

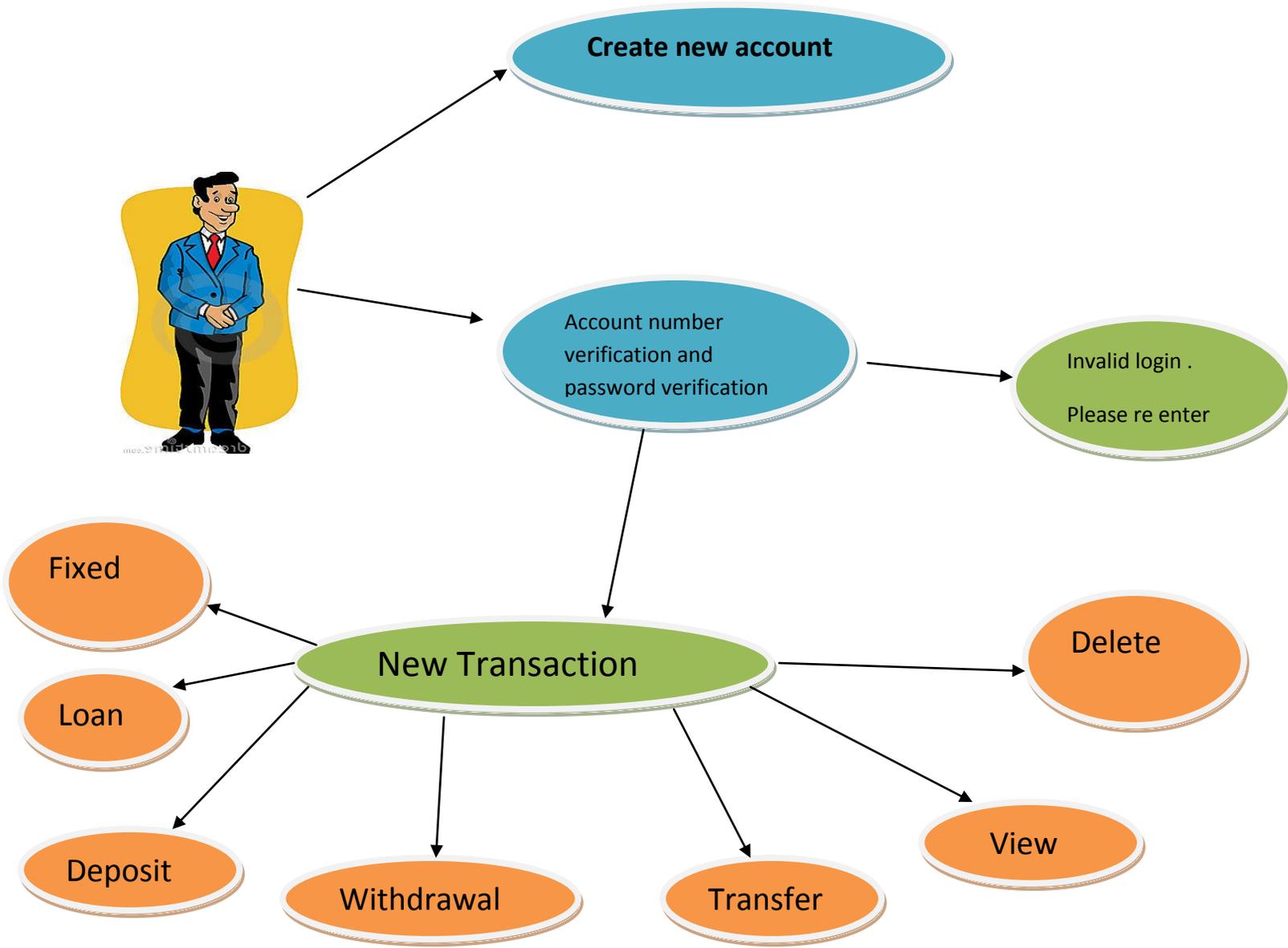
BANKING PROJECT

Team Members:

- Tejesh Raut
- Siddharth Birla
- Amit Kumar Bhagat

Introduction:

Our project provides an interface for the user to create their account in the bank and make transactions, modifications, taking loan, creating fixed deposits, and other useful functions of a bank. It also automatically updates the account just after debit, credit or any other transaction is done from that account. It provides a Graphical User Interface (GUI), like buttons are used to do various tasks performed by our program. It basically helps the user to perform all the tasks related to banking in a quick and efficient way.



Function Implementation:

`void create_account() :`

This function is used to create new account for the customer. The customer is required to fill a form asking for his personal details like name, date of birth, address, and asks him to enter a PIN for his account. The account will be created after the correct information is filled in the form.

```
void create_account()
{
    cout<<"Please fill the following form:"<<endl;//storing information filled by customer in a temporary
object c
    cout<<"Enter your name(Upto maximum 50 characters): "<<endl;
    cin.clear();
    cin.sync();
    cin.getline(Name,50);
    TryAgain19:
    cout<<"Enter date of birth(dd/mm/yyyy format): "<<endl;
```

```
        cin.clear();

        cin.sync();

        cin.getline(DOB,11);

if((DOB[2]!='/') || (DOB[5]!='/'))

    {

        cout<<"You have not entered date of birth in the given format"<<endl;

        cout<<"Please re-enter it in the given format "<<endl;

system("pause");

        system("cls");

        goto TryAgain19;

    }

cout<<"Enter complete residential address(Upto maximum 70 characters in a single line): ";

        cin.clear();

        cin.sync();

        cin.getline(Address,70);

        TryAgain3://label
```

```
        cout<<"Enter a new PIN(4 digits) for your account"<<endl;
        cin>>PIN;
if((PIN>9999) || (PIN<1000))//Invalid input
    {
        system("cls");
        cout<<"Invalid input"<<endl;
        goto TryAgain3;//Re-enter a new PIN
    }
    system("cls");
cout<<"Re-Enter PIN:"<<endl;
    int p;
    cin>>p;
if(p!=PIN)//Invalid re-entering
    {
        system("cls");
        cout<<"The two PINs entered by you don't match"<<endl;
```

```
    cout<<"Please try again "<<endl;
    goto TryAgain3;//Try entering new PIN again
}
```

TryAgain12:

```
    cout<<"Enter Initial balance in the account: "<<endl;
    cin>>Balance;
```

```
if(Balance<1000)
```

```
{
```

```
    cout<<"Any Account must have a minimum balance of 1000"<<endl;
```

```
        system("pause");
```

```
        system("cls");
```

```
        goto TryAgain12;
```

```
}
```

```
Account_num=count+1000;
```

```
system("cls");
```

```
cout<<"Account created successfully "<<endl;
```

```
}
```

```
void deposit_account() :
```

It is used to deposit some amount of money in the customer's account.

```
void deposit_account()
```

```
{
```

```
    float amount;
```

```
    system("cls");
```

```
    cout<<"Enter the amount to be deposited"<<endl;
```

```
    cin>>amount;
```

```
    Balance=Balance+amount;
```

```
    }//end of deposit_account()
```

```
void withdrawal_account() :
```

This is used by the customer to withdraw some amount of money from his account.

```
void withdrawal_account()
```

```
{
```

```
float amount;
```

```
TryAgain13:
```

```
system("cls");
```

```
cout<<"Enter the amount to withdraw"<<endl;
```

```
cin>>amount;
```

```
if((Balance-amount)<1000)
```

```
{
```

```
    cout<<"Any Account must have a minimum balance of 1000"<<endl;
```

```
    system("pause");
```

```
    system("cls");
```

```
    goto TryAgain13;
```

```
}
```

```
Balance=Balance-amount;
```

```
}//end of withdrawal_account()
```

```
void view_account() :
```

To view the details of an account.

```
void view_account()
{
    cout<<"Name\t:\t"<<Name<<endl;
    cout<<"Date of birth\t:\t"<<DOB<<endl;
    cout<<"Residential address\t:\t"<<Address<<endl;
    cout<<"Current balance\t:\t"<<Balance<<endl<<endl<<endl;
} //end of view_account()
```

`void modify_account() :`

To modify any details of the account.

```
void modify_account()
{
```

TryAgain8 :

```
system("cls");
```

To change name:

```
cout<<"Enter your new name(Upto maximum 50 characters): "<<endl;
    cin.clear();
    cin.sync();
cin.getline(Name,50);
return;
```

To change date of birth:

TryAgain20:

```
    cout<<"Enter new date of birth(dd/mm/yyyy format): "<<endl;
    cin.clear();
    cin.sync();
    cin.getline(DOB,11);
if((DOB[2]!='/') || (DOB[5]!='/'))
    {
        cout<<"You have not entered date of birth in the given format"<<endl;
        cout<<"Please re-enter it in the given format "<<endl;
system("pause");
```

```
system("cls");  
goto TryAgain20;  
}  
return;
```

To modify residential address:

```
cout<<"Enter new complete residential address(Upto maximum 70 characters in a single line): ";  
    cin.clear();  
    cin.sync();  
    cin.getline(Address,70);
```

```
return;
```

To modify PIN:

TryAgain6 :

```
int pin;  
cout<<"Enter current PIN : "<<endl;  
cin>>pin;
```

```
if(PIN!=pin)
{
    cout<<"You have entered incorrect PIN"<<endl;
    cout<<"Please try again"<<endl;
    system("pause");
    system("cls");
    goto TryAgain6;
}
```

```
TryAgain7://label
```

```
    system("cls");
    cout<<"Enter a new PIN(4 digits) for your account"<<endl;
    cin>>PIN;
```

```
if((PIN>9999) || (PIN<1000))//Invalid input
```

```
{
    system("cls");
    cout<<"Invalid input"<<endl;
```

```
        goto TryAgain7;//Re-enter a new PIN
    }
    system("cls");
    cout<<"Re-Enter PIN:"<<endl;
    int p;
    cin>>p;
    if(p!=PIN)//Invalid re-entering
    {
        system("cls");
        cout<<"The two PINs entered by you don't match"<<endl;
        cout<<"Please try again "<<endl;
        goto TryAgain7;//Try entering new PIN again
    }
    return;
}
```

```
void delete_account()
```

To delete an existing account

```
void delete_account()
{
Text t(175,81," Name ");
Name[0]='-';Name[1]='\0';
    DOB[0]='-';DOB[1]='\0';
    Address[0]='-';Address[1]='\0';
PIN =0;
    Balance=0;
} //end of delete_account()
```

void loan_account() :

To take a loan from the bank

```
void loan_account()
{
    int year;float principal,amount;
```

```
cout<<"Enter the amount to be taken as loan "<<endl;
    cin>>principal;
cout<<"Enter the time in years to take the loan "<<endl;
    cout<<"(You can take only for integral number of years"<<endl;
    cout<<"Our interest rate is only 10% per annum )"<<endl;
    cin>>year;
amount=principal*(pow(1.1,year));
system("cls");
Text t(175,55," You will have to repay ");
    Text t3(75,115,amount);
    Text t4(150,115," after ");
    Text t5(225,115,year);
    Text t6(300,115," years ");
    Text t7(175,225," Confirm ? ");
Balance=Balance+principal;
system("cls");
```

```
cout<<"Loan received successfully "<<endl;
```

```
return;
```

```
void fixeddeposit_account() :
```

To create a fixed deposit

```
void fixeddeposit_account()
```

```
{
```

```
    int year;float principal,amount;
```

TryAgain18:

```
    cout<<"Enter the amount to be given as fixed deposit "<<endl;
```

```
    cin>>principal;
```

```
    cout<<"Enter the time in years to keep fixed deposit "<<endl;
```

```
    cout<<"(You can take only for integral number of years"<<endl;
```

```
    cout<<"You will get profit of 8% per annum )"<<endl;
```

```
    cin>>year;
```

```
    amount=principal*(pow(1.08,year));
```

```
system("cls");
Text t(175,55," You will get ");
    Text t3(75,115,amount);
    Text t4(150,115," after ");
    Text t5(225,115,year);
    Text t6(300,115," years ");
    Text t7(175,225," Confirm ? ");
if((Balance-principal)<1000)
    {
        cout<<"Any Account must have a minimum balance of 1000"<<endl;
        system("pause");
        system("cls");
        goto TryAgain18;
    }
    Balance=Balance-principal;
system("cls");
```

```
cout<<"Fixed deposit created successfully "<<endl;
```

```
return;
```

```
}
```

```
void transfer_acc():
```

To transfer money from one account to another

```
void transfer_acc(long POS)
```

```
{
```

```
    long pos;
```

```
    fp=fopen("customers_data.txt","rb+");
```

```
    if(fp==NULL)//Unable to open the file
```

```
    {
```

```
        cout<<"Could not open database file"<<endl;
```

```
        return;
```

```
    }//end of if
```

```
    fseek(fp,POS,SEEK_SET);
```

```
fread(&c,rec_size,1,fp);//reading record into c
```

```
int amount,acc_no1;
```

```
    TryAgain21:
```

```
        system("cls");
```

```
cout<<"Enter the account number of the account to which money is to be transferred"<<endl;
```

```
    cin>>acc_no1;
```

```
if(acc_no1==(1000+POS/rec_size))
```

```
{
```

```
    cout<<"You cant enter your own account number for transfer "<<endl;
```

```
    cout<<"Please enter correct information "<<endl;
```

```
system("pause");
```

```
    goto TryAgain21;
```

```
}
```

```
fseek ( fp , 0 , SEEK_END );//Set cursor to the end of the file
```

```
pos=ftell(fp);
```

```
count=pos/rec_size;
```

```
if((acc_no1>(count+1000))||(acc_no1<1000))
{
    cout<<"You have entered incorrect account number"<<endl;
    cout<<"Please try again"<<endl;
system("pause");
    goto TryAgain21;
}
```

TryAgain15:

```
cout<<"Enter the amount to be transferred :"<<endl;
cin>>amount;
if((c.Balance-amount)<1000)
{
    cout<<"Any Account must have a minimum balance of 1000"<<endl;
    system("pause");
    system("cls");
    goto TryAgain15;
```

```
    }  
    c.Balance=c.Balance-amount;  
    fseek(fp,POS,SEEK_SET);//taking back to same position for update of record  
        fwrite(&c,rec_size,1,fp);//updating record  
        customer c1;  
    long POS1;//to store position of the record of account to which transfer is being done  
    POS1=(acc_no1-1000)*rec_size;  
    fseek(fp,POS1,SEEK_SET);  
        fread(&c1,rec_size,1,fp);//reading record into c1  
    c1.Balance=c1.Balance+amount;  
    fseek(fp,POS1,SEEK_SET);  
        fwrite(&c1,rec_size,1,fp);  
    system("cls");  
        cout<<"Your transfer was done successfully."<<endl;  
        fclose(fp);  
    }//end of transfer_acc()
```

Data Specification:

The data specification needed in our program are:

Erase (Rectangle) : This is a variable of type rectangle and is used to erase the canvas by covering it with itself.

Clickpos (int) : To store the clicked position on the canvas .

cx (int) : To store the x-coordinate of the clicked position.

cy (int) : To store the y-coordinate of the clicked position.

button (Rectangle) : To denote as a button on the screen.

acc_no (int) : To store the account number of the user.

pin (int) : To store the PIN code as provided by the user.

c (customer) :To refer to a record(a structure) from the file.

POS (long) : To denote position in the current file.

rec_size (sizeof(customer)) : To store size of the record.

fp (FILE*) : To refer as a pointer to the file customers_data.txt

count (int) : To denote the number of accounts in the file.

Data members of the structure customer :

Account_num (int) : Denotes the account number.

PIN (int) : Denotes the PIN of a customer

Balance (float) : To store the balance of the customer

Name[50] (char[]) : To store the name of the customer

DOB[11] (char[]) : To store date of birth in dd/mm/yyyy format

Address[70] (char[]) : To store address of the customer

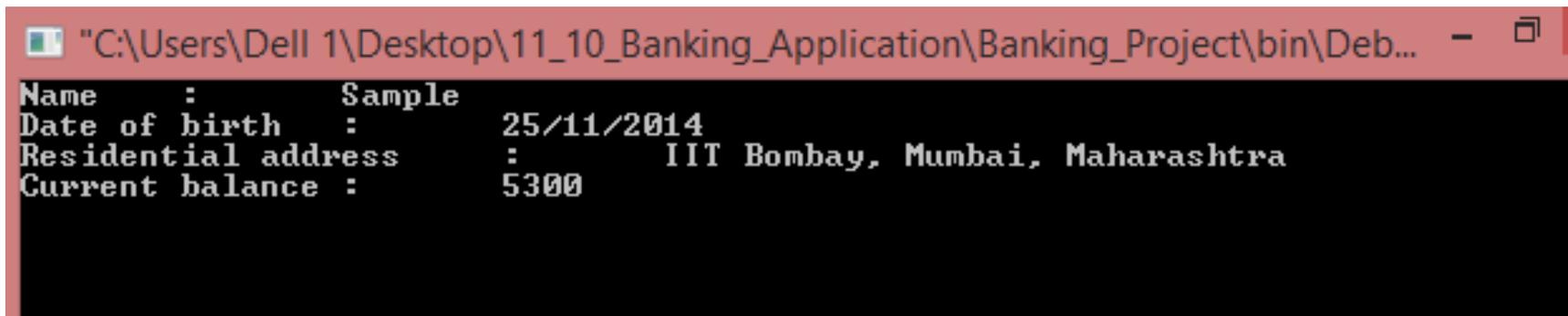
Output :

Our program includes several graphical features to handle the project easily.

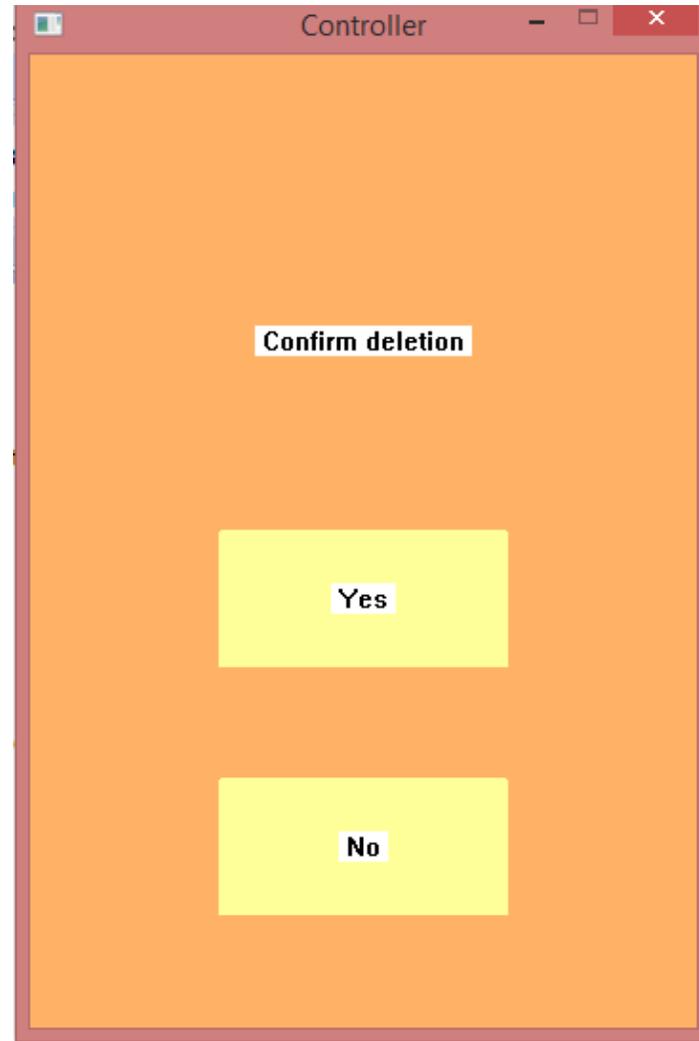
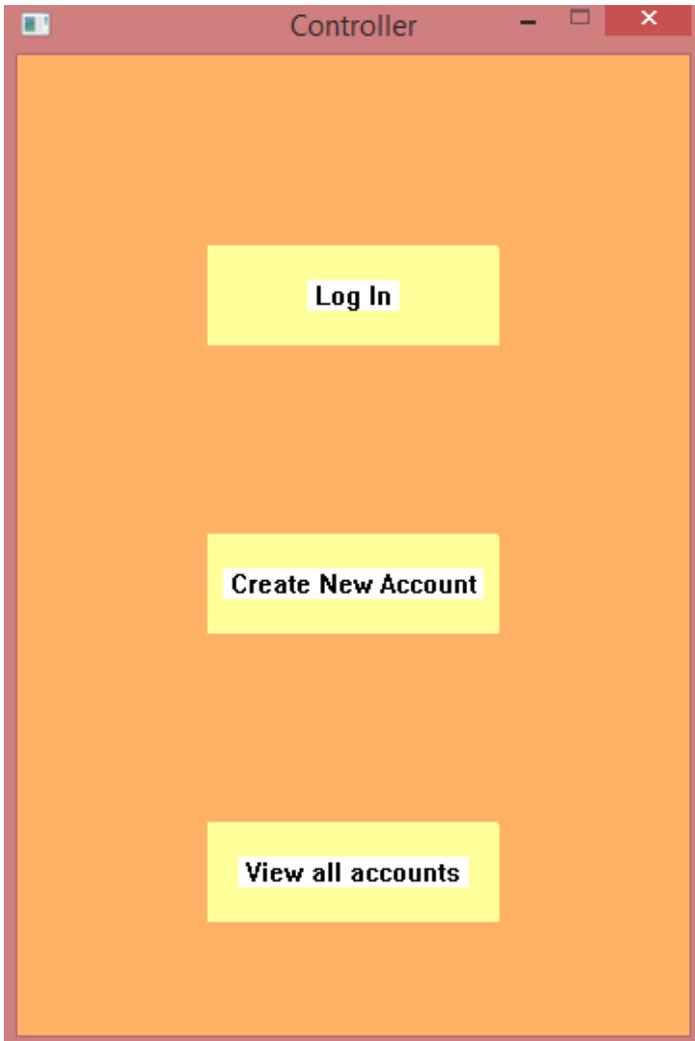
It takes keyboard input from the console screen but as far as choosing a task to be performed is considered, we tried our best to include that part in the canvas window.

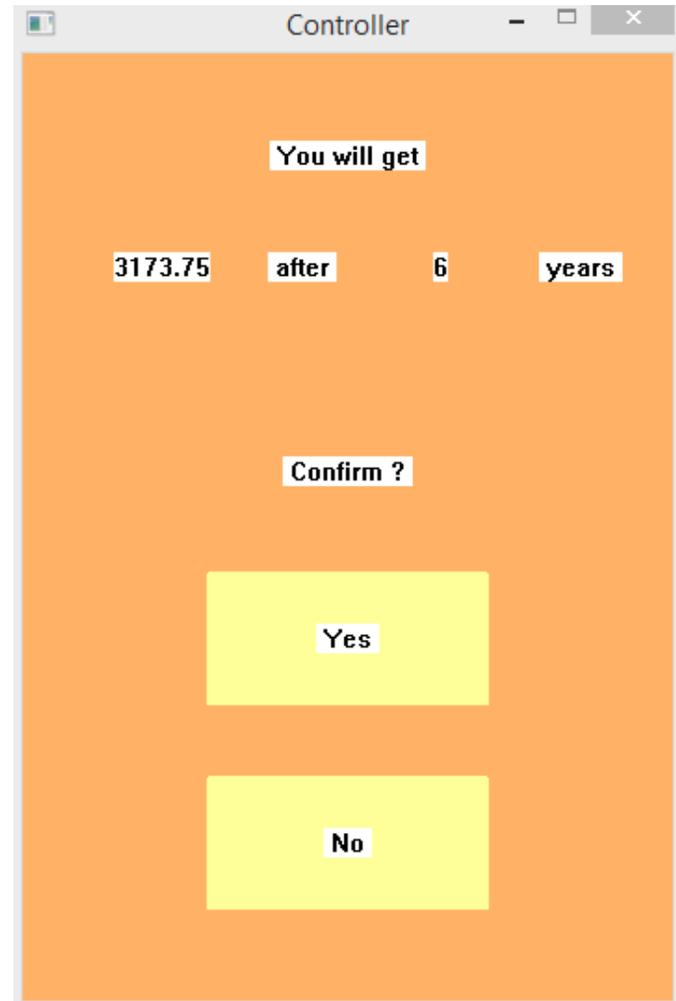
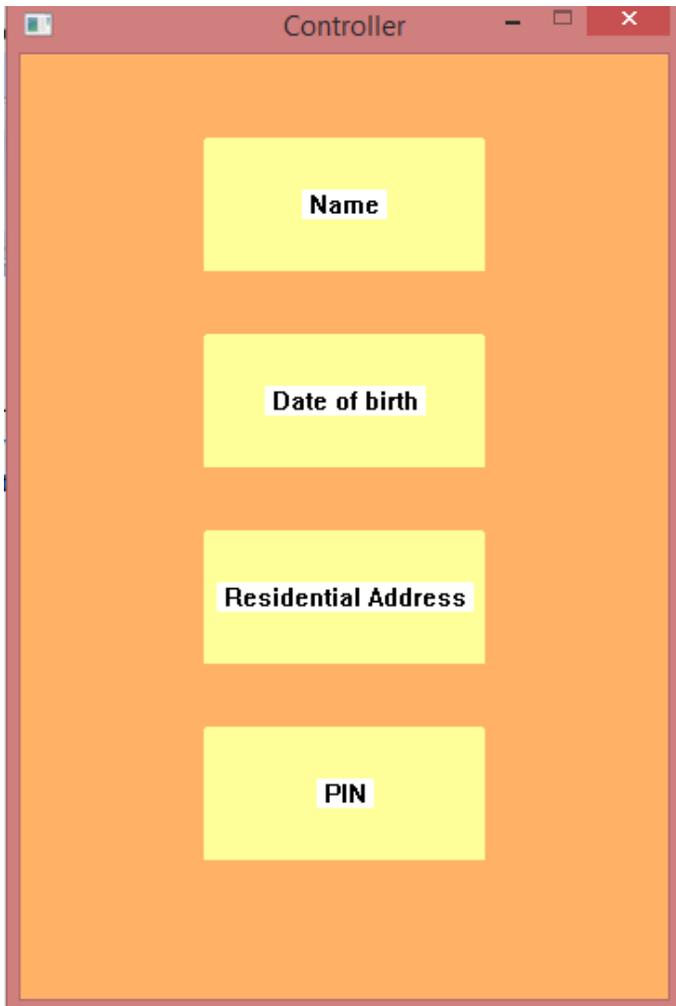
Most of our screenshots of how the program runs are displayed in the user's manual.

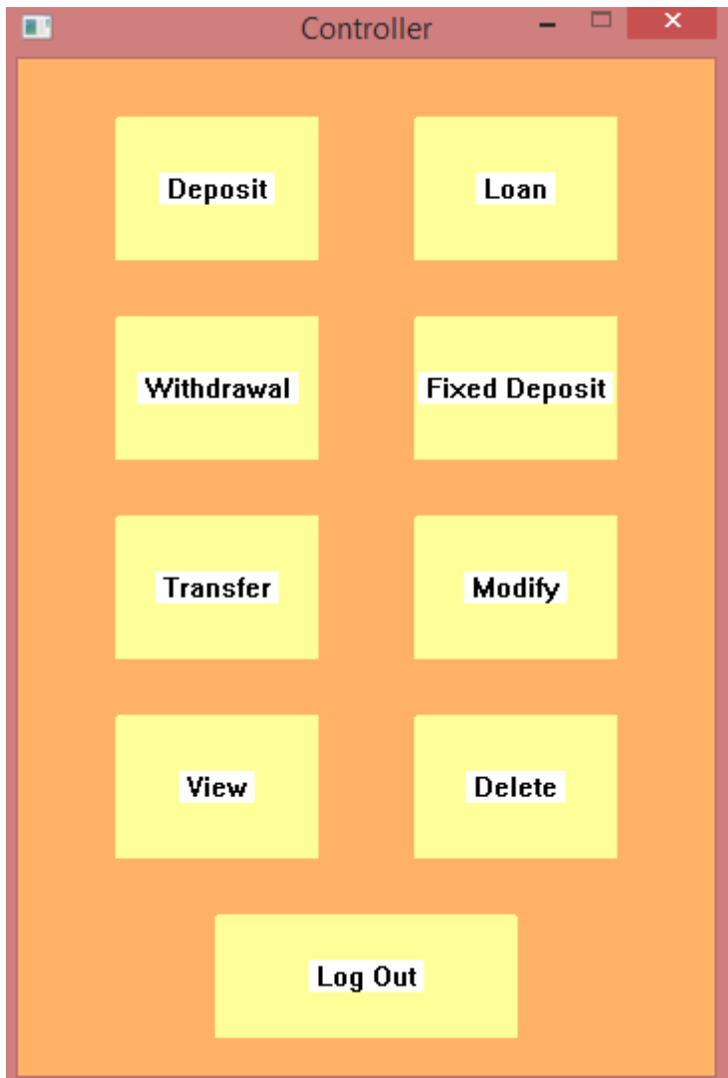
Some of the screenshots are :

A screenshot of a Windows command prompt window. The title bar shows the path "C:\Users\Dell 1\Desktop\11_10_Banking_Application\Banking_Project\bin\Deb...". The command prompt displays the following text:

```
Name      :      Sample
Date of birth  :      25/11/2014
Residential address  :      IIT Bombay, Mumbai, Maharashtra
Current balance :      5300
```







FEATURES:

- The special feature of our program is that it uses graphical interface to take inputs from the user, i.e., the user can perform different tasks by just clicking on the buttons on the canvas screen. It makes the program attractive and easy to use.
- Our program includes some good features like viewing all accounts on a single page, taking loans, creating fixed deposits, and modifying account information.
- It has all the basic features of banking like depositing and withdrawing money, transferring money from one account to another, viewing the account info, and deleting the account.
- The “creating new account” feature helps to create a new account for the customer after he fills the required form and then he is given the account number which he can use to log in his account. After logging in, he can use all the features the program provides.
- In case the user enters any wrong information our program outputs that incorrect information has been entered and asks the user to re-enter the information instead of terminating the complete program.
- Our program clears the console and the canvas screen from time to time wherever needed so that previous output is not repeatedly displayed on the screen and a fresh screen is displayed every time.
- Prompts are used everywhere when an input is taken so that a new user may easily understand what is the next field required by the compiler.
- Our program doesn't allow the customer to reduce his/her account balance below 1000 –the criterion followed by most of the real banks.

REFERENCES:

- www.wikipedia.com for getting what a SRS contains
- www.cplusplus.com for searching new functions which were not taught in the course such as goto,system("cls"),system("pause")
- www.stackoverflow.com for getting help on usage of various functions in c++.
- Book-Introduction to Problem Solving and Programming through C++ by Abhiram Ranade.
- Lecture videos on operations on files and using simplecpp for developing buttons from the link- <http://www.cse.iitb.ac.in/~cs101/lecture-slides.html>
- http://www.rapidtables.com/web/color/RGB_Color.htm for getting the RGB colour codes for different colours