Banking Management System

21CSC101T – Object Oriented Design and Programming

Mini Project Report

Submitted by

ADITYA SHARMA [Reg. No.: RA2211026010295] B. Tech CSE-AI&ML

KHUSHI UPADHYAY [Reg. No.: RA2211026010312] B. Tech

CSE-AI&ML

GAURI GUPTA [REG. NO.RA2211026010359] B. Tech CSE-AI&ML

NEELANSH BHARGAVA [REG. NO. RA2211026010360]] B.Tech CSE AI&ML

SRM INSTITUTION OF SCIENCE AND TECHNOLOGY
KATTANKULATHUR-603203

BONAFIDE CERTIFICATE

Certified that this Course Project Report titled **"Banking Management system in C++** is the bonafide work done by **GAURI GUPTA[RA2211026010359], ADITYA SHARMA[RA2211026010295], KHUSHI UPADHYAY[RA2211026010312], NEELANSH BHARGAVA[RA2211026010360]** who carried out under my supervision. Certified further, that to the best of my knowledge, the work reported herein does not form part of any other work.

SIGNATURE	**HEAD OF THE DEPARTMENT**
Faculty In-Charge	**Dr.Annie Uthra**
Dr. S. Aasha Nandhini	**Professor and Head,**
Assistant Professor	**Department of Computational Intelligence**
Department of Computing Technologies	**SRM Institute of Science and Technology**
SRM Institute of Science and Technology	

Aim:

To design an object-oriented model for banking management system using star UML application and to complement it using C++.

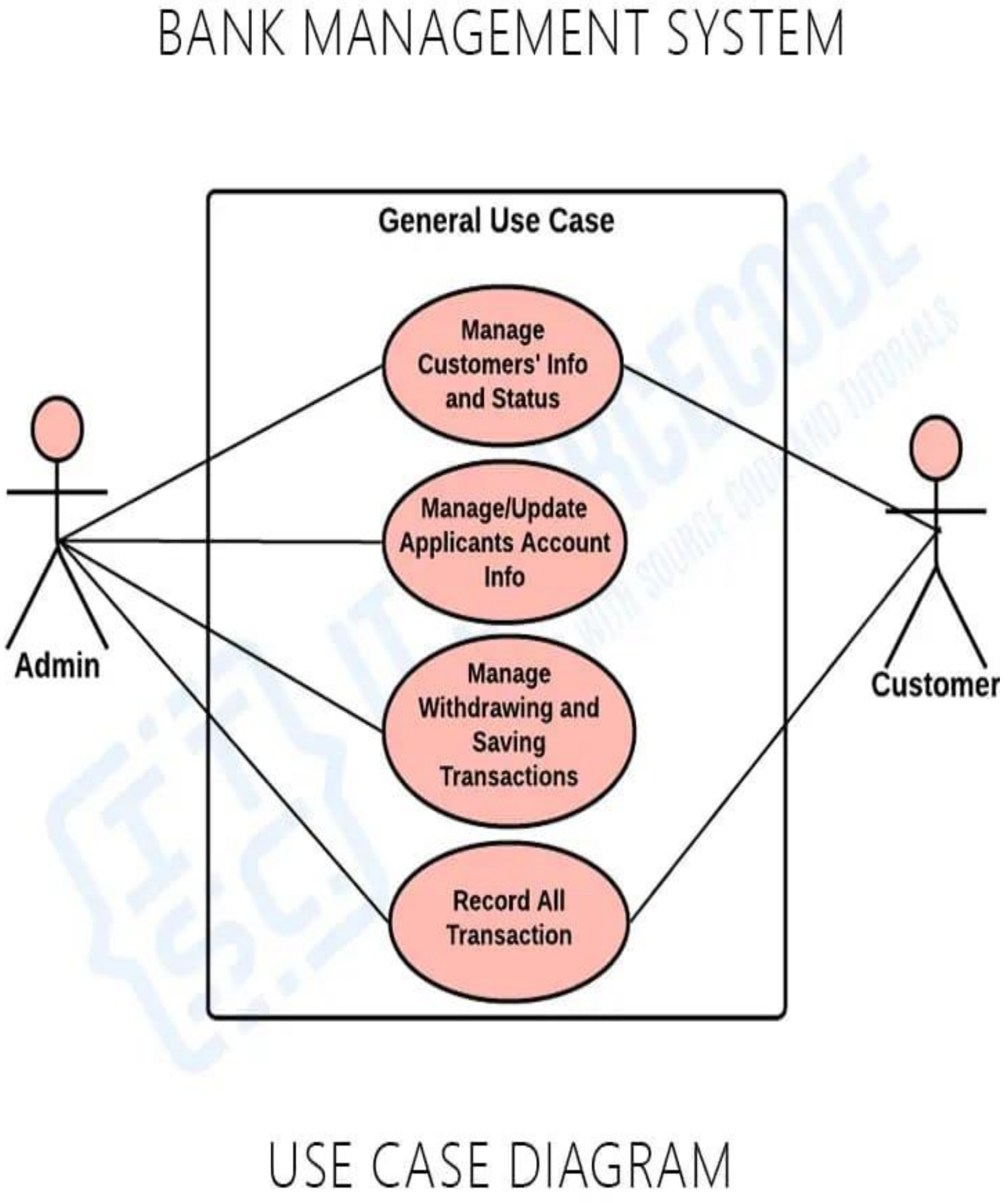
Problem statement:

The first thing that you must do before building UML diagrams for the system is to collect as much bank management data as you can. You will need to know how they manage bank information, processing bank transactions and distribution of information. You must complete all the necessary information in order to provide a quality system for them. Then you will apply the data gathered in designing the UML diagram.

The purpose of building the system is to provide the bank employees and owners the easier and efficient way of doing their tasks. And in order to do that, you must complete the needed diagrams to perfectly furnish your project. Build your project completely using UML diagrams to help you and keep you from repeating its development.

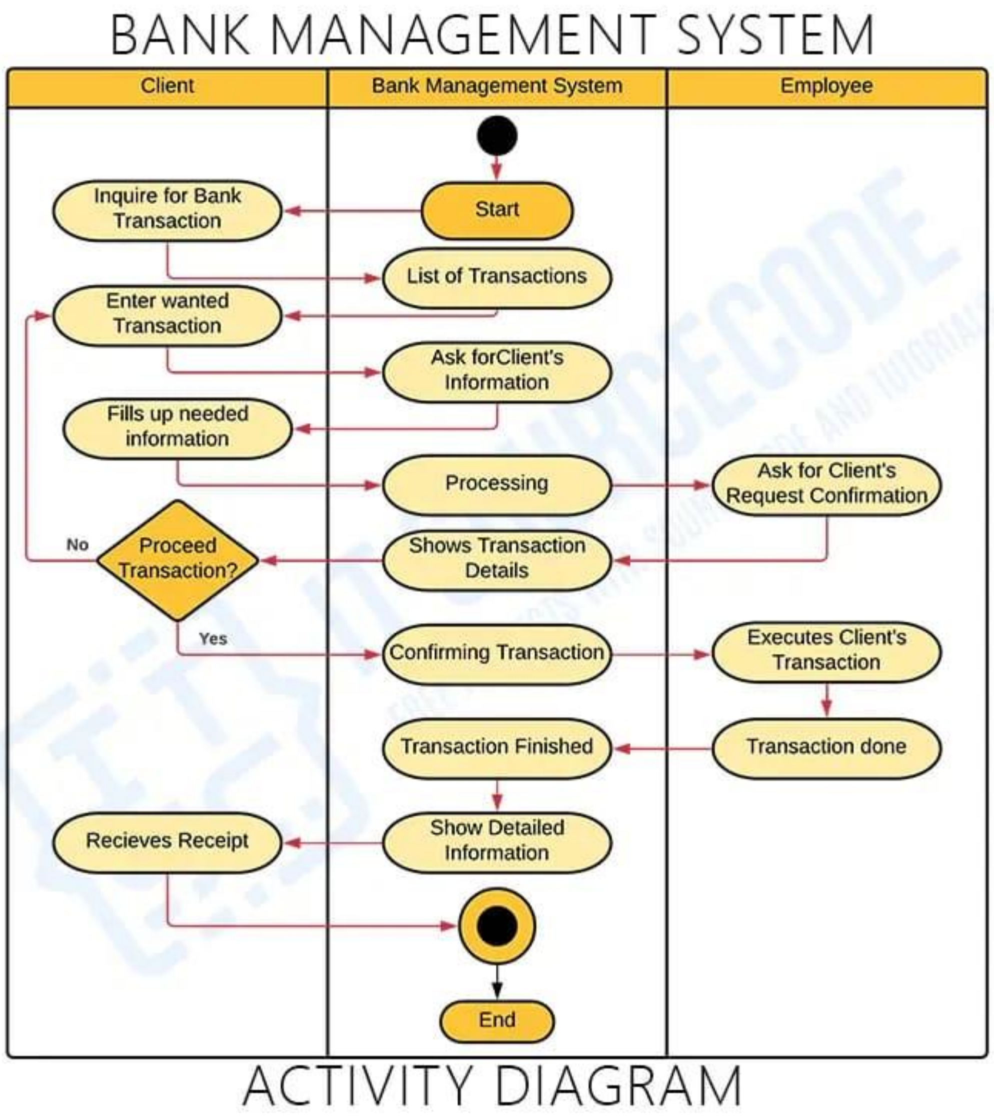
Use case Diagram

The use cases in the diagram represents the main processes in Bank Management System. Then they will be broken down into more specific use cases depending on the included processes of the main use case. Each of these use cases explains how the system handles the actions or scenarios requested by the user.


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

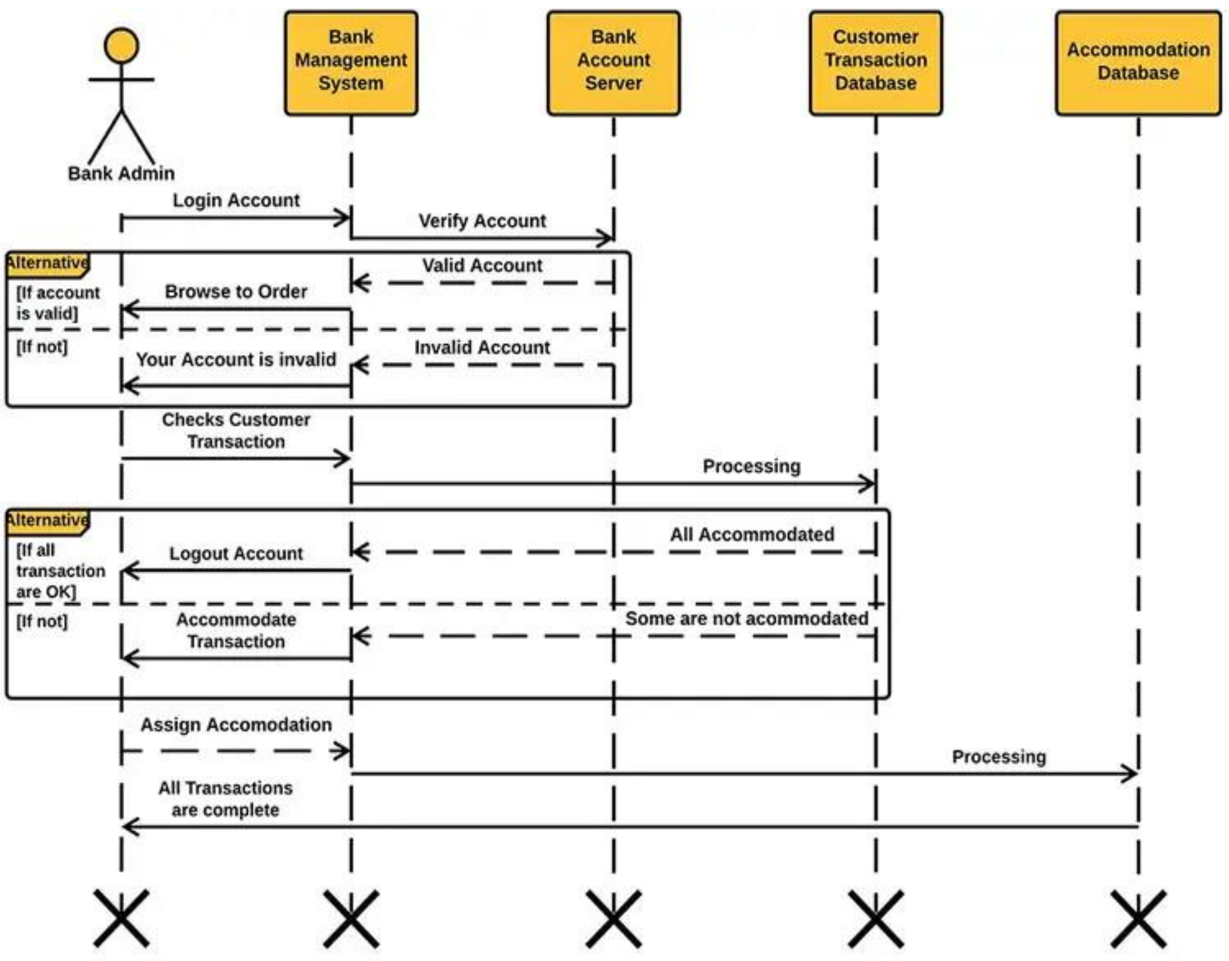
Activity Diagram

Here's the UML activity diagram design of Bank Management System that you can use for your own Final year Project. The UML activity Diagram is used to show the interaction of the user and the system. By creating it, you'll be able to see the flaws of the system and you may avoid it once you apply it to the project development. So it is important to have your diagrams designed first before jumping into its development.



Sequence Diagram

The designed sequence diagram illustrates the series of events that occurs in Bank Management System. In this illustration, the actors are represented by a stick man and the transactions or classes are represented by objects. It will give you clear explanation about the behavior of an Bank Management System in terms of processing the flow of instructions

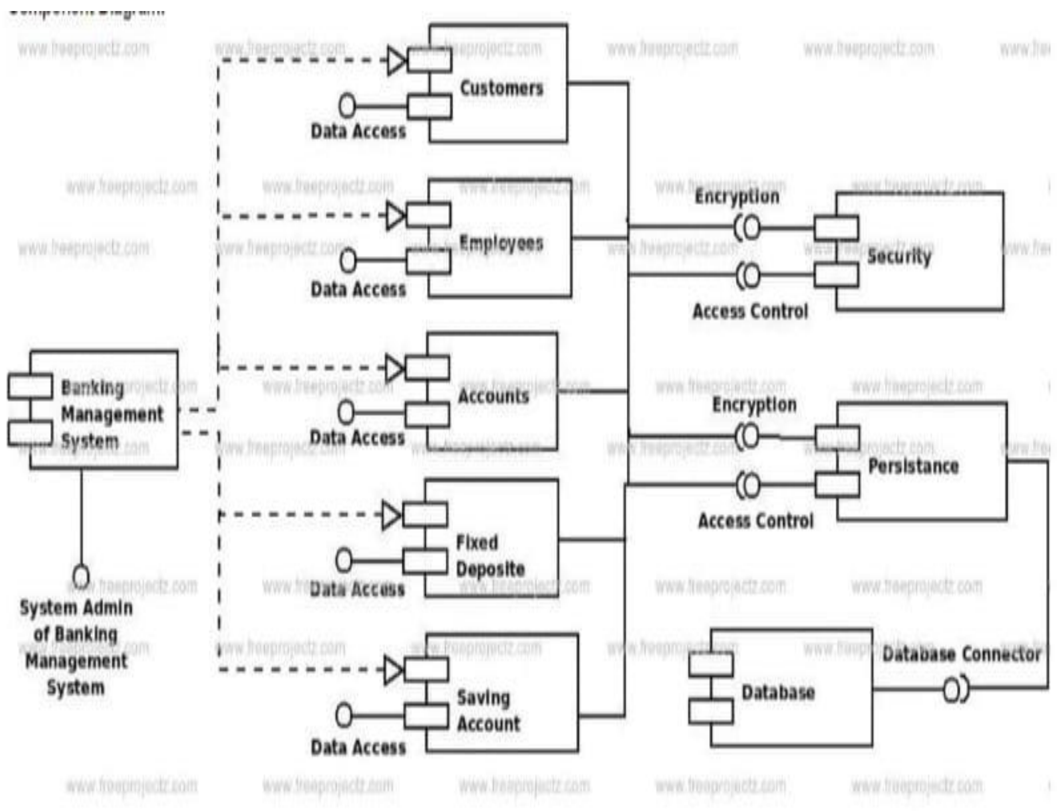


.

Component Diagram

The component diagram of bank management system is used to show how the parts work together to make the bank system operate correctly. A component diagram shows how the software's parts are organized and how they depend on each other. This diagram gives a high-level look at the parts of a system.

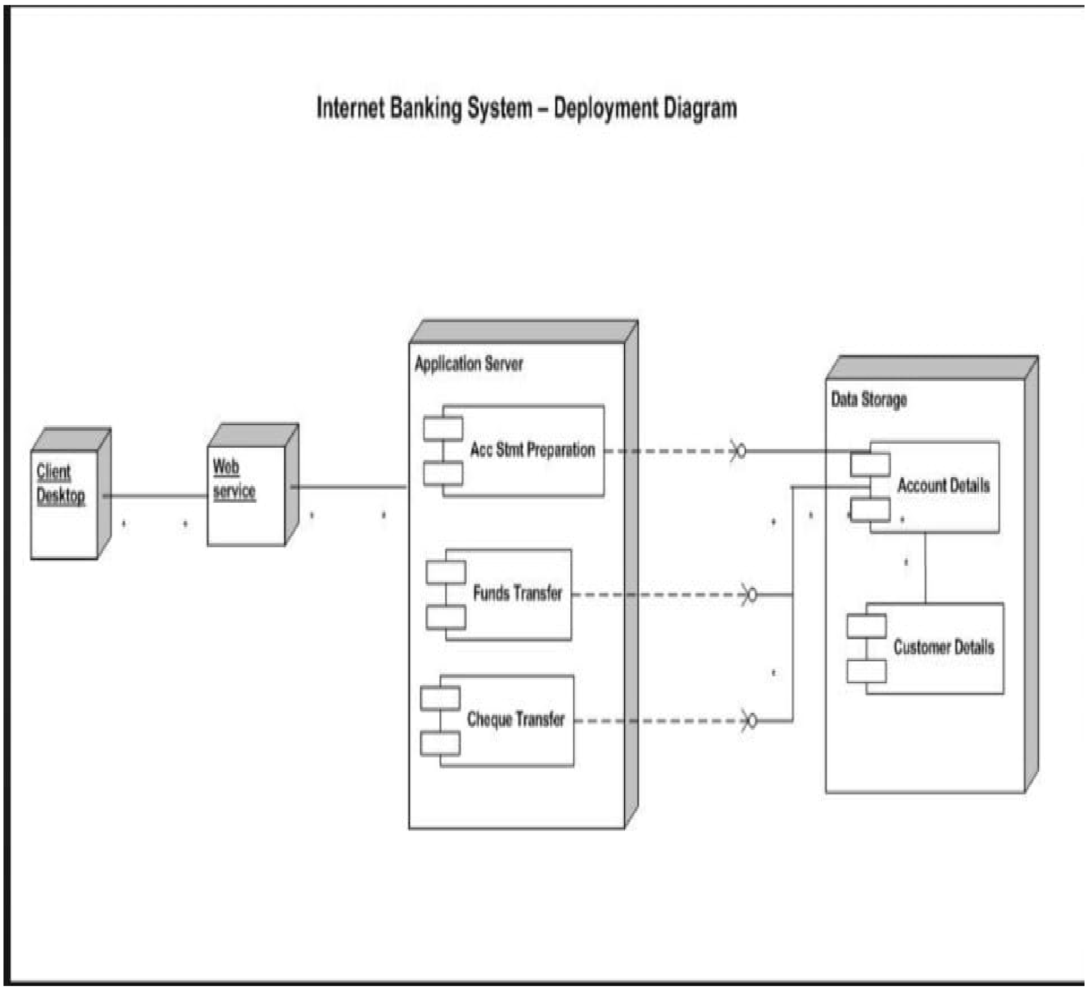
Components of a bank management component diagram can be part of software or hardware. They could be a database, a user interface, or something else that helps the bank management system work.



Deployment Diagram

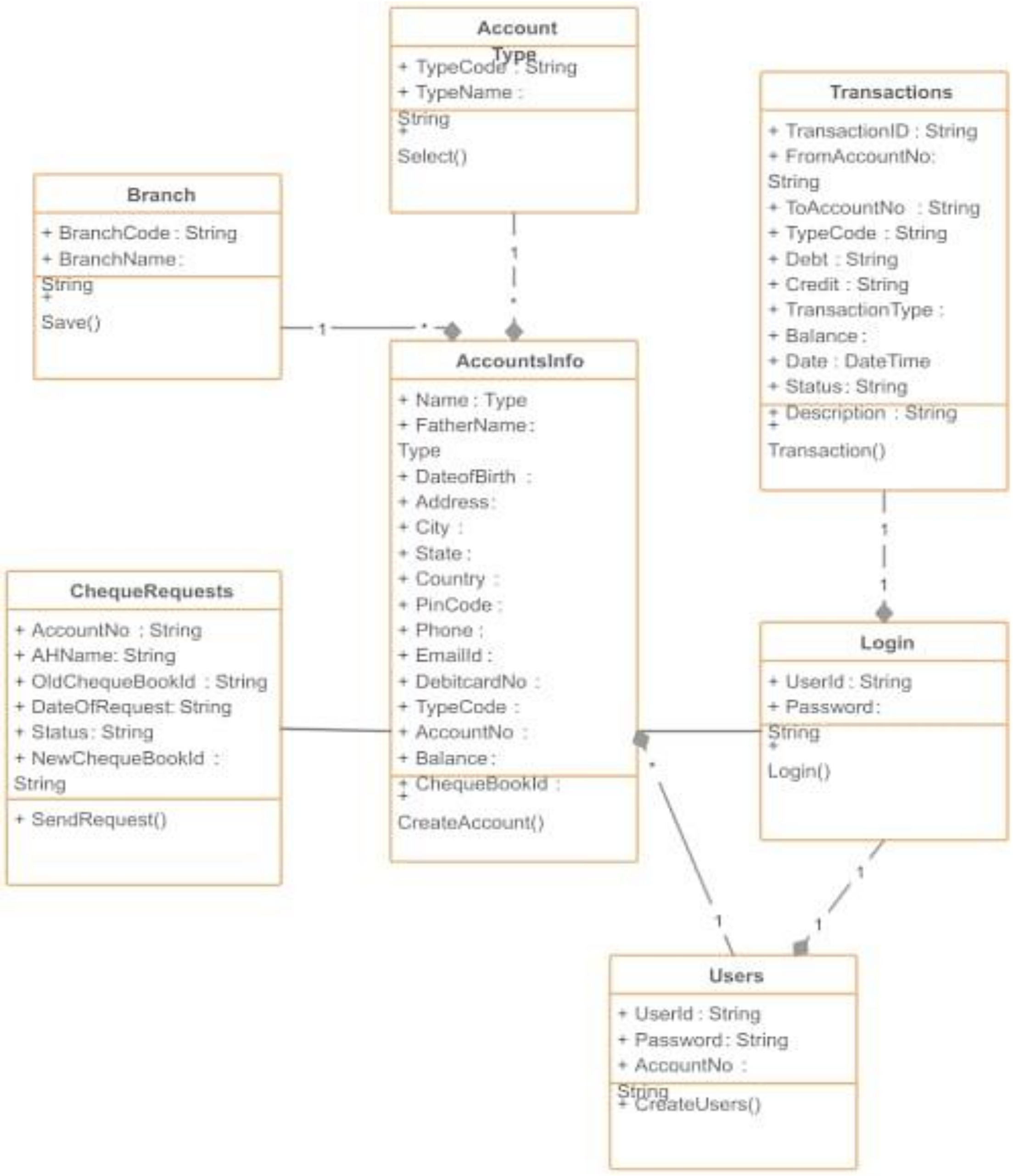
A deployment diagram for banking system layouts the system's tangibles elements that carry out the project execution. It is the projects' execution architecture that reveals the included hardware, software, and the connection between them. This banking system UML deployment diagram is used to define how the system works across included nodes and how are they connected.

The deployment diagram clarifies the communications between links(nodes) present in the banking system. This concept enables the banking system to work according to the design given to it. Deployment diagrams depict the setup of run-time processing nodes and the components that reside on them.



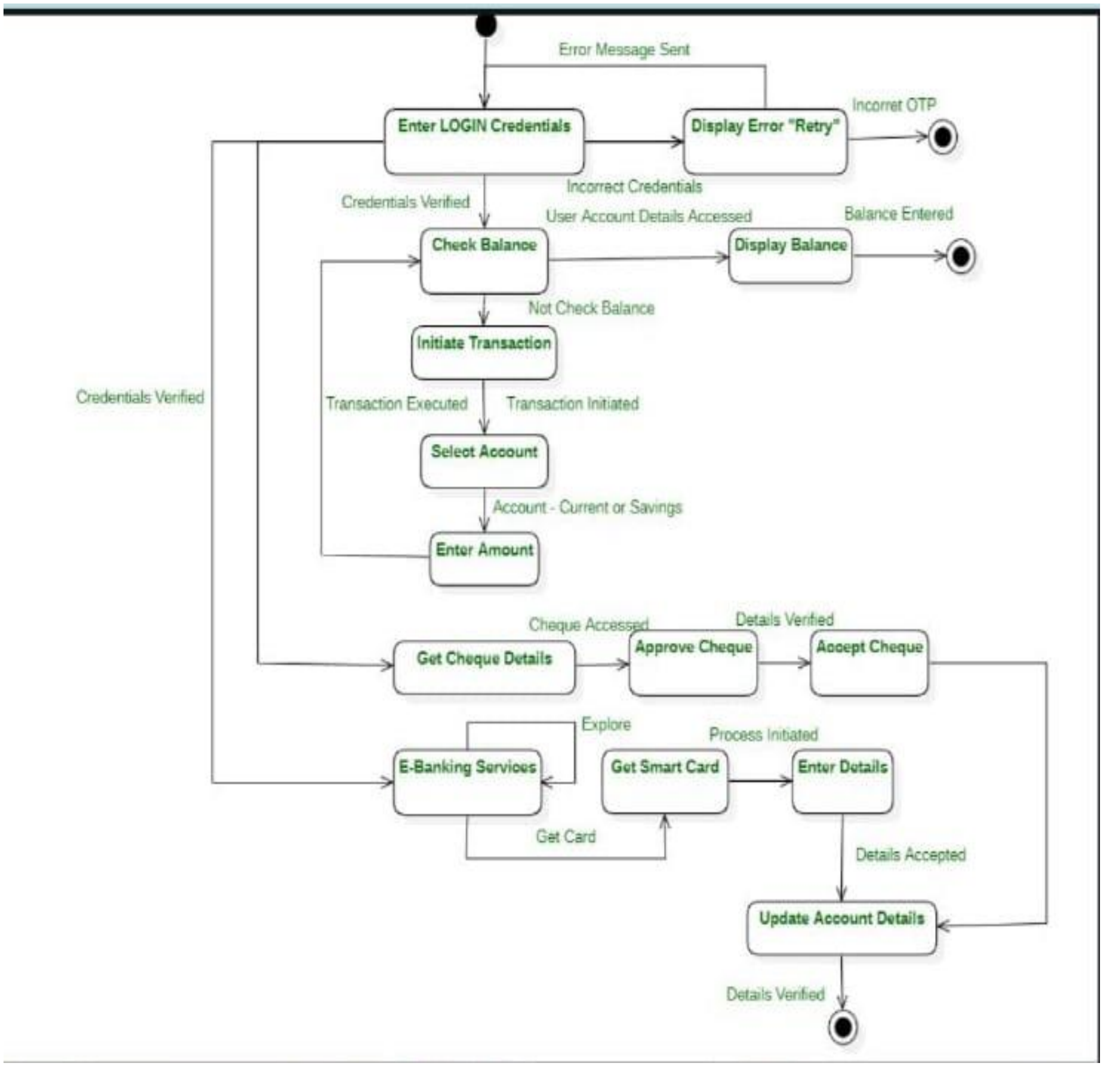
Package Diagram

Package diagrams are structural diagrams used to show the organization and arrangement of various model elements in the form of packages. A package is a grouping of related UML elements, such as diagrams, documents, classes, or even other packages.



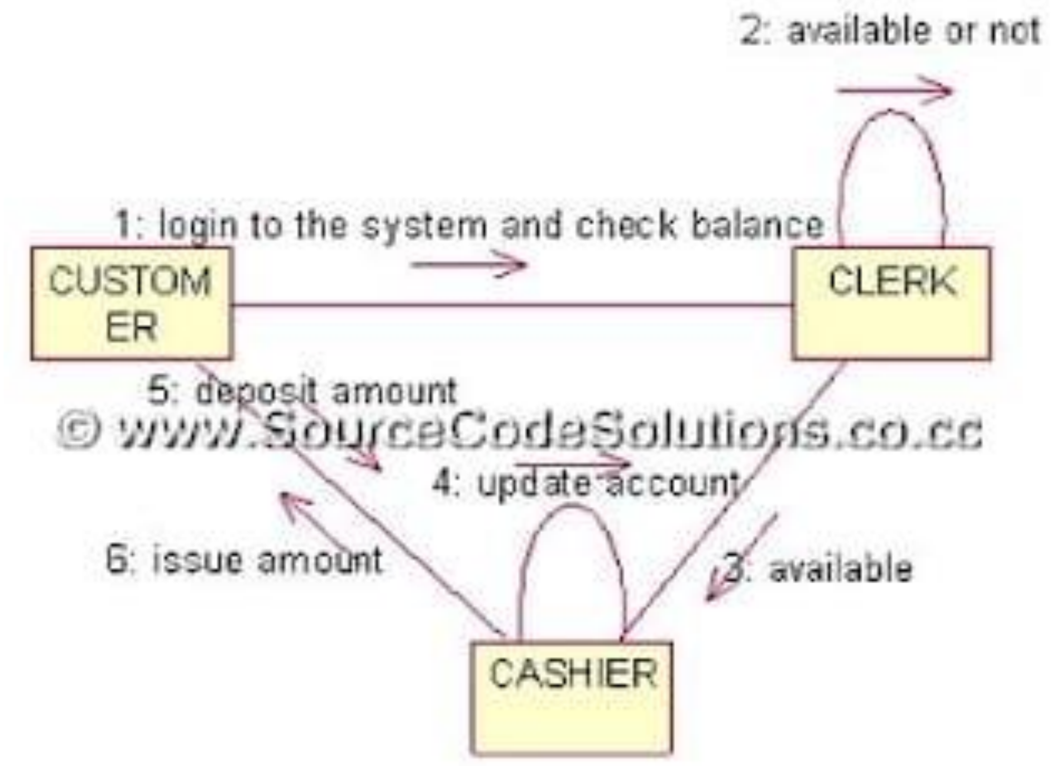
State Chart Diagram

State Diagrams are used to model and present the dynamic nature of a system. State Diagrams consists of different states which represent an activity or an action corresponding to an event. An event causes the transitions from a state to another state in the state diagram. Hence a state diagram is a pictorial representation of the flow of control with respect to either internal or external events.



Collaboration Diagram

The collaboration diagram is used to show the relationship between the objects in a system. Both the sequence and the collaboration diagrams represent the same information but differently. Instead of showing the flow of messages, it depicts the architecture of the object residing in the system as it is based on object-oriented programming. An object consists of several features. Multiple objects present in the system are connected to each other. The collaboration diagram, which is also known as a communication diagram, is used to portray the object's architecture in the system



Source Code

C++ Code of BANK MANAGEMENT SYSTEM

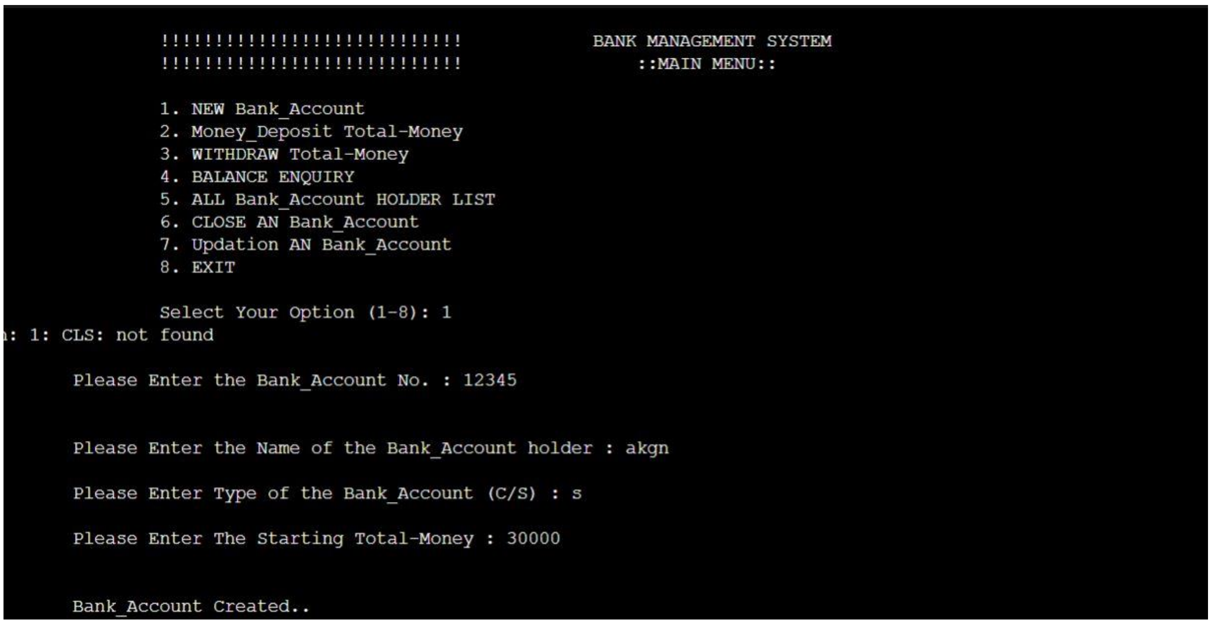
```
#include<iostream>
#include<fstream>
#include<cctype>
#include<iomanip>
using namespace std;
class Bank_Account
{

        int Money_Deposit;
        char type;
        int acno;
        char name[70];
public:

        void report() const;
        int retMoney_Deposit() const;
        void create_Bank_Account();
        void dep(int);
        int  retacno() const;
        void Display_Account() const;
        void Updation();
        char rettype() const;
        void draw(int);
};


void Bank_Account::Updation()
{
        cout<<"\n\tBank_Account No. : "<<acno;
        cout<<"\n\tUpdation Bank_Account Holder Name : ";
        cin.ignore();
        cin.getline(name,50);
        cout<<"\n\tUpdation Type of Bank_Account : ";
        cin>>type;
        type=toupper(type);
        cout<<"\n\tUpdation Balance Total-Money : ";
        cin>>Money_Deposit;
}
void Bank_Account::create_Bank_Account()
{
        system("CLS");
        cout<<"\n\tPlease Enter the Bank_Account No. : ";
        cin>>acno;
        cout<<"\n\n\tPlease Enter the Name of the Bank_Account holder : ";
        cin.ignore();
        cin.getline(name,50);
        cout<<"\n\tPlease Enter Type of the Bank_Account (C/S) : ";
        cin>>type;
        type=toupper(type);
        cout<<"\n\tPlease Enter The Starting Total-Money : ";
        cin>>Money_Deposit;
        cout<<"\n\n\tBank_Account Created..";
}

void Bank_Account::Display_Account() const
{
        cout<<"\n\tBank_Account No. : "<<acno;
        cout<<"\n\tBank_Account Holder Name : ";
        cout<<name;
        cout<<"\n\tType of Bank_Account : "<<type;
        cout<<"\n\tBalance Total-Money : "<<Money_Deposit;
}
int Bank_Account::retacno() const
{
        return acno;
}



char Bank_Account::rettype() const
{
        return type;
}
void Bank_Account::report() const
{
        cout<<acno<<setw(10)<<" "<<name<<setw(10)<<"
"<<type<<setw(6)<<Money_Deposit<<endl;
}
void Bank_Account::dep(int x)
```

```
{
	Money_Deposit+=x;
}
void Bank_Account::draw(int x)
{
	Money_Deposit-=x;
}
int Bank_Account::retMoney_Deposit() const
{
	return Money_Deposit;
}


void write_Bank_Account();
void display_sp(int);
void display_all();

void delete_Bank_Account(int);
void Money_Deposit_withdraw(int, int);
void Updation_Bank_Account(int);
int main()
{
	char ch;
	int num;
	do
	{
	system("CLS");
	cout<<"\n\n\t\t!!!!!!!!!!!!!!!!!!!!!!!!!!!!!";

	cout<<"\t\tBANK MANAGEMENT SYSTEM";
	cout<<"\n\t\t!!!!!!!!!!!!!!!!!!!!!!!!!!!!!";

		cout<<"\t\t    ::MAIN MENU::\n";
		cout<<"\n\t\t1. NEW Bank_Account";
		cout<<"\n\t\t2. Money_Deposit Total-Money";
		cout<<"\n\t\t3. WITHDRAW Total-Money";
		cout<<"\n\t\t4. BALANCE ENQUIRY";
		cout<<"\n\t\t5. ALL Bank_Account HOLDER LIST";
		cout<<"\n\t\t6. CLOSE AN Bank_Account";
		cout<<"\n\t\t7. Updation AN Bank_Account";
		cout<<"\n\t\t8. EXIT";
		cout<<"\n\n\t\tSelect Your Option (1-8): ";
		cin>>ch;
```

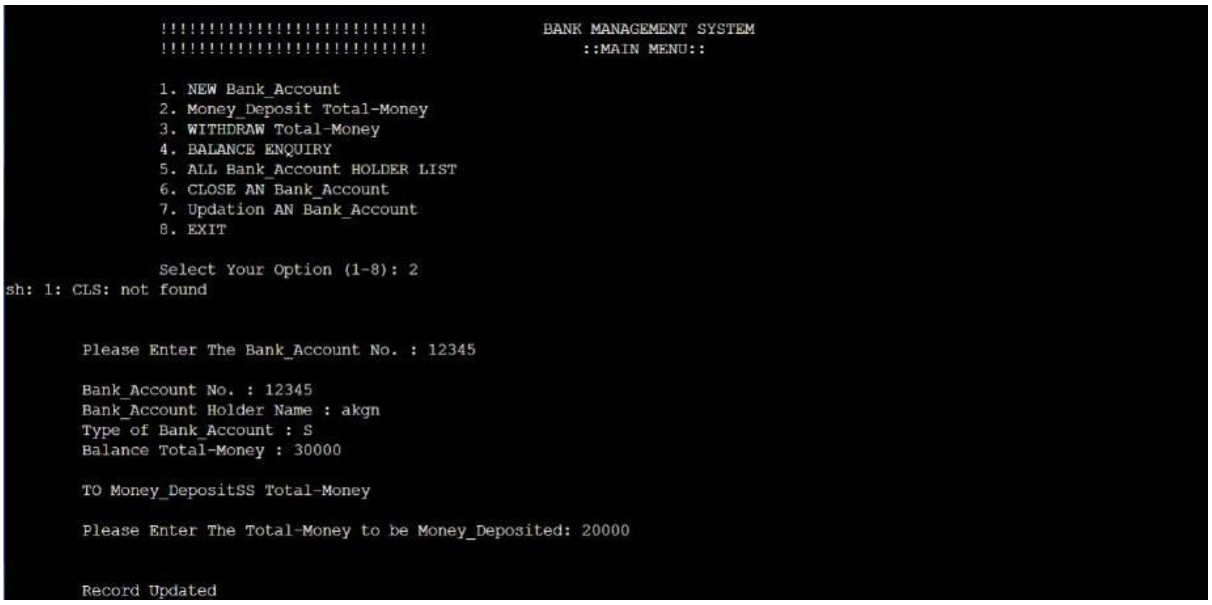
```
                switch(ch)
                {
                case '1':
                        write_Bank_Account();
                        break;
                case '2':
                        system("CLS");
                        cout<<"\n\n\tPlease Enter The Bank_Account No. : ";
cin>>num;
                        Money_Deposit_withdraw(num, 1);
                        break;
                case '3':
                        system("CLS");
                        cout<<"\n\n\tPlease Enter The Bank_Account No. : ";
cin>>num;
                        Money_Deposit_withdraw(num, 2);
                        break;
                case '4':
                        system("CLS");
                        cout<<"\n\n\tPlease Enter The Bank_Account No. : ";
cin>>num;
                        display_sp(num);
                        break;
                case '5':
                        display_all();
                        break;
                case '6':
                        system("CLS");
                        cout<<"\n\n\tPlease Enter The Bank_Account No. : ";
cin>>num;
                        delete_Bank_Account(num);
                        break;
                 case '7':
                        system("CLS");
                        cout<<"\n\n\tPlease Enter The Bank_Account No. : ";
cin>>num;
                        Updation_Bank_Account(num);
                        break;
                 case '8':
                        system("CLS");
                        break;
                 default :cout<<"\a";
                }
                cin.ignore();
                cin.get();
    }while(ch!='8');
        return 0;
}




void write_Bank_Account()
{
        Bank_Account ac;
        ofstream outFile;
        outFile.open("Bank_Account.dat",ios::binary|ios::app);
        ac.create_Bank_Account();
        outFile.write(reinterpret_cast<char *> (&ac),
sizeof(Bank_Account));
        outFile.close();
}
void delete_Bank_Account(int n)
{
        Bank_Account ac;
        ifstream inFile;
        ofstream outFile;
        inFile.open("Bank_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_Account)))
        {
                if(ac.retacno()!=n)
                {
                        outFile.write(reinterpret_cast<char *> (&ac),
sizeof(Bank_Account));
                }
        }
    inFile.close();
        outFile.close();
        remove("Bank_Account.dat");
        rename("Temp.dat","Bank_Account.dat");
        cout<<"\n\nRecord Deleted ..";
}

void display_sp(int n)
{
        Bank_Account ac;
        bool flag=false;
        ifstream inFile;
        inFile.open("Bank_Account.dat",ios::binary);
        if(!inFile)
        {
                cout<<"File could not be open !! Press any Key...";
                return;
        }
        cout<<"\n\tBALANCE DETAILS\n";
        while(inFile.read(reinterpret_cast<char *> (&ac),
sizeof(Bank_Account)))
        {
                if(ac.retacno()==n)
                {
                        ac.Display_Account();
                        flag=true;
                }
        }
    inFile.close();
        if(flag==false)
                cout<<"\n\n\tBank_Account number does not exist";
}


void display_all()
{
        system("CLS");
        Bank_Account ac;
        ifstream inFile;
        inFile.open("Bank_Account.dat",ios::binary);
        if(!inFile)
        {
                cout<<"File could not be open !! Press any Key...";
                return;
        }
        cout<<"\n\n\t\tBank_Account 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_Account)))
        {
                ac.report();
        }
        inFile.close();
}
void Updation_Bank_Account(int n)
{
        bool found=false;
        Bank_Account ac;
        fstream File;
    File.open("Bank_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_Account));
                if(ac.retacno()==n)
                {
                        ac.Display_Account();
                        cout<<"\n\n\tPlease Enter The New Details of
Bank_Account"<<endl;
                        ac.Updation();
                        int pos=(-1)*static_cast<int>(sizeof(Bank_Account));
                        File.seekp(pos,ios::cur); //fncallat1353
                    File.write(reinterpret_cast<char *> (&ac),
sizeof(Bank_Account));
                    cout<<"\n\n\tRecord Updated";
                    found=true;
                  }
        }
        File.close();
        if(found==false)
                cout<<"\n\n\tRecord Not Found ";
```

```
}

void Money_Deposit_withdraw(int n, int option)
{
        int amt;
        bool found=false;
        Bank_Account ac;
        fstream File;
    File.open("Bank_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_Account));
                if(ac.retacno()==n)
                {
                        ac.Display_Account();
                        if(option==1)
                        {
                                cout<<"\n\n\tTO Money_DepositSS Total-
Money";
                                cout<<"\n\n\tPlease Enter The Total-Money to
be Money_Deposited: ";
                                cin>>amt;
                                ac.dep(amt);
                        }
                    if(option==2)
                        {
                                cout<<"\n\n\tTO WITHDRAW Total-Money";
                                cout<<"\n\n\tPlease Enter The Total-Money to
be withdraw: ";
                                cin>>amt;
                                int bal=ac.retMoney_Deposit()-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(Bank_Account));
                        cout<<"\n\n\tRecord Updated";
                        found=true;
                }
          }
    File.close();
        if(found==false)
                cout<<"\n\n\tRecord Not Found ";
}
```

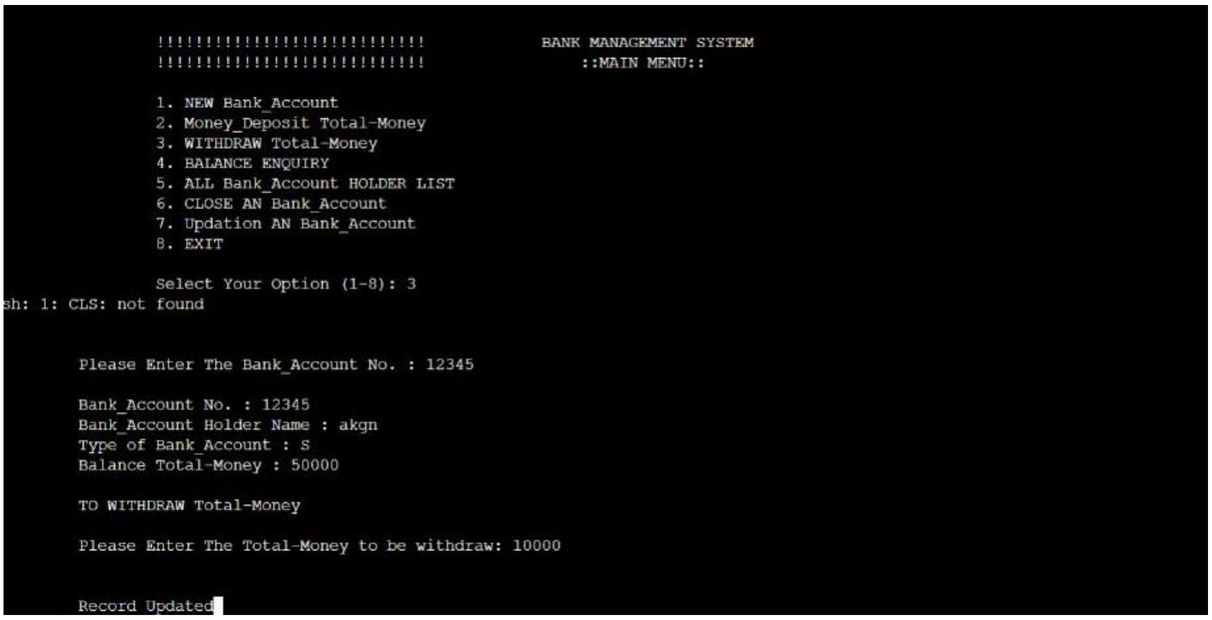
OUTPUT:



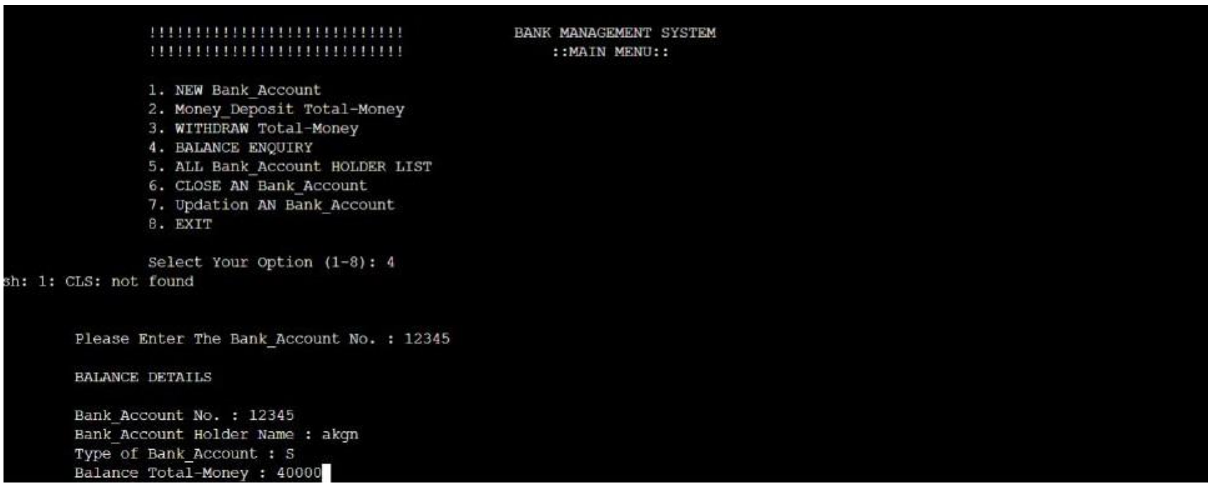
```
          !!!!!!!!!!!!!!!!!!!!!!!!!!!!           BANK MANAGEMENT SYSTEM
          !!!!!!!!!!!!!!!!!!!!!!!!!!!!               ::MAIN MENU::

          1. NEW Bank_Account
          2. Money_Deposit Total-Money
          3. WITHDRAW Total-Money
          4. BALANCE ENQUIRY
          5. ALL Bank_Account HOLDER LIST
          6. CLOSE AN Bank_Account
          7. Updation AN Bank_Account
          8. EXIT

          Select Your Option (1-8): 1
: 1: CLS: not found

    Please Enter the Bank_Account No. : 12345


    Please Enter the Name of the Bank_Account holder : akgn

    Please Enter Type of the Bank_Account (C/S) : s

    Please Enter The Starting Total-Money : 30000


    Bank_Account Created..
```



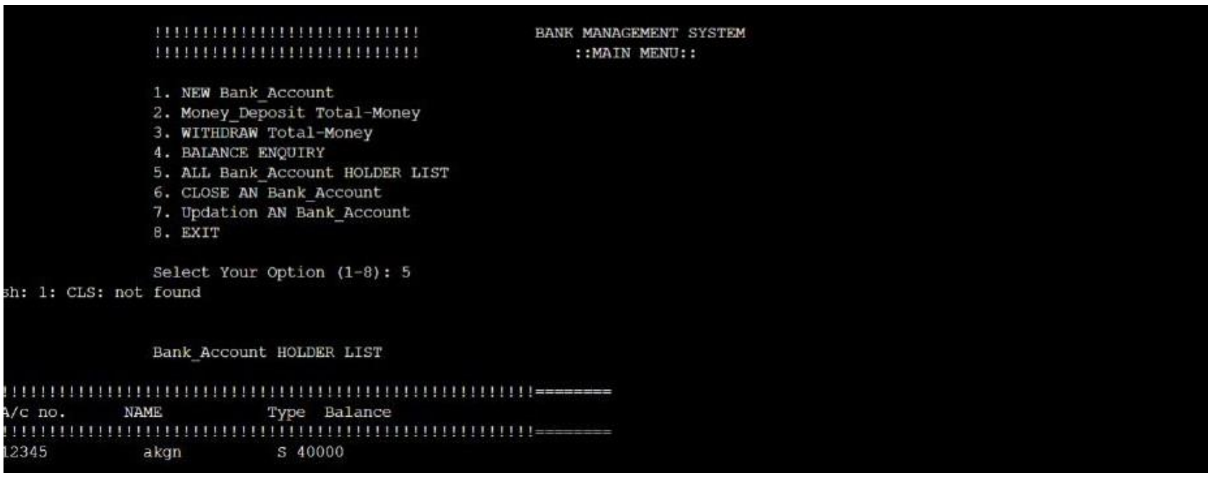
```
            !!!!!!!!!!!!!!!!!!!!!!!!!!!!           BANK MANAGEMENT SYSTEM
            !!!!!!!!!!!!!!!!!!!!!!!!!!!!               ::MAIN MENU::

            1. NEW Bank_Account
            2. Money_Deposit Total-Money
            3. WITHDRAW Total-Money
            4. BALANCE ENQUIRY
            5. ALL Bank_Account HOLDER LIST
            6. CLOSE AN Bank_Account
            7. Updation AN Bank_Account
            8. EXIT

            Select Your Option (1-8): 2
sh: 1: CLS: not found


      Please Enter The Bank_Account No. : 12345

      Bank_Account No. : 12345
      Bank_Account Holder Name : akgn
      Type of Bank_Account : S
      Balance Total-Money : 30000

      TO Money_DepositSS Total-Money

      Please Enter The Total-Money to be Money_Deposited: 20000


      Record Updated
```



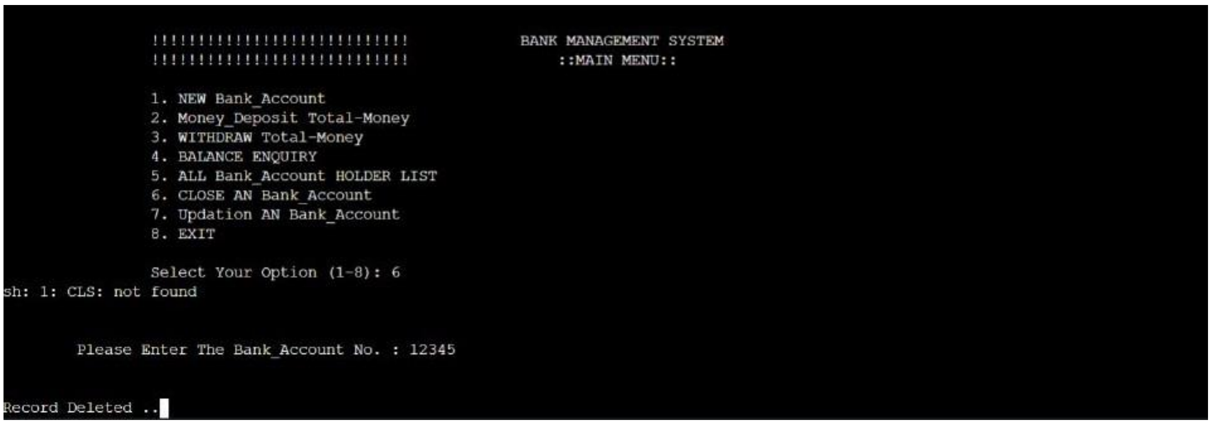
```
            !!!!!!!!!!!!!!!!!!!!!!!!!!!!           BANK MANAGEMENT SYSTEM
            !!!!!!!!!!!!!!!!!!!!!!!!!!!!               ::MAIN MENU::

            1. NEW Bank_Account
            2. Money_Deposit Total-Money
            3. WITHDRAW Total-Money
            4. BALANCE ENQUIRY
            5. ALL Bank_Account HOLDER LIST
            6. CLOSE AN Bank_Account
            7. Updation AN Bank_Account
            8. EXIT

            Select Your Option (1-8): 3
sh: 1: CLS: not found


      Please Enter The Bank_Account No. : 12345

      Bank_Account No. : 12345
      Bank_Account Holder Name : akgn
      Type of Bank_Account : S
      Balance Total-Money : 50000

      TO WITHDRAW Total-Money

      Please Enter The Total-Money to be withdraw: 10000


      Record Updated
```



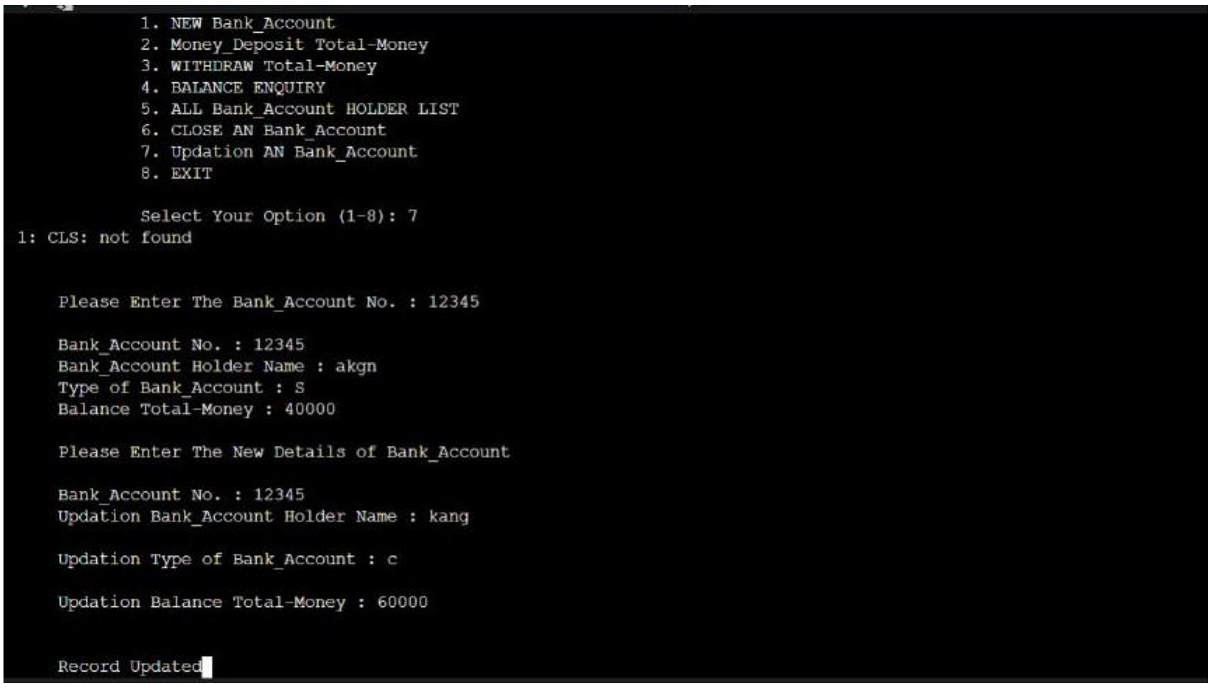
```
            !!!!!!!!!!!!!!!!!!!!!!!!!!!!           BANK MANAGEMENT SYSTEM
            !!!!!!!!!!!!!!!!!!!!!!!!!!!!               ::MAIN MENU::

            1. NEW Bank_Account
            2. Money_Deposit Total-Money
            3. WITHDRAW Total-Money
            4. BALANCE ENQUIRY
            5. ALL Bank_Account HOLDER LIST
            6. CLOSE AN Bank_Account
            7. Updation AN Bank_Account
            8. EXIT

            Select Your Option (1-8): 4
sh: 1: CLS: not found


      Please Enter The Bank_Account No. : 12345

      BALANCE DETAILS

      Bank_Account No. : 12345
      Bank_Account Holder Name : akgn
      Type of Bank_Account : S
      Balance Total-Money : 40000
```



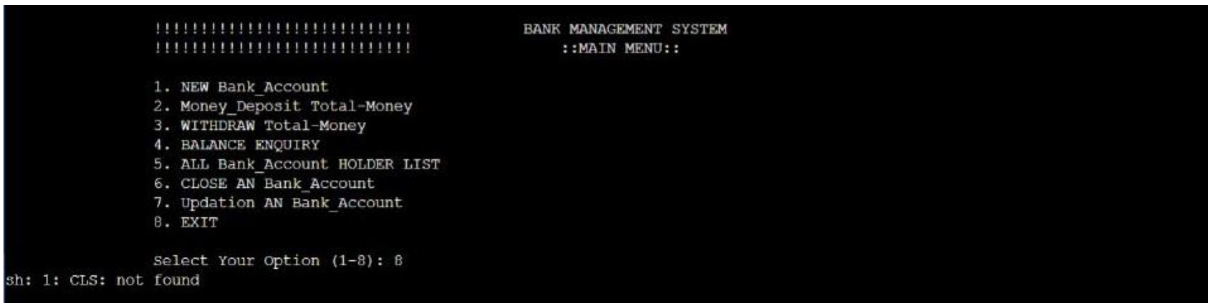
```
              !!!!!!!!!!!!!!!!!!!!!!!!!!!!           BANK MANAGEMENT SYSTEM
              !!!!!!!!!!!!!!!!!!!!!!!!!!!!               ::MAIN MENU::

              1. NEW Bank_Account
              2. Money_Deposit Total-Money
              3. WITHDRAW Total-Money
              4. BALANCE ENQUIRY
              5. ALL Bank_Account HOLDER LIST
              6. CLOSE AN Bank_Account
              7. Updation AN Bank_Account
              8. EXIT

              Select Your Option (1-8): 5
sh: 1: CLS: not found


              Bank_Account HOLDER LIST

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!========
A/c no.      NAME          Type  Balance
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!========
12345          akgn          S 40000
```



```
              !!!!!!!!!!!!!!!!!!!!!!!!!!!!           BANK MANAGEMENT SYSTEM
              !!!!!!!!!!!!!!!!!!!!!!!!!!!!               ::MAIN MENU::

              1. NEW Bank_Account
              2. Money_Deposit Total-Money
              3. WITHDRAW Total-Money
              4. BALANCE ENQUIRY
              5. ALL Bank_Account HOLDER LIST
              6. CLOSE AN Bank_Account
              7. Updation AN Bank_Account
              8. EXIT

              Select Your Option (1-8): 6
sh: 1: CLS: not found


       Please Enter The Bank_Account No. : 12345


Record Deleted ..
```



```
             1. NEW Bank_Account
             2. Money_Deposit Total-Money
             3. WITHDRAW Total-Money
             4. BALANCE ENQUIRY
             5. ALL Bank_Account HOLDER LIST
             6. CLOSE AN Bank_Account
             7. Updation AN Bank_Account
             8. EXIT

             Select Your Option (1-8): 7
 1: CLS: not found


     Please Enter The Bank_Account No. : 12345

     Bank_Account No. : 12345
     Bank_Account Holder Name : akgn
     Type of Bank_Account : S
     Balance Total-Money : 40000

     Please Enter The New Details of Bank_Account

     Bank_Account No. : 12345
     Updation Bank_Account Holder Name : kang

     Updation Type of Bank_Account : c

     Updation Balance Total-Money : 60000


     Record Updated
```



```
            !!!!!!!!!!!!!!!!!!!!!!!!!!!!            BANK MANAGEMENT SYSTEM
            !!!!!!!!!!!!!!!!!!!!!!!!!!!!                ::MAIN MENU::

            1. NEW Bank_Account
            2. Money_Deposit Total-Money
            3. WITHDRAW Total-Money
            4. BALANCE ENQUIRY
            5. ALL Bank_Account HOLDER LIST
            6. CLOSE AN Bank_Account
            7. Updation AN Bank_Account
            8. EXIT

            Select Your Option (1-8): 8
sh: 1: CLS: not found
```

Conclusion:

Thus, the various UML diagrams for Banking management system was drawn and the code was generated successfully. This software has been developed designed to reduce the time taken to handle the sales activity. It is designed to replace an existing manual record system for reducing time taken for calculations and for storing data. This system has been developed with oops concepts. The system is strong to handle daily operations where the database is cleared over certain time. This system will reduce manual work, calculations and will also provide periodic reports any time.