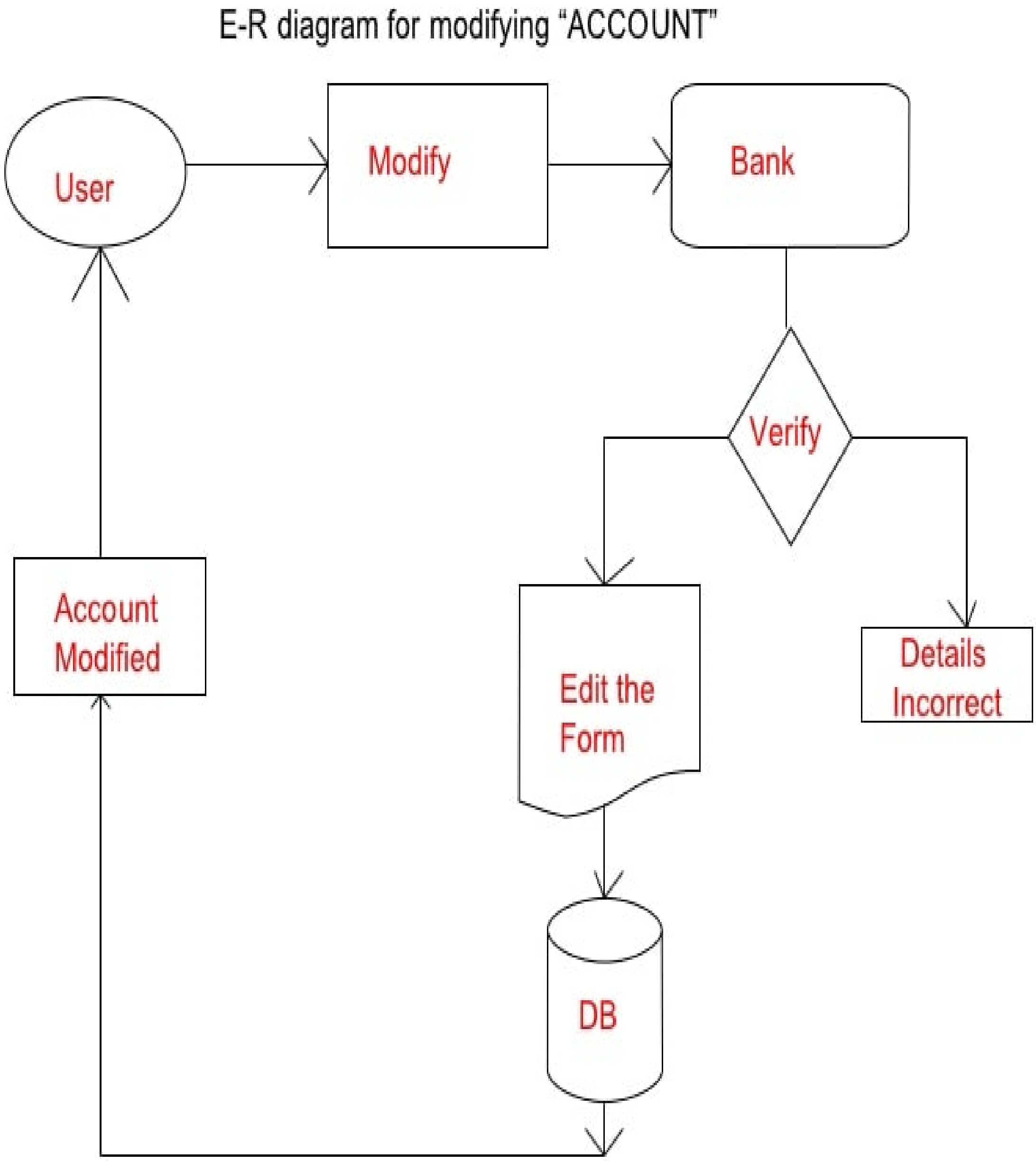


Fig1.3: E-R diagram for modifying "account"

USAGES OF PROGRAMMING LANGUAGE:

C++ FINDS VARIED USAGE IN APPLICATIONS SUCH AS:OPERATING SYSTEMS & SYSTEMS PROGRAMMING. E.G. LINUX-BASED OS (UBUNTU ETC.)BROWSERS (CHROME & FIREFOX)GRAPHICS & GAME ENGINES (PHOTOSHOP, BLENDER, UNREAL-ENGINE) DATABASE ENGINES (MYSQL, MONGODB, REDIS ETC.)

SOME INTERESTING FACTS ABOUT C++:

HERE ARE SOME AWESOME FACTS ABOUT C++ THAT MAY INTEREST YOU:

1. THE NAME OF C++ SIGNIFIES THE EVOLUTIONARY NATURE OF THE CHANGES FROM C. “++” IS THE C INCREMENT OPERATOR.

2. C++ IS ONE OF THE PREDOMINANT LANGUAGES FOR THEDEVELOPMENT OF ALL KIND OF TECHNICAL AND COMMERCIAL SOFTWARE.

3. C++ INTRODUCES OBJECT-ORIENTED PROGRAMMING, NOT PRESENT INC. LIKE OTHER THINGS, C++ SUPPORTS THE FOUR PRIMARY FEATURESOF OOP: ENCAPSULATION, POLYMORPHISM, ABSTRACTION, ANDINHERITANCE.

4. C++ GOT THE OOP FEATURES FROM SIMULA67 PROGRAMMINGLANGUAGE.

5. A FUNCTION IS A MINIMUM REQUIREMENT FOR A C++ PROGRAM TORUN.(AT LEAST MAIN() FUNCTION).

OBJECTIVES:

The researcher aims to create or develop a system that is capable and reliable in the whole record about the customer, retrieving and storing data in an appropriate way.

In particular it aims to. This **banking record system** project in C++ is a simple console application developed without the use of graphics component. It is more of a database project in C++, and is built using the language's file handling mechanism. It is suitable for beginners who want to learn how to add, edit, search, delete or modify records in a file, and how to use file as database overall .

The **source code** for this project is short–just over 300 lines. The coding has been presented in a very understandable manner. The source code needs to be compiled in **Dev-C++** IDE Version 5.11.You can use this application to keep the records such as Account number, First Name, Last Name, Balance etc. of your regular costumer. Moreover, if you have a new customer, you can add and edit the account at any time.

Many banks need an effective and accurate record system to be able to assure their records. The record involves receiving banking records from various systems, determining the record rates associated with the customer's records, calculating the amount foreach customers, aggregating these records periodically to generate invoices, showing invoices to the customer, and collecting balance received from the customer Banking record System application is so simple to use.

Banking Record System serves the following objectives:

Provide a database that will store information. Develop a system that will lessen process delay in terms of customer's record. Make an easy to use environment for users and customers .

Provides a convenient solution of record pattern. Add and maintain new entered category of records

. Add and maintain customer details. Search the customer using numbers of existing record.

Show the details of record from files. Show the details of programmer after exit.

The researcher aims to create or develop a system that is capable and reliable in the whole record about the customer, retrieving and storing data in an appropriate way.

In particular it aims to.

A bank account is a financial account maintained by a bank or other financial institution in which the_financial transactions between the bank and a customer are recorded.

Each financial institution sets the terms and conditions for each type of account it offers, which are classified in commonly understood types,such as deposit accounts, credit card accounts,_current accounts, loan accounts or many other types of account.

A customer may have more than one account. Once an account is opened, funds entrusted by the customer to the financial institution on deposit are recorded in the account designated by the customer Funds can be withdrawn from loan loaders.

In particular it aims to. This **banking record system** project in C++ is a simple console application developed without the use of graphics component. It is more of a database project in C++, and is built using the language's file handling mechanism. It is suitable for beginners who want to learn how to add, edit, search, delete or modify records in a file, and how to use file as database overall .

USER DEFINED FUNCTIONS USED:

File handling has been effectively used for each feature of this project. Here, I am going to describe these features in brief.

Add Record: For this feature void **read_data()** function has been used to add banking record into the file. It asks for information such as account number, first name, last name and balance to be entered.

Show/List Data: With the information provided in add record, the void **show_data()** function in this banking record system project in C++ show the record corresponding to a particular account number, first name and last name. Current balance of the account holder is displayed.

Search Record: When the function for this feature is first executed, it shows the total records in the file, and the user can then search by record number. If the record searched for is not found, the banking record system project in C++displays the message "Error in opening! File Not Found!!"

Edit Record: This works in similar manner to the Search feature. When the function for Edit Record is first executed ,I t shows the total records in the file, and the user can edit the information by providing record number.

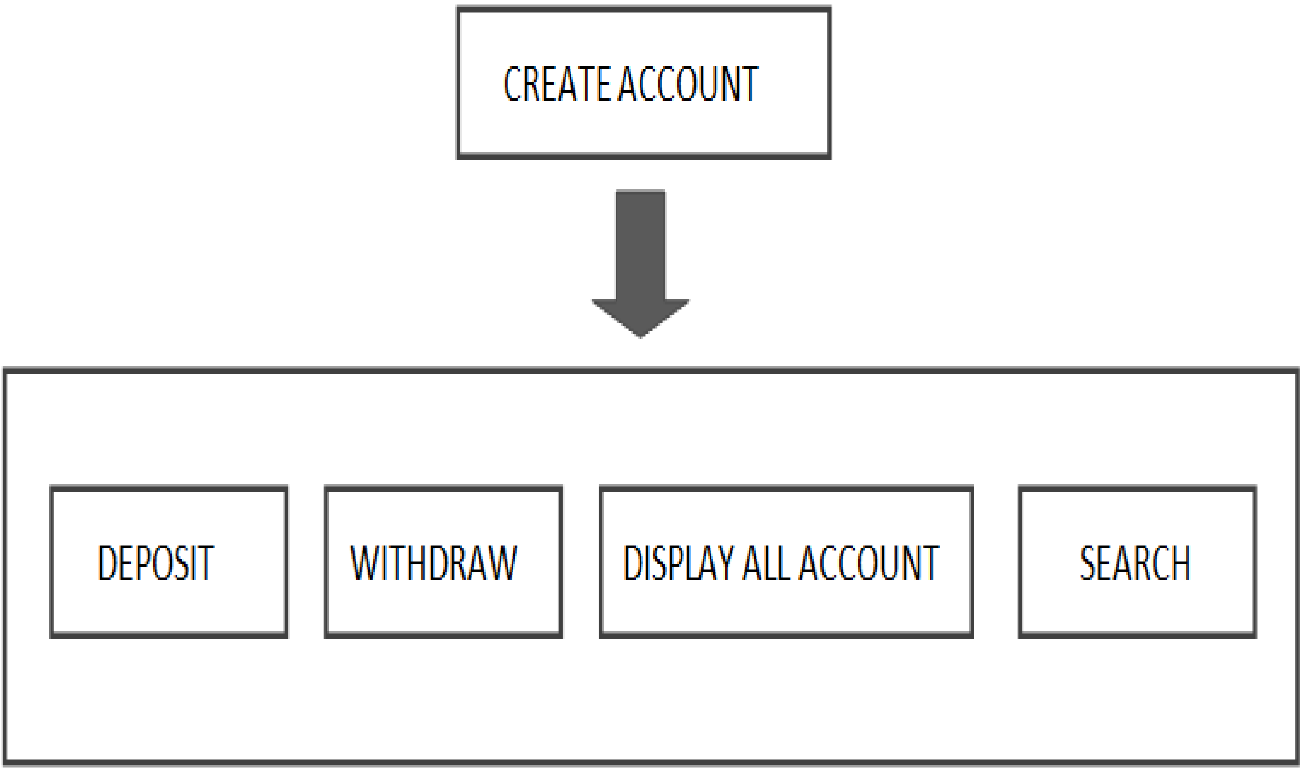


Fig1.4 UML diagram

HEADER FILES USED:

```
#include<conio.h>

#include<iostream>

#include<fstream>

#include<cstdlib>

#include<windows.h>
```

The word **conio.h** stands for **Con** sole- **I**nput **O**utput. The **conio. h** is a non-standard header file used in C and C++programming.

This file contains console input-output functions which are mostly used by MS-DOS compilers. Here we have explained some of the important and most widely used functions of **conio.h** header file.

Some of its most commonly used functions are clrscr, getch, get che, kb hit etc.

MODULES USED IN PROJECT Banking Record System :

Application is so simple to use. In order to use the application, click at the .exe file or run it directly using source code and then, you will have three options to

:1: Add record to file

2: Show record from file

3: Search Record from file

4: Update Record

5: Delete Record

6: Quit As per your need, enter 1, 2, 3, 4, 5 or 6 and follow the instructions provided by the application itself.

The project of **Banking Report system** can be used in many aspects, Firstly the application file generated can be used. Secondly the source code of Customer Billing System project in C++ can be used to learn C++ programming and its different features such as use of user defined functions, structures etc.

TECHNOLOGIES AND TOOLS: Software Used:

Languages Used : C++ Programming Language

Editor : Notepad++

IDE Used : Dev-C++ 5.11

Operating System: Windows XP Windows 7 Windows 8 Or any other version of windows

Hardware Used: CPU configuration o Processor : Intel Pentium or later

o RAM : 512 MB or later

o Hard Disk : 1 Gb Hard Disk Space or more

o Monitor : Any monitor

Code

```
#include <iostream>
#include <fstream>
#include <cctype>
#include <iomanip>
#include <time.h>
#include <stdlib.h>
using namespace std;
class account
{
	int accountnumber;
	char name[50];
	int deposit;
	char type;

public:
	void create_account();
	void show_account();
	void modify_account();
	void deposit_funds(int);
	void draw_funds(int);
	void get_report();
	int get_accountnumber();
	int get_funds();
	char get_accounttype();
};
```

```
void account::create_account()
{
	accountnumber = rand();
	cout << "\nThe account Number is : " << accountnumber;
	cout << "\nEnter The Name of The account Holder : ";
	cin.ignore();
	cin.getline(name, 50);
	cout << "\nEnter account type (enter s - saving or c - credit): ";
	cin >> type;
	type = toupper(type);
	cout << "\nEnter The Initial amount(500 or more for Saving and 1000 or more for
current ): ";
	cin >> deposit;
	cout << "\n\n\nAccount Created Successfully...";
}

void account::show_account()
{
	cout << "\nAccount Number: " << accountnumber;
	cout << "\nAccount Holder Name: " << name;
	cout << "\nType of Account: " << type;
	cout << "\nBalance amount: " << deposit;
}

void account::modify_account()
{
	cout << "\nAccount Number: " << accountnumber;
	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::deposit_funds(int x)
{
	deposit += x;
}

void account::draw_funds(int x)
{
	deposit -= x;
}

void account::get_report()
{
	cout << accountnumber << setw(10) << " " << name << setw(10) << " " << type << 
setw(6) << deposit << endl;
}

int account::get_accountnumber()
{
	return accountnumber;
}
```

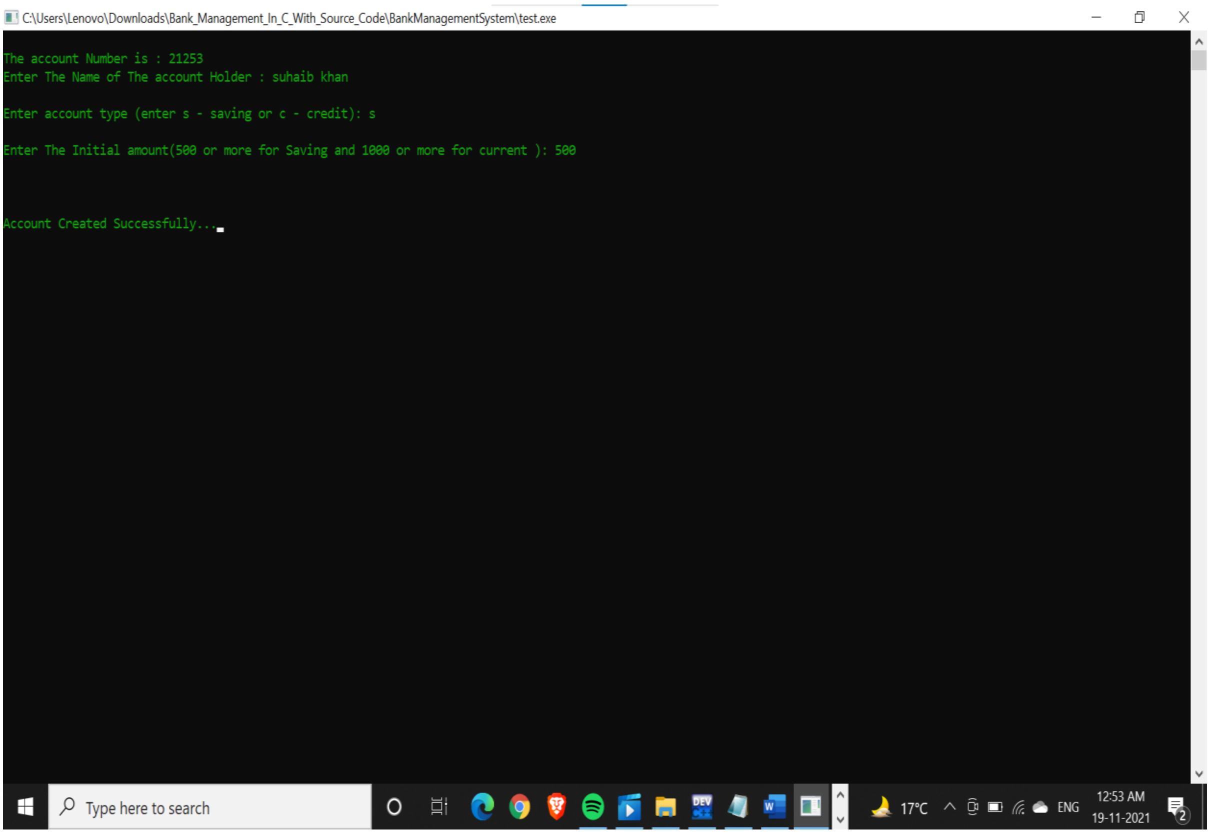
```
int account::get_funds()
{
	return deposit;
}

char account::get_accounttype()
{
	return type;
}

void write_account();
void display_sp(int);
void modify_account(int);
void delete_account(int);
void display_all();
void deposit_withdraw(int, int);

int main()
{
	srand(time(0));
	char opt;
	int num;
	for (;;)
	{
		system("color 05");
		system("cls");

		cout << "\n\t\tPress 1 to Create New Account";
```

```
        cout << "\n\t\tPress 2 to Deposit Funds";
        cout << "\n\t\tPress 3 to Withdraw Funds";
        cout << "\n\t\tPress 4 to Check Balance";
        cout << "\n\t\tPress 5 to View All Account Holder";
        cout << "\n\t\tPress 6 to Close an Account";
        cout << "\n\t\tPress 7 to Modify an Account";
        cout << "\n\t\tPress 8 to Exit the Program";
        cout <<
"\n\t\t@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@@@@@\n";
        cout << "\n\t\tOption: ";
        cin >> opt;
        system("cls");
        switch (opt)
        {
        case '1':
            system("color 02");
            write_account();
            break;
        case '2':
            system("color 03");
            cout << "\n\n\tEnter The account No. : ";
            cin >> num;
            deposit_withdraw(num, 1);
            break;
        case '3':
            system("color 06");
            cout << "\n\n\tEnter The account No. : ";
            cin >> num;
            deposit_withdraw(num, 2);
            break;
        case '4':
            system("color 08");
            cout << "\n\n\tEnter The account No. : ";
            cin >> num;
            display_sp(num);
            break;
        case '5':
            system("color 9");
            display_all();
            break;
        case '6':
            system("color 10");
            cout << "\n\n\tEnter The account No. : ";
            cin >> num;
            delete_account(num);
            break;
        case '7':
            system("color 11");
            cout << "\n\n\tEnter The account No. : ";
            cin >> num;
            modify_account(num);
            break;
        case '8':
            system("color 04");
            cout << "\n\n\tThanks for using Bank Management System";
            break;
        default:
            cout << "Invalid Option\n";
        }
        cin.ignore();
        cin.get();
        if (opt == '8')
            break;
    }
    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 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.get_accountnumber() == n)
        {
            ac.show_account();
            flag = true;
        }
    }
    inFile.close();
    if (flag == false)
        cout << "\n\nAccount number does not exist";
}

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.get_accountnumber() == n)
        {
            ac.show_account();
            cout << "\n\nEnter The New Details of account: " << endl;
            ac.modify_account();
            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 ";
}

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.get_accountnumber() != 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 display_all()
{
    int bankPassword = 111, enteredPassword;
    cout << "Enter the BANK PASSWORD: \n";
    cin >> enteredPassword;
    if (enteredPassword == bankPassword)
    {
        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 << "Account No.      Name          Type      Balance\n";
        cout << "========================================================\n";
        while (inFile.read(reinterpret_cast<char *>(&ac), sizeof(account)))
        {
            ac.get_report();
        }
        inFile.close();

    }
    else {
        cout << "Please enter the correct password.\n";
    }
}

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.get_accountnumber() == n)
        {
            ac.show_account();
            if (option == 1)
            {
                cout << "\n\n\tTO DEPOSITE AMOUNT ";
                cout << "\n\nEnter The amount to be deposited: ";
                cin >> amt;
                ac.deposit_funds(amt);
            }
            if (option == 2)
            {
                cout << "\n\n\tTO WITHDRAW AMOUNT ";
                cout << "\n\nEnter The amount to be withdraw: ";
                cin >> amt;
                int bal = ac.get_funds() - amt;
                if ((bal < 500 && ac.get_accounttype() == 'S') || (bal < 1000 && ac.get_accounttype() == 'C'))
                    cout << "Insufficience balance";
                else
                    ac.draw_funds(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 ";

}
```

OUTPUT SCREEN FIGURE

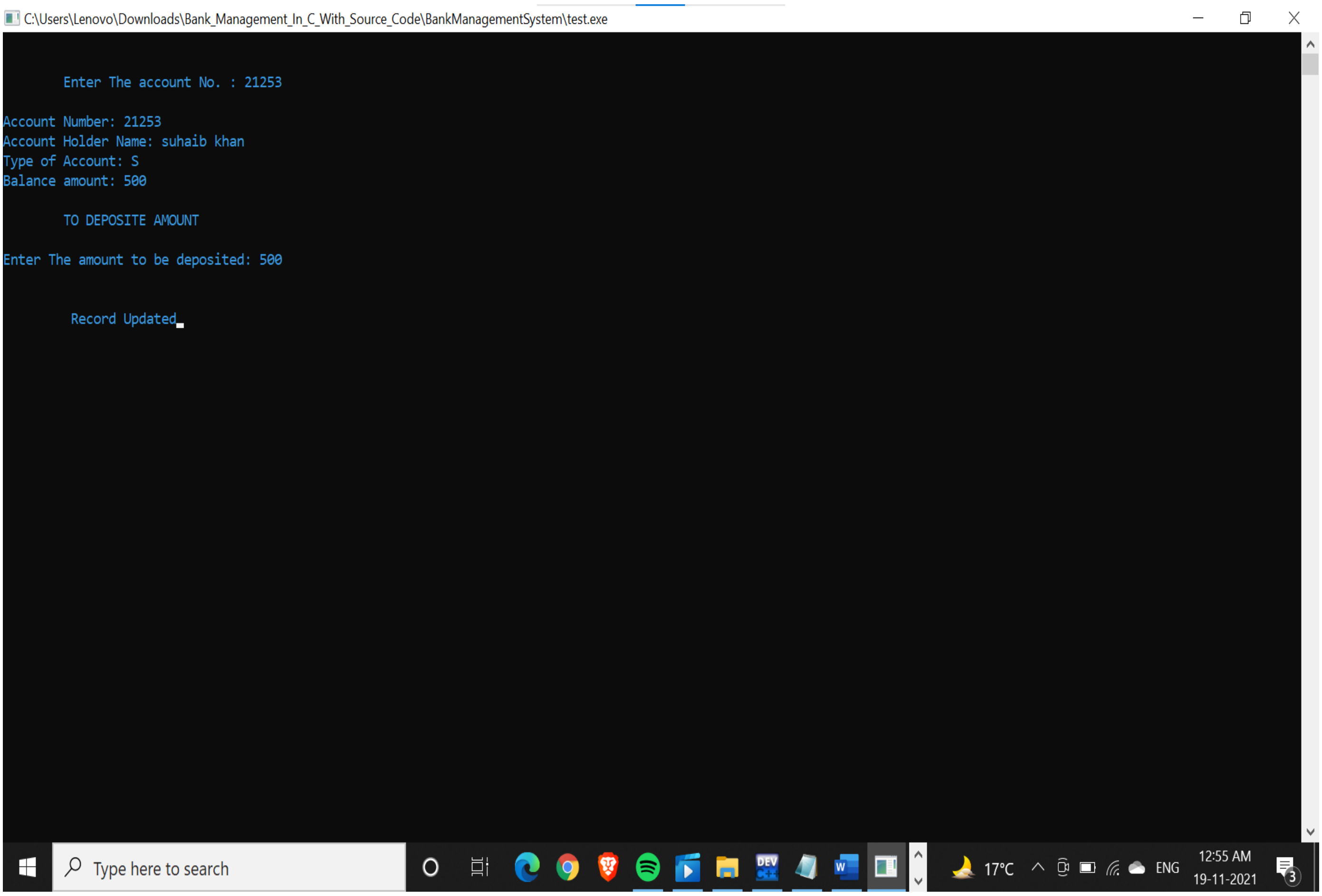
Press 1= to create account:


C:\Users\Lenovo\Downloads\Bank_Management_In_C_With_Source_Code\BankManagementSystem\test.exe
Type here to search
17°C
ENG
12:53 AM
19-11-2021

```
The account Number is : 21253
Enter The Name of The account Holder : suhaib khan

Enter account type (enter s - saving or c - credit): s

Enter The Initial amount(500 or more for Saving and 1000 or more for current ): 500



Account Created Successfully...
```

Fig2.1: to create account

Press 2= to deposit fund :

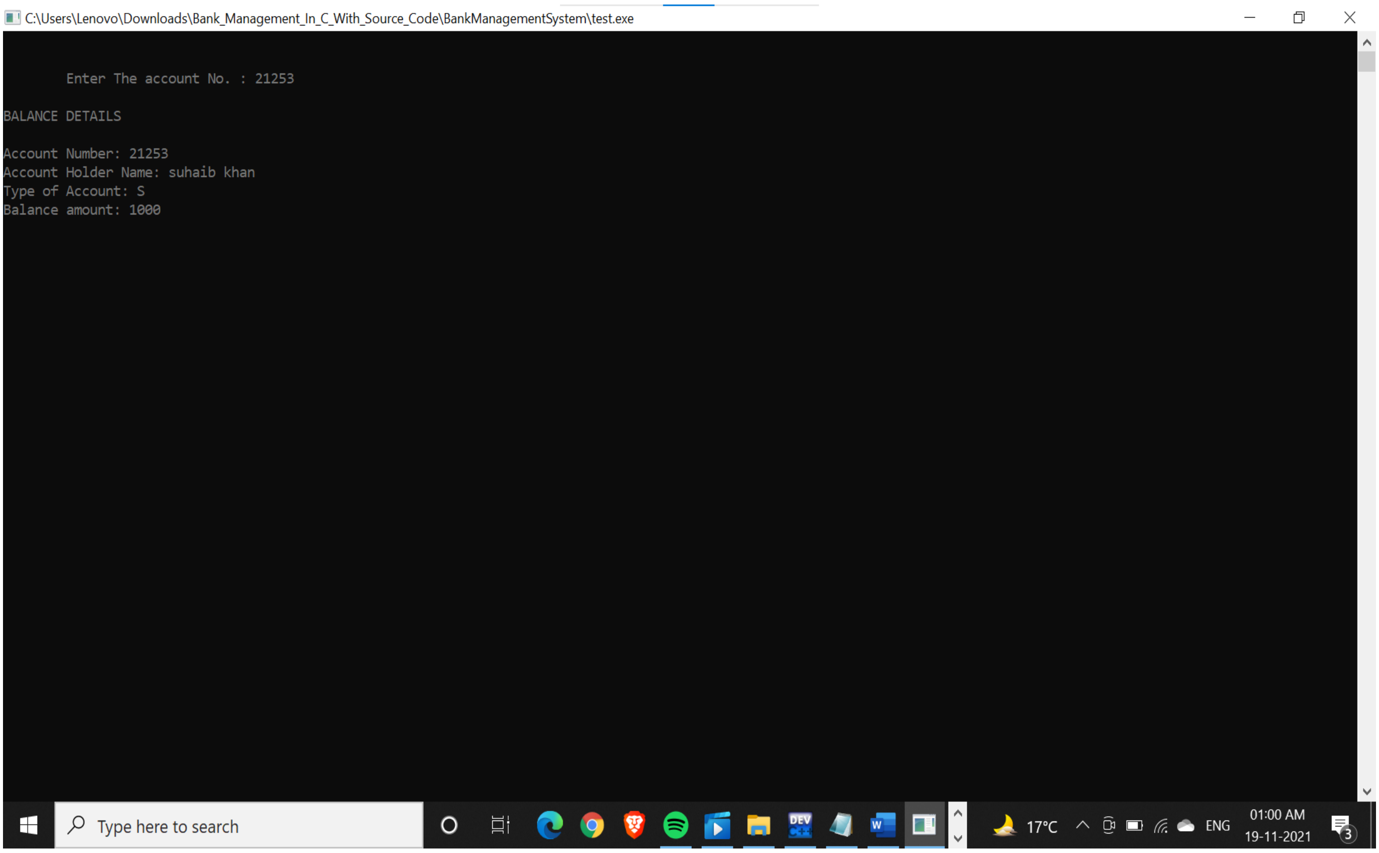


C:\Users\Lenovo\Downloads\Bank_Management_In_C_With_Source_Code\BankManagementSystem\test.exe

```
        Enter The account No. : 21253

Account Number: 21253
Account Holder Name: suhaib khan
Type of Account: S
Balance amount: 500

        TO DEPOSITE AMOUNT

Enter The amount to be deposited: 500


         Record Updated_
```

Fig2.2: to deposit fund

Press 3 = to check balance :

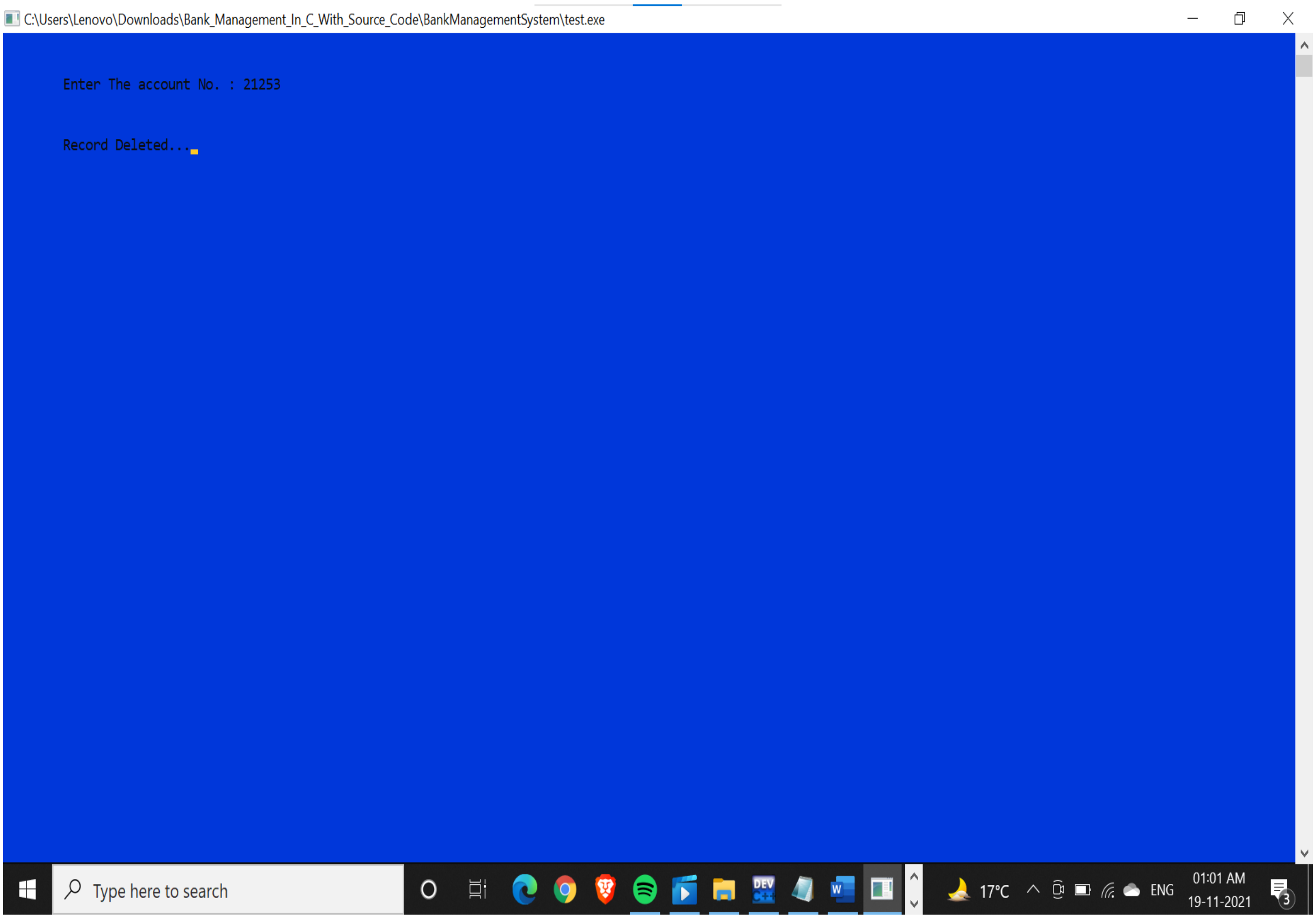


```
        Enter The account No. : 21253

BALANCE DETAILS

Account Number: 21253
Account Holder Name: suhaib khan
Type of Account: S
Balance amount: 1000
```

C:\Users\Lenovo\Downloads\Bank_Management_In_C_With_Source_Code\BankManagementSystem\test.exe
Type here to search
17°C
ENG
01:00 AM
19-11-2021

Fig2.3: to check balance

Press 4 = to close an account:

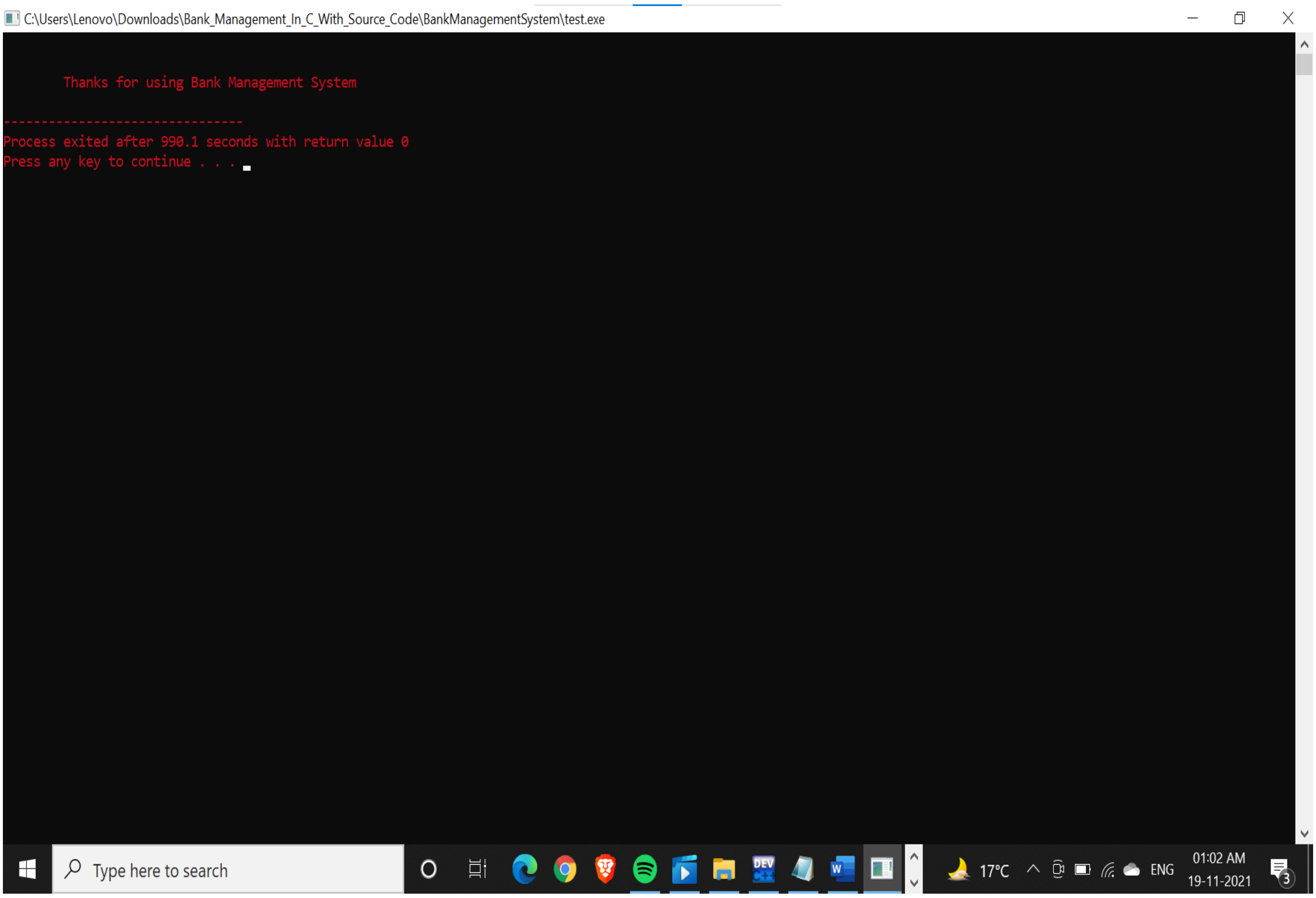


```
Enter The account No. : 21253


Record Deleted...
```

C:\Users\Lenovo\Downloads\Bank_Management_In_C_With_Source_Code\BankManagementSystem\test.exe
Type here to search
17°C
ENG
01:01 AM
19-11-2021

Fig2.4: to close an account

Press 5 = to exit the program:



C:\Users\Lenovo\Downloads\Bank_Management_In_C_With_Source_Code\BankManagementSystem\test.exe

```
        Thanks for using Bank Management System

--------------------------------
Process exited after 990.1 seconds with return value 0
Press any key to continue . . .
```

Fig2.5: to exit the program

FUTURE SCOPE

1. This project will help the bankers in fast reporting.

2.This project enable banker to maintain a great data base of all Customer ’ s details from the software.

3. Project will enable to see report regarding query.

4. It is easy to maintain in future prospect.

CONCLUSION

This was my project of System Design about “Banking Record System”. Development of this System takes a lot of efforts.

I think this system gave a lot of satisfaction. Though every task is never said to be perfect in this development field even more improvement may be possible in this system.

I learnt so many things and gained a lot of knowledge about development field.

I hope this will prove fruitful.

BIBLIOGRAPHY

1.Books referred = the complete reference 4th edition by Herbert schildt .

2. c/c++ programming book from pragya publication

3.Websites referred = google.com , stackoverflow.com, geeksforgeeks.com , javatpoint.com.