**Final Project Report**

**On**

**C++ Bank Application Systems**

****

By

Michael Stanley Chinaza

L2AC

21th june, 2022

C++ Bank Application Systems

**Abstract.**

The adoption of Electronic Banking through industrial businesses has been in life since the mid 90s, an awful lot more in variety because of lower operating expenses related to it. Electronic banking has initially been within the shape of automated teller machines and cellphone transactions. These days, it's been switched with the aid of the Internet, a brand new delivery channel for banking offerings that advantages each customers and banks. Internet banking device offerings can encompass: Open an account, Balance enquiry, Request for Cheque ebook, Beneficiary bills (EFT), Viewing monthly. Furthermore, purchaser’s applications for electronic banking centres are increasing as the value of financial savings on transactions over the Internet are huge.

**Table of Contents**

CoverPage------------------------------------1

Title Page------------------------------------ 2

Abstract--------------------------------------3

1. Introduction--------------------------------- 6

*1.1. Use case diagram*------------------------------7

*1.2. Programming language used*----------------------8

1. Aims and benefits-----------------------------9
2. Proposed Objectives--------------------------10
3. Problem Description--------------------------11
4. Module description---------------------------12
5. Database Design----------------------------- 13
6. Data structures used------------------------- 14
7. Technologies and tools used-------------------15
8. screen shots and testing---------------------- 16
9. conclusion--------------------------------29

**List of Figures**

Figure 1.1. Transaction between bank and account holders stored in the database----------------------7

Figure 1.2. Use case diagram------------------------7

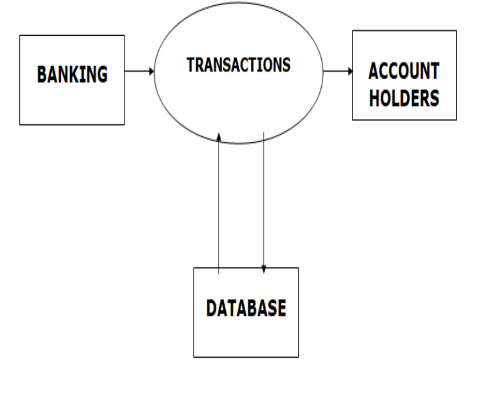
Figure 5.1 UML Diagram---------------------------12

Figure 9.1. Main menu---------------------------- 28

1. **Introduction**

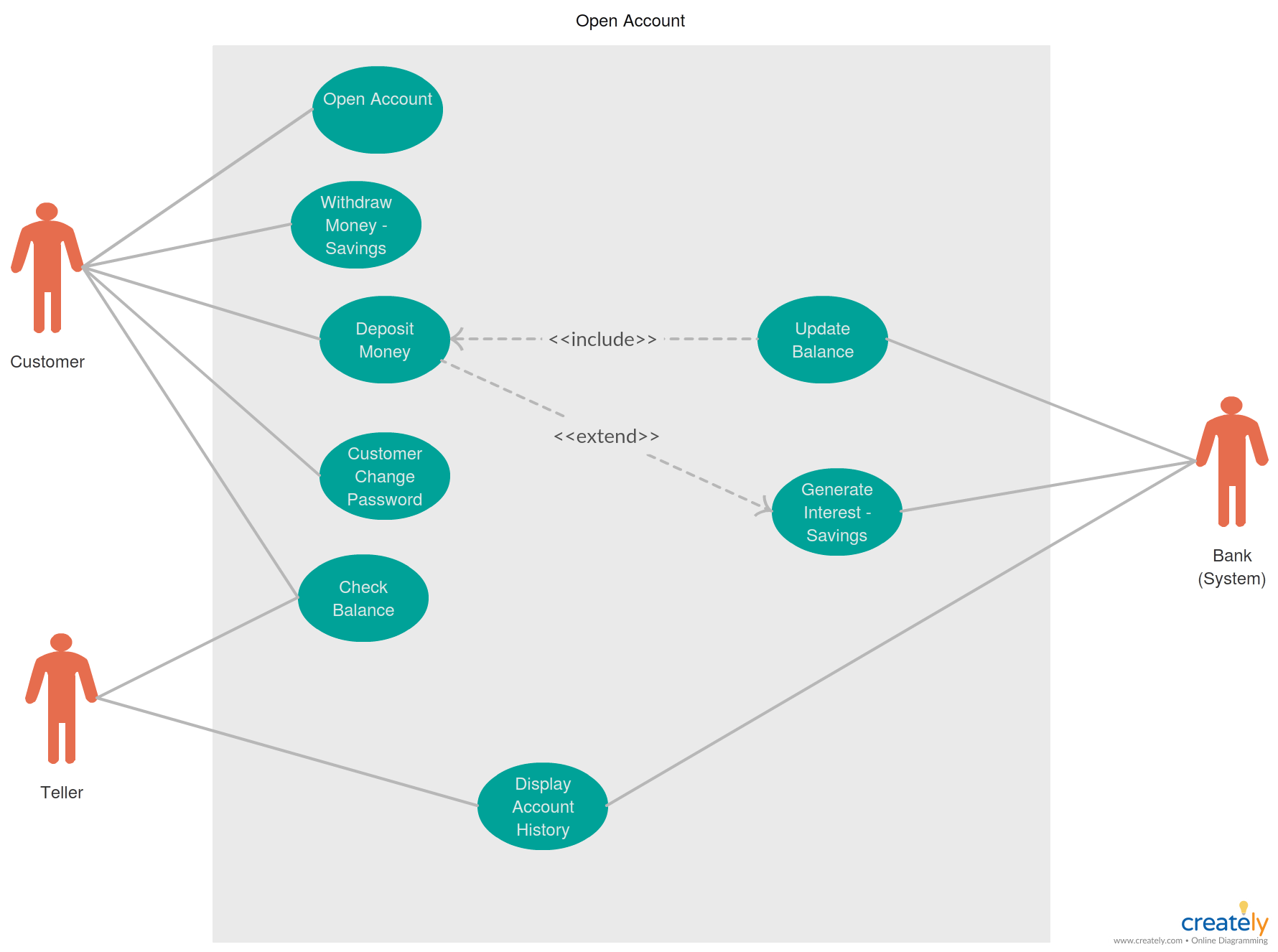
The C++ Bank application software can keep the statistics of account type, account commencing form, deposit , and look at the transaction, transaction report, individual account commencing form, organisation account as a document. It displays statistics of transaction reviews, statistical precis of account kind and interest information. This enables them to provide the bendy answer to the consumer. Right here the risk of incidence of mistakes is less while as compared with the existing gadget. It's miles fast, green and reliable. Smooth accessibility of facts and avoids statistics redundancy and inconsistency. Banking record system challenge in C++ is an easy console application advanced without the use of photos. It is more of a database task in C++, and is built using the language’s document dealing with mechanism. It is appropriate for beginners who want to learn how to upload, edit, seek, delete or alter facts in a report, and a way to use documents as database universals.

A financial institution statement is sort of a non-public p&l statement. It lets account holders maintain music in their budget and plan for future fees. Bank statements also are extremely useful for budgeting, as they allow account holders to decipher how a whole lot they are spending on unique categories.



**figure 1.1 :** Transaction between bank and account holders stored in the database.

***1.1. Use Case diagram***

******

**Figure 1.2.** Use case diagram,

***1.2. Programming language used***

C++ is a well known purpose programming language that was advanced as an enhancement of the C language to consist of an object-oriented paradigm. It is a vital and a compiled language. C++ is a middle-degree language rendering it the benefit of programming low-stage (drivers, kernels) or even higher-level packages (games, GUI, desktop apps and so on.). The simple syntax and code structure of both C and C++ are the same. Some of the capabilities key-factors to observe approximately the programming language are as follows:

SIMPLE: It is an easy language in the sense that applications may be broken down into logical devices and parts, has a rich library assist and a ramification of statistics-kinds.

MID-LEVEL LANGUAGE: it is a mid-level language as we will do each structure-programming (drivers, kernels, networking and so on.) and construct big-scale person applications (media gamers, photoshop, sport engines and many others.)

RICH LIBRARY SUPPORT: has a wealthy library support (both preferred ~ built-in statistics structures, algorithms and many others.) as well 3rd party libraries (e.G. Enhance libraries).

**2.Aim and Benefits**

The major aim is to create or increase an app that is capable and reliable within the entire report, approximately the purchaser, retrieving and storing information in the ideal way. Also, to develop a software for solving financial applications for a customer in a banking environment in order to nurture the needs of an end banking user by providing various ways to perform banking tasks. Also to enable the user’s workspace to have additional functionalities which are not provided under conventional banking software. Many banks want a powerful and accurate document gadget so as to guarantee their statistics. The file includes receiving banking statistics from numerous systems, determining the record quotes associated with the purchaser’s facts, calculating the amount foreach clients, aggregating those records periodically to generate invoices, showing invoices to the patron, and gathering stability received from the consumer Banking document System application is so simple to use.

The application may be extraordinarily useful for the Customers intending to apply and function their financial institution account and will get diverse advantages within the discipline of management of debts on a clean and user-friendly platform. “Bank Application System”, is a simple software, that's specially generated and designed for the financial institution with a purpose to input the applicant facts approximately his or her bank account and may carry out different characteristics like foreign money trade. It is the consumer name and ID included as properly. Following are the predominant objectives at the back of the new proposed machine: It creates a person friendly surroundings, where an ordinary person can access through all the advantages of the system.

**3.proposed objectives**

In particular it aims to. This banking record system project in C++ is a simple console application developed without the use of graphics components. 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 the file as a 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 visual basics with C++ compiler. You can use this application to keep the records such as Account number, First Name, Last Name, Balance etc. of your regular customer. Moreover, if you have a new customer, you can add and edit the account at any time.

**4.problem description**

In recent times, people leave their homes to the bank in search of many bank activities, like check account balance deposit, withdrawals, transfers, cash collections and loan collections. Sometimes this causes too many crowds at the bank making people lose their other day to day activities and some risk of getting robbed. Most of the time there are some people with busy schedules that don't have much time to go to the bank to make transactions every day! So a fix for all this will be a bank application to make it easier and comfortable for everyone anywhere.

**5. Module description**

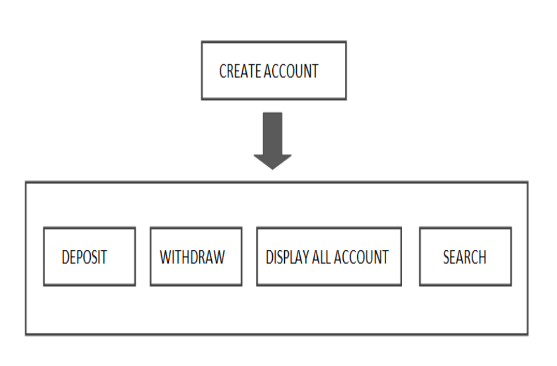


Figure 5.1; UML diagram

* Create Account: Selecting this creates a new record for the consumer by accepting input inclusive of account variety, call and amount Open Account Opens the new account for the person by accepting input including account range, name and minimum stability.
* Deposit: Provides options to deposit amount from the given account number.
* Withdraw: This feature provides partitions for withdrawing any amount from the given account number.
* Search: Search Enables to search for the information of the given account variety
* Display all created accounts: this feature displays every account created and stored in the database.

**6. Database Design**

A database is an organised mechanism that has the capability of storing facts via which a user can retrieve saved records in an effective and green way. The records are the cause of any database and should be blanketed. The database layout is a two level manner. In step one, consumer necessities are accrued together and a database is designed in order to meet those requirements as really as possible. This step is referred to as Information Level Design and it is taken impartially by any DBMS. In the second step, this Information level layout is transferred into a layout for the particular DBMS to be able to be used to enforce the machine in question. This step is referred to as Physical Level Design, concerned with the traits of the precise DBMS so as to be used. A database design runs parallel with the system design. The organisation of the facts within the database is aimed to gain the subsequent two major goals. Data Integrity Data independence Normalisation is the technique of decomposing the attributes in a utility, which ends up in a hard and fast table with quite simple shape. The reason for normalisation is to make tables as simple as feasible. Normalisation is carried out on this machine for the subsequent motives. To shape the information in order that there is no repetition of information , this enables saving. To permit easy retrieval of records in reaction to question and record requests. To simplify the renovation of the information via updates, insertions, deletions.

**7. Data Structures used**

The data structures used in the project are listed below;

* Queue
* Stacks
* Linked-list
* Tree [Binary search Tree]

QUEUE; Queue in C++ is a sort of data structure that is designed to work as a First in First Out (FIFO) information box. Data entered from one facet of a queue is extracted from the alternative aspect of a queue in a FIFO manner. In C++, std:: queue class provides all queue related functionalities to programmers.

STACKS; A stack is a preferred C++ container adapter, designed for use in a LIFO context, and is implemented with an interface/wrapper to the type surpassed to it as a template argument, which defaults to a deque.

Linked-List; A Link-listing is a set of nodes that include a information part and a subsequent pointer that includes the reminiscence deal with the following detail in the list. The closing element in the listing has its subsequent pointer set to NULL, thereby indicating the give up of the listing. Linked listing is used to delete an Account.

Tree[Binary Search Tree]; Binary search tree in C++ is defined as a statistics shape that consists of the node-based totally binary tree in which every node consists of at maximum 2 nodes which can be referred to as child nodes.

Binary Search Tree is used for the records creation as it will be green in phrases of space and complexity to look at facts.

**8. Technologies And Tools Used**

Software used

Languages Used : C++ Programming Language

Editor : Notepad++

IDE Used : Dev-C++, visual basics

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

Hardware Used

CPU configuration of Processor : Intel Pentium or later

RAM : 512 MB or later Hard Disk : 1 Gb Hard Disk Space or more

Monitor : Any monitor

**9. Screenshot and Testing**

***Source Code***

*#include<iostream>*

*#include<fstream>*

*#include<cctype>*

*#include<iomanip>*

*using namespace std;*

*class account*

*{*

*int acno;*

*char name[50];*

*int deposit;*

*char type;*

*public:*

*void create\_account();*

*void show\_account() const;*

*void modify();*

*void dep(int);*

*void draw(int);*

*void report() const;*

*int retacno() const;*

*int retdeposit() const;*

*char rettype() const;*

*};*

*void account::create\_account()*

*{*

*system("CLS");*

*cout<<"\n\t\t\tEnter the Account No. : ";*

*cin>>acno;*

*cout<<"\n\n\t\t\tEnter the Name of the Account holder : ";*

*cin.ignore();*

*cin.getline(name,50);*

*cout<<"\n\t\t\tEnter Type of the Account (C/S) : ";*

*cin>>type;*

*type=toupper(type);*

*cout<<"\n\t\t\tEnter The Initial amount : ";*

*cin>>deposit;*

*cout<<"\n\n\t\t\tAccount Created..";*

*}*

*void account::show\_account() const*

*{*

*cout<<"\n\t\t\tAccount No. : "<<acno;*

*cout<<"\n\t\t\tAccount Holder Name : ";*

*cout<<name;*

*cout<<"\n\t\t\tType of Account : "<<type;*

*cout<<"\n\t\t\tBalance amount : "<<deposit;*

*}*

*void account::modify()*

*{*

*cout<<"\n\t\t\tAccount No. : "<<acno;*

*cout<<"\n\t\t\tModify Account Holder Name : ";*

*cin.ignore();*

*cin.getline(name,50);*

*cout<<"\n\t\t\tModify Type of Account : ";*

*cin>>type;*

*type=toupper(type);*

*cout<<"\n\t\t\tModify Balance amount : ";*

*cin>>deposit;*

*}*

*void account::dep(int x)*

*{*

*deposit+=x;*

*}*

*void account::draw(int x)*

*{*

*deposit-=x;*

*}*

*void account::report() const*

*{*

*cout<<acno<<setw(10)<<" "<<name<<setw(10)<<" "<<type<<setw(6)<<deposit<<endl;*

*}*

*int account::retacno() const*

*{*

*return acno;*

*}*

*int account::retdeposit() const*

*{*

*return deposit;*

*}*

*char account::rettype() const*

*{*

*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()*

*{*

*char ch;*

*int num;*

*do*

*{*

*system("CLS");*

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

*cout<<"\t\t\t\tBANK APPLICATION C++";*

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

*cout<<"\t\t\t\t ::MAIN MENU::\n";*

*cout<<"\n\t\t\t\t1. NEW ACCOUNT";*

*cout<<"\n\t\t\t\t2. DEPOSIT AMOUNT";*

*cout<<"\n\t\t\t\t3. WITHDRAW AMOUNT";*

*cout<<"\n\t\t\t\t4. BALANCE ENQUIRY";*

*cout<<"\n\t\t\t\t5. ALL ACCOUNT HOLDER LIST";*

*cout<<"\n\t\t\t\t6. CLOSE AN ACCOUNT";*

*cout<<"\n\t\t\t\t7. MODIFY AN ACCOUNT";*

*cout<<"\n\t\t\t\t8. EXIT";*

*cout<<"\n\n\t\t\t\tSelect Your Option (1-8): ";*

*cin>>ch;*

*switch(ch)*

*{*

*case '1':*

*write\_account();*

*break;*

*case '2':*

*system("CLS");*

*cout<<"\n\n\t\t\tEnter The account No. : "; cin>>num;*

*deposit\_withdraw(num, 1);*

*break;*

*case '3':*

*system("CLS");*

*cout<<"\n\n\t\t\tEnter The account No. : "; cin>>num;*

*deposit\_withdraw(num, 2);*

*break;*

*case '4':*

*system("CLS");*

*cout<<"\n\n\t\t\tEnter The account No. : "; cin>>num;*

*display\_sp(num);*

*break;*

*case '5':*

*display\_all();*

*break;*

*case '6':*

*system("CLS");*

*cout<<"\n\n\t\t\tEnter The account No. : "; cin>>num;*

*delete\_account(num);*

*break;*

*case '7':*

*system("CLS");*

*cout<<"\n\n\t\t\tEnter The account No. : "; cin>>num;*

*modify\_account(num);*

*break;*

*case '8':*

*system("CLS");*

*cout<<"\n\n\t\t\tBrought To You By code-projects.org";*

*break;*

*default :cout<<"\a";*

*}*

*cin.ignore();*

*cin.get();*

*}while(ch!='8');*

*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<<"\n\t\t\tBALANCE 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\n\t\t\tAccount 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.retacno()==n)*

*{*

*ac.show\_account();*

*cout<<"\n\n\t\t\tEnter The New Details of account"<<endl;*

*ac.modify();*

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

*File.seekp(pos,ios::cur); //fncallat1353*

*File.write(reinterpret\_cast<char \*> (&ac), sizeof(account));*

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

*found=true;*

*}*

*}*

*File.close();*

*if(found==false)*

*cout<<"\n\n\t\t\tRecord 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.retacno()!=n)*

*{*

*outFile.write(reinterpret\_cast<char \*> (&ac), sizeof(account));*

*}*

*}*

*inFile.close();*

*outFile.close();*

*remove("account.dat");*

*rename("Temp.dat","account.dat");*

*cout<<"\n\nRecord Deleted ..";*

*}*

*void display\_all()*

*{*

*system("CLS");*

*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 Type Balance\n";*

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

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

*{*

*ac.report();*

*}*

*inFile.close();*

*}*

*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\t\t\tTO DEPOSITSS AMOUNT";*

*cout<<"\n\n\t\t\tEnter The amount to be deposited: ";*

*cin>>amt;*

*ac.dep(amt);*

*}*

*if(option==2)*

*{*

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

*cout<<"\n\n\t\t\tEnter 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);//fn1353*

*File.write(reinterpret\_cast<char \*> (&ac), sizeof(account));*

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

*found=true;*

*}*

*}*

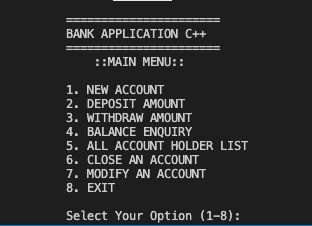
*File.close();*

*if(found==false)*

*cout<<"\n\n\t\t\tRecord Not Found ";*

*}*

***Output***

******

**Figure 9.1**: main menu

**10. Conclusion**

This became my challenge of System Design approximately “Banking Application Systems”. Development of this System takes plenty of effort. I suppose this device gave me a whole lot of satisfaction. Though every mission is by no means stated to be best in this improvement subject even more improvement may be feasible on this device. I learnt so many matters and gained a lot of know-how about the development field. I hope this can prove fruitful. Some few additional information listed below;

* This project will help the bankers in speedy reporting.
* This project allows the banker to preserve a fantastic information base of all Customer’s info from the software program.
* Project will enable me to see documents regarding the query.
* It is simple to hold in the future.