**Review – III**



**Project title**

**The Ultimate Banking Management System.**

Name: Hari Krishna Shah

VIT ID: 21BCS0167

Team Members: Hari Krishna Shah only.

Contact Number: +977 9844523112

Email: [harikrishna.shah@vitstudent2021.ac.in](mailto:harikrishna.shah@vitstudent2021.ac.in)

**Abstract**

Banking sector is regarded as the backbone of the economy. A more digital and technologized banking system significantly helps the nation’s economy to grow and flourish. In this project, I will be digitalizing the traditional banking system in order to boost the output of the bank. To do so, I will be taking object-oriented approach. My program will also be data centric which will try to maximize the protection of data of the bank and the clients of the bank because such data are extremely confidential and can cause huge damage if they are corrupted or manipulated by the wrong person. Hence, the concept of classes and object, that are part of object-oriented programming, which provides us data protection and abstraction along with encapsulation will be beneficial. We will also be using the concept of polymorphism to create multiple functions and operators of same name but different functions by the means of function overloading and operator overloading. This will make our program hassle free.

I aim to achieve below features in my program.

* Bank account creation.
* Deposit and withdrawal feature
* Overdraft feature
* Loan disbursement feature
* Advanced loan tracking and analysis
* Client risk profiling for the applied loan
* Collateral management and analysis
* Collateral evaluation features
* Decision support system
* Data Protection

As you can see there are a lot of features mentioned above. My program will have all those features and make banking system more efficient and effective. It will aid the bank in all the possible ways to carry out their functions.

These all features make my program the ultimate solution for the banking management system.

**Introduction**

The program will be made using the concepts of objected oriented programming.

**List of Modules**

The following Modules are used in the program:

Module 1:

1. **Client Details Class**

It stores all the details of the client like, client name, client’s national id, location, and date of birth.

It contains functions such as get and display to take input and show output regarding client details.

1. **Saving Account**

Saving Account inherits the client details class and further stores all the details about client’s saving account like saving account number, saving account security pin and balance.

It contains member functions such as get() to take input, display() to show output, withdraw() to facilitate money withdrawal, deposit() to facilitate money deposit and search() to find the searched bank account according to the master key.

1. **Current Account Class**

Current Account inherits the client details class and further stores all the details about client’s current account like current account number, current account security pin and balance.

It contains member functions such as get() to take input, display() to show output, withdraw() to facilitate money withdrawal, deposit() to facilitate money deposit and search() to find the searched bank account according to the master key.

1. **Asset Class**

Asset class has been used in this program to stores the details about the assets owned by the client. The worth of this asset will be used to calculate the optimal amount of loan that can be granted to him.

It contains member functions such as get() to take input and display() to show output.

1. **Bank Class**

Bank class inherits saving account class, current account class and asset class. This inheritance is an example of multi-path inheritance. Further, it contains master\_key data member which will be used to access the client’s details. It also contains other data members like total\_balance, net\_balance, loan amount and account type.

Account type specifies if the client has saving account or current account or both with the bank.

This class contains inherited member functions along with it’s own get() function to get input, display() to show output, calculate\_optimal\_loan() to find the optimal amount of loan that can be granted to a client and loan() to grant loan to a client.

**Implementation of the project**

#include <iostream>

**using** **namespace** std**;**

*//Coded by Hari Shah*

#define max\_account 100

#define min\_balance 1000

**class** client\_details**{**

**protected:**

**char** name**[**100**];**

*// 1 for citizenship, 2 for driving license, 3 for passport\_id;*

**int** national\_id\_type**;**

**union** national\_id**{**

**long** **long** **int** citizenship**;**

**long** **long** **int** driving\_license**;**

**long** **long** **int** passport\_id**;**

**}**id**;**

**struct** date\_of\_birth**{**

**int** day**;**

**int** month**;**

**int** year**;**

**}**dob**;**

**struct** address**{**

**char** city**[**100**];**

**char** country**[**100**];**

**}**location**;**

**public:**

**void** get\_details**(){**

cout**<<"Enter the client details below."<<**endl**;**

cout**<<"Name: ";**

cin**>>**name**;**

cout**<<"Choose a national id type."<<**endl**;**

cout**<<"Enter 1 for citizenship, 2 for driving license or 3 for passport: ";**

cin**>>**national\_id\_type**;**

cout**<<"Enter your national id: ";**

cin**>>**id**.**citizenship**;**

cout**<<"Enter your birth details in integer below."<<**endl**;**

cout**<<"Day of Birth: ";**

cin**>>**dob**.**day**;**

cout**<<"Month of Birth: ";**

cin**>>**dob**.**month**;**

cout**<<"Year of Birth: ";**

cin**>>**dob**.**year**;**

cout**<<"Enter your address details below."<<**endl**;**

cout**<<"Enter your city: ";**

cin**>>**location**.**city**;**

cout**<<"Enter your country: ";**

cin**>>**location**.**country**;**

**}**

**void** display**(){**

cout**<<"The client details are given below: "<<**endl**;**

cout**<<"Name: "<<**name**<<**endl**;**

**if(**national\_id\_type**==**1**){**

cout**<<"Citizenship number: "<<**id**.**citizenship**<<**endl**;**

**}**

**else** **if(**national\_id\_type**==**2**){**

cout**<<"Driving License number: "<<**id**.**driving\_license**<<**endl**;**

**}**

**else** **if(**national\_id\_type**==**3**){**

cout**<<"Passport number: "<<**id**.**passport\_id**<<**endl**;**

**}**

cout**<<"Date of birth is: "<<**dob**.**year**<<"/"<<**dob**.**month**<<"/"<<**dob**.**day**<<**endl**;**

cout**<<"Address is: "<<**location**.**city**<<", "<<**location**.**country**<<**endl**;**

**}**

**};**

*/\**

*...- / -.-- -... .. .-. / .-.. -... .... / .--. ..- ...- --. . -. .-.-.- .-.. -... .... . ..-. / -. -.-- .--- -. .-.. ..-. / ..- -. . ...- / ..-. ..- -. ..- .-.-.-*

*\*/*

**class** saving\_account**:public** **virtual** client\_details**{**

**protected:**

**int** acc\_no**;**

**float** balance**;**

**int** pin**;**

**public:**

saving\_account**(){**

balance **=** 0**;**

**}**

**void** get**(){**

cout**<<"Enter saving account number: ";**

cin**>>**acc\_no**;**

cout**<<"Enter a PIN for security: ";**

cin**>>**pin**;**

cout**<<"The saving account has been created successfully created.";**

**}**

**void** display**(){**

cout**<<"\nThe saving account details is given below: "<<**endl**;**

cout**<<"Account number: "<<**acc\_no**<<**endl**;**

cout**<<"Balance Amount: "<<**balance**<<**endl**;**

**}**

**int** search**(int** search\_acc**,** **int** pin\_search**){**

**if(**search\_acc **==** acc\_no **&&** pin\_search **==** pin**){**

**return** 1**;**

**}**

**return** 0**;**

**}**

**void** withdraw**(){**

**float** withdraw\_req**,** after\_balance**;**

cout**<<"Enter the amount to withdraw: ";**

cin**>>**withdraw\_req**;**

after\_balance **=** balance **-** withdraw\_req**;**

**if(**after\_balance **>**min\_balance**){**

balance **=** after\_balance**;**

cout**<<"Withdrawl Sucessful."<<**endl**;**

**}**

**else{**

cout**<<"You don't have sufficient balance."<<**endl**;**

**}**

**}**

**void** deposit**(){**

**float** deposit\_req**;**

cout**<<"Enter the amount to deposit: ";**

cin**>>**deposit\_req**;**

balance **+=** deposit\_req**;**

cout**<<"Amount Deposited Sucessfully."<<**endl**;**

**}**

**int** return\_saving\_pin**(){**

**return** pin**;**

**}**

**};**

**class** current\_account**:public** **virtual** client\_details**{**

**protected:**

**int** acc\_no**;**

**float** balance**;**

**int** pin**;**

**public:**

current\_account**(){**

balance **=** 0**;**

**}**

**void** get**(){**

cout**<<"Enter current account number: ";**

cin**>>**acc\_no**;**

cout**<<"Enter a PIN for security: ";**

cin**>>**pin**;**

cout**<<"The current account has been created successfully created.";**

**}**

**void** display**(){**

cout**<<"The current account details is given below: "<<**endl**;**

cout**<<"Account number: "<<**acc\_no**<<**endl**;**

cout**<<"Balance Amount: "<<**balance**<<**endl**;**

**}**

**int** search**(int** search\_acc**,** **int** pin\_search**){**

**if(**search\_acc **==** acc\_no **&&** pin\_search **==** pin**){**

**return** 1**;**

**}**

**return** 0**;**

**}**

**void** withdraw**(){**

**float** withdraw\_req**,** after\_balance**;**

cout**<<"Enter the amount to withdraw: ";**

cin**>>**withdraw\_req**;**

after\_balance **=** balance **-** withdraw\_req**;**

**if(**after\_balance **>**min\_balance**){**

balance **=** after\_balance**;**

cout**<<"Withdrawl Sucessful."<<**endl**;**

**}**

**else{**

cout**<<"You don't have sufficient balance."<<**endl**;**

**}**

**}**

**void** deposit**(){**

**float** deposit\_req**;**

cout**<<"Enter the amount to deposit: ";**

cin**>>**deposit\_req**;**

balance **+=** deposit\_req**;**

cout**<<"Amount Deposited Sucessfully."<<**endl**;**

**}**

**int** return\_current\_pin**(){**

**return** pin**;**

**}**

**};**

**class** asset**:** **public** **virtual** client\_details**{**

**protected:**

**char** asset\_name**[**100**];**

**float** worth**;**

**public:**

asset**(){**

worth **=** 0**;**

**}**

**void** get**(){**

cout**<<"Enter the name of the asset: ";**

cin**>>**asset\_name**;**

cout**<<"Enter the worth of the asset: ";**

cin**>>**worth**;**

**}**

**void** display**(){**

cout**<<"The details of the asset owned by the client is given below."<<**endl**;**

cout**<<"Name of the asset: "<<**asset\_name**<<**endl**;**

cout**<<"Worth of the asset: "<<**worth**<<**endl**;**

**}**

**};**

**class** Bank**:** **public** saving\_account**,** **public** current\_account**,** **public** asset**{**

**protected:**

*// 1 for saving account, 2 for current account, 3 for both saving and current account*

**int** account\_type**;**

**float** loan\_amount**;**

**float** total\_balance**;**

**float** net\_balance**;**

**int** master\_pin**;**

**public:**

Bank**(){**

loan\_amount **=** 0**;**

total\_balance **=** 0**;**

net\_balance **=** 0**;**

**}**

**void** get**(){**

cout**<<"Enter your master key for the bank security: ";**

cin**>>**master\_pin**;**

cout**<<"Please choose an option from the below menu.\n \**

**Enter 1 to create a saving account\n \**

**Enter 2 to create a current account\n \**

**Enter 3 to create both saving and current account\n"<<**endl**;**

cout**<<"Enter your option here: ";**

cin**>>**account\_type**;**

**if(**account\_type **==** 1**){**

saving\_account**::**get**();**

**}**

**else** **if(**account\_type **==** 2**){**

current\_account**::**get**();**

**}**

**else** **if** **(**account\_type **==** 3**){**

saving\_account**::**get**();**

cout**<<**endl**<<**endl**;**

current\_account**::**get**();**

**}**

**else{**

cout**<<"Please enter a valid option."<<**endl**;**

**}**

**}**

**void** display**(){**

**if(**account\_type **==** 1**){**

saving\_account**::**display**();**

**}**

**else** **if(**account\_type **==** 2**){**

current\_account**::**display**();**

**}**

**else** **if(**account\_type **==** 3**){**

saving\_account**::**display**();**

cout**<<**endl**;**

current\_account**::**display**();**

**}**

asset**::**display**();**

total\_balance **=** saving\_account**::**balance **+** current\_account**::**balance**;**

cout**<<"Total Balance: "<<**total\_balance**<<**endl**;**

cout**<<"Loan Amount: "<<**loan\_amount**<<**endl**;**

cout**<<"Net Balance: "<<**total\_balance **-** loan\_amount**<<**endl**;**

**}**

**int** search\_master**(int** master\_pin\_search**){**

**if(**master\_pin\_search **==** master\_pin**){**

**return** 1**;**

**}**

**else{**

**return** 0**;**

**}**

**}**

**float** calculate\_optimal\_loan**(){**

**float** optimal\_loan\_amount**;**

asset**::**get**();**

**float** total\_net\_worth**;**

total\_net\_worth **=** saving\_account**::**balance **+** current\_account**::**balance **+** asset**::**worth**;**

optimal\_loan\_amount **=** 0.5**\***total\_net\_worth**;**

**return** optimal\_loan\_amount**;**

**}**

**void** loan**(){**

**float** loan\_req**;**

**int** credit\_acc**;**

cout**<<"Enter the loan amount you want: "<<**endl**;**

cin**>>**loan\_req**;**

**if(**loan\_req**<**calculate\_optimal\_loan**()){**

cout**<<"Enter 1 to credit the loan to your saving account, 2 to credit it to your current account: ";**

cin**>>**credit\_acc**;**

**if(**credit\_acc **==** 1 **&&(**account\_type **==** 1 **||** account\_type **==** 3**)){**

saving\_account**::**balance **+=** loan\_amount**;**

cout**<<"Loan granted."<<**endl**;**

loan\_amount **=** loan\_req**;**

saving\_account**::**balance **+=** loan\_amount**;**

**}**

**else** **if** **(**credit\_acc **==** 2 **&&(**account\_type **==** 2 **||** account\_type **==** 3**)){**

current\_account**::**balance **+=** loan\_amount**;**

cout**<<"Loan granted."<<**endl**;**

loan\_amount **=** loan\_req**;**

current\_account**::**balance **+=** loan\_amount**;**

**}**

**else{**

cout**<<"Please enter a valid account type."<<**endl**;**

**}**

**}**

**else{**

cout**<<"You can't take that much loan."<<**endl**;**

cout**<<"Loan request rejected."<<**endl**;**

**}**

**}**

**int** return\_account\_type**(){**

**return** account\_type**;**

**}**

**};**

**int** main**(){**

**static** **int** user\_count **=** **-**1**;**

**class** Bank client**[**max\_account**];**

**int** option **=** **-**1**;**

**do{**

cout**<<"\n\t\t\*\*\*Welcome to the Shah Bank Corp\*\*"<<**endl**;**

cout**<<"Please choose an option from menu below: \n \**

**Enter 1 to create a bank account\n \**

**Enter 2 to check your account profile\n \**

**Enter 3 to check your balance\n \**

**Enter 4 to withdraw money\n \**

**Enter 5 to deposit money\n \**

**Enter 6 to calculate your loan credit capactiy\n \**

**Enter 7 to take loan \n \**

**Enter -1 to quit the program"<<**endl**;**

cout**<<"Enter your option here: ";**

cin**>>**option**;**

**switch(**option**){**

**case** **-**1**:{**

cout**<<"\nThe program exited successfully.";**

**break;**

**}**

**case** 1**:{**

user\_count **+=** 1**;**

client**[**user\_count**].**saving\_account**::**client\_details**::**get\_details**();**

client**[**user\_count**].**get**();**

**break;**

**}**

**case** 2 **:{**

**int** master\_search\_pin**,** master\_found**,** i**;**

cout**<<"Enter the master key for your account: ";**

cin**>>**master\_search\_pin**;**

**for(**i **=** 0**;** i**<=**user\_count**;** i**++){**

master\_found **=** client**[**i**].**search\_master**(**master\_search\_pin**);**

**if(**master\_found **==** 1**){**

cout**<<"Account Found."<<**endl**;**

**break;**

**}**

**}**

**if(**master\_found **==** 1**){**

client**[**i**].**saving\_account**::**client\_details**::**display**();**

client**[**i**].**display**();**

**}**

**else** **if(**master\_found **==** 0**){**

cout**<<"The searched account didn't match with any account."<<**endl**;**

**}**

**break;**

**}**

**case** 3 **:{**

**int** master\_search\_pin**,** master\_found**,** i**;**

cout**<<"Enter the master key for your account: ";**

cin**>>**master\_search\_pin**;**

**for(**i **=** 0**;** i**<=**user\_count**;** i**++){**

master\_found **=** client**[**i**].**search\_master**(**master\_search\_pin**);**

**if(**master\_found **==** 1**){**

cout**<<"Account Found."<<**endl**;**

**break;**

**}**

**}**

**if(**master\_found **==** 1**){**

client**[**i**].**display**();**

**}**

**else** **if(**master\_found **==** 0**){**

cout**<<"The searched account didn't match with any account."<<**endl**;**

**}**

**break;**

**}**

**case** 4 **:{**

**int** master\_search\_pin**,** master\_found**,** i**,** withdrawl\_acc\_type**,** saving\_pin**,** current\_pin**;**

cout**<<"Enter the master key for your account: ";**

cin**>>**master\_search\_pin**;**

**for(**i **=** 0**;** i**<=**user\_count**;** i**++){**

master\_found **=** client**[**i**].**search\_master**(**master\_search\_pin**);**

**if(**master\_found **==** 1**){**

cout**<<"Account Found."<<**endl**;**

**break;**

**}**

**}**

**if(**master\_found **==** 1**){**

cout**<<"Enter the account from where you want to withdraw money."<<**endl**;**

cout**<<"Enter 1 to withdraw from saving account or 2 to withdraw from current account: ";**

cin**>>**withdrawl\_acc\_type**;**

**if(**withdrawl\_acc\_type **==** client**[**i**].**return\_account\_type**()** **||** client**[**i**].**return\_account\_type**()==**3**){**

**if(**withdrawl\_acc\_type**==**1**){**

cout**<<"Enter your saving account PIN: ";**

cin**>>**saving\_pin**;**

**if(**saving\_pin **==** client**[**i**].**saving\_account**::**return\_saving\_pin**()){**

client**[**i**].**saving\_account**::**withdraw**();**

**}**

**else{**

cout**<<"Incorrect PIN. Withdraw Request Rejected."<<**endl**;**

**}**

**}**

**else** **if(**withdrawl\_acc\_type**==**2**){**

cout**<<"Enter your current account PIN: ";**

cin**>>**current\_pin**;**

**if(**current\_pin **==** client**[**i**].**current\_account**::**return\_current\_pin**()){**

client**[**i**].**current\_account**::**withdraw**();**

**}**

**else{**

cout**<<"Incorrect PIN. Withdraw Request Rejected."<<**endl**;**

**}**

**}**

**}**

**else{**

cout**<<"You don't have such acccount with the bank."<<**endl**;**

**}**

**}**

**else** **if(**master\_found **==** 0**){**

cout**<<"The searched account didn't match with any account."<<**endl**;**

**}**

**break;**

**}**

**case** 5**:** **{**

**int** master\_search\_pin**,** master\_found**,** i**,** deposit\_acc\_type**;**

cout**<<"Enter the master key for your account: ";**

cin**>>**master\_search\_pin**;**

**for(**i **=** 0**;** i**<=**user\_count**;** i**++){**

master\_found **=** client**[**i**].**search\_master**(**master\_search\_pin**);**

**if(**master\_found **==** 1**){**

cout**<<"Account Found."<<**endl**;**

**break;**

**}**

**}**

**if(**master\_found **==** 1**){**

cout**<<"Enter the account to where you want to deposit money."<<**endl**;**

cout**<<"Enter 1 to deposit to saving account or 2 to deposit to current account: ";**

cin**>>**deposit\_acc\_type**;**

**if(**deposit\_acc\_type **==** client**[**i**].**return\_account\_type**()** **||** client**[**i**].**return\_account\_type**()==**3 **){**

**if(**deposit\_acc\_type**==**1**){**

client**[**i**].**saving\_account**::**deposit**();**

**}**

**else** **if(**deposit\_acc\_type**==**2**){**

client**[**i**].**current\_account**::**deposit**();**

**}**

**}**

**else{**

cout**<<"You don't have such acccount with the bank."<<**endl**;**

**}**

**}**

**else** **if(**master\_found **==** 0**){**

cout**<<"The searched account didn't match with any account."<<**endl**;**

**}**

**break;**

**}**

**case** 6**:** **{**

**float** loan\_capacity**;**

**int** master\_search\_pin**,** master\_found**,** i**;**

cout**<<"Enter the master key for your account: ";**

cin**>>**master\_search\_pin**;**

**for(**i **=** 0**;** i**<=**user\_count**;** i**++){**

master\_found **=** client**[**i**].**search\_master**(**master\_search\_pin**);**

**if(**master\_found **==** 1**){**

cout**<<"Account Found."<<**endl**;**

**break;**

**}**

**}**

loan\_capacity **=** client**[**i**].**calculate\_optimal\_loan**();**

cout**<<"Your optimal loan credit capacity is "<<**loan\_capacity**<<"."<<**endl**;**

**break;**

**}**

**case** 7**:** **{**

**int** key**;**

cout**<<"You need to a bank account to take loan."<<**endl**;**

cout**<<"Press -1 to go to main menu and create bank account or any other key to proceed: ";**

cin**>>**key**;**

**if(**key **==** **-**1**){**

cout**<<"Taking you to main menu."<<**endl**;**

**break;**

**}**

**else{**

**float** loan\_capacity**;**

**int** master\_search\_pin**,** master\_found**,** i**;**

cout**<<"Enter the master key for your account: ";**

cin**>>**master\_search\_pin**;**

**for(**i **=** 0**;** i**<=**user\_count**;** i**++){**

master\_found **=** client**[**i**].**search\_master**(**master\_search\_pin**);**

**if(**master\_found **==** 1**){**

cout**<<"Account Found."<<**endl**;**

**break;**

**}**

**}**

**if(**master\_found **==** 1**){**

client**[**i**].**loan**();**

**}**

**else{**

cout**<<"Account not found. Try again."<<**endl**;**

**}**

**}**

**break;**

**}**

**default:{**

cout**<<"Please enter a valid option."<<**endl**;**

**break;**

**}**

cout**<<**endl**;**

**}**

**}**

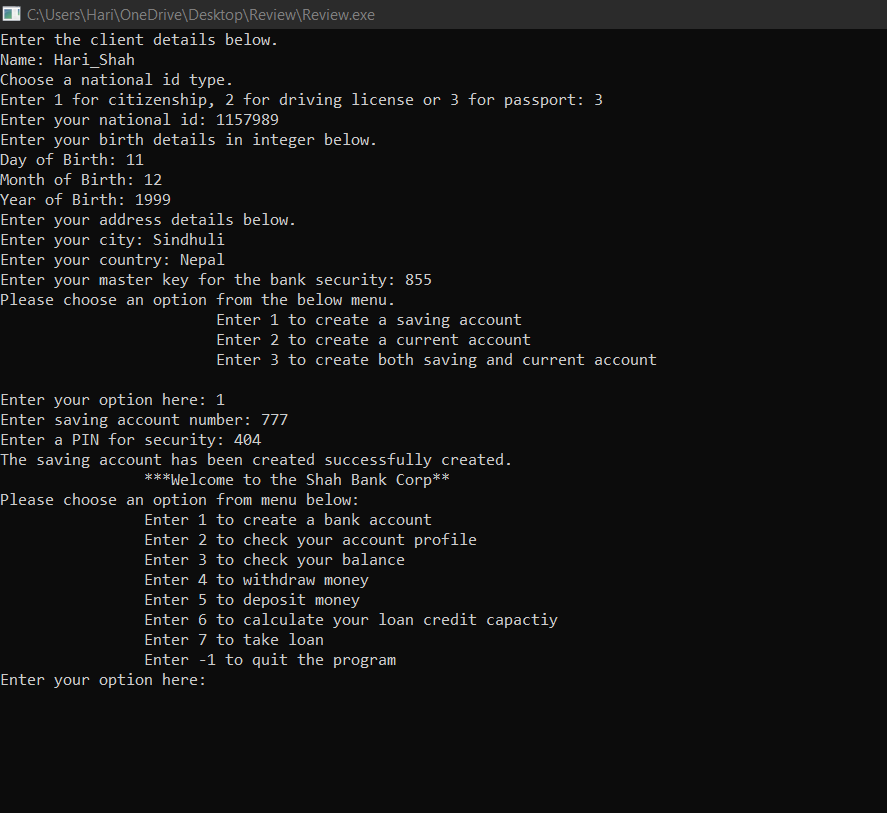
**while(**option **!=** **-**1**);**

**}**

**Result and Discussion with various output**

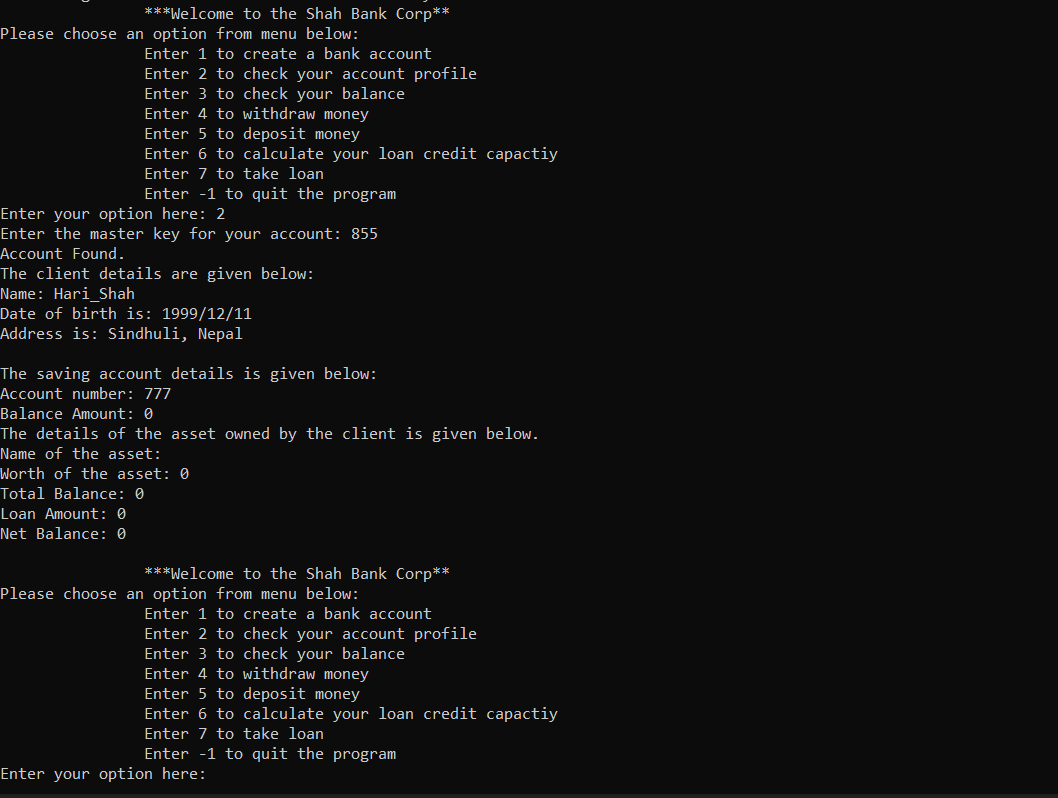
1. Saving account creation

I created a saving account through the menu options. I gave all the required input and finally a saving account was created. In this case, the master security PIN was 855 and security PIN for saving account was 404. These PINs will be further used to access my bank account and perform various banking operations like withdrawal, deposit, balance enquiry etc. **Below the snapshot of the output:**



1. Checking Profile Details

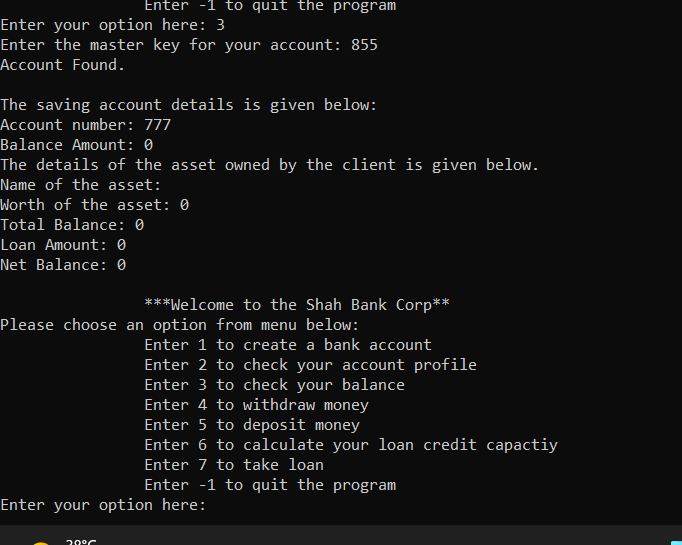
To check client’s profile, the client must enter his master key. Once, the master key matches with the bank account in the database, the client’s detail is printed.



1. Check Balance

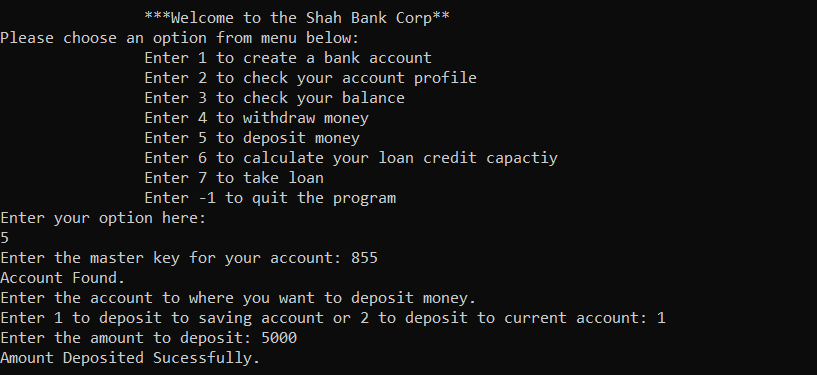
Client can check his/her balance using his/her master key.

Below is the attached snapshot of balance inquiry screen:

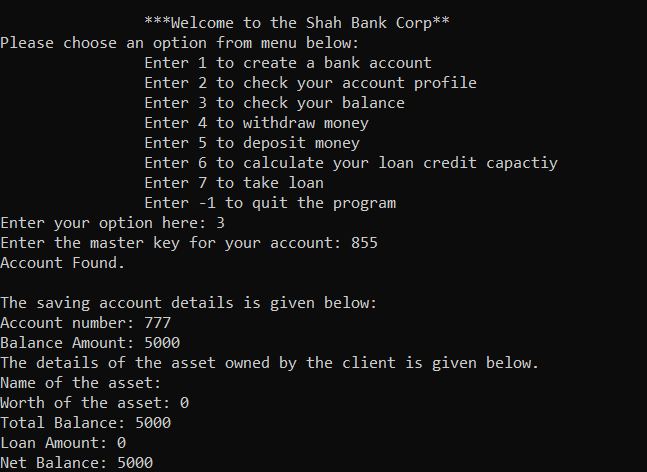


1. Deposit Money

Client needs to enter the master key to deposit money in his/her account. Then, client needs to select the account type i.e saving account or current account. If user tries to deposit money in any account that s/he doesn’t own, the system shows him a warning that he doesn’t own such account with the bank.

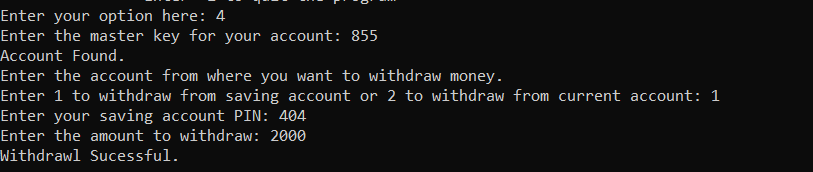


After Deposit, the balance is reflected in the saving account.

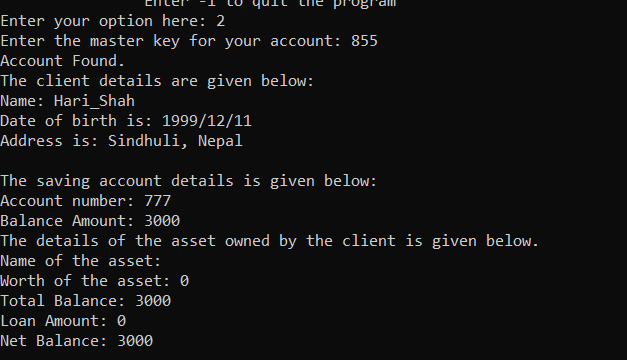


1. Withdrawal

User needs to enter both master key and securtiy key of the account type to withdraw money from the account. Furthermore, the user needs to maintain atleast Rs 1000 in the account. The system is designed in such a way that it restricts the client to withdraw money in such amount that violets this minimum balance.

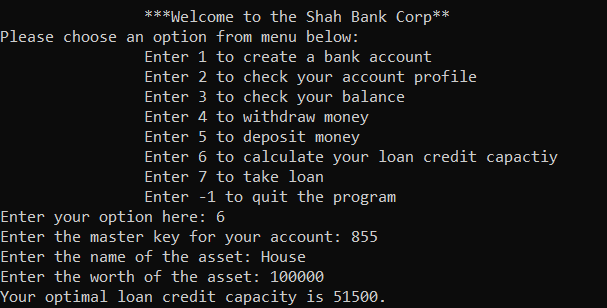


After Withdrawal, the client profile is :



1. Calculate optimal Loan credit capacity

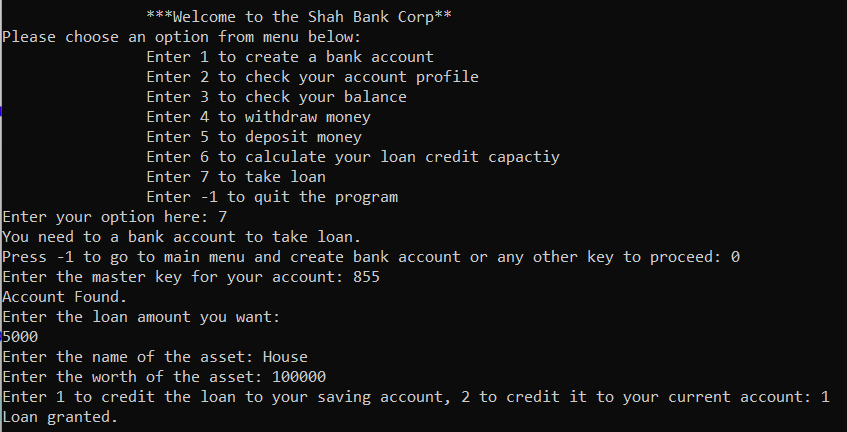
The program calculates the optimal loan capacity of the client as the 50% of the total net worth of the client where the net worth of the client is given as the net balance of the client summed up with his total assets.



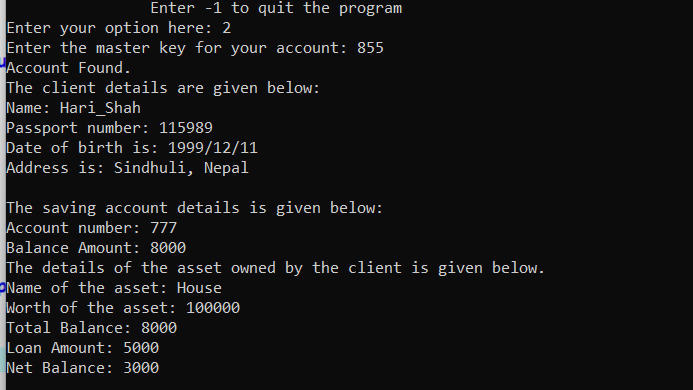
1. Loan

Loan option is used to grant loan to a client. A client must own a bank account with the bank to take loan. If user doesn’t have an account with bank, the system will take the user back to the main menu to create a bank account. If the user has a bank account, the user can proceed to take loan from the bank. While taking the loan, the user’s total net worth including the net balance and the client’s asset will be evaluated and only upto 50% of the amount will be granted. If the user requests for more loan, the user’s loan request will be rejected.

**Below the snapshot of loan screenshot.**



Below is the screenshot of the client’s profile after taking loan.



**Future Work Discussion**

I plan on adding additional features like loan interest calculation, total loan disbursement of the bank, the risk profile of the bank, analysis of the bad loan of the bank and bad loan recovery features. I will keep learning more and more concepts of object oriented programming and introduce these features to my program.